Big Data Applications and Analytics Fall 2016 Documentation

Release 1.0

Gregor von Laszewski

December 06, 2016

1 Overview **3**
1.1 About the Course . . . 4
1.2 Course Numbers . . . 4
1.3 Meeting Times . . . 5
1.4 Office Hours . . . 5
1.5 Discussions and communication . . . 6
1.6 Calendar . . . 6
1.7 Common Mistakes . . . 8
1.8 Systems Usage . . . 8
1.9 Term Paper or Project . . . 8
1.10 Software Project . . . 9
1.11 Term Paper . . . 9
1.12 Report Format . . . 10
1.13 Code Repositories Deliverables . . . 11
1.14 Prerequisites . . . 11
1.15 Learning Outcomes . . . 12
1.16 Grading . . . 12
1.17 Academic Integrity Policy . . . 12
1.18 Instructors . . . 12
1.19 Teaching Assistants . . . 15
1.20 Links . . . 16
1.21 Updates . . . 16

2 Syllabus **17**
2.1 Errata . . . 19
2.2 Section 1 - Introduction . . . 19
2.3 Section 2 - Overview of Data Science: What is Big Data, Data Analytics and X-Informatics? . . . 24
2.4 Section 3 - Health Informatics Case Study . . . 30
2.5 Section 4 - Sports Case Study . . . 35
2.6 Section 5 - Technology Training - Python & FutureSystems (will be updated) . . . 39
2.7 Section 6 - Physics Case Study . . . 42
2.8 Section 7 - Big Data Use Cases Survey . . . 48
2.9 Section 8 - Technology Training - Plotviz . . . 60
2.10 Section 9 - e-Commerce and LifeStyle Case Study . . . 61
2.11 Section 10 - Technology Training - kNN & Clustering . . . 66
2.12 Section 11 - Cloud Computing Technology for Big Data Applications & Analytics (will be updated) . . . 69
2.13 Section 12 - Web Search and Text Mining and their technologies . . . 77
2.14 Section 13 - Technology for Big Data Applications and Analytics . . . 81

2.15 Section 14 - Sensors Case Study . . . 85
2.16 Section 15 - Radar Case Study . . . 87

3 FAQ **89**
3.1 I am full time student at IUPUI? Can I take the online version? . . . 89
3.2 I am a residential student can I take the online version only? . . . 89
3.3 Do I need to buy a textbook? . . . 89
3.4 Do I need a computer to participate in this class? . . . 89
3.5 How to write a research article on computer science . . . 90
3.6 How to you use bibliography managers JabRef & Endnote or Mendeley . . . 90
3.7 Plagiarism test and resources related to that . . . 90
3.8 How many hours will this course take to work on every week? . . . 90

4 Homework **93**
4.1 Assignments . . . 95
4.2 Assignment Guidelines . . . 104
4.3 Homework upload . . . 108

5 Using GitLab **111**
5.1 Getting an account . . . 111
5.2 Upload your public key . . . 111
5.3 How to configure Git and Gitlab for your computer . . . 111
5.4 Using Web browsers to upload . . . 112
5.5 Using Git GUI tools . . . 112
5.6 Submission of homework . . . 112
5.7 Git Resources . . . 113

6 Software Projects **115**
6.1 Common Requirements . . . 115
6.2 Deployment Projects . . . 117
6.3 IaaS . . . 117
6.4 Analytics Projects . . . 118
6.5 Project Idea: World wide road kill . . . 119
6.6 Project Idea: Author disambiguty problem . . . 120

7 Introduction to Python **121**
7.1 Acknowledgments . . . 122
7.2 Description . . . 122
7.3 Installation . . . 123
7.4 Alternative Installations . . . 123
7.5 Resources . . . 124
7.6 Prerequisite . . . 124
7.7 Dependencies . . . 125
7.8 Learning Goals . . . 125
7.9 Using Python on FutureSystems . . . 125
7.10 Interactive Python . . . 125
7.11 Syntax . . . 126
7.12 Writing and Saving Programs . . . 130
7.13 Installing Libraries . . . 133
7.14 Further Learning . . . 135
7.15 Exercises . . . 136
7.16 Ecosystem . . . 136

8 Python for Big Data **139**
8.1 Managing Data . . . 139

8.2 Numpy . . . 140
8.3 Graphics Libraries . . . 140
8.4 Network and Graphs . . . 141
8.5 Examples . . . 141

9 Python Fingerprint Example **143**
9.1 Utility functions . . . 144
9.2 Dataset . . . 145
9.3 Data Model . . . 146
9.4 Plotting . . . 149
9.5 Main Entry Point . . . 150
9.6 Running . . . 150

10 Datasets **153**

11 Refcards **155**

12 Linux **157**
12.1 File commands . . . 157
12.2 Search commands . . . 157
12.3 Help . . . 157
12.4 Keyboard Shortcuts . . . 158
12.5 Assignments . . . 158

13 LaTeX **159**
13.1 Sharelatex . . . 159
13.2 Overleaf . . . 159
13.3 jabref . . . 159
13.4 References . . . 160
13.5 Introduction . . . 160
13.6 Manual pages and programs . . . 160
13.7 The LaTeX Cycle . . . 161
13.8 Generating Images . . . 161
13.9 Editing LaTeX . . . 161
13.10 How to edit Bibliographies? . . . 162
13.11 How to produce Slides? . . . 162

14 Reference Managers **163**
14.1 jabref . . . 163
14.2 Endnote . . . 163
14.3 Mendeley . . . 163
14.4 Zotero . . . 163

15 Ubuntu Virtual Machine **165**
15.1 Creation . . . 165
15.2 Guest additions . . . 166
15.3 Development Configuration . . . 166
15.4 Homework Virtualbox . . . 166

16 Using SSH Keys **167**
16.1 Using SSH from Windows . . . 167
16.2 Using SSH on Mac OS X . . . 168
16.3 Generate a SSH key . . . 169
16.4 Add or Replace Passphrase for an Already Generated Key . . . 170
16.5 Upload the key to gitlab . . . 170

17 Links Report **171**

18 Homework References **173**

19 Drafts (TODO) **179**
19.1 Additional Programming Assignments . . . 179
19.2 Preview Course Examples . . . 181
19.3 Additional Programming Assignemts 2 . . . 182
19.4 Installing Cloudmesh Client . . . 186
19.5 Hadoop . . . 186
19.6 Refernces . . . 187
19.7 Cloud Resources . . . 187
19.8 QuickStart for OpenStack on FutureSystems . . . 187
19.9 Chameleon Cloud . . . 194
19.10 Example . . . 195

20 Contributing **197**

21 Todos **199**
21.1 General . . . 199

22 Changelog **201**
22.1 %%version%% (unreleased) . . . 201

Bibliography **203**

- Semester: Fall 2016
- Link to Read the Docs: http://bdaafall2016.readthedocs.io/en/latest/
- Link to OpenEdX

CHAPTER 1

Overview

Page Contents

- *About the Course*
- *Course Numbers*
- *Meeting Times*
- *Office Hours*
- *Discussions and communication*
- *Calendar*
- *Common Mistakes*
- *Systems Usage*
- *Term Paper or Project*
- *Software Project*
- *Term Paper*
- *Report Format*
- *Code Repositories Deliverables*
- *Prerequisites*
- *Learning Outcomes*
- *Grading*
- *Academic Integrity Policy*
- *Instructors*
  - *Dr. Gregor von Laszewski*
  - *Dr. Geoffrey Fox*
  - *Dr. Badi' Abdul-Wahid*
- *Teaching Assistants*
  - *Hyungro Lee*
  - *Jerome Mitchell*
  - *Prashanth Balasubramani*
- *Links*
- *Updates*

This page may be updated throughout Fall 2016, we recommend to review this page weekly for changes.

1.1 About the Course

The Big Data Applications and Analytics course is an overview course in Data Science and covers the applications and technologies (data analytics and clouds) needed to process the application data. It is organized around rallying cry: Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics.

1.2 Course Numbers

This course is offered for Graduate and Undergraduate students at Indiana University and as an online course. To Register, for University credit please go to:

- http://registrar.indiana.edu/browser/soc4168/INFO/INFO-I523.shtml
- http://registrar.indiana.edu/browser/soc4168/INFO/INFO-I423.shtml
- http://registrar.indiana.edu/browser/soc4168/ENGR/ENGR-E599.shtml

Please, select the course that is most suitable for your program:

- **INFO-I 423 - BIG DATA APPLS & ANALYTICS**
  - 34954 online undergraduate students
  - 34955 Discussion Friday 9:30 - 10:45AM Informatics East (I2) 150
- **INFO-I 523 - BIG DATA APPLS & ANALYTICS**
  - 32863 online graduate students
  - 32864 Discussion Friday 9:30 - 10:45AM Informatics East (I2) 150
  - 32866 Data Science majors only
- **ENGR-E 599 - TOPICS IN INTELLIGENT SYSTEMS ENGINEERING**
  - 36362 online graduate engineering students
  - 36363 Discussion Friday 9:30 - 10:45AM Informatics East (I2) 150

Warning: Please note that all discussion sections for residential students have been merged to:
- Friday 9:30 - 10:45AM Informatics East (I2) 150

Please ignore postings in CANVAS and the REGISTRAR about this.

From Registrar (however with updated meeting times and location):

```
INFO-I 523  BIG DATA APPLS & ANALYTICS (3 CR)
     CLSD *****          ARR             ARR    ARR       Von Laszewski G          50    0    2
              Above class open to graduates only
              Above class taught online
              Discussion (DIS)
     CLSD 32864          09:30A-10:45A   F      I2 150    Von Laszewski G          50    0    2
              Above class meets with INFO-I 423

INFO-I 523  BIG DATA APPLS & ANALYTICS (3 CR)
    I 523 : P - Data Science majors only
           32866 RSTR     ARR             ARR    ARR       Von Laszewski G          90   72    0
              This is a 100% online class taught by IU Bloomington. No
              on-campus class meetings are required. A distance education
              fee may apply; check your campus bursar website for more
              information
```

```
             Above class for students not in residence on the Bloomington
             campus

INFO-I 423  BIG DATA APPLS & ANALYTICS (3 CR)
     CLSD ***** RSTR     ARR             ARR    ARR       Von Laszewski G          10    0    6
             Above class open to undergraduates only
             Above class taught online
             Discussion (DIS)
     CLSD 34955 RSTR     09:30A-10:45A   F      I2 150    Von Laszewski G          10    0    6
             Above class meets with INFO-I 523

ENGR-E 599  TOPICS IN INTELL SYS ENGINEER (3 CR)
       VT: BG DATA APPLICATNS & ANLYTCS ISE
          ***** RSTR     ARR             ARR    ARR       Von Laszewski G          25   25    0
             Above class open to graduate engineering students only
             Above class taught online
             Discussion (DIS)
       VT: BG DATA APPLICATNS & ANLYTCS ISE
          36363 RSTR     01:00P-02:15P   F      HD TBA    Von Laszewski G          25   25    0
              Above class meets with INFO-I 523
```

1.3 Meeting Times

The classes are published online. Residential students at Indiana University will participate in a discussion taking place at the following time:

- Fridays 09:30am - 10:45am EST, I2 150

For the 100% online students see the office hours.

1.4 Office Hours

Office hours will be held every week

- Tue 10-11am EST, typically Gregor
- Thu 6-7pm EST, typically Gregor
- Sun 4-6pm EST, either Jerome or Prashanth
- Tue 7-8pm, either Jerome or Prashanth
- Wed 7-8pm, either Jerome or Prashanth

These are live sessions that will allow you to interact in group or one-on-one with either an instructor or a TA. Office hours sessions may be recorded. All importan FAQs will be either posted on the Web page or in Piazza ASAP. During these times, we can be reached via zoom with the following information for the call:

Join from PC, Mac, Linux, iOS or Android:

- https://IU.zoom.us/j/195576919

Or Telephone:

- However as we are most likely sharing documents phone participation may not be too useful.
- Dial: +1 646 558 8656 (US Toll) or +1 408 638 0968 (US Toll)
- Meeting ID: 195 576 919

- International numbers available: https://IU.zoom.us/zoomconference?m=GUZ8CEVGWPB_312js4gnzkGM_QvcVUy3
- Or a H.323/SIP room system:
  - H.323: 162.255.37.11 (US West) or 162.255.36.11 (US East)
  - Meeting ID: 195 576 919
  - SIP: 195576919@zoomcrc.com

Please use a headphone with microphone to increase sound quality.

1.5 Discussions and communication

Online discussions and communication will be conducted in piazza at the following URL:

https://piazza.com/iu/fall2016/infoi523/home

Discussions are conducted in clearly marked folders/topics. For example "Discussion d1" will be conducted in the piazza folder "d1". Students are responsible for posting their content to the right folder. No credit will be given if the post has been filed wrongly.

Please note that the communications to instructors can be seen by all instructors. In matters that are sensitive, please use gvonlasz@indiana.edu. Please, never share your university ID number or your social security number or any other sensitive information with us either in e-mail or in the discussion lists.

1.6 Calendar

All sessions refer to Sections, Discussions and Units

- This document supersedes any assignment dates and comments regarding assignments made in videos or stated elsewhere
- Official and additional announcements will be send via CANVAS
- All lectures are assigned Friday's
- All discussions and homework are due 3 weeks after the assignment + the next weekend, e.g. Monday Morning if not specified otherwise. Precise dates will be published in CANVAS
- Note calendar and content may change

Assigned	Wk	Week	Descriptions
08/22/2016	1	W1	*Section 1 - Introduction* (due in W1) *Section 2 - Overview of Data Science: What is Big Data, Data Analytics and X-Informatics?* (due in W1) d1 (due in W1) SURVEY1 (due in W1) *Paper p1* (due in W2)
08/26/2016	2	W2	*Section 3 - Health Informatics Case Study* d3, *Paper p2*
09/02/2016	3	W3	*Section 4 - Sports Case Study* d4 Geolocation Quiz on Canvas *Paper p3* *References R1*
09/05/2016	3	Holiday	Labor Day
09/09/2016	4	W4	d5 Preparation reports Preparation: *Software Projects* *Ubuntu Virtual Machine* *Programming prg1: Python* (recom. 10/14 (3) *Programming prg1: Python* (due 12/02)
09/16/2016	5	W5	*Section 6 - Physics Case Study* d6 Work on Project Learn enough Python(2)
09/23/2016	6	W6	
1.6. Calendar			*Section 7 - Big Data Use Cases Survey* d7 Work on Project

- (1) Use lecture free time wisely
- (2) Improve your python knowledge while you do your project
- (3) If you can not do PRG by Oct 10/14 or have difficulties with it, we recommend that you do a paper
- (4) we will not do PRG2, and PRG3 in this class
- (5) if you have homework late past Dec 2nd you may run the risk of obtaining an incomplete in the class as grading may need time and will be conducted in January.
- (6) *Paper p11* has been canceled so you can focus on your project

The following sections will be replaced:

- TBD: *Section 5 - Technology Training - Python & FutureSystems (will be updated)*

1.7 Common Mistakes

- Starting the Project late.
- Not using gitlab for homework submission
- Not using the 2 column ACM report template
- Not using jabref or endnote for References
- Not understanding plagiarism
- Being in a team where one team member does not perform
- Violating university policy by doing another students work
- Not using frequent checkins to gitlab and pushing the commits

1.8 Systems Usage

Projects may be executed on your local computer, a cloud or other resources you may have access to. This may include:

- chameleoncloud.org
- furturesystems.org
- AWS (you will be responsible for charges)
- Azure (you will be responsible for charges)
- virtualbox if you have a powerful computer and like to prototype
- other clouds

1.9 Term Paper or Project

You have a choice to write a term paper or do a software project. This will constitute to **50%** of your class grade.

In case you chose a project your maximum grade could be an A+. However, an A+ project must be truly outstanding and include an exceptional project report. Such a project and report will have the potential quality of being able to be published in a conference.

In case you chose a Term Paper your maximum Grade for the entire class will be an A-.

Please note that a project includes also writing a project report/paper. However the length is a bit lower than for a term paper.

1.10 Software Project

In case of a software project, we encourage a group project with up to three members. You can use the discussion forum in the folder project to form project teams or just communicate privately with other class members to formulate a team. The following artifacts are part of the deliverables for a project

Code: You must deliver the code in gitlab. The code must be compilable and a TA may try to replicate to run your code. You MUST avoid lengthy install descriptions and everything must be installable from the command line. We will check submission. All team members must be responsible for one part of the project.

Project Report: A report must be produced while using the format discussed in the Report Format section. The following length is required:

- 4 pages, one student in the project
- 6 pages, two students in the project
- 8 pages, three students in the project

Work Breakdown: This document is only needed for team projects. A one page PDF document describing who did what. It includes pointers to the git history that documents the statistics that demonstrate not only one student has worked on the project.

In addition the graders will go into gitlab, which provides a history of checkins to verify each team member has used gitlab to checkin their contributions frequently. E.g. if we find that one of the students has not checked in code or documentation at all, it will be questioned.

License: All projects are developed under an open source license such as Apache 2.0 License, or similar. You will be required to add a LICENCE.txt file and if you use other software identify how it can be reused in your project. If your project uses different licenses, please add in a README.rst file which packages are used and which license these packages have.

Additional links:

- *Software Projects*

1.11 Term Paper

Teams: Up to three people. You can use the discussion forum in the folder term-project to build teams.

Term Report: A report must be produced while using the format discussed in the Report Format section. The following length is required:

In case you chose the term paper, you or your team will pick a topic relevant for the class. You will write a high quality scholarly paper about this topic. The following artifacts are part of the deliverables for a term paper. A report must be produced while using the format discussed in the Report Format section. The following length is required:

- 6 pages, one student in the project
- 9 pages, two student in the project
- 12 pages, three student in the project

Work Breakdown: This document is only needed for team projects. A one page PDF document describing who did what.

Grading: As stated above the maximum grade for the entire class will be A- if you deliver a very good paper. However, axceptional term papers are possible and could result in higher grades. They must contain significant contributions and novel ideas so that the paper could be published in a conference or journal. A comprehensive survey would be an example. The page limitation will most likely be exceeded by such work. Number of pages is not reflecting quallity. Refernces must be outstanding.

Additional links:

- reports
- A sample report directory can be looked at at

  https://gitlab.com/cloudmesh/project-000/tree/master

  Please make sure to follow exact the guiedlines given in report/README.rst and that you have a report/report.pdf, as well as submit all images in an image folder.

1.12 Report Format

All reports will be using the ACM proceedings format. The MSWord template can be found here:

- `paper-report.docx`

A LaTeX version can be found at

- https://www.acm.org/publications/proceedings-template

however you have to remove the ACM copyright notice in the LaTeX version.

There will be **NO EXCEPTION** to this format. In case you are in a team, you can use either gitlab while collaboratively developing the LaTeX document or use MicrosoftOne Drive which allows collaborative editing features. All bibliographical entries must be put into a bibliography manager such as jabref, endnote, or Mendeley. This will guarantee that you follow proper citation styles. You can use either ACM or IEEE reference styles. Your final submission will include the bibliography file as a separate document.

Documents that do not follow the ACM format and are not accompanied by references managed with jabref or endnote or are not spell checked will be returned without review.

Report Checklist:

- [ ] Have you written the report in word or LaTeX in the specified format.
- [ ] In case of LaTeX, have you removed the ACM copyright information
- [ ] Have you included the report in gitlab.
- [ ] Have you specified the names and e-mails of all team members in your report. E.g. the username in Canvas.
- [ ] Have you included all images in native and PDF format in gitlab in the images folder.
- [ ] Have you added the bibliography file (such as endnote or bibtex file e.g. jabref) in a directory bib.
- [ ] Have you submitted an additional page that describes who did what in the project or report.
- [ ] Have you spellchecked the paper.
- [ ] Have you made sure you do not plagiarize.
- [ ] Have you structured your directory as given in the sample at https://gitlab.com/cloudmesh/project-000/tree/master

- [ ] Have you followed the guiedlines given in report/README.rst of project-000 and that you have a report/report.pdf, as well as submit all images in an image folder.

1.13 Code Repositories Deliverables

Code repositories are for code, if you have additional libraries that are needed you need to develop a script or use a DevOps framework to install such software. Thus zip files and .class, .o files are not permissible in the project. Each project must be reproducible with a simple script. An example is:

```
git clone ....
make install
make run
make view
```

Which would use a simple make file to install, run, and view the results. Naturally you can use ansible or shell scripts. It is not permissible to use GUI based DevOps preinstalled frameworks. Everything must be installable form the command line.

1.14 Prerequisites

Python or Java experience is expected. The programming load is modest.

In case you elect a programming project we will assume that you are familiar with the programming languages required as part of the project you suggest. We will limit the languages to Python and JavaScript if you like to do interactive visualization. If you do not know the required technologies, we will expect you to learn it outside of class. For example, Python has a reputation for being easy to learn, and those with strong programming background in another general-purpose programming language (like C/C++, Java, Ruby, etc.) can learn it within a few hours to days dependent on experience level. Please consult the instructor if you have concerns about your programming background. In addition, we may encounter math of various kinds, including linear algebra, probability theory, and basic calculus. We expect that you know them on an elementary level. Students with limited math backgrounds may need to do additional reading outside of class.

In case you are interested in further development of cloudmesh for big data strong Python or JavaScript experience is needed.

You will also need a sufficiently modern and powerful computer to do the class work. Naturally if you expect that you want to to the course only on your cell phone or iPad, or your windows 98 computer, this does not work. We recommend that you have a relatively new and updated computer with sufficient memory. In some cases its easier to not use Windows and for example use Linux via virtualbox, so your machine should have sufficient memory to comfortably run it. If you do not have such a machine we are at this time trying to get virtual machines that you can use on our cloud. However, runtime of these VMs is limited to 6 hours and they will be terminated after that. Naturally you can run new VMs. This is done in order to avoid resource "hogging" of idle VMs. In contrast to AWS you are not paying for our VMs so we enforce a rule to encourage proper community spirit while not occupying resources that could be used by others. Certainly you can naturally also use AWS or other clouds where you can run virtual machines, but in that case you need to pay for the usage yourself.

Please remember that this course does not have a required text books and the money you safe on this you can be used to buy a new or upgrade your current computer if needed.

1.15 Learning Outcomes

Students will gain broad understanding of Big Data application areas and approaches used. This course is a good preparation for any student likely to be involved with Big Data in their future.

1.16 Grading

Grading for homework will be done within a week of submission on the due date. For homework that were submitted beyond the due date, the grading will be done within 2-3 weeks after the submission. A 10% grade reduction will be given. Some homework can not be delivered late (which will be clearly marked and 0 points will be given if late; these are mostly related to setting up your account and communicating to us your account names.)

It is the student's responsibility to upload submissions well ahead of the deadline to avoid last minute problems with network connectivity, browser crashes, cloud issues, etc. It is a very good idea to make early submissions and then upload updates as the deadline approaches; we will grade the last submission received before the deadline.

Note that paper and project will take a considerable amount of time and doing proper time management is a must for this class. Avoid starting your project late. Procrastination does not pay off. Late Projects or term papers will receive a 10% grade reduction.

- 40% Homework
- 50% Term Paper or Project
- 10% Participation/Discussion

Details about the assignments can be found in the Section *Homework*.

1.17 Academic Integrity Policy

We take academic integrity very seriously. You are required to abide by the Indiana University policy on academic integrity, as described in the Code of Student Rights, Responsibilities, and Conduct, as well as the Computer Science Statement on Academic Integrity (http://www.soic.indiana.edu/doc/graduate/graduate-forms/Academic-Integrity-Guideline-FINAL-2015.pdf). It is your responsibility to understand these policies. Briefly summarized, the work you submit for course assignments, projects, quizzes, and exams must be your own or that of your group, if group work is permitted. You may use the ideas of others but you must give proper credit. You may discuss assignments with other students but you must acknowledge them in the reference section according to scholarly citation rules. Please also make sure that you know how to not plagiarize text from other sources while reviewing citation rules.

We will respond to acts of plagiarism and academic misconduct according to university policy. Sanctions typically involve a grade of 0 for the assignment in question and/or a grade of F in the course. In addition, University policy requires us to report the incident to the Dean of Students, who may apply additional sanctions, including expulsion from the university.

Students agree that by taking this course, papers and source code submitted to us may be subject to textual similarity review, for example by Turnitin.com. These submissions may be included as source documents in reference databases for the purpose of detecting plagiarism of such papers or codes.

1.18 Instructors

The course presents lectures in online form given by the instructors listed bellow. Many others have helped making this material available and may not be listed here.

For this class support is provided by

- Gregor von Laszewski (PhD)
- Badi' Abdul-Wahid (PhD)
- Jerome Mitchell (Teaching Assistant)
- Prashanth Balasubramani (Teaching Assistant)
- Hyungro Lee (Teaching Assistant)

1.18.1 Dr. Gregor von Laszewski

Gregor von Laszewski is an Assistant Director of Cloud Computing in the DSC. He held a position at Argonne National Laboratory from Nov. 1996 – Aug. 2009 where he was last a scientist and a fellow of the Computation Institute at University of Chicago. During the last two years of that appointment he was on sabbatical and held a position as Associate Professor and the Director of a Lab at Rochester Institute of Technology focussing on Cyberinfrastructure. He received a Masters Degree in 1990 from the University of Bonn, Germany, and a Ph.D. in 1996 from Syracuse University in computer science. He was involved in Grid computing since the term was coined. He was the lead of the Java Commodity Grid Kit (http://www.cogkit.org) which provides till today a basis for many Grid related projects including the Globus toolkit. Current research interests are in the areas of Cloud computing. He is leading the effort to develop a simple IaaS client available at as OpenSource project at http://cloudmesh.github.io/client/

His Web page is located at http://gregor.cyberaide.org. To contact him please send mail to laszewski@gmail.com. For class related e-mail please use PIazza for this class.

In his free time he teaches Lego Robotics to high school students. In 2015 the team won the 2nd prize in programming design in Indiana. If you like to volunteer helping in this effort please contact him.

He offers also the opportunity to work with him on interesting independent studies. Current topics include but are not limited to

- cloudmesh
- big data benchmarking
- scientific impact of supercomputer and data centers.
- STEM and other educational activities while using robotics or big data

Please contact me if you are interested in this.

1.18.2 Dr. Geoffrey Fox

Fox received a Ph.D. in Theoretical Physics from Cambridge University and is now distinguished professor of Informatics and Computing, and Physics at Indiana University where he is director of the Digital Science Center, Chair of Department of Intelligent Systems Engineering and Director of the Data Science program at the School of Informatics and Computing. He previously held positions at Caltech, Syracuse University and Florida State University after being a postdoc at the Institute of Advanced Study at Princeton, Lawrence Berkeley Laboratory and Peterhouse College Cambridge. He has supervised the PhD of 68 students and published around 1200 papers in physics and computer science with an index of 70 and over 26000 citations. He currently works in applying computer science from infrastructure to analytics in Biology, Pathology, Sensor Clouds, Earthquake and Ice-sheet Science, Image processing, Deep Learning, Manufacturing, Network Science and Particle Physics. The infrastructure work is built around Software Defined Systems on Clouds and Clusters. The analytics focuses on scalable parallelism.

He is involved in several projects to enhance the capabilities of Minority Serving Institutions. He has experience in online education and its use in MOOCs for areas like Data and Computational Science. He is a Fellow of APS (Physics) and ACM (Computing).

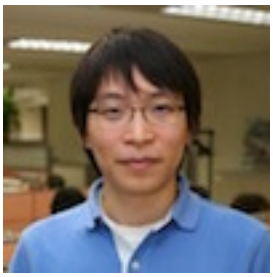
1.18.3 Dr. Badi' Abdul-Wahid

Badi' received a Ph.D. in Computer Science at the University of Notre Dame under Professor Jesus Izaguirre. The primary focus of his graduate work was the development of scalable, fault-tolerant, elastic distributed applications for running Molecular Dynamics simulations.

At Indiana University, Badi' works with the FutureSystems project on a NIST-funded study whose goal is to understand patterns in the development and usage of Big Data Analysis pipelines.

1.19 Teaching Assistants

1.19.1 Hyungro Lee



Hyungro Lee is a PhD candidate in Computer Science at Indiana University working with Dr. Geoffrey C. Fox. Prior to beginning the PhD program, Hyungro worked as a software engineer in the Cyworld Group (social networking platform in South Korea) at SK Communications, developing communications platforms including emails, texts and messaging at large scale to support over 40 million users. From this work he developed an interest in how distributed systems achieve scalability and high availability along with managing resources efficiently. He is currently working on the FutureSystems project to support Big Data Analysis Software Stacks in Virtual Clusters. He was also working on the FutureGrid project, an NSF funded significant new experimental computing grid and cloud test-bed to the research community, together with user supports. His research interests are parallel and distributed systems, and cloud computing

1.19.2 Jerome Mitchell

Jerome Mitchell is a Ph.D candidate in computer science at Indiana University and is interested in coupling the fields of computer and polar science. He has participated in the United State Antarctic Program, (USAP), where he collaborated with a multidisciplinary team of engineers and scientists to design a mobile robot for harsh polar environments to autonomously collect ice sheet data, decrease the human footprint of polar expeditions, and enhance measurement precision. His current work include: using machine learning techniques to help polar scientists identify bedrock and internal layers in radar imagery. He has also been involved in facilitating workshops to educate faculty and students on the importance of parallel and distributed computing at minority-serving institutions.

1.19.3 Prashanth Balasubramani

Prashanth Balasubramani is an MS student in Computer Science at Indiana University working with Gregor von Laszewski, Assistant Director of Cloud Computing at DSC. He has been working under Professor Gregor and Dr.Geoffrey Fox for the past year as an Associate Instructor for the course Big Data Analytics and Applications during the Fall 2015 and Spring 2016 semesters. Before joining Indiana University, he worked as a ETL developer for Capital One Banking firm (Wipro Technologies, Bangalore) developing Hadoop MR and Spark jobs for real time migration of Historical Data into virtual clusters on the Cloud. He is currently working as an Teaching Assistant for the Big Data Applications and Analytics course for the Fall 2016 semester. He is also working on NIST benchmarking project for recording benchmarks on different cloud platforms His research interests include Big Data applications, Cloud computing and Data Warehousing.

1.20 Links

This page is published at the following locations:

- OpenEdX: http://openedx.scholargrid.org/courses/SoIC/INFO-I-523/Fall_2016/about
- Readthedocs: http://bdaafall2016.readthedocs.io/en/latest/
- Source: https://gitlab.com/cloudmesh/fall2016

1.21 Updates

This page is conveniently managed with git. The location for the changes can be found at

- https://gitlab.com/cloudmesh/fall2016/commits/master

The repository is at

- https://gitlab.com/cloudmesh/fall2016/tree/master

Issues can be submitted at

- https://gitlab.com/cloudmesh/fall2016/issues

Or better use piazza so you notify us in our discussion lists. If you detect errors, you could also create a merge request at

- https://gitlab.com/cloudmesh/fall2016/merge_requests

CHAPTER 2

Syllabus

- *Errata*
- *Section 1 - Introduction*
  - *Unit 1.1 - Course Introduction*
    - *Lesson 1*
    - *Lesson 2 - Overall Introduction*
    - *Lesson 3 - Course Topics I*
    - *Lesson 4 - Course Topics II*
    - *Lesson 5 - Course Topics III*
  - *Unit 1.2 - Course Motivation*
    - *Unit Overview*
    - *Slides*
    - *Lesson 1 - Introduction*
    - *Lesson 2: Data Deluge*
    - *Lesson 3 - Jobs*
    - *Lesson 4 - Industrial Trends*
    - *Lesson 5 - Digital Disruption of Old Favorites*
    - *Lesson 6 - Computing Model: Industry adopted clouds which are attractive for data analytics*
    - *Lesson 7 - Research Model: 4th Paradigm; From Theory to Data driven science?*
    - *Lesson 8 - Data Science Process*
    - *Lesson 9 - Physics-Informatics Looking for Higgs Particle with Large Hadron Collider LHC*
    - *Lesson 10 - Recommender Systems I*
    - *Lesson 11 - Recommender Systems II*
    - *Lesson 12 - Web Search and Information Retrieval*
    - *Lesson 13 - Cloud Application in Research*
    - *Lesson 14 - Parallel Computing and MapReduce*
    - *Lesson 15 - Data Science Education*
    - *Lesson 16 - Conclusions*
    - *Resources*
- *Section 2 - Overview of Data Science: What is Big Data, Data Analytics and X-Informatics?*
  - *Section Overview*
  - *Unit 3 - Part I: Data Science generics and Commercial Data Deluge*
    - *Unit Overview*
    - *Slides*
    - *Lesson 1 - What is X-Informatics and its Motto*
    - *Lesson 2 - Jobs*
    - *Lesson 3 - Data Deluge ~~ General Structure*
    - *Lesson 4 - Data Science ~~ Process*
    - *Lesson 5 - Data Deluge ~~ Internet*
    - *Lesson 6 - Data Deluge ~~ Business I*
    - *Lesson 7 - Data Deluge ~~ Business II*
    - *Lesson 8 - Data Deluge ~~ Business III*
    - *Resources*
  - *Unit 4 - Part II: Data Deluge and Scientific Applications and Methodology*
    - *Unit Overview*
    - *Slides*
    - *Lesson 1 - Science & Research I*
    - *Lesson 2 - Science & Research II*
    - *Lesson 3 - Implications for Scientific Method*
    - *Lesson 4 - Long Tail of Science*
    - *Lesson 5 - Internet of Things*
    - *Resources*
  - *Unit 5 - Part III: Clouds and Big Data Processing; Data Science Process and Analytics*
    - *Unit Overview*
    - *Slides*
    - *Lesson 1 - Clouds*
    - *Lesson 2 - Features of Data Deluge I*
    - *Lesson 3 - Features of Data Deluge II*
    - *Lesson 4 - Data Science Process*
    - *Lesson 5 - Data Analytics I*
    - *Lesson 6 - Data Analytics II*

2.1 Errata

Note: You may find that some videos may have a different lesson, section or unit number. Please ignore this. In case the content does not correspond to the title, please let us know.

2.2 Section 1 - Introduction

This section has a technical overview of course followed by a broad motivation for course.

The course overview covers it's content and structure. It presents the X-Informatics fields (defined values of X) and the Rallying cry of course: Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics (or e-X). The courses is set up as a MOOC divided into units that vary in length but are typically around an hour and those are further subdivided into 5-15 minute lessons.

The course covers a mix of applications (the X in X-Informatics) and technologies needed to support the field electronically i.e. to process the application data. The overview ends with a discussion of course content at highest level. The course starts with a longish Motivation unit summarizing clouds and data science, then units describing applications (X = Physics, e-Commerce, Web Search and Text mining, Health, Sensors and Remote Sensing). These are interspersed with discussions of infrastructure (clouds) and data analytics (algorithms like clustering and collaborative filtering used in applications). The course uses either Python or Java and there are Side MOOCs discussing Python and Java tracks.

The course motivation starts with striking examples of the data deluge with examples from research, business and the consumer. The growing number of jobs in data science is highlighted. He describes industry trend in both clouds and big data. Then the cloud computing model developed at amazing speed by industry is introduced. The 4 paradigms of scientific research are described with growing importance of data oriented version.He covers 3 major X-informatics areas: Physics, e-Commerce and Web Search followed by a broad discussion of cloud applications. Parallel computing in general and particular features of MapReduce are described. He comments on a data science education and the benefits of using MOOC's.

2.2.1 Unit 1.1 - Course Introduction

Lesson 1

We provide a short introduction to the course covering it's content and structure. It presents the X-Informatics fields (defined values of X) and the Rallying cry of course: Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics (or e-X). The courses is set up as a MOOC divided into units that vary in length but are typically around an hour and those are further subdivided into 5-15 minute lessons. It follows discussion of mechanics of course with a list of all the units offered.

Video: https://youtu.be/CRYz3iTJxRQ

VIdeo with cc: https://www.youtube.com/watch?v=WZxnCa9Ltoc

Lesson 2 - Overall Introduction

This course gives an overview of big data from a use case (application) point of view noting that big data in field X drives the concept of X-Informatics. It covers applications, algorithms and infrastructure/technology (cloud computing). We are providing a short overview of the Syllabus

Video: https://youtu.be/Gpivfx4v5eY

Video with cc: https://www.youtube.com/watch?v=aqgDnu5fRMM

Lesson 3 - Course Topics I

Discussion of some of the available units:

- Motivation: Big Data and the Cloud; Centerpieces of the Future Economy
- Introduction: What is Big Data, Data Analytics and X-Informatics
- Python for Big Data Applications and Analytics: NumPy, SciPy, MatPlotlib
- Using FutureGrid for Big Data Applications and Analytics Course
- X-Informatics Physics Use Case, Discovery of Higgs Particle; Counting Events and Basic Statistics Parts I-IV.

Video: http://youtu.be/9NgG-AUOpYQ

Lesson 4 - Course Topics II

Discussion of some more of the available units:

- X-Informatics Use Cases: Big Data Use Cases Survey
- Using Plotviz Software for Displaying Point Distributions in 3D
- X-Informatics Use Case: e-Commerce and Lifestyle with recommender systems
- Technology Recommender Systems - K-Nearest Neighbors, Clustering and heuristic methods
- Parallel Computing Overview and familiar examples
- Cloud Technology for Big Data Applications & Analytics

Video http://youtu.be/pxuyjeLQc54

Lesson 5 - Course Topics III

Discussion of the remainder of the available units:

- X-Informatics Use Case: Web Search and Text Mining and their technologies
- Technology for X-Informatics: PageRank
- Technology for X-Informatics: Kmeans
- Technology for X-Informatics: MapReduce
- Technology for X-Informatics: Kmeans and MapReduce Parallelism
- X-Informatics Use Case: Sports
- X-Informatics Use Case: Health
- X-Informatics Use Case: Sensors
- X-Informatics Use Case: Radar for Remote Sensing.

Video: http://youtu.be/rT4thK_i5ig

2.2.2 Unit 1.2 - Course Motivation

Unit Overview

We motivate the study of X-informatics by describing data science and clouds. He starts with striking examples of the data deluge with examples from research, business and the consumer. The growing number of jobs in data science is highlighted. He describes industry trend in both clouds and big data.

He introduces the cloud computing model developed at amazing speed by industry. The 4 paradigms of scientific research are described with growing importance of data oriented version. He covers 3 major X-informatics areas: Physics, e-Commerce and Web Search followed by a broad discussion of cloud applications. Parallel computing in general and particular features of MapReduce are described. He comments on a data science education and the benefits of using MOOC's.

Slides

https://iu.box.com/s/muldo1qkcdlpdeiog3zo

Lesson 1 - Introduction

This presents the overview of talk, some trends in computing and data and jobs. Gartner's emerging technology hype cycle shows many areas of Clouds and Big Data. We highlight 6 issues of importance: economic imperative, computing model, research model, Opportunities in advancing computing, Opportunities in X-Informatics, Data Science Education

Video: http://youtu.be/kyJxstTivoI

Lesson 2: Data Deluge

We give some amazing statistics for total storage; uploaded video and uploaded photos; the social media interactions every minute; aspects of the business big data tidal wave; monitors of aircraft engines; the science research data sizes from particle physics to astronomy and earth science; genes sequenced; and finally the long tail of science. The next slide emphasizes applications using algorithms on clouds. This leads to the rallying cry "Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics educated in data science" with a catalog of the many values of X ''Astronomy, Biology, Biomedicine, Business, Chemistry, Climate, Crisis, Earth Science, Energy, Environment, Finance, Health, Intelligence, Lifestyle, Marketing, Medicine, Pathology, Policy, Radar, Security, Sensor, Social, Sustainability, Wealth and Wellness"

Video: http://youtu.be/sVNV0NxlQ6A

Lesson 3 - Jobs

Jobs abound in clouds and data science. There are documented shortages in data science, computer science and the major tech companies advertise for new talent.

Video: http://youtu.be/h9u7YeKkHHU

Lesson 4 - Industrial Trends

Trends include the growing importance of mobile devices and comparative decrease in desktop access, the export of internet content, the change in dominant client operating systems, use of social media, thriving Chinese internet companies.

Video: http://youtu.be/EIRIPDYN5nM

Lesson 5 - Digital Disruption of Old Favorites

Not everything goes up. The rise of the Internet has led to declines in some traditional areas including Shopping malls and Postal Services.

Video: http://youtu.be/RxGopRuMWOE

Lesson 6 - Computing Model: Industry adopted clouds which are attractive for data analytics

Clouds and Big Data are transformational on a 2-5 year time scale. Already Amazon AWS is a lucrative business with almost a $4B revenue. We describe the nature of cloud centers with economies of scale and gives examples of importance of virtualization in server consolidation. Then key characteristics of clouds are reviewed with expected high growth in Infrastructure, Platform and Software as a Service.

Video: http://youtu.be/NBZPQqXKbiw

Lesson 7 - Research Model: 4th Paradigm; From Theory to Data driven science?

We introduce the 4 paradigms of scientific research with the focus on the new fourth data driven methodology.

Video: http://youtu.be/2ke459BRBhw

Lesson 8 - Data Science Process

We introduce the DIKW data to information to knowledge to wisdom paradigm. Data flows through cloud services transforming itself and emerging as new information to input into other transformations.

Video: http://youtu.be/j9ytOaBoe2k

Lesson 9 - Physics-Informatics Looking for Higgs Particle with Large Hadron Collider LHC

We look at important particle physics example where the Large hadron Collider has observed the Higgs Boson. He shows this discovery as a bump in a histogram; something that so amazed him 50 years ago that he got a PhD in this field. He left field partly due to the incredible size of author lists on papers.

Video: http://youtu.be/qUB0q4AOavY

Lesson 10 - Recommender Systems I

Many important applications involve matching users, web pages, jobs, movies, books, events etc. These are all optimization problems with recommender systems one important way of performing this optimization. We go through the example of Netflix ~~ everything is a recommendation and muses about the power of viewing all sorts of things as items in a bag or more abstractly some space with funny properties.

Video: http://youtu.be/Aj5k0Sa7XGQ

Lesson 11 - Recommender Systems II

Continuation of Lesson 10 - Part 2

Video: http://youtu.be/VHS7il5OdjM

Lesson 12 - Web Search and Information Retrieval

This course also looks at Web Search and here we give an overview of the data analytics for web search, Pagerank as a method of ranking web pages returned and uses material from Yahoo on the subtle algorithms for dynamic personalized choice of material for web pages.

Video: http://youtu.be/i9gR9PdVXUU

Lesson 13 - Cloud Application in Research

We describe scientific applications and how they map onto clouds, supercomputers, grids and high throughput systems. He likes the cloud use of the Internet of Things and gives examples.

Video: http://youtu.be/C19-5WQH2TU

Lesson 14 - Parallel Computing and MapReduce

We define MapReduce and gives a homely example from fruit blending.

Video: http://youtu.be/BbW1PFNnKrE

Lesson 15 - Data Science Education

We discuss one reason you are taking this course ~~ Data Science as an educational initiative and aspects of its Indiana University implementation. Then general; features of online education are discussed with clear growth spearheaded by MOOC's where we use this course and others as an example. He stresses the choice between one class to 100,000 students or 2,000 classes to 50 students and an online library of MOOC lessons. In olden days he suggested ''hermit's cage virtual university" ~~ gurus in isolated caves putting together exciting curricula outside the traditional university model. Grading and mentoring models and important online tools are discussed. Clouds have MOOC's describing them and MOOC's are stored in clouds; a pleasing symmetry.

Video: http://youtu.be/x2LuiX8DYLs

Lesson 16 - Conclusions

The conclusions highlight clouds, data-intensive methodology, employment, data science, MOOC's and never forget the Big Data ecosystem in one sentence "Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics educated in data science"

Video: http://youtu.be/C0GszJg-MjE

Resources

- http://www.gartner.com/technology/home.jsp and many web links
- Meeker/Wu May 29 2013 Internet Trends D11 Conference http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013
- http://cs.metrostate.edu/~sbd/slides/Sun.pdf
- Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, Bill Franks Wiley ISBN: 978-1-118-20878-6
- Bill Ruh http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html

- http://www.genome.gov/sequencingcosts/
- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of Science October 2012 Dennis Gannon
- http://www.microsoft.com/en-us/news/features/2012/mar12/03-05CloudComputingJobs.aspx
- http://www.mckinsey.com/mgi/publications/big_data/index.asp
- Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- http://research.microsoft.com/en-us/people/barga/sc09_cloudcomp_tutorial.pdf
- http://research.microsoft.com/pubs/78813/AJ18_EN.pdf
- http://www.google.com/green/pdfs/google-green-computing.pdf
- http://www.wired.com/wired/issue/16-07
- http://research.microsoft.com/en-us/collaboration/fourthparadigm/
- Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf
- http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf
- http://www.interactions.org/cms/?pid=1032811
- http://www.quantumdiaries.org/2012/09/07/why-particle-detectors-need-a-trigger/atlasmgg/
- http://www.sciencedirect.com/science/article/pii/S037026931200857X
- http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-tutorial
- http://www.ifi.uzh.ch/ce/teaching/spring2012/16-Recommender-Systems_Slides.pdf
- http://en.wikipedia.org/wiki/PageRank
- http://pages.cs.wisc.edu/~beechung/icml11-tutorial/
- https://sites.google.com/site/opensourceiotcloud/
- http://datascience101.wordpress.com/2013/04/13/new-york-times-data-science-articles/
- http://blog.coursera.org/post/49750392396/on-the-topic-of-boredom
- http://x-informatics.appspot.com/course
- http://iucloudsummerschool.appspot.com/preview
- https://www.youtube.com/watch?v=M3jcSCA9_hM

2.3 Section 2 - Overview of Data Science: What is Big Data, Data Analytics and X-Informatics?

2.3.1 Section Overview

The course introduction starts with X-Informatics and its rallying cry. The growing number of jobs in data science is highlighted. The first unit offers a look at the phenomenon described as the Data Deluge starting with its broad features. Data science and the famous DIKW (Data to Information to Knowledge to Wisdom) pipeline are covered. Then more detail is given on the flood of data from Internet and Industry applications with eBay and General Electric discussed in most detail.

In the next unit, we continue the discussion of the data deluge with a focus on scientific research. He takes a first peek at data from the Large Hadron Collider considered later as physics Informatics and gives some biology examples.

He discusses the implication of data for the scientific method which is changing with the data-intensive methodology joining observation, theory and simulation as basic methods. Two broad classes of data are the long tail of sciences: many users with individually modest data adding up to a lot; and a myriad of Internet connected devices ~~ the Internet of Things.

We give an initial technical overview of cloud computing as pioneered by companies like Amazon, Google and Microsoft with new centers holding up to a million servers. The benefits of Clouds in terms of power consumption and the environment are also touched upon, followed by a list of the most critical features of Cloud computing with a comparison to supercomputing. Features of the data deluge are discussed with a salutary example where more data did better than more thought. Then comes Data science and one part of it ~~ data analytics ~~ the large algorithms that crunch the big data to give big wisdom. There are many ways to describe data science and several are discussed to give a good composite picture of this emerging field.

2.3.2 Unit 3 - Part I: Data Science generics and Commercial Data Deluge

Unit Overview

We start with X-Informatics and its rallying cry. The growing number of jobs in data science is highlighted. This unit offers a look at the phenomenon described as the Data Deluge starting with its broad features. Then he discusses data science and the famous DIKW (Data to Information to Knowledge to Wisdom) pipeline. Then more detail is given on the flood of data from Internet and Industry applications with eBay and General Electric discussed in most detail.

Slides

https://iu.box.com/s/rmnw3soy81kc82a5qzow

Lesson 1 - What is X-Informatics and its Motto

This discusses trends that are driven by and accompany Big data. We give some key terms including data, information, knowledge, wisdom, data analytics and data science. WE introduce the motto of the course: Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics. We list many values of X you can defined in various activities across the world.

Video: http://youtu.be/AKkyWF95Fp4

Lesson 2 - Jobs

Big data is especially important as there are some many related jobs. We illustrate this for both cloud computing and data science from reports by Microsoft and the McKinsey institute respectively. We show a plot from LinkedIn showing rapid increase in the number of data science and analytics jobs as a function of time.

Video: http://youtu.be/pRlfEigUJAc

Lesson 3 - Data Deluge ~~ General Structure

We look at some broad features of the data deluge starting with the size of data in various areas especially in science research. We give examples from real world of the importance of big data and illustrate how it is integrated into an enterprise IT architecture. We give some views as to what characterizes Big data and why data science is a science that is needed to interpret all the data.

Video: http://youtu.be/mPJ9twAFRQU

Lesson 4 - Data Science ~~ Process

We stress the DIKW pipeline: Data becomes information that becomes knowledge and then wisdom, policy and decisions. This pipeline is illustrated with Google maps and we show how complex the ecosystem of data, transformations (filters) and its derived forms is.

Video: http://youtu.be/ydH34L-z0Rk

Lesson 5 - Data Deluge ~~ Internet

We give examples of Big data from the Internet with Tweets, uploaded photos and an illustration of the vitality and size of many commodity applications.

Video: http://youtu.be/rtuq5y2Bx2g

Lesson 6 - Data Deluge ~~ Business I

We give examples including the Big data that enables wind farms, city transportation, telephone operations, machines with health monitors, the banking, manufacturing and retail industries both online and offline in shopping malls. We give examples from ebay showing how analytics allowing them to refine and improve the customer experiences.

Video: http://youtu.be/PJz38t6yn_s

Lesson 7 - Data Deluge ~~ Business II

Continuation of Lesson 6 - Part 2

Video: http://youtu.be/fESm-2Vox9M

Lesson 8 - Data Deluge ~~ Business III

Continuation of Lesson 6 - Part 3

Video: http://youtu.be/fcvn-IxPO00

Resources

- http://www.microsoft.com/en-us/news/features/2012/mar12/03-05CloudComputingJobs.aspx
- http://www.mckinsey.com/mgi/publications/big_data/index.asp
- Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Anjul Bhambhri http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf
- http://www.economist.com/node/15579717
- http://cs.metrostate.edu/~sbd/slides/Sun.pdf
- http://jess3.com/geosocial-universe-2/
- Bill Ruhhttp://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- http://www.hsph.harvard.edu/ncb2011/files/ncb2011-z03-rodriguez.pptx
- Hugh Williams http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html

2.3.3 Unit 4 - Part II: Data Deluge and Scientific Applications and Methodology

Unit Overview

We continue the discussion of the data deluge with a focus on scientific research. He takes a first peek at data from the Large Hadron Collider considered later as physics Informatics and gives some biology examples. He discusses the implication of data for the scientific method which is changing with the data-intensive methodology joining observation, theory and simulation as basic methods. We discuss the long tail of sciences; many users with individually modest data adding up to a lot. The last lesson emphasizes how everyday devices ~~ the Internet of Things ~~ are being used to create a wealth of data.

Slides

https://iu.box.com/s/e73lyv9sx7xcaqymb2n6

Lesson 1 - Science & Research I

We look into more big data examples with a focus on science and research. We give astronomy, genomics, radiology, particle physics and discovery of Higgs particle (Covered in more detail in later lessons), European Bioinformatics Institute and contrast to Facebook and Walmart.

Video: http://youtu.be/u1h6bAkuWQ8

Lesson 2 - Science & Research II

Continuation of Lesson 1 - Part 2

Video: http://youtu.be/_JfcUg2cheg

Lesson 3 - Implications for Scientific Method

We discuss the emergences of a new fourth methodology for scientific research based on data driven inquiry. We contrast this with third ~~ computation or simulation based discovery - methodology which emerged itself some 25 years ago.

Video: http://youtu.be/srEbOAmU_g8

Lesson 4 - Long Tail of Science

There is big science such as particle physics where a single experiment has 3000 people collaborate!.Then there are individual investigators who don't generate a lot of data each but together they add up to Big data.

Video: http://youtu.be/dwzEKEGYhqE

Lesson 5 - Internet of Things

A final category of Big data comes from the Internet of Things where lots of small devices ~~ smart phones, web cams, video games collect and disseminate data and are controlled and coordinated in the cloud.

Video: http://youtu.be/K2anbyxX48w

Resources

- http://www.economist.com/node/15579717
- Geoffrey Fox and Dennis Gannon Using Clouds for Technical Computing To be published in Proceedings of HPC 2012 Conference at Cetraro, Italy June 28 2012
- http://grids.ucs.indiana.edu/ptliupages/publications/Clouds_Technical_Computing_FoxGannonv2.pdf
- http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf
- http://www.genome.gov/sequencingcosts/
- http://www.quantumdiaries.org/2012/09/07/why-particle-detectors-need-a-trigger/atlasmgg
- http://salsahpc.indiana.edu/dlib/articles/00001935/
- http://en.wikipedia.org/wiki/Simple_linear_regression
- http://www.ebi.ac.uk/Information/Brochures/
- http://www.wired.com/wired/issue/16-07
- http://research.microsoft.com/en-us/collaboration/fourthparadigm/
- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of Science October 2012 Dennis Gannon https://sites.google.com/site/opensourceiotcloud/

2.3.4 Unit 5 - Part III: Clouds and Big Data Processing; Data Science Process and Analytics

Unit Overview

We give an initial technical overview of cloud computing as pioneered by companies like Amazon, Google and Microsoft with new centers holding up to a million servers. The benefits of Clouds in terms of power consumption and the environment are also touched upon, followed by a list of the most critical features of Cloud computing with a comparison to supercomputing.

He discusses features of the data deluge with a salutary example where more data did better than more thought. He introduces data science and one part of it ~~ data analytics ~~ the large algorithms that crunch the big data to give big wisdom. There are many ways to describe data science and several are discussed to give a good composite picture of this emerging field.

Slides

https://iu.box.com/s/38z9ryldgi3b8dgcbuan

Lesson 1 - Clouds

We describe cloud data centers with their staggering size with up to a million servers in a single data center and centers built modularly from shipping containers full of racks. The benefits of Clouds in terms of power consumption and the environment are also touched upon, followed by a list of the most critical features of Cloud computing and a comparison to supercomputing.

Video: http://youtu.be/8RBzooC_2Fw

Lesson 2 - Features of Data Deluge I

Data, Information, intelligence algorithms, infrastructure, data structure, semantics and knowledge are related. The semantic web and Big data are compared. We give an example where "More data usually beats better algorithms". We discuss examples of intelligent big data and list 8 different types of data deluge

Video: http://youtu.be/FMktnTQGyrw

Lesson 3 - Features of Data Deluge II

Continuation of Lesson 2 - Part 2

Video: http://youtu.be/QNVZobXHiZw

Lesson 4 - Data Science Process

We describe and critique one view of the work of a data scientists. Then we discuss and contrast 7 views of the process needed to speed data through the DIKW pipeline.

Note: You may find that some videos may have a different lesson, section or unit number. Please ignore this. In case the content does not correspond to the title, please let us know.

Video: http://youtu.be/lpQ-Q9ZidR4

Lesson 5 - Data Analytics I

We stress the importance of data analytics giving examples from several fields. We note that better analytics is as important as better computing and storage capability.

Video: http://youtu.be/RPVojR8jrb8

Lesson 6 - Data Analytics II

Continuation of Lesson 5 - Part 2

Link to the slide: http://archive2.cra.org/ccc/files/docs/nitrdsymposium/keyes.pdf

High Performance Computing in Science and Engineering: the Tree and the Fruit

Video: http://youtu.be/wOSgywqdJDY

2.3.5 Resources

- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of Science October 2012 Dennis Gannon
- Dan Reed Roger Barga Dennis Gannon Rich Wolskihttp://research.microsoft.com/en-us/people/barga/sc09_cloudcomp_tutorial.pdf
- http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pue-is-1-8/
- http://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data-centers-one-way-of-getting-it-just-right/

- http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf
- Bina Ramamurthy http://www.cse.buffalo.edu/~bina/cse487/fall2011/
- Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf
- Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf
- Anjul Bhambhri http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- http://cs.metrostate.edu/~sbd/slides/Sun.pdf
- Hugh Williams http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- http://www.mckinsey.com/mgi/publications/big_data/index.asp
- http://cra.org/ccc/docs/nitrdsymposium/pdfs/keyes.pdf

2.4 Section 3 - Health Informatics Case Study

2.4.1 Section Overview

This section starts by discussing general aspects of Big Data and Health including data sizes, different areas including genomics, EBI, radiology and the Quantified Self movement. We review current state of health care and trends associated with it including increased use of Telemedicine. We summarize an industry survey by GE and Accenture and an impressive exemplar Cloud-based medicine system from Potsdam. We give some details of big data in medicine. Some remarks on Cloud computing and Health focus on security and privacy issues.

We survey an April 2013 McKinsey report on the Big Data revolution in US health care; a Microsoft report in this area and a European Union report on how Big Data will allow patient centered care in the future. Examples are given of the Internet of Things, which will have great impact on health including wearables. A study looks at 4 scenarios for healthcare in 2032. Two are positive, one middle of the road and one negative. The final topic is Genomics, Proteomics and Information Visualization.

2.4.2 Unit 6 - X-Informatics Case Study: Health Informatics

Unit Overview

Slides:

https://iu.app.box.com/s/4v7omhmfpzd4y1bkpy9iab6o4jyephoa

This section starts by discussing general aspects of Big Data and Health including data sizes, different areas including genomics, EBI, radiology and the Quantified Self movement. We review current state of health care and trends associated with it including increased use of Telemedicine. We summarize an industry survey by GE and Accenture and an impressive exemplar Cloud-based medicine system from Potsdam. We give some details of big data in medicine. Some remarks on Cloud computing and Health focus on security and privacy issues.

We survey an April 2013 McKinsey report on the Big Data revolution in US health care; a Microsoft report in this area and a European Union report on how Big Data will allow patient centered care in the future. Examples are given of the Internet of Things, which will have great impact on health including wearables. A study looks at 4 scenarios for healthcare in 2032. Two are positive, one middle of the road and one negative. The final topic is Genomics, Proteomics and Information Visualization.

Lesson 1 - Big Data and Health

This lesson starts with general aspects of Big Data and Health including listing subareas where Big data important. Data sizes are given in radiology, genomics, personalized medicine, and the Quantified Self movement, with sizes and access to European Bioinformatics Institute.

Video: http://youtu.be/i7volfOVAmY

Lesson 2 - Status of Healthcare Today

This covers trends of costs and type of healthcare with low cost genomes and an aging population. Social media and government Brain initiative.

Video: http://youtu.be/tAT3pux4zeg

Lesson 3 - Telemedicine (Virtual Health)

This describes increasing use of telemedicine and how we tried and failed to do this in 1994.

Video: http://youtu.be/4JbGim9FFXg

Lesson 4 - Big Data and Healthcare Industry

Summary of an industry survey by GE and Accenture.

Video: http://youtu.be/wgK9JIUiWpQ

Lesson 5 - Medical Big Data in the Clouds

An impressive exemplar Cloud-based medicine system from Potsdam.

Video: http://youtu.be/-D9mEdM62uY

Lesson 6 - Medical image Big Data

Video: http://youtu.be/aaNplveyKf0

Lesson 7 - Clouds and Health

Video: http://youtu.be/9Whkl_UPS5g

Lesson 8 - McKinsey Report on the big-data revolution in US health care

This lesson covers 9 aspects of the McKinsey report. These are the convergence of multiple positive changes has created a tipping point for innovation; Primary data pools are at the heart of the big data revolution in healthcare; Big data is changing the paradigm: these are the value pathways; Applying early successes at scale could reduce US healthcare costs by $300 billion to $450 billion; Most new big-data applications target consumers and providers across pathways; Innovations are weighted towards influencing individual decision-making levers; Big data innovations use a range of public, acquired, and proprietary data types; Organizations implementing a big data transformation should provide the leadership required for the associated cultural transformation; Companies must develop a range of big data capabilities.

Video: http://youtu.be/bBoHzRjMEmY

Lesson 9 - Microsoft Report on Big Data in Health

This lesson identifies data sources as Clinical Data, Pharma & Life Science Data, Patient & Consumer Data, Claims & Cost Data and Correlational Data. Three approaches are Live data feed, Advanced analytics and Social analytics.

Video: http://youtu.be/PjffvVgj1PE

Lesson 10 - EU Report on Redesigning health in Europe for 2020

This lesson summarizes an EU Report on Redesigning health in Europe for 2020. The power of data is seen as a lever for change in My Data, My decisions; Liberate the data; Connect up everything; Revolutionize health; and Include Everyone removing the current correlation between health and wealth.

Video: http://youtu.be/9mbt_ZSs0iw

Lesson 11 - Medicine and the Internet of Things

The Internet of Things will have great impact on health including telemedicine and wearables. Examples are given.

Video: http://youtu.be/QGRfWlvw584

Lesson 12 - Extrapolating to 2032

A study looks at 4 scenarios for healthcare in 2032. Two are positive, one middle of the road and one negative.

Video: http://youtu.be/Qel4gmBxy8U

Lesson 13 - Genomics, Proteomics and Information Visualization I

A study of an Azure application with an Excel frontend and a cloud BLAST backend starts this lesson. This is followed by a big data analysis of personal genomics and an analysis of a typical DNA sequencing analytics pipeline. The Protein Sequence Universe is defined and used to motivate Multi dimensional Scaling MDS. Sammon's method is defined and its use illustrated by a metagenomics example. Subtleties in use of MDS include a monotonic mapping of the dissimilarity function. The application to the COG Proteomics dataset is discussed. We note that the MDS approach is related to the well known chisq method and some aspects of nonlinear minimization of chisq (Least Squares) are discussed.

Video: http://youtu.be/r1yENstaAUE

Lesson 14 - Genomics, Proteomics and Information Visualization II

This lesson continues the discussion of the COG Protein Universe introduced in the last lesson. It is shown how Proteomics clusters are clearly seen in the Universe browser. This motivates a side remark on different clustering methods applied to metagenomics. Then we discuss the Generative Topographic Map GTM method that can be used in dimension reduction when original data is in a metric space and is in this case faster than MDS as GTM computational complexity scales like N not N squared as seen in MDS.

Examples are given of GTM including an application to topic models in Information Retrieval. Indiana University has developed a deterministic annealing improvement of GTM. 3 separate clusterings are projected for visualization and show very different structure emphasizing the importance of visualizing results of data analytics. The final slide shows an application of MDS to generate and visualize phylogenetic trees.

Video: http://youtu.be/_F1Eo6bfN0w

Lesson 15 - Genomics, Proteomics and Information Visualization III

This lesson continues the discussion of the COG Protein Universe introduced in the last lesson. It is shown how Proteomics clusters are clearly seen in the Universe browser. This motivates a side remark on different clustering methods applied to metagenomics. Then we discuss the Generative Topographic Map GTM method that can be used in dimension reduction when original data is in a metric space and is in this case faster than MDS as GTM computational complexity scales like N not N squared as seen in MDS.

Examples are given of GTM including an application to topic models in Information Retrieval. Indiana University has developed a deterministic annealing improvement of GTM. 3 separate clusterings are projected for visualization and show very different structure emphasizing the importance of visualizing results of data analytics. The final slide shows an application of MDS to generate and visualize phylogenetic trees.

Video: http://youtu.be/R1svGGKipkc

Resources

- https://wiki.nci.nih.gov/display/CIP/CIP+Survey+of+Biomedical+Imaging+Archives
- http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf
- http://www.ieee-icsc.org/ICSC2010/Tony%20Hey%20-%2020100923.pdf
- http://quantifiedself.com/larry-smarr/
- http://www.ebi.ac.uk/Information/Brochures/
- http://www.kpcb.com/internet-trends
- http://www.slideshare.net/drsteventucker/wearable-health-fitness-trackers-and-the-quantified-self
- http://www.siam.org/meetings/sdm13/sun.pdf
- http://en.wikipedia.org/wiki/Calico_%28company%29
- http://www.slideshare.net/GSW_Worldwide/2015-health-trends
- http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Competitive-Landscape-Industries.pdf
- http://www.slideshare.net/schappy/how-realtime-analysis-turns-big-medical-data-into-precision-medicine
- http://medcitynews.com/2013/03/the-body-in-bytes-medical-images-as-a-source-of-healthcare-big-data-infographic/
- http://healthinformatics.wikispaces.com/file/view/cloud_computing.ppt
- http://www.mckinsey.com/~/media/McKinsey/dotcom/Insights/Health%20care/The%20big-data%20revolution%20in%20US%20health%20care/The%20big-data%20revolution%20in%20US%20health%20care%20Accel
- https://partner.microsoft.com/download/global/40193764
- http://ec.europa.eu/information_society/activities/health/docs/policy/taskforce/redesigning_health-eu-for2020-ehtf-report2012.pdf
- http://www.kpcb.com/internet-trends
- http://www.liveathos.com/apparel/app
- http://debategraph.org/Poster.aspx?aID=77
- http://www.oerc.ox.ac.uk/downloads/presentations-from-events/microsoftworkshop/gannon

- http://www.delsall.org
- http://salsahpc.indiana.edu/millionseq/mina/16SrRNA_index.html
- http://www.geatbx.com/docu/fcnindex-01.html
- https://wiki.nci.nih.gov/display/CIP/CIP+Survey+of+Biomedical+Imaging+Archives
- http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf
- http://www.ieee-icsc.org/ICSC2010/Tony%20Hey%20-%2020100923.pdf
- http://quantifiedself.com/larry-smarr/
- http://www.ebi.ac.uk/Information/Brochures/
- http://www.kpcb.com/internet-trends
- http://www.slideshare.net/drsteventucker/wearable-health-fitness-trackers-and-the-quantified-self
- http://www.siam.org/meetings/sdm13/sun.pdf
- http://en.wikipedia.org/wiki/Calico_%28company%29
- http://www.slideshare.net/GSW_Worldwide/2015-health-trends
- http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Competitive-Landscape-Industries.pdf
- http://www.slideshare.net/schappy/how-realtime-analysis-turns-big-medical-data-into-precision-medicine
- http://medcitynews.com/2013/03/the-body-in-bytes-medical-images-as-a-source-of-healthcare-big-data-infographic/
- http://healthinformatics.wikispaces.com/file/view/cloud_computing.ppt
- http://www.mckinsey.com/~/media/McKinsey/dotcom/Insights/Health%20care/The%20big-data%20revolution%20in%20US%20health%20care/The%20big-data%20revolution%20in%20US%20health%20care%20Accele
- https://partner.microsoft.com/download/global/40193764
- http://ec.europa.eu/information_society/activities/health/docs/policy/taskforce/redesigning_health-eu-for2020-ehtf-report2012.pdf
- http://www.kpcb.com/internet-trends
- http://www.liveathos.com/apparel/app
- http://debategraph.org/Poster.aspx?aID=77
- http://www.oerc.ox.ac.uk/downloads/presentations-from-events/microsoftworkshop/gannon
- http://www.delsall.org
- http://salsahpc.indiana.edu/millionseq/mina/16SrRNA_index.html
- http://www.geatbx.com/docu/fcnindex-01.html

Slides

- https://iu.app.box.com/s/4v7omhmfpzd4y1bkpy9iab6o4jyephoa

2.5 Section 4 - Sports Case Study

2.5.1 Section Overview

Sports sees significant growth in analytics with pervasive statistics shifting to more sophisticated measures. We start with baseball as game is built around segments dominated by individuals where detailed (video/image) achievement measures including PITCHf/x and FIELDf/x are moving field into big data arena. There are interesting relationships between the economics of sports and big data analytics. We look at Wearables and consumer sports/recreation. The importance of spatial visualization is discussed. We look at other Sports: Soccer, Olympics, NFL Football, Basketball, Tennis and Horse Racing.

2.5.2 Unit 7 - Sports Informatics I : Sabermetrics (Basic)

Unit Overview

This unit discusses baseball starting with the movie Moneyball and the 2002-2003 Oakland Athletics. Unlike sports like basketball and soccer, most baseball action is built around individuals often interacting in pairs. This is much easier to quantify than many player phenomena in other sports. We discuss Performance-Dollar relationship including new stadiums and media/advertising. We look at classic baseball averages and sophisticated measures like Wins Above Replacement.

Slides

https://iu.box.com/s/trsxko7icktb7htqfickfsws0cqmvt2j

Lesson 1 - Introduction and Sabermetrics (Baseball Informatics) Lesson

Introduction to all Sports Informatics, Moneyball The 2002-2003 Oakland Athletics, Diamond Dollars economic model of baseball, Performance - Dollar relationship, Value of a Win.

Video: http://youtu.be/oviNJ-_fLto

Lesson 2 - Basic Sabermetrics

Different Types of Baseball Data, Sabermetrics, Overview of all data, Details of some statistics based on basic data, OPS, wOBA, ERA, ERC, FIP, UZR.

Video: http://youtu.be/-5JYfQXC2ew

Lesson 3 - Wins Above Replacement

Wins above Replacement WAR, Discussion of Calculation, Examples, Comparisons of different methods, Coefficient of Determination, Another, Sabermetrics Example, Summary of Sabermetrics.

Video: http://youtu.be/V5uzUS6jdHw

Resources

- http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-powered-storytelling
- http://www.sloansportsconference.com/
- http://sabr.org/
- http://en.wikipedia.org/wiki/Sabermetrics
- http://en.wikipedia.org/wiki/Baseball_statistics
- http://www.sportvision.com/baseball
- http://m.mlb.com/news/article/68514514/mlbam-introduces-new-way-to-analyze-every-play
- http://www.fangraphs.com/library/offense/offensive-statistics-list/
- http://en.wikipedia.org/wiki/Component_ERA
- http://www.fangraphs.com/library/pitching/fip/
- http://nomaas.org/2012/05/a-look-at-the-defense-the-yankees-d-stinks-edition/
- http://en.wikipedia.org/wiki/Wins_Above_Replacement
- http://www.fangraphs.com/library/misc/war/
- http://www.baseball-reference.com/about/war_explained.shtml
- http://www.baseball-reference.com/about/war_explained_comparison.shtml
- http://www.baseball-reference.com/about/war_explained_position.shtml
- http://www.baseball-reference.com/about/war_explained_pitch.shtml
- http://www.fangraphs.com/leaders.aspx?pos=all&stats=bat&lg=all&qual=y&type=8&season=2014&month=0&season1=1871&i
- http://battingleadoff.com/2014/01/08/comparing-the-three-war-measures-part-ii/
- http://battingleadoff.com/2014/01/08/comparing-the-three-war-measures-part-ii/
- http://en.wikipedia.org/wiki/Coefficient_of_determination
- http://www.sloansportsconference.com/wp-content/uploads/2014/02/2014_SSAC_Data-driven-Method-for-In-game-Decision-Making.pdf
- https://courses.edx.org/courses/BUx/SABR101x/2T2014/courseware/10e616fc7649469ab4457ae18df92b20/

2.5.3 Unit 8 - Sports Informatics II : Sabermetrics (Advanced)

Unit Overview

This unit discusses 'advanced sabermetrics' covering advances possible from using video from PITCHf/X, FIELDf/X, HITf/X, COMMANDf/X and MLBAM.

Slides

https://iu.box.com/s/o2kikemoh2580ohzt2pn3y3jps4f7wr3

Lesson 1 - Pitching Clustering

A Big Data Pitcher Clustering method introduced by Vince Gennaro, Data from Blog and video at 2013 SABR conference.

Video: http://youtu.be/I06_AOKyB20

Lesson 2 - Pitcher Quality

Results of optimizing match ups, Data from video at 2013 SABR conference.

Video: http://youtu.be/vAPJx8as4_0

Lesson 3 - PITCHf/X

Examples of use of PITCHf/X.

Video: http://youtu.be/JN1-sCa9Bjs

Lesson 4 - Other Video Data Gathering in Baseball

FIELDf/X, MLBAM, HITf/X, COMMANDf/X.

Video: http://youtu.be/zGGThkkIJg8

Resources

- http://vincegennaro.mlblogs.com/
- https://www.youtube.com/watch?v=H-kx-x_d0Mk
- http://www.sportvision.com/media/pitchfx-how-it-works
- http://www.baseballprospectus.com/article.php?articleid=13109
- http://baseball.physics.illinois.edu/FastPFXGuide.pdf
- http://baseball.physics.illinois.edu/FieldFX-TDR-GregR.pdf
- http://www.sportvision.com/baseball/fieldfx
- http://regressing.deadspin.com/mlb-announces-revolutionary-new-fielding-tracking-syste-1534200504
- http://grantland.com/the-triangle/mlb-advanced-media-play-tracking-bob-bowman-interview/
- http://www.sportvision.com/baseball/hitfx
- https://www.youtube.com/watch?v=YkjtnuNmK74

2.5.4 Unit 9 - Sports Informatics III : Other Sports

Unit Overview

We look at Wearables and consumer sports/recreation. The importance of spatial visualization is discussed. We look at other Sports: Soccer, Olympics, NFL Football, Basketball, Tennis and Horse Racing.

Slides

https://iu.box.com/s/ho0ktliih8cj0oyl929axwwu6083e8ck

Lesson 1 - Wearables

Consumer Sports, Stake Holders, and Multiple Factors.

Video: http://youtu.be/1UzvNHzFCFQ

Lesson 2 - Soccer and the Olympics

Soccer, Tracking Players and Balls, Olympics.

Video: http://youtu.be/01mlZ2KBkzE

Lesson 3 - Spatial Visualization in NFL and NBA

NFL, NBA, and Spatial Visualization.

Video: http://youtu.be/Q0Pt97BwRlo

Lesson 4 - Tennis and Horse Racing

Tennis, Horse Racing, and Continued Emphasis on Spatial Visualization.

Video: http://youtu.be/EuXrtfHG3cY

Resources

- http://www.sloansportsconference.com/?page_id=481&sort_cate=Research%20Paper
- http://www.slideshare.net/Tricon_Infotech/big-data-for-big-sports
- http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-powered-storytelling
- http://www.liveathos.com/apparel/app
- http://www.slideshare.net/elew/sport-analytics-innovation
- http://www.wired.com/2013/02/catapault-smartball/
- http://www.sloansportsconference.com/wp-content/uploads/2014/06/Automated_Playbook_Generation.pdf
- http://autoscout.adsc.illinois.edu/publications/football-trajectory-dataset/
- http://www.sloansportsconference.com/wp-content/uploads/2012/02/Goldsberry_Sloan_Submission.pdf
- http://gamesetmap.com/
- http://www.trakus.com/technology.asp#tNetText

2.6 Section 5 - Technology Training - Python & FutureSystems (will be updated)

2.6.1 Section Overview

This section is meant to give an overview of the python tools needed for doing for this course.

These are really powerful tools which every data scientist who wishes to use python must know.

NumPy - It is popular library on top of which many other libraries (like pandas, scipy) are built. It provides a way a vectorizing data. This helps to organize in a more intuitive fashion and also helps us use the various matrix operations which are popularly used by the machine learning community. Matplotlib: This a data visualization package. It allows you to create graphs charts and other such diagrams. It supports Images in JPEG, GIF, TIFF format. SciPy: SciPy is a library built above numpy and has a number of off the shelf algorithms / operations implemented. These include algorithms from calculus(like integration), statistics, linear algebra, image-processing, signal processing, machine learning, etc.

2.6.2 Unit 10 - Python for Big Data and X-Informatics: NumPy, SciPy, MatPlotlib

Unit Overview

This section is meant to give an overview of the python tools needed for doing for this course. These are really powerful tools which every data scientist who wishes to use python must know.

Lesson 1 - Introduction

This section is meant to give an overview of the python tools needed for doing for this course. These are really powerful tools which every data scientist who wishes to use python must know. This section covers NumPy, MatPlotLib, and Scipy.

Pycharm

is an Integrated Development Environment (IDE) used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with git.

Video: https://youtu.be/X8ZpbZweJcw

Python in 45 minutes

Here is an introductory video about the Python programming language that we found on the internet. Naturally there are many alternatives to this video, but the video is probably a good start. It also uses PyCharm which we recommend.

https://www.youtube.com/watch?v=N4mEzFDjqtA

How much you want to understand of python is actually a bit up to your, while its goot to know classes and inheritance, you may be able for this class to get away without using it. However, we do recommend that you learn it.

Lesson 3 - Numpy 1

NumPy - It is popular library on top of which many other libraries (like pandas, scipy) are built. It provides a way a vectorizing data. This helps to organize in a more intuitive fashion and also helps us use the various matrix operations which are popularly used by the machine learning community.

Video: http://youtu.be/mN_JpGO9Y6s

Lesson 4 - Numpy 2

Continuation of Lesson 3 - Part 2

Video: http://youtu.be/7QfW7AT7UNU

Lesson 5 - Numpy 3

Continuation of Lesson 3 - Part 3

Video: http://youtu.be/Ccb67Q5gpsk

Lesson 6 - Matplotlib 1

Matplotlib: This a data visualization package. It allows you to create graphs charts and other such diagrams. It supports Images in JPEG, GIF, TIFF format.

Video: http://youtu.be/3UOvB5OmtYE

Lesson 7 - Matplotlib 2

Continuation of Lesson 6 - Part 2

Video: http://youtu.be/9ONSnsN4hcg

Lesson 8 - Scipy 1

SciPy: SciPy is a library built above numpy and has a number of off the shelf algorithms / operations implemented. These include algorithms from calculus(like integration), statistics, linear algebra, image-processing, signal processing, machine learning, etc.

Video: http://youtu.be/lpC6Mn-09jY

Lesson 9 - Scipy 2

Continuation of Lesson 8 - Part 2

Video: http://youtu.be/-XKBz7qCUqw

2.6.3 Unit 11 - Using FutureSystems (Please do not do yet)

Unit Overview

This section is meant to give an overview of the FutureSystems and how to use for the Big Data Course. In addition to this creating FutureSystems Account, Uploading OpenId and SSH Key and how to instantiate and log into Virtual Machine and accessing Ipython are covered. In the end we discuss about running Python and Java on Virtual Machine.

Lesson 1 - FutureSystems Overview

In this video we introduce FutureSystems in terms of its services and features.

FirstProgram.java: http://openedx.scholargrid.org:18010/c4x/SoIC/INFO-I-523/asset/FirstProgram.java

Video: http://youtu.be/RibpNSyd4qg

Lesson 2 - Creating Portal Account

This lesson explains how to create a portal account, which is the first step in gaining access to FutureSystems.

See Lesson 4 and 7 for SSH key generation on Linux, OSX or Windows.

Video: http://youtu.be/X6zeVEALzTk

Lesson 3 - Upload an OpenId

This lesson explains how to upload and use OpenID to easily log into the FutureSystems portal.

Video: http://youtu.be/rZzpCYWDEpI

Lesson 4 - SSH Key Generation using ssh-keygen command

SSH keys are used to identify user accounts in most systems including FutureSystems. This lesson walks you through generating an SSH key via ssh-keygen command line tool.

Video: http://youtu.be/pQb2VV1zNIc

Lesson 5 - Shell Access via SSH

This lesson explains how to get access FutureSystems resources vis SSH terminal with your registered SSH key.

Video: http://youtu.be/aJDXfvOrzRE

Lesson 6 - Advanced SSH

This lesson shows you how to write SSH 'config' file in advanced settings.

Video: http://youtu.be/eYanElmtqMo

Lesson 7 - SSH Key Generation via putty (Windows user only)

This lesson is for Windows users.

You will learn how to create an SSH key using PuTTYgen, add the public key to you FutureSystems portal, and then login using the PuTTY SSH client.

Video: http://youtu.be/irmVJKwWQCU

Lesson 8 - Using FS - Creating VM using Cloudmesh and running IPython

This lesson explains how to log into FutureSystems and our customized shell and menu options that will simplify management of the VMs for this upcoming lessons.

Instruction is at: http://cloudmesh.github.io/introduction_to_cloud_computing/class/cm-mooc/cm-mooc.html

Video: http://youtu.be/nbZbJxheLwc

2.7 Section 6 - Physics Case Study

2.7.1 Section Overview

This section starts by describing the LHC accelerator at CERN and evidence found by the experiments suggesting existence of a Higgs Boson. The huge number of authors on a paper, remarks on histograms and Feynman diagrams is followed by an accelerator picture gallery. The next unit is devoted to Python experiments looking at histograms of Higgs Boson production with various forms of shape of signal and various background and with various event totals. Then random variables and some simple principles of statistics are introduced with explanation as to why they are relevant to Physics counting experiments. The unit introduces Gaussian (normal) distributions and explains why they seen so often in natural phenomena. Several Python illustrations are given. Random Numbers with their Generators and Seeds lead to a discussion of Binomial and Poisson Distribution. Monte-Carlo and accept-reject methods. The Central Limit Theorem concludes discussion.

2.7.2 Unit 12 - I: Looking for Higgs Particles, Bumps in Histograms, Experiments and Accelerators

Unit Overview

This unit is devoted to Python and Java experiments looking at histograms of Higgs Boson production with various forms of shape of signal and various background and with various event totals. The lectures use Python but use of Java is described.

Slides

https://iu.app.box.com/s/6uz4ofnnd9usv75cab71

Files

- `HiggsClassI-Sloping.py`

Lesson 1 - Looking for Higgs Particle and Counting Introduction I

We return to particle case with slides used in introduction and stress that particles often manifested as bumps in histograms and those bumps need to be large enough to stand out from background in a statistically significant fashion.

Video: http://youtu.be/VQAupoFUWTg

Lesson 2 - Looking for Higgs Particle and Counting Introduction II

We give a few details on one LHC experiment ATLAS. Experimental physics papers have a staggering number of authors and quite big budgets. Feynman diagrams describe processes in a fundamental fashion.

Video: http://youtu.be/UAMzmOgjj7I

Lesson 3 - Physics-Informatics Looking for Higgs Particle Experiments

We give a few details on one LHC experiment ATLAS. Experimental physics papers have a staggering number of authors and quite big budgets. Feynman diagrams describe processes in a fundamental fashion.

Video: http://youtu.be/BW12d780qT8

Lesson 4 - Accelerator Picture Gallery of Big Science

This lesson gives a small picture gallery of accelerators. Accelerators, detection chambers and magnets in tunnels and a large underground laboratory used fpr experiments where you need to be shielded from background like cosmic rays.

Video: http://youtu.be/WLJIxWWMYi8

Resources

- http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf
- http://www.interactions.org/cms/?pid=6002
- http://www.interactions.org/cms/?pid=1032811
- http://www.sciencedirect.com/science/article/pii/S037026931200857X
- http://biologos.org/blog/what-is-the-higgs-boson
- http://www.atlas.ch/pdf/ATLAS_fact_sheets.pdf
- http://www.nature.com/news/specials/lhc/interactive.html

2.7.3 Unit 13 - II: Looking for Higgs Particles: Python Event Counting for Signal and Background

Unit Overview

This unit is devoted to Python experiments looking at histograms of Higgs Boson production with various forms of shape of signal and various background and with various event totals.

Slides

https://iu.app.box.com/s/77iw9brrugz2pjoq6fw1

Files

- `HiggsClassI-Sloping.py`
- `HiggsClassIII.py`
- `HiggsClassIIUniform.py`

Lesson 1 - Physics Use Case II 1: Class Software

We discuss how this unit uses Java and Python on both a backend server (FutureGrid) or a local client. WE point out useful book on Python for data analysis. This builds on technology training in Section 3.

Video: http://youtu.be/tOFJEUM-Vww

Lesson 2 - Physics Use Case II 2: Event Counting

We define ''event counting" data collection environments. We discuss the python and Java code to generate events according to a particular scenario (the important idea of Monte Carlo data). Here a sloping background plus either a Higgs particle generated similarly to LHC observation or one observed with better resolution (smaller measurement error).

Video: http://youtu.be/h8-szCeFugQ

Lesson 3 - Physics Use Case II 3: With Python examples of Signal plus Background

This uses Monte Carlo data both to generate data like the experimental observations and explore effect of changing amount of data and changing measurement resolution for Higgs.

Video: http://youtu.be/bl2f0tAzLj4

Lesson 4 - Physics Use Case II 4: Change shape of background & num of Higgs Particles

This lesson continues the examination of Monte Carlo data looking at effect of change in number of Higgs particles produced and in change in shape of background.

Video: http://youtu.be/bw3fd5cfQhk

Resources

- Python for Data Analysis: Agile Tools for Real World Data By Wes McKinney, Publisher: O'Reilly Media, Released: October 2012, Pages: 472.
- http://jwork.org/scavis/api/
- https://en.wikipedia.org/wiki/DataMelt

2.7.4 Unit 14 - III: Looking for Higgs Particles: Random Variables, Physics and Normal Distributions

Unit Overview

We introduce random variables and some simple principles of statistics and explains why they are relevant to Physics counting experiments. The unit introduces Gaussian (normal) distributions and explains why they seen so often in natural phenomena. Several Python illustrations are given. Java is currently not available in this unit.

Slides

https://iu.app.box.com/s/bcyze7h8knj6kvhyr05y

`HiggsClassIII.py`

Lesson 1 - Statistics Overview and Fundamental Idea: Random Variables

We go through the many different areas of statistics covered in the Physics unit. We define the statistics concept of a random variable.

Video: http://youtu.be/0oZzALLzYBM

Lesson 2 - Physics and Random Variables I

We describe the DIKW pipeline for the analysis of this type of physics experiment and go through details of analysis pipeline for the LHC ATLAS experiment. We give examples of event displays showing the final state particles seen in a few events. We illustrate how physicists decide whats going on with a plot of expected Higgs production experimental cross sections (probabilities) for signal and background.

Video: http://youtu.be/Tn3GBxgplxg

Lesson 3 - Physics and Random Variables II

We describe the DIKW pipeline for the analysis of this type of physics experiment and go through details of analysis pipeline for the LHC ATLAS experiment. We give examples of event displays showing the final state particles seen in a few events. We illustrate how physicists decide whats going on with a plot of expected Higgs production experimental cross sections (probabilities) for signal and background.

Video: http://youtu.be/qWEjp0OtvdA

Lesson 4 - Statistics of Events with Normal Distributions

We introduce Poisson and Binomial distributions and define independent identically distributed (IID) random variables. We give the law of large numbers defining the errors in counting and leading to Gaussian distributions for many things. We demonstrate this in Python experiments.

Video: http://youtu.be/LMBtpWOOQLo

Lesson 5 - Gaussian Distributions

We introduce the Gaussian distribution and give Python examples of the fluctuations in counting Gaussian distributions.

Video: http://youtu.be/LWIbPa-P5W0

Lesson 6 - Using Statistics

We discuss the significance of a standard deviation and role of biases and insufficient statistics with a Python example in getting incorrect answers.

Video: http://youtu.be/n4jlUrGwgic

Resources

- http://indico.cern.ch/event/20453/session/6/contribution/15?materialId=slides
- http://www.atlas.ch/photos/events.html
- http://cms.web.cern.ch/

2.7.5 Unit 15 - IV: Looking for Higgs Particles: Random Numbers, Distributions and Central Limit Theorem

Unit Overview

We discuss Random Numbers with their Generators and Seeds. It introduces Binomial and Poisson Distribution. Monte-Carlo and accept-reject methods are discussed. The Central Limit Theorem and Bayes law concludes discussion. Python and Java (for student - not reviewed in class) examples and Physics applications are given.

Slides

https://iu.app.box.com/s/me7738igixwzc9h9qwe1

Files

- `HiggsClassIII.py`

Lesson 1 - Generators and Seeds I

We define random numbers and describe how to generate them on the computer giving Python examples. We define the seed used to define to specify how to start generation.

Video: http://youtu.be/76jbRphjRWo

Lesson 2 - Generators and Seeds II

We define random numbers and describe how to generate them on the computer giving Python examples. We define the seed used to define to specify how to start generation.

Video: http://youtu.be/9QY5qkQj2Ag

Lesson 3 - Binomial Distribution

We define binomial distribution and give LHC data as an example of where this distribution valid.

Video: http://youtu.be/DPd-eVI_twQ

Lesson 4 - Accept-Reject

We introduce an advanced method ~~ accept/reject ~~ for generating random variables with arbitrary distributions.

Video: http://youtu.be/GfshkKMKCj8

Lesson 5 - Monte Carlo Method

We define Monte Carlo method which usually uses accept/reject method in typical case for distribution.

Video: http://youtu.be/kIQ-BTyDfOQ

Lesson 6 - Poisson Distribution

We extend the Binomial to the Poisson distribution and give a set of amusing examples from Wikipedia.

Video: http://youtu.be/WFvgsVo-k4s

Lesson 7 - Central Limit Theorem

We introduce Central Limit Theorem and give examples from Wikipedia.

Video: http://youtu.be/ZO53iKlPn7c

Lesson 8 - Interpretation of Probability: Bayes v. Frequency

This lesson describes difference between Bayes and frequency views of probability. Bayes's law of conditional probability is derived and applied to Higgs example to enable information about Higgs from multiple channels and multiple experiments to be accumulated.

Video: http://youtu.be/jzDkExAQI9M

Resources

- https://en.wikipedia.org/wiki/Pseudorandom_number_generator
- https://en.wikipedia.org/wiki/Mersenne_Twister
- https://en.wikipedia.org/wiki/Mersenne_prime
- CMS-PAS-HIG-12-041 Updated results on the new boson discovered in the search for the standard model Higgs boson in the ZZ to 4 leptons channel in pp collisions at sqrt(s) = 7 and 8 TeV http://cds.cern.ch/record/1494488?ln=en
- https://en.wikipedia.org/wiki/Poisson_distribution
- https://en.wikipedia.org/wiki/Central_limit_theorem
- http://jwork.org/scavis/api/

- https://en.wikipedia.org/wiki/DataMelt

2.8 Section 7 - Big Data Use Cases Survey

2.8.1 Section Overview

This section covers 51 values of X and an overall study of Big data that emerged from a NIST (National Institute for Standards and Technology) study of Big data. The section covers the NIST Big Data Public Working Group (NBD-PWG) Process and summarizes the work of five subgroups: Definitions and Taxonomies Subgroup, Reference Architecture Subgroup, Security and Privacy Subgroup, Technology Roadmap Subgroup and the Requirements andUse Case Subgroup. 51 use cases collected in this process are briefly discussed with a classification of the source of parallelism and the high and low level computational structure. We describe the key features of this classification.

2.8.2 Unit 16 - Overview of NIST Big Data Public Working Group (NBD-PWG) Process and Results

Unit Overview

This unit covers the NIST Big Data Public Working Group (NBD-PWG) Process and summarizes the work of five subgroups: Definitions and Taxonomies Subgroup, Reference Architecture Subgroup, Security and Privacy Subgroup, Technology Roadmap Subgroup and the Requirements and Use Case Subgroup. The work of latter is continued in next two units.

Slides

https://iu.app.box.com/s/bgr7lyaz7uazcarangqd

Lesson 1 - Introduction to NIST Big Data Public Working Group (NBD-PWG) Process

The focus of the (NBD-PWG) is to form a community of interest from industry, academia, and government, with the goal of developing a consensus definitions, taxonomies, secure reference architectures, and technology roadmap. The aim is to create vendor-neutral, technology and infrastructure agnostic deliverables to enable big data stakeholders to pick-and-choose best analytics tools for their processing and visualization requirements on the most suitable computing platforms and clusters while allowing value-added from big data service providers and flow of data between the stakeholders in a cohesive and secure manner.

Video: http://youtu.be/ofRfHBKpyvg

Lesson 2 - Definitions and Taxonomies Subgroup

The focus is to gain a better understanding of the principles of Big Data. It is important to develop a consensus-based common language and vocabulary terms used in Big Data across stakeholders from industry, academia, and government. In addition, it is also critical to identify essential actors with roles and responsibility, and subdivide them into components and sub-components on how they interact/ relate with each other according to their similarities and differences.

For Definitions: Compile terms used from all stakeholders regarding the meaning of Big Data from various standard bodies, domain applications, and diversified operational environments. For Taxonomies: Identify key actors with their roles and responsibilities from all stakeholders, categorize them into components and subcomponents based on their similarities and differences. In particular data Science and Big Data terms are discussed.

Video: http://youtu.be/sGshHN-DdbE

Lesson 3 - Reference Architecture Subgroup

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus-based approach to orchestrate vendor-neutral, technology and infrastructure agnostic for analytics tools and computing environments. The goal is to enable Big Data stakeholders to pick-and-choose technology-agnostic analytics tools for processing and visualization in any computing platform and cluster while allowing value-added from Big Data service providers and the flow of the data between the stakeholders in a cohesive and secure manner. Results include a reference architecture with well defined components and linkage as well as several exemplars.

Video: http://youtu.be/JV596ZH36YA

Lesson 4 - Security and Privacy Subgroup

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus secure reference architecture to handle security and privacy issues across all stakeholders. This includes gaining an understanding of what standards are available or under development, as well as identifies which key organizations are working on these standards. The Top Ten Big Data Security and Privacy Challenges from the CSA (Cloud Security Alliance) BDWG are studied. Specialized use cases include Retail/Marketing, Modern Day Consumerism, Nielsen Homescan, Web Traffic Analysis, Healthcare, Health Information Exchange, Genetic Privacy, Pharma Clinical Trial Data Sharing, Cyber-security, Government, Military and Education.

Video: http://youtu.be/Gbk0LaWE3lM

Lesson 5 - Technology Roadmap Subgroup

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus vision with recommendations on how Big Data should move forward by performing a good gap analysis through the materials gathered from all other NBD subgroups. This includes setting standardization and adoption priorities through an understanding of what standards are available or under development as part of the recommendations. Tasks are gather input from NBD subgroups and study the taxonomies for the actors' roles and responsibility, use cases and requirements, and secure reference architecture; gain understanding of what standards are available or under development for Big Data; perform a thorough gap analysis and document the findings; identify what possible barriers may delay or prevent adoption of Big Data; and document vision and recommendations.

Video: http://youtu.be/GCc9yfErmd0

Lesson 6 - Requirements and Use Case Subgroup Introduction I

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus list of Big Data requirements across all stakeholders. This includes gathering and understanding various use cases from diversified application domains.Tasks are gather use case input from all stakeholders; derive Big Data requirements from each use case; analyze/prioritize a list of challenging general requirements that may delay or prevent adoption of Big Data deployment; develop a set of general patterns capturing the ‘’essence” of use cases (not done yet) and work with Reference Architecture to validate requirements and reference architecture by explicitly implementing some patterns based on use cases. The progress of gathering use cases (discussed in next two units) and requirements systemization are discussed.

Video: http://youtu.be/sztqNXJ9P6c

Lesson 7 - Requirements and Use Case Subgroup Introduction II

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus list of Big Data requirements across all stakeholders. This includes gathering and understanding various use cases from diversified application domains.Tasks are gather use case input from all stakeholders; derive Big Data requirements from each use case; analyze/prioritize a list of challenging general requirements that may delay or prevent adoption of Big Data deployment; develop a set of general patterns capturing the ''essence" of use cases (not done yet) and work with Reference Architecture to validate requirements and reference architecture by explicitly implementing some patterns based on use cases. The progress of gathering use cases (discussed in next two units) and requirements systemization are discussed.

Video: http://youtu.be/0sbfIqHUauI

Lesson 8 - Requirements and Use Case Subgroup Introduction III

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus list of Big Data requirements across all stakeholders. This includes gathering and understanding various use cases from diversified application domains.Tasks are gather use case input from all stakeholders; derive Big Data requirements from each use case; analyze/prioritize a list of challenging general requirements that may delay or prevent adoption of Big Data deployment; develop a set of general patterns capturing the ''essence" of use cases (not done yet) and work with Reference Architecture to validate requirements and reference architecture by explicitly implementing some patterns based on use cases. The progress of gathering use cases (discussed in next two units) and requirements systemization are discussed.

Video: http://youtu.be/u59559nqjiY

Resources

- NIST Big Data Public Working Group (NBD-PWG) Process https://www.nist.gov/el/cyber-physical-systems/big-data-pwg
- Big Data Definitions: http://dx.doi.org/10.6028/NIST.SP.1500-1 (link is external)
- Big Data Taxonomies: http://dx.doi.org/10.6028/NIST.SP.1500-2 (link is external)
- Big Data Use Cases and Requirements: http://dx.doi.org/10.6028/NIST.SP.1500-3 (link is external)
- Big Data Security and Privacy: http://dx.doi.org/10.6028/NIST.SP.1500-4 (link is external)
- Big Data Architecture White Paper Survey: http://dx.doi.org/10.6028/NIST.SP.1500-5 (link is external)
- Big Data Reference Architecture: http://dx.doi.org/10.6028/NIST.SP.1500-6 (link is external)
- Big Data Standards Roadmap: http://dx.doi.org/10.6028/NIST.SP.1500-7 (link is external)

Some of the links bellow may be outdated. Please let us know the new links and notify us of the outdated links.

- DCGSA Standard Cloud: https://www.youtube.com/watch?v=l4Qii7T8zeg
- On line 51 Use Cases http://bigdatawg.nist.gov/usecases.php
- Summary of Requirements Subgroup http://bigdatawg.nist.gov/_uploadfiles/M0245_v5_6066621242.docx
- Use Case 6 Mendeley http://mendeley.com%20http//dev.mendeley.com
- Use Case 7 Netflix http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-tutoria

- Use Case 8 Search http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013, http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html, http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws, http://www.slideshare.net/beechung/recommender-systems-tutorialpart1intro, http://www.worldwidewebsize.com/
- Use Case 9 IaaS (Infrastructure as a Service) Big Data Business Continuity & Disaster Recovery (BC/DR) Within A Cloud Eco-System provided by Cloud Service Providers (CSPs) and Cloud Brokerage Service Providers (CBSPs) http://www.disasterrecovery.org/
- Use Case 11 and Use Case 12 Simulation driven Materials Genomics https://www.materialsproject.org/
- Use Case 13 Large Scale Geospatial Analysis and Visualization http://www.opengeospatial.org/standards, http://geojson.org/ , http://earth-info.nga.mil/publications/specs/printed/CADRG/cadrg.html
- Use Case 14 Object identification and tracking from Wide Area Large Format Imagery (WALF) Imagery or Full Motion Video (FMV) - Persistent Surveillance http://www.militaryaerospace.com/topics/m/video/79088650/persistent-surveillance-relies-on-extracting-relevant-data-points-and-connecting-the-dots.htm, http://www.defencetalk.com/wide-area-persistent-surveillance-revolutionizes-tactical-isr-45745/
- Use Case 15 Intelligence Data Processing and Analysis http://www.afcea-aberdeen.org/files/presentations/AFCEAAberdeen_DCGSA_COLWells_PS.pdf, http://stids.c4i.gmu.edu/papers/STIDSPapers/STIDS2012_T14_SmithEtAl_HorizontalIntegrationOfWarfighterIntel.pdf, http://stids.c4i.gmu.edu/STIDS2011/papers/STIDS2011_CR_T1_SalmenEtAl.pdf, https://www.youtube.com/watch?v=l4Qii7T8zeg, http://dcgsa.apg.army.mil/
- Use Case 16 Electronic Medical Record (EMR) Data: Regenstrief Institute , Logical observation identifiers names and codes , Indiana Health Information Exchange , Institute of Medicine Learning Healthcare System
- Use Case 17 Pathology Imaging/digital pathology; https://web.cci.emory.edu/confluence/display/PAIS , https://web.cci.emory.edu/
- Use Case 19 Genome in a Bottle Consortium: www.genomeinabottle.org
- Use Case 20 Comparative analysis for metagenomes and genomes http://img.jgi.doe.gov/
- Use Case 25 Biodiversity and LifeWatch
- Use Case 26 Deep Learning: Recent popular press coverage of deep learning technology: http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-intelligence.html , http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html , http://www.wired.com/2013/06/andrew_ng/,

  A recent research paper on HPC for Deep Learning: http://www.stanford.edu/~acoates/papers/CoatesHuvalWangWuNgCatanzaro_
  Widely-used tutorials and references for Deep Learning: http://ufldl.stanford.edu/wiki/index.php/Main_Page, http://deeplearning.net/
- Use Case 27 Organizing large-scale, unstructured collections of consumer photos http://vision.soic.indiana.edu/projects/disco/
- Use Case 28 Truthy: Information diffusion research from Twitter Data http://truthy.indiana.edu/ , http://cnets.indiana.edu/groups/nan/truthy/ , http://cnets.indiana.edu/groups/nan/despic/
- Use Case 30 CINET: Cyberinfrastructure for Network (Graph) Science and Analytics http://cinet.vbi.vt.edu/cinet_new/
- Use Case 31 NIST Information Access Division analytic technology performance measurement, evaluations, and standards http://www.nist.gov/itl/iad/
- Use Case 32 DataNet Federation Consortium DFC: The DataNet Federation Consortium , iRODS
- Use Case 33 The 'Discinnet process', metadata < - > big data global experiment http://www.discinnet.org/

- Use Case 34 Semantic Graph-search on Scientific Chemical and Text-based Data http://www.eurekalert.org/pub_releases/2013-07/aiop-ffm071813.php , http://xpdb.nist.gov/chemblast/pdb.pl
- Use Case 35 Light source beamlines http://www-als.lbl.gov/ , https://www1.aps.anl.gov/
- Use Case 36 CRTS survey , CSS survey ; For an overview of the classification challenges, see, e.g., http://arxiv.org/abs/1209.1681
- Use Case 37 DOE Extreme Data from Cosmological Sky Survey and Simulations http://www.lsst.org/lsst/ , http://www.nersc.gov/ , http://www.nersc.gov/assets/Uploads/HabibcosmosimV2.pdf
- Use Case 38 Large Survey Data for Cosmology http://desi.lbl.gov/ , http://www.darkenergysurvey.org/
- Use Case 39 Particle Physics: Analysis of LHC Large Hadron Collider Data: Discovery of Higgs particle http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf , http://www.es.net/assets/pubs_presos/High-throughput-lessons-from-the-LHC-experience.Johnston.TNC2013.pdf
- Use Case 40 Belle II High Energy Physics Experiment http://belle2.kek.jp/
- Use Case 41 EISCAT 3D incoherent scatter radar system https://www.eiscat3d.se/
- Use Case 42 ENVRI, Common Operations of Environmental Research Infrastructure, ENVRI Project website , ENVRI Reference Model , ENVRI deliverable D3.2 : Analysis of common requirements of Environmental Research Infrastructures , ICOS , Euro - Argo , EISCAT 3D , LifeWatch , EPOS , EMSO
- Use Case 43 Radar Data Analysis for CReSIS Remote Sensing of Ice Sheets https://www.cresis.ku.edu/
- Use Case 44 UAVSAR Data Processing, Data Product Delivery, and Data Services http://uavsar.jpl.nasa.gov/ , http://www.asf.alaska.edu/program/sdc , http://geo-gateway.org/main.html
- Use Case 47 Atmospheric Turbulence - Event Discovery and Predictive Analytics http://oceanworld.tamu.edu/resources/oceanography-book/teleconnections.htm , http://www.forbes.com/sites/toddwoody/2012/03/21/meet-the-scientists-mining-big-data-to-predict-the-weather/
- Use Case 48 Climate Studies using the Community Earth System Model at DOE.s NERSC center http://www-pcmdi.llnl.gov/ , http://www.nersc.gov/ , http://science.energy.gov/ber/research/cesd/ , http://www2.cisl.ucar.edu/
- Use Case 50 DOE-BER AmeriFlux and FLUXNET Networks http://ameriflux.lbl.gov/ , http://www.fluxdata.org/default.aspx
- Use Case 51 Consumption forecasting in Smart Grids http://smartgrid.usc.edu/, http://ganges.usc.edu/wiki/Smart_Grid, https://www.ladwp.com/ladwp/faces/ladwp/aboutus/a-power/a-p-smartgridla?_afrLoop=157401916661989&_afrWindowMode=0&_afrWindowId=null#%40%3F_afrWindowId%3Dnull%26_afr state%3Db7yulr4rl_17, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6475927

2.8.3 Unit 17 - 51 Big Data Use Cases

Unit Overview

This units consists of one or more slides for each of the 51 use cases - typically additional (more than one) slides are associated with pictures. Each of the use cases is identified with source of parallelism and the high and low level computational structure. As each new classification topic is introduced we briefly discuss it but full discussion of topics is given in following unit.

Slides

https://iu.app.box.com/s/cvki350s0a12o404a524

Lesson 1 - Government Use Cases I

This covers Census 2010 and 2000 - Title 13 Big Data; National Archives and Records Administration Accession NARA, Search, Retrieve, Preservation; Statistical Survey Response Improvement (Adaptive Design) and Non-Traditional Data in Statistical Survey Response Improvement (Adaptive Design).

Video: http://youtu.be/gCqBFYDDzSQ

Lesson 2 - Government Use Cases II

This covers Census 2010 and 2000 - Title 13 Big Data; National Archives and Records Administration Accession NARA, Search, Retrieve, Preservation; Statistical Survey Response Improvement (Adaptive Design) and Non-Traditional Data in Statistical Survey Response Improvement (Adaptive Design).

Video: http://youtu.be/y0nIed-Nxjw

Lesson 3 - Commercial Use Cases I

This covers Cloud Eco-System, for Financial Industries (Banking, Securities & Investments, Insurance) transacting business within the United States; Mendeley - An International Network of Research; Netflix Movie Service; Web Search; IaaS (Infrastructure as a Service) Big Data Business Continuity & Disaster Recovery (BC/DR) Within A Cloud Eco-System; Cargo Shipping; Materials Data for Manufacturing and Simulation driven Materials Genomics.

Video: http://youtu.be/P1iuViI-AKc

Lesson 4 - Commercial Use Cases II

This covers Cloud Eco-System, for Financial Industries (Banking, Securities & Investments, Insurance) transacting business within the United States; Mendeley - An International Network of Research; Netflix Movie Service; Web Search; IaaS (Infrastructure as a Service) Big Data Business Continuity & Disaster Recovery (BC/DR) Within A Cloud Eco-System; Cargo Shipping; Materials Data for Manufacturing and Simulation driven Materials Genomics.

Video: http://youtu.be/epFH4w_Q9lc

Lesson 5 - Commercial Use Cases III

This covers Cloud Eco-System, for Financial Industries (Banking, Securities & Investments, Insurance) transacting business within the United States; Mendeley - An International Network of Research; Netflix Movie Service; Web Search; IaaS (Infrastructure as a Service) Big Data Business Continuity & Disaster Recovery (BC/DR) Within A Cloud Eco-System; Cargo Shipping; Materials Data for Manufacturing and Simulation driven Materials Genomics.

Video: http://youtu.be/j5kWjL4y7Bo

Lesson 6 - Defense Use Cases I

This covers Large Scale Geospatial Analysis and Visualization; Object identification and tracking from Wide Area Large Format Imagery (WALF) Imagery or Full Motion Video (FMV) - Persistent Surveillance and Intelligence Data Processing and Analysis.

Video: http://youtu.be/8hXG7dinhjg

Lesson 7 - Defense Use Cases II

This covers Large Scale Geospatial Analysis and Visualization; Object identification and tracking from Wide Area Large Format Imagery (WALF) Imagery or Full Motion Video (FMV) - Persistent Surveillance and Intelligence Data Processing and Analysis.

Video: http://youtu.be/MplyAfmuxko

Lesson 8 - Healthcare and Life Science Use Cases I

This covers Electronic Medical Record (EMR) Data; Pathology Imaging/digital pathology; Computational Bioimaging; Genomic Measurements; Comparative analysis for metagenomes and genomes; Individualized Diabetes Management; Statistical Relational Artificial Intelligence for Health Care; World Population Scale Epidemiological Study; Social Contagion Modeling for Planning, Public Health and Disaster Management and Biodiversity and LifeWatch.

Video: http://youtu.be/jVARCWVeYxQ

Lesson 9 - Healthcare and Life Science Use Cases II

This covers Electronic Medical Record (EMR) Data; Pathology Imaging/digital pathology; Computational Bioimaging; Genomic Measurements; Comparative analysis for metagenomes and genomes; Individualized Diabetes Management; Statistical Relational Artificial Intelligence for Health Care; World Population Scale Epidemiological Study; Social Contagion Modeling for Planning, Public Health and Disaster Management and Biodiversity and LifeWatch.

Video: http://youtu.be/y9zJzrH4P8k

Lesson 10 - Healthcare and Life Science Use Cases III

This covers Electronic Medical Record (EMR) Data; Pathology Imaging/digital pathology; Computational Bioimaging; Genomic Measurements; Comparative analysis for metagenomes and genomes; Individualized Diabetes Management; Statistical Relational Artificial Intelligence for Health Care; World Population Scale Epidemiological Study; Social Contagion Modeling for Planning, Public Health and Disaster Management and Biodiversity and LifeWatch.

Video: http://youtu.be/eU5emeI3AmM

Lesson 11 - Deep Learning and Social Networks Use Cases

This covers Large-scale Deep Learning; Organizing large-scale, unstructured collections of consumer photos; Truthy: Information diffusion research from Twitter Data; Crowd Sourcing in the Humanities as Source for Bigand Dynamic Data; CINET: Cyberinfrastructure for Network (Graph) Science and Analytics and NIST Information Access Division analytic technology performance measurement, evaluations, and standards.

Video: http://youtu.be/WLSe6MF4ha4

Lesson 12 - Research Ecosystem Use Cases

DataNet Federation Consortium DFC; The 'Discinnet process', metadata - big data global experiment; Semantic Graph-search on Scientific Chemical and Text-based Data and Light source beamlines.

Video: http://youtu.be/pZ6JucTCKcw

Lesson 13 - Astronomy and Physics Use Cases I

This covers Catalina Real-Time Transient Survey (CRTS): a digital, panoramic, synoptic sky survey; DOE Extreme Data from Cosmological Sky Survey and Simulations; Large Survey Data for Cosmology; Particle Physics: Analysis of LHC Large Hadron Collider Data: Discovery of Higgs particle and Belle II High Energy Physics Experiment.

Video: http://youtu.be/rWqkF-b3Kwk

Lesson 14 - Astronomy and Physics Use Cases II

This covers Catalina Real-Time Transient Survey (CRTS): a digital, panoramic, synoptic sky survey; DOE Extreme Data from Cosmological Sky Survey and Simulations; Large Survey Data for Cosmology; Particle Physics: Analysis of LHC Large Hadron Collider Data: Discovery of Higgs particle and Belle II High Energy Physics Experiment.

Video: http://youtu.be/RxLCB6yLmpk

Lesson 15 - Environment, Earth and Polar Science Use Cases I

EISCAT 3D incoherent scatter radar system; ENVRI, Common Operations of Environmental Research Infrastructure; Radar Data Analysis for CReSIS Remote Sensing of Ice Sheets; UAVSAR Data Processing, DataProduct Delivery, and Data Services; NASA LARC/GSFC iRODS Federation Testbed; MERRA Analytic Services MERRA/AS; Atmospheric Turbulence - Event Discovery and Predictive Analytics; Climate Studies using the Community Earth System Model at DOE's NERSC center; DOE-BER Subsurface Biogeochemistry Scientific Focus Area and DOE-BER AmeriFlux and FLUXNET Networks.

Video: http://youtu.be/u2zTIGwsJwU

Lesson 16 - Environment, Earth and Polar Science Use Cases II

EISCAT 3D incoherent scatter radar system; ENVRI, Common Operations of Environmental Research Infrastructure; Radar Data Analysis for CReSIS Remote Sensing of Ice Sheets; UAVSAR Data Processing, DataProduct Delivery, and Data Services; NASA LARC/GSFC iRODS Federation Testbed; MERRA Analytic Services MERRA/AS; Atmospheric Turbulence - Event Discovery and Predictive Analytics; Climate Studies using the Community Earth System Model at DOE's NERSC center; DOE-BER Subsurface Biogeochemistry Scientific Focus Area and DOE-BER AmeriFlux and FLUXNET Networks.

Video: http://youtu.be/sH3B3gXuJ7E

Lesson 17 - Energy Use Case

This covers Consumption forecasting in Smart Grids.

Video: http://youtu.be/ttmVypmgWmw

Resources

- DCGSA Standard Cloud: https://www.youtube.com/watch?v=l4Qii7T8zeg
- NIST Big Data Public Working Group (NBD-PWG) Process http://bigdatawg.nist.gov/home.php
- On line 51 Use Cases http://bigdatawg.nist.gov/usecases.php
- Summary of Requirements Subgroup http://bigdatawg.nist.gov/_uploadfiles/M0245_v5_6066621242.docx
- Use Case 6 Mendeley http://mendeley.com%20http//dev.mendeley.com

- Use Case 7 Netflix http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-tutoria
- Use Case 8 Search http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013 , http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html , http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws , http://www.slideshare.net/beechung/recommender-systems-tutorialpart1intro , http://www.worldwidewebsize.com/
- Use Case 9 IaaS (Infrastructure as a Service) Big Data Business Continuity & Disaster Recovery (BC/DR) Within A Cloud Eco-System provided by Cloud Service Providers (CSPs) and Cloud Brokerage Service Providers (CBSPs) http://www.disasterrecovery.org/
- Use Case 11 and Use Case 12 Simulation driven Materials Genomics https://www.materialsproject.org/
- Use Case 13 Large Scale Geospatial Analysis and Visualization http://www.opengeospatial.org/standards , http://geojson.org/ , http info.nga.mil/publications/specs/printed/CADRG/cadrg.html
- Use Case 14 Object identification and tracking from Wide Area Large Format Imagery (WALF) Imagery or Full Motion Video (FMV) - Persistent Surveillance http://www.militaryaerospace.com/topics/m/video/79088650/persistent-surveillance-relies-on-extracting-relevant-data-points-and-connecting-the-dots.htm , http://www.defencetalk.com/wide-area-persistent-surveillance-revolutionizes-tactical-isr-45745/
- Use Case 15 Intelligence Data Processing and Analysis http://www.afcea-aberdeen.org/files/presentations/AFCEAAberdeen_DCGSA_COLWells_PS.pdf ,‘ http://stids.c4i.gmu.edu/papers/STIDSPapers/S <http://stids.c4i.gmu.edu/papers/STIDSPapers/STIDS2012_T14_SmithEtAl_HorizontalIntegrationOfWarfighterIntel.pdf>‘__ , ht
- Use Case 16 Electronic Medical Record (EMR) Data: Regenstrief Institute , Logical observation identifiers names and codes , Indiana Health Information Exchange , Institute of Medicine Learning Healthcare System
- Use Case 17 Pathology Imaging/digital pathology; https://web.cci.emory.edu/confluence/display/PAIS , https://web.cci.emory.edu/
- Use Case 19 Genome in a Bottle Consortium: www.genomeinabottle.org
- Use Case 20 Comparative analysis for metagenomes and genomes http://img.jgi.doe.gov/
- Use Case 25 Biodiversity and LifeWatch
- Use Case 26 Deep Learning: Recent popular press coverage of deep learning technology: http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-intelligence.html , http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html , http://www.wired.com/2013/06/andrew_ng/ ; A recent research paper on HPC for Deep Learning: http://www.stanford.edu/~acoates/papers/CoatesHuvalWangWuNgCatanzaro_icml2013.pdf Widely-used tutorials and references for Deep Learning: http://ufldl.stanford.edu/wiki/index.php/Main_Page , http://deeplearning.net/
- Use Case 27 Organizing large-scale, unstructured collections of consumer photos http://vision.soic.indiana.edu/projects/disco/
- Use Case 28 Truthy: Information diffusion research from Twitter Data http://truthy.indiana.edu/ , http://cnets.indiana.edu/groups/nan/truthy/ , http://cnets.indiana.edu/groups/nan/despic/
- Use Case 30 CINET: Cyberinfrastructure for Network (Graph) Science and Analytics http://cinet.vbi.vt.edu/cinet_new/
- Use Case 31 NIST Information Access Division analytic technology performance measurement, evaluations, and standards http://www.nist.gov/itl/iad/
- Use Case 32 DataNet Federation Consortium DFC: The DataNet Federation Consortium , iRODS
- Use Case 33 The ‘Discinnet process’, metadata < - > big data global experiment http://www.discinnet.org/

- Use Case 34 Semantic Graph-search on Scientific Chemical and Text-based Data http://www.eurekalert.org/pub_releases/2013-07/aiop-ffm071813.php , http://xpdb.nist.gov/chemblast/pdb.pl
- Use Case 35 Light source beamlines http://www-als.lbl.gov/ , https://www1.aps.anl.gov/
- Use Case 36 CRTS survey , CSS survey ; For an overview of the classification challenges, see, e.g., http://arxiv.org/abs/1209.1681
- Use Case 37 DOE Extreme Data from Cosmological Sky Survey and Simulations http://www.lsst.org/lsst/ , http://www.nersc.gov/ , http://www.nersc.gov/assets/Uploads/HabibcosmosimV2.pdf
- Use Case 38 Large Survey Data for Cosmology http://desi.lbl.gov/ , http://www.darkenergysurvey.org/
- Use Case 39 Particle Physics: Analysis of LHC Large Hadron Collider Data: Discovery of Higgs particle http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf , http://
throughput-lessons-from-the-LHC-experience.Johnston.TNC2013.pdf
- Use Case 40 Belle II High Energy Physics Experiment http://belle2.kek.jp/
- Use Case 41 EISCAT 3D incoherent scatter radar system https://www.eiscat3d.se/
- Use Case 42 ENVRI, Common Operations of Environmental Research Infrastructure, ENVRI Project website , ENVRI Reference Model , ENVRI deliverable D3.2 : Analysis of common requirements of Environmental Research Infrastructures , ICOS , Euro - Argo , EISCAT 3D , LifeWatch , EPOS , EMSO
- Use Case 43 Radar Data Analysis for CReSIS Remote Sensing of Ice Sheets https://www.cresis.ku.edu/
- Use Case 44 UAVSAR Data Processing, Data Product Delivery, and Data Services http://uavsar.jpl.nasa.gov/ , http://www.asf.alaska.edu/program/sdc , http://geo-gateway.org/main.html
- Use Case 47 Atmospheric Turbulence - Event Discovery and Predictive Analytics http://oceanworld.tamu.edu/resources/oceanography-book/teleconnections.htm , http://www.forbes.com/sites/toddwoody/2012/03/21/meet-the-scientists-mining-big-data-to-predict-the-weather/
- Use Case 48 Climate Studies using the Community Earth System Model at DOE.s NERSC center http://www-pcmdi.llnl.gov/ , http://www.nersc.gov/ , http://science.energy.gov/ber/research/cesd/ , http://www2.cisl.ucar.edu/
- Use Case 50 DOE-BER AmeriFlux and FLUXNET Networks http://ameriflux.lbl.gov/ , http://www.fluxdata.org/default.aspx
- Use Case 51 Consumption forecasting in Smart Grids http://smartgrid.usc.edu/ , http://ganges.usc.edu/wiki/Smart_Grid , https://ww
power/a-p-smartgridla?_afrLoop=157401916661989&_afrWindowMode=0&_afrWindowId=null#%40%3F_afrWindowId%3Dn
state%3Db7yulr4rl_17 , http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6475927

2.8.4 Unit 18 - Features of 51 Big Data Use Cases

Unit Overview

This unit discusses the categories used to classify the 51 use-cases. These categories include concepts used for parallelism and low and high level computational structure. The first lesson is an introduction to all categories and the further lessons give details of particular categories.

Slides

https://iu.app.box.com/s/azpn47brv4o46iij9xvb

Lesson 1 - Summary of Use Case Classification I

This discusses concepts used for parallelism and low and high level computational structure. Parallelism can be over People (users or subjects), Decision makers; Items such as Images, EMR, Sequences; observations, contents of online store; Sensors – Internet of Things; Events; (Complex) Nodes in a Graph; Simple nodes as in a learning network; Tweets, Blogs, Documents, Web Pages etc.; Files or data to be backed up, moved or assigned metadata; Particles/cells/mesh points. Low level computational types include PP (Pleasingly Parallel); MR (MapReduce); MRStat; MRIter (Iterative MapReduce); Graph; Fusion; MC (Monte Carlo) and Streaming. High level computational types include Classification; S/Q (Search and Query); Index; CF (Collaborative Filtering); ML (Machine Learning); EGO (Large Scale Optimizations); EM (Expectation maximization); GIS; HPC; Agents. Patterns include Classic Database; NoSQL; Basic processing of data as in backup or metadata; GIS; Host of Sensors processed on demand; Pleasingly parallel processing; HPC assimilated with observational data; Agent-based models; Multi-modal data fusion or Knowledge Management; Crowd Sourcing.

Video: http://youtu.be/dfgH6YvHCGE

Lesson 2 - Summary of Use Case Classification II

This discusses concepts used for parallelism and low and high level computational structure. Parallelism can be over People (users or subjects), Decision makers; Items such as Images, EMR, Sequences; observations, contents of online store; Sensors – Internet of Things; Events; (Complex) Nodes in a Graph; Simple nodes as in a learning network; Tweets, Blogs, Documents, Web Pages etc.; Files or data to be backed up, moved or assigned metadata; Particles/cells/mesh points. Low level computational types include PP (Pleasingly Parallel); MR (MapReduce); MRStat; MRIter (Iterative MapReduce); Graph; Fusion; MC (Monte Carlo) and Streaming. High level computational types include Classification; S/Q (Search and Query); Index; CF (Collaborative Filtering); ML (Machine Learning); EGO (Large Scale Optimizations); EM (Expectation maximization); GIS; HPC; Agents. Patterns include Classic Database; NoSQL; Basic processing of data as in backup or metadata; GIS; Host of Sensors processed on demand; Pleasingly parallel processing; HPC assimilated with observational data; Agent-based models; Multi-modal data fusion or Knowledge Management; Crowd Sourcing.

Video: http://youtu.be/TjHus5-HaMQ

Lesson 3 - Summary of Use Case Classification III

This discusses concepts used for parallelism and low and high level computational structure. Parallelism can be over People (users or subjects), Decision makers; Items such as Images, EMR, Sequences; observations, contents of online store; Sensors – Internet of Things; Events; (Complex) Nodes in a Graph; Simple nodes as in a learning network; Tweets, Blogs, Documents, Web Pages etc.; Files or data to be backed up, moved or assigned metadata; Particles/cells/mesh points. Low level computational types include PP (Pleasingly Parallel); MR (MapReduce); MRStat; MRIter (Iterative MapReduce); Graph; Fusion; MC (Monte Carlo) and Streaming. High level computational types include Classification; S/Q (Search and Query); Index; CF (Collaborative Filtering); ML (Machine Learning); EGO (Large Scale Optimizations); EM (Expectation maximization); GIS; HPC; Agents. Patterns include Classic Database; NoSQL; Basic processing of data as in backup or metadata; GIS; Host of Sensors processed on demand; Pleasingly parallel processing; HPC assimilated with observational data; Agent-based models; Multi-modal data fusion or Knowledge Management; Crowd Sourcing.

Video: http://youtu.be/EbuNBbt4rQc

Lesson 4 - Database(SQL) Use Case Classification

This discusses classic (SQL) datbase approach to data handling with Search&Query and Index features. Comparisons are made to NoSQL approaches.

Video: http://youtu.be/8QDcUWjA9Ok

Lesson 5 - NoSQL Use Case Classification

This discusses NoSQL (compared in previous lesson) with HDFS, Hadoop and Hbase. The Apache Big data stack is introduced and further details of comparison with SQL.

Video: http://youtu.be/aJ127gkHQUs

Lesson 6 - Use Case Classifications I

This discusses a subset of use case features: GIS, Sensors. the support of data analysis and fusion by streaming data between filters.

Video: http://youtu.be/STAoaS1T2bM

Lesson 7 - Use Case Classifications II Part 1

This discusses a subset of use case features: Pleasingly parallel, MRStat, Data Assimilation, Crowd sourcing, Agents, data fusion and agents, EGO and security.

Video: http://youtu.be/_tJRzG-jS4A

Lesson 8 - Use Case Classifications II Part 2

This discusses a subset of use case features: Pleasingly parallel, MRStat, Data Assimilation, Crowd sourcing, Agents, data fusion and agents, EGO and security.

Video: http://youtu.be/5iHdzMNviZo

Lesson 9 - Use Case Classifications III Part 1

This discusses a subset of use case features: Classification, Monte Carlo, Streaming, PP, MR, MRStat, MRIter and HPC(MPI), global and local analytics (machine learning), parallel computing, Expectation Maximization, graphs and Collaborative Filtering.

Video: http://youtu.be/tITbuwCRVzs

Lesson 10 - Use Case Classifications III Part 2

This discusses a subset of use case features: Classification, Monte Carlo, Streaming, PP, MR, MRStat, MRIter and HPC(MPI), global and local analytics (machine learning), parallel computing, Expectation Maximization, graphs and Collaborative Filtering.

Video: http://youtu.be/0zaXWo8A4Co

Resources

See previous section

2.9 Section 8 - Technology Training - Plotviz

2.9.1 Section Overview

We introduce Plotviz, a data visualization tool developed at Indiana University to display 2 and 3 dimensional data. The motivation is that the human eye is very good at pattern recognition and can ''see" structure in data. Although most Big data is higher dimensional than 3, all can be transformed by dimension reduction techniques to 3D. He gives several examples to show how the software can be used and what kind of data can be visualized. This includes individual plots and the manipulation of multiple synchronized plots.Finally, he describes the download and software dependency of Plotviz.

2.9.2 Unit 19 - Using Plotviz Software for Displaying Point Distributions in 3D

Unit Overview

We introduce Plotviz, a data visualization tool developed at Indiana University to display 2 and 3 dimensional data. The motivation is that the human eye is very good at pattern recognition and can ''see" structure in data. Although most Big data is higher dimensional than 3, all can be transformed by dimension reduction techniques to 3D. He gives several examples to show how the software can be used and what kind of data can be visualized. This includes individual plots and the manipulation of multiple synchronized plots. Finally, he describes the download and software dependency of Plotviz.

Slides

https://iu.app.box.com/s/jypomnrz755xgps5e6iw

Files

- `Fungi_LSU_3_15_to_3_26_zeroidx.pviz`
- `DatingRatings-OriginalLabels.pviz`
- `ClusterFinal-M30-C28.pviz`
- `clusterFinal-M3-C3Dating-ReClustered.pviz`

Lesson 1 - Motivation and Introduction to use

The motivation of Plotviz is that the human eye is very good at pattern recognition and can ''see" structure in data. Although most Big data is higher dimensional than 3, all data can be transformed by dimension reduction techniques to 3D and one can check analysis like clustering and/or see structure missed in a computer analysis. The motivations shows some Cheminformatics examples. The use of Plotviz is started in slide 4 with a discussion of input file which is either a simple text or more features (like colors) can be specified in a rich XML syntax. Plotviz deals with points and their classification (clustering). Next the protein sequence browser in 3D shows the basic structure of Plotviz interface. The next two slides explain the core 3D and 2D manipulations respectively. Note all files used in examples are available to students.

Video: http://youtu.be/4aQlCmQ1jfY

Lesson 2 - Example of Use I: Cube and Structured Dataset

Initially we start with a simple plot of 8 points ~~ the corners of a cube in 3 dimensions ~~ showing basic operations such as size/color/labels and Legend of points. The second example shows a dataset (coming from GTM dimension reduction) with significant structure. This has .pviz and a .txt versions that are compared.

Video: http://youtu.be/nCTT5mI_j_Q

Lesson 3 - Example of Use II: Proteomics and Synchronized Rotation

This starts with an examination of a sample of Protein Universe Browser showing how one uses Plotviz to look at different features of this set of Protein sequences projected to 3D. Then we show how to compare two datasets with synchronized rotation of a dataset clustered in 2 different ways; this dataset comes from k Nearest Neighbor discussion.

Video: http://youtu.be/lDbIhnLrNkk

Lesson 4 - Example of Use III: More Features and larger Proteomics Sample

This starts by describing use of Labels and Glyphs and the Default mode in Plotviz. Then we illustrate sophisticated use of these ideas to view a large Proteomics dataset.

Video: http://youtu.be/KBkUW_QNSvs

Lesson 5 - Example of Use IV: Tools and Examples

This lesson starts by describing the Plotviz tools and then sets up two examples ~~ Oil Flow and Trading ~~ described in PowerPoint. It finishes with the Plotviz viewing of Oil Flow data.

Video: http://youtu.be/zp_709imR40

Lesson 6 - Example of Use V: Final Examples

This starts with Plotviz looking at Trading example introduced in previous lesson and then examines solvent data. It finishes with two large biology examples with 446K and 100K points and each with over 100 clusters. We finish remarks on Plotviz software structure and how to download. We also remind you that a picture is worth a 1000 words.

Video: http://youtu.be/FKoCfTJ_cDM

Resources

Download files from http://salsahpc.indiana.edu/pviz3/

2.10 Section 9 - e-Commerce and LifeStyle Case Study

2.10.1 Section Overview

Recommender systems operate under the hood of such widely recognized sites as Amazon, eBay, Monster and Netflix where everything is a recommendation. This involves a symbiotic relationship between vendor and buyer whereby the buyer provides the vendor with information about their preferences, while the vendor then offers recommendations tailored to match their needs. Kaggle competitions h improve the success of the Netflix and other recommender systems. Attention is paid to models that are used to compare how changes to the systems affect their overall performance. It

is interesting that the humble ranking has become such a dominant driver of the world's economy. More examples of recommender systems are given from Google News, Retail stores and in depth Yahoo! covering the multi-faceted criteria used in deciding recommendations on web sites.

The formulation of recommendations in terms of points in a space or bag is given where bags of item properties, user properties, rankings and users are useful. Detail is given on basic principles behind recommender systems: user-based collaborative filtering, which uses similarities in user rankings to predict their interests, and the Pearson correlation, used to statistically quantify correlations between users viewed as points in a space of items. Items are viewed as points in a space of users in item-based collaborative filtering. The Cosine Similarity is introduced, the difference between implicit and explicit ratings and the k Nearest Neighbors algorithm. General features like the curse of dimensionality in high dimensions are discussed. A simple Python k Nearest Neighbor code and its application to an artificial data set in 3 dimensions is given. Results are visualized in Matplotlib in 2D and with Plotviz in 3D. The concept of a training and a testing set are introduced with training set pre labeled. Recommender system are used to discuss clustering with k-means based clustering methods used and their results examined in Plotviz. The original labelling is compared to clustering results and extension to 28 clusters given. General issues in clustering are discussed including local optima, the use of annealing to avoid this and value of heuristic algorithms.

2.10.2 Unit 20 - Recommender Systems: Introduction

Unit Overview

We introduce Recommender systems as an optimization technology used in a variety of applications and contexts online. They operate in the background of such widely recognized sites as Amazon, eBay, Monster and Netflix where everything is a recommendation. This involves a symbiotic relationship between vendor and buyer whereby the buyer provides the vendor with information about their preferences, while the vendor then offers recommendations tailored to match their needs, to the benefit of both.

There follows an exploration of the Kaggle competition site, other recommender systems and Netflix, as well as competitions held to improve the success of the Netflix recommender system. Finally attention is paid to models that are used to compare how changes to the systems affect their overall performance. It is interesting how the humble ranking has become such a dominant driver of the world's economy.

Slides

https://iu.app.box.com/s/v2coa6mxql12iax4yc8f

Lesson 1 - Recommender Systems as an Optimization Problem

We define a set of general recommender systems as matching of items to people or perhaps collections of items to collections of people where items can be other people, products in a store, movies, jobs, events, web pages etc. We present this as "yet another optimization problem".

https://youtu.be/rymBt1kdyVU

Lesson 2 - Recommender Systems Introduction

We give a general discussion of recommender systems and point out that they are particularly valuable in long tail of tems (to be recommended) that aren't commonly known. We pose them as a rating system and relate them to information retrieval rating systems. We can contrast recommender systems based on user profile and context; the most familiar collaborative filtering of others ranking; item properties; knowledge and hybrid cases mixing some or all of these.

https://youtu.be/KbjBKrzFYKg

Lesson 3 - Kaggle Competitions

We look at Kaggle competitions with examples from web site. In particular we discuss an Irvine class project involving ranking jokes.

https://youtu.be/DFH7GPrbsJA

Lesson 4 - Examples of Recommender Systems

We go through a list of 9 recommender systems from the same Irvine class.

https://youtu.be/1Eh1epQj-EQ

Lesson 5 - Netflix on Recommender Systems I

This is Part 1.

We summarize some interesting points from a tutorial from Netflix for whom ''everything is a recommendation''. Rankings are given in multiple categories and categories that reflect user interests are especially important. Criteria used include explicit user preferences, implicit based on ratings and hybrid methods as well as freshness and diversity. Netflix tries to explain the rationale of its recommendations. We give some data on Netflix operations and some methods used in its recommender systems. We describe the famous Netflix Kaggle competition to improve its rating system. The analogy to maximizing click through rate is given and the objectives of optimization are given.

https://youtu.be/tXsU5RRAD-w

Lesson 6 - Netflix on Recommender Systems II

This is Part 2 of "Netflix on Recommender Systems"

https://youtu.be/GnAol5aGuEo

Lesson 7 - Consumer Data Science

Here we go through Netflix's methodology in letting data speak for itself in optimizing the recommender engine. An example iis given on choosing self produced movies. A/B testing is discussed with examples showing how testing does allow optimizing of sophisticated criteria. This lesson is concluded by comments on Netflix technology and the full spectrum of issues that are involved including user interface, data, AB testing, systems and architectures. We comment on optimizing for a household rather than optimizing for individuals in household.

https://youtu.be/B8cjaOQ57LI

Resources

- http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-tutorial
- http://www.ifi.uzh.ch/ce/teaching/spring2012/16-Recommender-Systems_Slides.pdf
- https://www.kaggle.com/
- http://www.ics.uci.edu/~welling/teaching/CS77Bwinter12/CS77B_w12.html
- Jeff Hammerbacher https://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf
- http://www.techworld.com/news/apps/netflix-foretells-house-of-cards-success-with-cassandra-big-data-engine-3437514/

- https://en.wikipedia.org/wiki/A/B_testing
- http://www.infoq.com/presentations/Netflix-Architecture

2.10.3 Unit 21 - Recommender Systems: Examples and Algorithms

Unit Overview

We continue the discussion of recommender systems and their use in e-commerce. More examples are given from Google News, Retail stores and in depth Yahoo! covering the multi-faceted criteria used in deciding recommendations on web sites. Then the formulation of recommendations in terms of points in a space or bag is given.

Here bags of item properties, user properties, rankings and users are useful. Then we go into detail on basic principles behind recommender systems: user-based collaborative filtering, which uses similarities in user rankings to predict their interests, and the Pearson correlation, used to statistically quantify correlations between users viewed as points in a space of items.

Slides

https://iu.app.box.com/s/pqa1xpk7g4jnr7k2xlbe

Lesson 1 - Recap and Examples of Recommender Systems

We start with a quick recap of recommender systems from previous unit; what they are with brief examples.

https://youtu.be/dcdm5AfGZ64

Lesson 2 - Examples of Recommender Systems

We give 2 examples in more detail: namely Google News and Markdown in Retail.

https://youtu.be/og07mH9fU0M

Lesson 3 - Recommender Systems in Yahoo Use Case Example I

This is Part 1.

We describe in greatest detail the methods used to optimize Yahoo web sites. There are two lessons discussing general approach and a third lesson examines a particular personalized Yahoo page with its different components. We point out the different criteria that must be blended in making decisions; these criteria include analysis of what user does after a particular page is clicked; is the user satisfied and cannot that we quantified by purchase decisions etc. We need to choose Articles, ads, modules, movies, users, updates, etc to optimize metrics such as relevance score, CTR, revenue, engagement.These lesson stress that if though we have big data, the recommender data is sparse. We discuss the approach that involves both batch (offline) and on-line (real time) components.

https://youtu.be/FBn7HpGFNvg

Lesson 4 - Recommender Systems in Yahoo Use Case Example II

This is Part 2 of "Recommender Systems in Yahoo Use Case Example"

https://youtu.be/VS2Y4lAiP5A

Lesson 5 - Recommender Systems in Yahoo Use Case Example III: Particular Module

This is Part 3 of "Recommender Systems in Yahoo Use Case Example"

https://youtu.be/HrRJWEF8EfU

Lesson 6 - User-based nearest-neighbor collaborative filtering I

This is Part 1.

Collaborative filtering is a core approach to recommender systems. There is user-based and item-based collaborative filtering and here we discuss the user-based case. Here similarities in user rankings allow one to predict their interests, and typically this quantified by the Pearson correlation, used to statistically quantify correlations between users.

https://youtu.be/lsf_AE-8dSk

Lesson 7 - User-based nearest-neighbor collaborative filtering II

This is Part 2 of "User-based nearest-neighbor collaborative filtering"

https://youtu.be/U7-qeX2ItPk

Lesson 8 - Vector Space Formulation of Recommender Systems

We go through recommender systems thinking of them as formulated in a funny vector space. This suggests using clustering to make recommendations.

https://youtu.be/IlQUZOXlaSU

Resources

- http://pages.cs.wisc.edu/~beechung/icml11-tutorial/

2.10.4 Unit 22 - Item-based Collaborative Filtering and its Technologies

Unit Overview

We move on to item-based collaborative filtering where items are viewed as points in a space of users. The Cosine Similarity is introduced, the difference between implicit and explicit ratings and the k Nearest Neighbors algorithm. General features like the curse of dimensionality in high dimensions are discussed.

Slides

https://iu.app.box.com/s/fvrwds7zd65m79a7uur3

Lesson 1 - Item-based Collaborative Filtering I

This is Part 1.

We covered user-based collaborative filtering in the previous unit. Here we start by discussing memory-based real time and model based offline (batch) approaches. Now we look at item-based collaborative filtering where items are viewed in the space of users and the cosine measure is used to quantify distances. WE discuss optimizations and how batch processing can help. We discuss different Likert ranking scales and issues with new items that do not have a significant number of rankings.

https://youtu.be/25sBgh3HwxY

Lesson 2 - Item-based Collaborative Filtering II

This is Part 2 of "Item-based Collaborative Filtering"

https://youtu.be/SM8EJdAa4mw

Lesson 3 - k Nearest Neighbors and High Dimensional Spaces

We define the k Nearest Neighbor algorithms and present the Python software but do not use it. We give examples from Wikipedia and describe performance issues. This algorithm illustrates the curse of dimensionality. If items were a real vectors in a low dimension space, there would be faster solution methods.

https://youtu.be/2NqUsDGQDy8

2.11 Section 10 - Technology Training - kNN & Clustering

2.11.1 Section Overview

This section is meant to provide a discussion on the kth Nearest Neighbor (kNN) algorithm and clustering using K-means. Python version for kNN is discussed in the video and instructions for both Java and Python are mentioned in the slides. Plotviz is used for generating 3D visualizations.

2.11.2 Unit 23 - Recommender Systems - K-Nearest Neighbors (Python & Java Track)

Unit Overview

We discuss simple Python k Nearest Neighbor code and its application to an artificial data set in 3 dimensions. Results are visualized in Matplotlib in 2D and with Plotviz in 3D. The concept of training and testing sets are introduced with training set pre-labelled.

Slides

https://iu.app.box.com/s/i9et3dxnhr3qt5gn14bg

Files

- `kNN.py`
- `kNN_Driver.py`
- `DatingTesting2.txt`
- `clusterFinal-M3-C3Dating-ReClustered.pviz`
- `DatingRating-OriginalLabels.pviz`
- `clusterFinal-M30-C28.pviz`

Lesson 1 - Python k'th Nearest Neighbor Algorithms I

This is Part 1.

This lesson considers the Python k Nearest Neighbor code found on the web associated with a book by Harrington on Machine Learning. There are two data sets. First we consider a set of 4 2D vectors divided into two categories (clusters) and use k=3 Nearest Neighbor algorithm to classify 3 test points. Second we consider a 3D dataset that has already been classified and show how to normalize. In this lesson we just use Matplotlib to give 2D plots.

https://youtu.be/o16L0EqsQ_g

Lesson 2 - Python k'th Nearest Neighbor Algorithms II

This is Part 2 of "Python k'th Nearest Neighbor Algorithms".

https://youtu.be/JK5p24mnTjs

Lesson 3 - 3D Visualization

The lesson modifies the online code to allow it to produce files readable by PlotViz. We visualize already classified 3D set and rotate in 3D.

https://youtu.be/fLtH-ZI1Jqk

Lesson 4 - Testing k'th Nearest Neighbor Algorithms

The lesson goes through an example of using k NN classification algorithm by dividing dataset into 2 subsets. One is training set with initial classification; the other is test point to be classified by k=3 NN using training set. The code records fraction of points with a different classification from that input. One can experiment with different sizes of the two subsets. The Python implementation of algorithm is analyzed in detail.

https://youtu.be/zLaPGMIQ9So

2.11.3 Unit 24 - Clustering and heuristic methods

Unit Overview

We use example of recommender system to discuss clustering. The details of methods are not discussed but k-means based clustering methods are used and their results examined in Plotviz. The original labelling is compared to clustering results and extension to 28 clusters given. General issues in clustering are discussed including local optima, the use of annealing to avoid this and value of heuristic algorithms.

Slides

https://iu.app.box.com/s/70qn6d61oln9b50jqobl

Files

- `Fungi_LSU_3_15_to_3_26_zeroidx.pviz`
- `DatingRating-OriginalLabels.pviz`
- `clusterFinal-M30-C28.pviz`
- `clusterFinal-M3-C3Dating-ReClustered.pviz`

Lesson 1 - Kmeans Clustering

We introduce the k means algorithm in a gentle fashion and describes its key features including dangers of local minima. A simple example from Wikipedia is examined.

https://youtu.be/3KTNJ0Okrqs

Lesson 2 - Clustering of Recommender System Example

Plotviz is used to examine and compare the original classification with an ''optimal" clustering into 3 clusters using a fancy deterministic annealing method that is similar to k means. The new clustering has centers marked.

https://youtu.be/yl_KZ86NT-A

Lesson 3 - Clustering of Recommender Example into more than 3 Clusters

The previous division into 3 clusters is compared into a clustering into 28 separate clusters that are naturally smaller in size and divide 3D space covered by 1000 points into compact geometrically local regions.

https://youtu.be/JWZmh48l0cw

Lesson 4 - Local Optima in Clustering

This lesson introduces some general principles. First many important processes are ''just" optimization problems. Most such problems are rife with local optima. The key idea behind annealing to avoid local optima is described. The pervasive greedy optimization method is described.

https://youtu.be/Zmq8O_axCmc

Lesson 5 - Clustering in General

The two different applications of clustering are described. First find geometrically distinct regions and secondly divide spaces into geometrically compact regions that may have no ''thin air" between them. Generalizations such as mixture models and latent factor methods are just mentioned. The important distinction between applications in vector spaces and those where only inter-point distances are defined is described. Examples are then given using PlotViz from 2D clustering of a mass spectrometry example and the results of clustering genomic data mapped into 3D with Multi Dimensional Scaling MDS.

https://youtu.be/JejNZhBxjRU

Lesson 6 - Heuristics

Some remarks are given on heuristics; why are they so important why getting exact answers is often not so important?

https://youtu.be/KT22YuX8ZMY

Resources

- https://en.wikipedia.org/wiki/Kmeans
- http://grids.ucs.indiana.edu/ptliupages/publications/DACIDR_camera_ready_v0.3.pdf
- http://salsahpc.indiana.edu/millionseq/
- http://salsafungiphy.blogspot.com/
- https://en.wikipedia.org/wiki/Heuristic

2.12 Section 11 - Cloud Computing Technology for Big Data Applications & Analytics (will be updated)

2.12.1 Section Overview

We describe the central role of Parallel computing in Clouds and Big Data which is decomposed into lots of ''Little data" running in individual cores. Many examples are given and it is stressed that issues in parallel computing are seen in day to day life for communication, synchronization, load balancing and decomposition. Cyberinfrastructure for e-moreorlessanything or moreorlessanything-Informatics and the basics of cloud computing are introduced. This includes virtualization and the important ''as a Service" components and we go through several different definitions of cloud computing.

Gartner's Technology Landscape includes hype cycle and priority matrix and covers clouds and Big Data. Two simple examples of the value of clouds for enterprise applications are given with a review of different views as to nature of Cloud Computing. This IaaS (Infrastructure as a Service) discussion is followed by PaaS and SaaS (Platform and Software as a Service). Features in Grid and cloud computing and data are treated. We summarize the 21 layers and almost 300 software packages in the HPC-ABDS Software Stack explaining how they are used.

Cloud (Data Center) Architectures with physical setup, Green Computing issues and software models are discussed followed by the Cloud Industry stakeholders with a 2014 Gartner analysis of Cloud computing providers. This is followed by applications on the cloud including data intensive problems, comparison with high performance computing, science clouds and the Internet of Things. Remarks on Security, Fault Tolerance and Synchronicity issues in cloud follow. We describe the way users and data interact with a cloud system. The Big Data Processing from an application perspective with commercial examples including eBay concludes section after a discussion of data system architectures.

2.12.2 Unit 25 - Parallel Computing: Overview of Basic Principles with familiar Examples

Unit Overview

We describe the central role of Parallel computing in Clouds and Big Data which is decomposed into lots of ''Little data" running in individual cores. Many examples are given and it is stressed that issues in parallel computing are seen in day to day life for communication, synchronization, load balancing and decomposition.

Slides

https://iu.app.box.com/s/nau0rsr39kyej240s4yz

Lesson 1 - Decomposition I

This is Part 1.

We describe why parallel computing is essential with Big Data and distinguishes parallelism over users to that over the data in problem. The general ideas behind data decomposition are given followed by a few often whimsical examples dreamed up 30 years ago in the early heady days of parallel computing. These include scientific simulations, defense outside missile attack and computer chess. The basic problem of parallel computing ~~ efficient coordination of separate tasks processing different data parts ~~ is described with MPI and MapReduce as two approaches. The challenges of data decomposition in irregular problems is noted.

https://youtu.be/R-wHQW2YuRE

Lesson 2 - Decomposition II

This is Part 2 of "Decomposition".

https://youtu.be/iIi9wdvlwCM

Lesson 3 - Decomposition III

This is Part 3 of "Decomposition".

https://youtu.be/F0aeeLeTD9I

Lesson 4 - Parallel Computing in Society I

This is Part 1.

This lesson from the past notes that one can view society as an approach to parallel linkage of people. The largest example given is that of the construction of a long wall such as that (Hadrian's wall) between England and Scotland. Different approaches to parallelism are given with formulae for the speed up and efficiency. The concepts of grain size (size of problem tackled by an individual processor) and coordination overhead are exemplified. This example also illustrates Amdahl's law and the relation between data and processor topology. The lesson concludes with other examples from nature including collections of neurons (the brain) and ants.

https://youtu.be/8rtjoe8AeJw

Lesson 5 - Parallel Computing in Society II

This is Part 2 of "Parallel Computing in Society".

https://youtu.be/7sCgH_TTPGk

Lesson 6 - Parallel Processing for Hadrian's Wall

This lesson returns to Hadrian's wall and uses it to illustrate advanced issues in parallel computing. First We describe the basic SPMD ~~ Single Program Multiple Data ~~ model. Then irregular but homogeneous and heterogeneous problems are discussed. Static and dynamic load balancing is needed. Inner parallelism (as in vector instruction or the multiple fingers of masons) and outer parallelism (typical data parallelism) are demonstrated. Parallel I/O for Hadrian's wall is followed by a slide summarizing this quaint comparison between Big data parallelism and the construction of a large wall.

https://youtu.be/ZD2AQ08cy8I

Resources

- Solving Problems in Concurrent Processors-Volume 1, with M. Johnson, G. Lyzenga, S. Otto, J. Salmon, D. Walker, Prentice Hall, March 1988.
- Parallel Computing Works!, with P. Messina, R. Williams, Morgan Kaufman (1994). http://www.netlib.org/utk/lsi/pcwLSI/text/
- The Sourcebook of Parallel Computing book edited by Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, and Andy White, Morgan Kaufmann, November 2002.
- Geoffrey Fox Computational Sciences and Parallelism to appear in Encyclopedia on Parallel Computing edited by David Padua and published by Springer. http://grids.ucs.indiana.edu/ptliupages/publications/SpringerEncyclopedia_Fox.pdf

2.12.3 Unit 26 - Cloud Computing Technology Part I: Introduction

Unit Overview

We discuss Cyberinfrastructure for e-moreorlessanything or moreorlessanything-Informatics and the basics of cloud computing. This includes virtualization and the important 'as a Service' components and we go through several different definitions of cloud computing.Gartner's Technology Landscape includes hype cycle and priority matrix and covers clouds and Big Data. The unit concludes with two simple examples of the value of clouds for enterprise applications. Gartner also has specific predictions for cloud computing growth areas.

Slides

https://iu.app.box.com/s/p3lztuu9kv240pdm66141or9b8p1uvzb

Lesson 1 - Cyberinfrastructure for E-MoreOrLessAnything

This introduction describes Cyberinfrastructure or e-infrastructure and its role in solving the electronic implementation of any problem where e-moreorlessanything is another term for moreorlessanything-Informatics and generalizes early discussion of e-Science and e-Business.

https://youtu.be/gHz0cu195ZM

Lesson 2 - What is Cloud Computing: Introduction

Cloud Computing is introduced with an operational definition involving virtualization and efficient large data centers that can rent computers in an elastic fashion. The role of services is essential ~~ it underlies capabilities being offered in the cloud. The four basic aaS's ~~ Software (SaaS), Platform (Paas), Infrastructure (IaaS) and Network (NaaS) ~~

are introduced with Research aaS and other capabilities (for example Sensors aaS are discussed later) being built on top of these.

https://youtu.be/Od_mYXRs5As

Lesson 3 - What and Why is Cloud Computing: Several Other Views I

This is Part 1.

This lesson contains 5 slides with diverse comments on ''what is cloud computing" from the web.

https://youtu.be/5VeqMjXKU_Y

Lesson 4 - What and Why is Cloud Computing: Several Other Views II

This is Part 2 of "What and Why is Cloud Computing: Several Other Views".

https://youtu.be/J963LR0PS_g

Lesson 5 - What and Why is Cloud Computing: Several Other Views III

This is Part 3 of "What and Why is Cloud Computing: Several Other Views".

https://youtu.be/_ryLXUnOAzo

Lesson 6 - Gartner's Emerging Technology Landscape for Clouds and Big Data

This lesson gives Gartner's projections around futures of cloud and Big data. We start with a review of hype charts and then go into detailed Gartner analyses of the Cloud and Big data areas. Big data itself is at the top of the hype and by definition predictions of doom are emerging. Before too much excitement sets in, note that spinach is above clouds and Big data in Google trends.

https://youtu.be/N7aEtU1mUwc

Lesson 7 - Simple Examples of use of Cloud Computing

This short lesson gives two examples of rather straightforward commercial applications of cloud computing. One is server consolidation for multiple Microsoft database applications and the second is the benefits of scale comparing gmail to multiple smaller installations. It ends with some fiscal comments.

https://youtu.be/VCctCP6BKEo

Lesson 8 - Value of Cloud Computing

Some comments on fiscal value of cloud computing.

https://youtu.be/HM1dZCxdsaA

Resources

- http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing
- http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis
- https://setandbma.wordpress.com/2012/08/10/hype-cycle-2012-emerging-technologies/
- http://insights.dice.com/2013/01/23/big-data-hype-is-imploding-gartner-analyst-2/
- http://research.microsoft.com/pubs/78813/AJ18_EN.pdf
- http://static.googleusercontent.com/media/www.google.com/en//green/pdfs/google-green-computing.pdf

2.12.4 Unit 27 - Cloud Computing Technology Part II: Software and Systems

Unit Overview

We cover different views as to nature of architecture and application for Cloud Computing. Then we discuss cloud software for the cloud starting at virtual machine management (IaaS) and the broad Platform (middleware) capabilities with examples from Amazon and academic studies. We summarize the 21 layers and almost 300 software packages in the HPC-ABDS Software Stack explaining how they are used.

Slides

https://iu.app.box.com/s/k61o0ff1w6jkn5zmpaaiw02yth4v4alh

Lesson 1 - What is Cloud Computing

This lesson gives some general remark of cloud systems from an architecture and application perspective.

https://youtu.be/h3Rpb0Eyj1c

Lesson 2 - Introduction to Cloud Software Architecture: IaaS and PaaS I

We discuss cloud software for the cloud starting at virtual machine management (IaaS) and the broad Platform (middleware) capabilities with examples from Amazon and academic studies.

https://youtu.be/1AnyJYyh490

Lesson 3 - Introduction to Cloud Software Architecture: IaaS and PaaS II

We cover different views as to nature of architecture and application for Cloud Computing. Then we discuss cloud software for the cloud starting at virtual machine management (IaaS) and the broad Platform (middleware) capabilities with examples from Amazon and academic studies. We summarize the 21 layers and almost 300 software packages in the HPC-ABDS Software Stack explaining how they are used.

https://youtu.be/hVpFAUHcAd4

Lesson 4 - Using the HPC-ABDS Software Stack

Using the HPC-ABDS Software Stack.

https://youtu.be/JuTQdRW78Pg

Resources

- http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis
- http://research.microsoft.com/en-us/people/barga/sc09_cloudcomp_tutorial.pdf
- http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_OpportunitiesAndChallenges_Youse
- http://cloudonomic.blogspot.com/2009/02/cloud-taxonomy-and-ontology.html

2.12.5 Unit 28 - Cloud Computing Technology Part III: Architectures, Applications and Systems

Unit Overview

We start with a discussion of Cloud (Data Center) Architectures with physical setup, Green Computing issues and software models. We summarize a 2014 Gartner analysis of Cloud computing providers. This is followed by applications on the cloud including data intensive problems, comparison with high performance computing, science clouds and the Internet of Things. Remarks on Security, Fault Tolerance and Synchronicity issues in cloud follow.

Slides

https://iu.app.box.com/s/0bn57opwe56t0rx4k18bswupfwj7culv

Lesson 1 - Cloud (Data Center) Architectures I

This is Part 1.

Some remarks on what it takes to build (in software) a cloud ecosystem, and why clouds are the data center of the future are followed by pictures and discussions of several data centers from Microsoft (mainly) and Google. The role of containers is stressed as part of modular data centers that trade scalability for fault tolerance. Sizes of cloud centers and supercomputers are discussed as is "green" computing.

https://youtu.be/j0P32DmQjI8

Lesson 2 - Cloud (Data Center) Architectures II

This is Part 2 of "Cloud (Data Center) Architectures".

https://youtu.be/3HAGqz34AB4

Lesson 3 - Analysis of Major Cloud Providers

Gartner 2014 Analysis of leading cloud providers.

https://youtu.be/Tu8hE1SeT28

Lesson 4 - Commercial Cloud Storage Trends

Use of Dropbox, iCloud, Box etc.

https://youtu.be/i5OI6R526kM

Lesson 5 - Cloud Applications I

This is Part 1.

This short lesson discusses the need for security and issues in its implementation. Clouds trade scalability for greater possibility of faults but here clouds offer good support for recovery from faults. We discuss both storage and program fault tolerance noting that parallel computing is especially sensitive to faults as a fault in one task will impact all other tasks in the parallel job.

https://youtu.be/nkeSOMTGbbo

Lesson 6 - Cloud Applications II

This is Part 2 of "Cloud Applications".

https://youtu.be/ORd3aBhc2Rc

Lesson 7 - Science Clouds

Science Applications and Internet of Things.

https://youtu.be/2PDvpZluyvs

Lesson 8 - Security

This short lesson discusses the need for security and issues in its implementation.

https://youtu.be/NojXG3fbrEo

Lesson 9 - Comments on Fault Tolerance and Synchronicity Constraints

Clouds trade scalability for greater possibility of faults but here clouds offer good support for recovery from faults. We discuss both storage and program fault tolerance noting that parallel computing is especially sensitive to faults as a fault in one task will impact all other tasks in the parallel job.

https://youtu.be/OMZiSiN7dlU

Resources

- http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing
- http://www.eweek.com/c/a/Cloud-Computing/AWS-Innovation-Means-Cloud-Domination-307831
- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of Science October 2012 Dennis Gannon.
- http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_OpportunitiesAndChallenges_Youse
- http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pue-is-1-8/
- https://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data-centers-one-way-of-getting-it-just-right/
- http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf
- http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis

- http://www.slideshare.net/botchagalupe/introduction-to-clouds-cloud-camp-columbus
- http://www.venus-c.eu/Pages/Home.aspx
- Geoffrey Fox and Dennis Gannon Using Clouds for Technical Computing To be published in Proceedings of HPC 2012 Conference at Cetraro, Italy June 28 2012 http://grids.ucs.indiana.edu/ptliupages/publications/Clouds_Technical_Computing_FoxGannonv2.pdf
- https://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf
- Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, Bill Franks Wiley ISBN: 978-1-118-20878-6
- Anjul Bhambhri, VP of Big Data, IBM http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Conquering Big Data with the Oracle Information Model, Helen Sun, Oracle
- Hugh Williams VP Experience, Search & Platforms, eBay http://businessinnovation.berkeley.edu/fisher-cio-leadership-program/
- Dennis Gannon, Scientific Computing Environments, http://www.nitrd.gov/nitrdgroups/images/7/73/D_Gannon_2025_scientific_c
- http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_OpportunitiesAndChallenges_Youse
- http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pue-is-1-8/
- https://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data-centers-one-way-of-getting-it-just-right/
- http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf
- http://searchcloudcomputing.techtarget.com/feature/Cloud-computing-experts-forecast-the-market-climate-in-2014
- http://www.slideshare.net/botchagalupe/introduction-to-clouds-cloud-camp-columbus
- http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing
- http://www.venus-c.eu/Pages/Home.aspx
- http://www.kpcb.com/internet-trends

2.12.6 Unit 29 - Cloud Computing Technology Part IV: Data Systems

Unit Overview

We describe the way users and data interact with a cloud system. The unit concludes with the treatment of data in the cloud from an architecture perspective and Big Data Processing from an application perspective with commercial examples including eBay.

Slides

https://iu.app.box.com/s/ftfpybxm8jzjepzp409vgair1fttv3m1

Lesson 1 - The 10 Interaction scenarios (access patterns) I

The next 3 lessons describe the way users and data interact with the system.

https://youtu.be/vB4rCNri_P0

Lesson 2 - The 10 Interaction scenarios - Science Examples

This lesson describes the way users and data interact with the system for some science examples.

https://youtu.be/cFX1PQpiSbk

Lesson 3 - Remaining general access patterns

This lesson describe the way users and data interact with the system for the final set of examples.

https://youtu.be/-dtE9zXB-I0

Lesson 4 - Data in the Cloud

Databases, File systems, Object Stores and NOSQL are discussed and compared. The way to build a modern data repository in the cloud is introduced.

https://youtu.be/HdtIOnk3qX4

Lesson 5 - Applications Processing Big Data

This lesson collects remarks on Big data processing from several sources: Berkeley, Teradata, IBM, Oracle and eBay with architectures and application opportunities.

https://youtu.be/d6A2m4GR-hw

Resources

- http://bigdatawg.nist.gov/_uploadfiles/M0311_v2_2965963213.pdf
- https://dzone.com/articles/hadoop-t-etl
- http://venublog.com/2013/07/16/hadoop-summit-2013-hive-authorization/
- https://indico.cern.ch/event/214784/session/5/contribution/410
- http://asd.gsfc.nasa.gov/archive/hubble/a_pdf/news/facts/FS14.pdf
- http://blogs.teradata.com/data-points/announcing-teradata-aster-big-analytics-appliance/
- http://wikibon.org/w/images/2/20/Cloud-BigData.png
- http://hortonworks.com/hadoop/yarn/
- https://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf
- http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html

2.13 Section 12 - Web Search and Text Mining and their technologies

2.13.1 Section Overview

This section starts with an overview of data mining and puts our study of classification, clustering and exploration methods in context. We examine the problem to be solved in web and text search and note the relevance of history with libraries, catalogs and concordances. An overview of web search is given describing the continued evolution of search engines and the relation to the field of Information Retrieval. The importance of recall, precision and diversity

is discussed. The important Bag of Words model is introduced and both Boolean queries and the more general fuzzy indices. The important vector space model and revisiting the Cosine Similarity as a distance in this bag follows. The basic TF-IDF approach is dis cussed. Relevance is discussed with a probabilistic model while the distinction between Bayesian and frequency views of probability distribution completes this unit.

We start with an overview of the different steps (data analytics) in web search and then goes key steps in detail starting with document preparation. An inverted index is described and then how it is prepared for web search. The Boolean and Vector Space approach to query processing follow. This is followed by Link Structure Analysis including Hubs, Authorities and PageRank. The application of PageRank ideas as reputation outside web search is covered. The web graph structure, crawling it and issues in web advertising and search follow. The use of clustering and topic models completes section

2.13.2 Unit 30 - Web Search and Text Mining I

Unit Overview

The unit starts with the web with its size, shape (coming from the mutual linkage of pages by URL's) and universal power laws for number of pages with particular number of URL's linking out or in to page. Information retrieval is introduced and compared to web search. A comparison is given between semantic searches as in databases and the full text search that is base of Web search. The origin of web search in libraries, catalogs and concordances is summarized. DIKW ~~ Data Information Knowledge Wisdom ~~ model for web search is discussed. Then features of documents, collections and the important Bag of Words representation. Queries are presented in context of an Information Retrieval architecture. The method of judging quality of results including recall, precision and diversity is described. A time line for evolution of search engines is given.

Boolean and Vector Space models for query including the cosine similarity are introduced. Web Crawlers are discussed and then the steps needed to analyze data from Web and produce a set of terms. Building and accessing an inverted index is followed by the importance of term specificity and how it is captured in TF-IDF. We note how frequencies are converted into belief and relevance.

Slides

https://iu.app.box.com/s/qo7itbtcxp2b58syz3jg

Lesson 1 - Web and Document/Text Search: The Problem

This lesson starts with the web with its size, shape (coming from the mutual linkage of pages by URL's) and universal power laws for number of pages with particular number of URL's linking out or in to page.

https://youtu.be/T12BccKe8p4

Lesson 2 - Information Retrieval leading to Web Search

Information retrieval is introduced A comparison is given between semantic searches as in databases and the full text search that is base of Web search. The ACM classification illustrates potential complexity of ontologies. Some differences between web search and information retrieval are given.

https://youtu.be/KtWhk2cdRa4

Lesson 3 - History behind Web Search

The origin of web search in libraries, catalogs and concordances is summarized.

https://youtu.be/J7D61uH5gVM

Lesson 4 - Key Fundamental Principles behind Web Search

This lesson describes the DIKW ~~ Data Information Knowledge Wisdom ~~ model for web search. Then it discusses documents, collections and the important Bag of Words representation.

https://youtu.be/yPFi6xFnDHE

Lesson 5 - Information Retrieval (Web Search) Components

This describes queries in context of an Information Retrieval architecture. The method of judging quality of results including recall, precision and diversity is described.

https://youtu.be/EGsnonXgb3Y

Lesson 6 - Search Engines

This short lesson describes a time line for evolution of search engines. The first web search approaches were directly built on Information retrieval but in 1998 the field was changed when Google was founded and showed the importance of URL structure as exemplified by PageRank.

https://youtu.be/kBV-99N6f7k

Lesson 7 - Boolean and Vector Space Models

This lesson describes the Boolean and Vector Space models for query including the cosine similarity.

https://youtu.be/JzGBA0OhsIk

Lesson 8 - Web crawling and Document Preparation

This describes a Web Crawler and then the steps needed to analyze data from Web and produce a set of terms.

https://youtu.be/Wv-r-PJ9lro

Lesson 9 - Indices

This lesson describes both building and accessing an inverted index. It describes how phrases are treated and gives details of query structure from some early logs.

https://youtu.be/NY2SmrHoBVM

Lesson 10 - TF-IDF and Probabilistic Models

It describes the importance of term specificity and how it is captured in TF-IDF. It notes how frequencies are converted into belief and relevance.

https://youtu.be/9P_HUmpselU

Resources

- http://saedsayad.com/data_mining_map.htm
- http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html
- The Web Graph: an Overview Jean-Loup Guillaume and Matthieu Latapy https://hal.archives-ouvertes.fr/file/index/docid/54458/filename/webgraph.pdf
- Constructing a reliable Web graph with information on browsing behavior, Yiqun Liu, Yufei Xue, Danqing Xu, Rongwei Cen, Min Zhang, Shaoping Ma, Liyun Ru http://www.sciencedirect.com/science/article/pii/S0167923612001844
- http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

2.13.3 Unit 31 - Web Search and Text Mining II

Unit Overview

We start with an overview of the different steps (data analytics) in web search. This is followed by Link Structure Analysis including Hubs, Authorities and PageRank. The application of PageRank ideas as reputation outside web search is covered. Issues in web advertising and search follow. his leads to emerging field of computational advertising. The use of clustering and topic models completes unit with Google News as an example.

Slides

https://iu.app.box.com/s/iuzc1qfep748z1o2kgx2

Lesson 1 - Data Analytics for Web Search

This short lesson describes the different steps needed in web search including: Get the digital data (from web or from scanning); Crawl web; Preprocess data to get searchable things (words, positions); Form Inverted Index mapping words to documents; Rank relevance of documents with potentially sophisticated techniques; and integrate technology to support advertising and ways to allow or stop pages artificially enhancing relevance.

https://youtu.be/ugyycKBjaBQ

Lesson 2 - Link Structure Analysis including PageRank I

This is Part 1.

The value of links and the concepts of Hubs and Authorities are discussed. This leads to definition of PageRank with examples. Extensions of PageRank viewed as a reputation are discussed with journal rankings and university department rankings as examples. There are many extension of these ideas which are not discussed here although topic models are covered briefly in a later lesson.

https://youtu.be/1oXdopVxqfI

Lesson 3 - Link Structure Analysis including PageRank II

This is Part 2 of "Link Structure Analysis including PageRank".

https://youtu.be/OCn-gCTxvrU

Lesson 4 - Web Advertising and Search

Internet and mobile advertising is growing fast and can be personalized more than for traditional media. There are several advertising types Sponsored search, Contextual ads, Display ads and different models: Cost per viewing, cost per clicking and cost per action. This leads to emerging field of computational advertising.

https://youtu.be/GgkmG0NzQvg

Lesson 5 - Clustering and Topic Models

We discuss briefly approaches to defining groups of documents. We illustrate this for Google News and give an example that this can give different answers from word-based analyses. We mention some work at Indiana University on a Latent Semantic Indexing model.

https://youtu.be/95cHMyZ-TUs

Resources

- http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws
- https://en.wikipedia.org/wiki/PageRank
- http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html
- Meeker/Wu May 29 2013 Internet Trends D11 Conference http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013

2.14 Section 13 - Technology for Big Data Applications and Analytics

2.14.1 Section Overview

We use the K-means Python code in SciPy package to show real code for clustering. After a simple example we generate 4 clusters of distinct centers and various choice for sizes using Matplotlib tor visualization. We show results can sometimes be incorrect and sometimes make different choices among comparable solutions. We discuss the ''hill" between different solutions and rationale for running K-means many times and choosing best answer. Then we introduce MapReduce with the basic architecture and a homely example. The discussion of advanced topics includes an extension to Iterative MapReduce from Indiana University called Twister and a generalized Map Collective model. Some measurements of parallel performance are given. The SciPy K-means code is modified to support a MapReduce execution style. This illustrates the key ideas of mappers and reducers. With appropriate runtime this code would run in parallel but here the ''parallel" maps run sequentially. This simple 2 map version can be generalized to scalable parallelism. Python is used to Calculate PageRank from Web Linkage Matrix showing several different formulations of the basic matrix equations to finding leading eigenvector. The unit is concluded by a calculation of PageRank for general web pages by extracting the secret from Google.

2.14.2 Unit 32 - Technology for X-Informatics: K-means (Python & Java Track)

Unit Overview

We use the K-means Python code in SciPy package to show real code for clustering. After a simple example we generate 4 clusters of distinct centers and various choice for sizes using Matplotlib tor visualization. We show results can sometimes be incorrect and sometimes make different choices among comparable solutions. We discuss the ''hill" between different solutions and rationale for running K-means many times and choosing best answer.

Slides

https://iu.app.box.com/s/ltgbehfjwvgh40l5d3w8

Files

- `xmean.py`
- `sample.csv`
- `parallel_kmeans.py`
- `kmeans_extra.py`

Lesson 1 - K-means in Python

We use the K-means Python code in SciPy package to show real code for clustering and applies it a set of 85 two dimensional vectors ~~ officially sets of weights and heights to be clustered to find T-shirt sizes. We run through Python code with Matplotlib displays to divide into 2-5 clusters. Then we discuss Python to generate 4 clusters of varying sizes and centered at corners of a square in two dimensions. We formally give the K means algorithm better than before and make definition consistent with code in SciPy.

https://youtu.be/I79ISV6XBbE

Lesson 2 - Analysis of 4 Artificial Clusters I

This is Part 1.

We present clustering results on the artificial set of 1000 2D points described in previous lesson for 3 choices of cluster sizes ''small" ''large" and ''very large''. We emphasize the SciPy always does 20 independent K means and takes the best result ~~ an approach to avoiding local minima. We allow this number of independent runs to be changed and in particular set to 1 to generate more interesting erratic results. We define changes in our new K means code that also has two measures of quality allowed. The slides give many results of clustering into 2 4 6 and 8 clusters (there were only 4 real clusters). We show that the ''very small" case has two very different solutions when clustered into two clusters and use this to discuss functions with multiple minima and a hill between them. The lesson has both discussion of already produced results in slides and interactive use of Python for new runs.

https://youtu.be/Srgq9VDg4C8

Lesson 3 - Analysis of 4 Artificial Clusters II

This is Part 2 of "Analysis of 4 Artificial Clusters".

https://youtu.be/rjyAXjA_mOk

Lesson 4 - Analysis of 4 Artificial Clusters III

This is Part 3 of "Analysis of 4 Artificial Clusters".

https://youtu.be/N6QKyrhNVAc

2.14.3 Unit 33 - Technology for X-Informatics: MapReduce

Unit Overview

We describe the basic architecture of MapReduce and a homely example. The discussion of advanced topics includes extension to Iterative MapReduce from Indiana University called Twister and a generalized Map Collective model. Some measurements of parallel performance are given.

Slides

https://iu.app.box.com/s/hqykdx1bquez7ers3d1j

Lesson 1 - Introduction

This introduction uses an analogy to making fruit punch by slicing and blending fruit to illustrate MapReduce. The formal structure of MapReduce and Iterative MapReduce is presented with parallel data flowing from disks through multiple Map and Reduce phases to be inspected by the user.

https://youtu.be/67qFY64aj7g

Lesson 2 - Advanced Topics I

This is Part 1.

This defines 4 types of MapReduce and the Map Collective model of Qiu. The Iterative MapReduce model from Indiana University called Twister is described and a few performance measurements on Microsoft Azure are presented.

https://youtu.be/lo4movzSyVw

Lesson 3 - Advanced Topics II

This is Part 2 of "Advanced Topics".

https://youtu.be/wnanWncQBow

2.14.4 Unit 34 - Technology: Kmeans and MapReduce Parallelism

Unit Overview

We modify the SciPy K-means code to support a MapReduce execution style and runs it in this short unit. This illustrates the key ideas of mappers and reducers. With appropriate runtime this code would run in parallel but here the ''parallel" maps run sequentially. We stress that this simple 2 map version can be generalized to scalable parallelism.

Slides

https://iu.app.box.com/s/zc9pckhyehn0cog8wy19

Files

- `ParallelKmeans`

Lesson 1 - MapReduce Kmeans in Python I

This is Part 1.

We modify the SciPy K-means code to support a MapReduce execution style and runs it in this short unit. This illustrates the key ideas of mappers and reducers. With appropriate runtime this code would run in parallel but here the ''parallel" maps run sequentially. We stress that this simple 2 map version can be generalized to scalable parallelism.

https://youtu.be/2El1oL3gKpQ

Lesson 2 - MapReduce Kmeans in Python II

This is Part 2 of "MapReduce Kmeans in Python"

https://youtu.be/LLrTWWdE3T0

2.14.5 Unit 35 - Technology: PageRank (Python & Java Track)

Unit Overview

We use Python to Calculate PageRank from Web Linkage Matrix showing several different formulations of the basic matrix equations to finding leading eigenvector. The unit is concluded by a calculation of PageRank for general web pages by extracting the secret from Google.

Slides

https://iu.app.box.com/s/gwq1qp0kmwbvilo0kjqq

Files

- `pagerank1.py`
- `pagerank2.py`

Lesson 1 - Calculate PageRank from Web Linkage Matrix I

This is Part 1.

We take two simple matrices for 6 and 8 web sites respectively to illustrate the calculation of PageRank.

https://youtu.be/rLWUvvcHrCQ

Lesson 2 - Calculate PageRank from Web Linkage Matrix II

This is Part 2 of "Calculate PageRank for Web linkage Matrix".

https://youtu.be/UzQRukCFQv8

Lesson 3 - Calculate PageRank of a real page

This tiny lesson presents a Python code that finds the Page Rank that Google calculates for any page on the web.

https://youtu.be/8L_72bRLQVk

2.15 Section 14 - Sensors Case Study

2.15.1 Section Overview

We start with the Internet of Things IoT giving examples like monitors of machine operation, QR codes, surveillance cameras, scientific sensors, drones and self driving cars and more generally transportation systems. We give examples of robots and drones. We introduce the Industrial Internet of Things IIoT and summarize surveys and expectations Industry wide. We give examples from General Electric. Sensor clouds control the many small distributed devices of IoT and IIoT. More detail is given for radar data gathered by sensors; ubiquitous or smart cities and homes including U-Korea; and finally the smart electric grid.

2.15.2 Unit 36 - Case Study: Sensors

Unit Overview

See Section Overview

Slides

https://iu.box.com/s/9a5y7p7xvhjqgrc9zjob8gorv3ft4kyq

Lesson 1 - Internet of Things

There are predicted to be 24-50 Billion devices on the Internet by 2020; these are typically some sort of sensor defined as any source or sink of time series data. Sensors include smartphones, webcams, monitors of machine operation, barcodes, surveillance cameras, scientific sensors (especially in earth and environmental science), drones and self driving cars and more generally transportation systems. The lesson gives many examples of distributed sensors, which form a Grid that is controlled by a cloud.

https://youtu.be/fFMvxYW6Yu0

Lesson 2 - Robotics and IOT Expectations

Examples of Robots and Drones.

https://youtu.be/VqXvn0dwqxs

Lesson 3 - Industrial Internet of Things I

This is Part 1.

We summarize surveys and expectations Industry wide.

https://youtu.be/jqQJjtTEsEo

Lesson 4 - Industrial Internet of Things II

This is Part 2 of "Industrial Internet of Things".

Examples from General Electric.

https://youtu.be/YiIvQRCi3j8

Lesson 5 - Sensor Clouds

We describe the architecture of a Sensor Cloud control environment and gives example of interface to an older version of it. The performance of system is measured in terms of processing latency as a function of number of involved sensors with each delivering data at 1.8 Mbps rate.

https://youtu.be/0egT1FsVGrU

Lesson 6 - Earth/Environment/Polar Science data gathered by Sensors

This lesson gives examples of some sensors in the Earth/Environment/Polar Science field. It starts with material from the CReSIS polar remote sensing project and then looks at the NSF Ocean Observing Initiative and NASA's MODIS or Moderate Resolution Imaging Spectroradiometer instrument on a satellite.

https://youtu.be/CS2gX7axWfI

Lesson 7 - Ubiquitous/Smart Cities

For Ubiquitous/Smart cities we give two examples: Iniquitous Korea and smart electrical grids.

https://youtu.be/MFFIItQ3SOo

Lesson 8 - U-Korea (U=Ubiquitous)

Korea has an interesting positioning where it is first worldwide in broadband access per capita, e-government, scientific literacy and total working hours. However it is far down in measures like quality of life and GDP. U-Korea aims to improve the latter by Pervasive computing, everywhere, anytime i.e. by spreading sensors everywhere. The example of a 'High-Tech Utopia' New Songdo is given.

https://youtu.be/wdot23r4YKs

Lesson 9 - Smart Grid

The electrical Smart Grid aims to enhance USA's aging electrical infrastructure by pervasive deployment of sensors and the integration of their measurement in a cloud or equivalent server infrastructure. A variety of new instruments include smart meters, power monitors, and measures of solar irradiance, wind speed, and temperature. One goal is autonomous local power units where good use is made of waste heat.

https://youtu.be/m3eX8act0GU

Resources

- https://www.gesoftware.com/minds-and-machines
- https://www.gesoftware.com/predix
- https://www.gesoftware.com/sites/default/files/the-industrial-internet/index.html
- https://developer.cisco.com/site/eiot/discover/overview/
- http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Competitive-Landscape-Industries.pdf
- http://www.gesoftware.com/ge-predictivity-infographic
- http://www.getransportation.com/railconnect360/rail-landscape

- http://www.gesoftware.com/sites/default/files/GE-Software-Modernizing-Machine-to-Machine-Interactions.pdf

2.16 Section 15 - Radar Case Study

2.16.1 Unit 37 - Case Study: Radar

Unit Overview

The changing global climate is suspected to have long-term effects on much of the world's inhabitants. Among the various effects, the rising sea level will directly affect many people living in low-lying coastal regions. While the ocean-s thermal expansion has been the dominant contributor to rises in sea level, the potential contribution of discharges from the polar ice sheets in Greenland and Antarctica may provide a more significant threat due to the unpredictable response to the changing climate. The Radar-Informatics unit provides a glimpse in the processes fueling global climate change and explains what methods are used for ice data acquisitions and analysis.

Slides

https://iu.app.box.com/s/njxktkb71e2cbroopsx2

Lesson 1 - Introduction

This lesson motivates radar-informatics by building on previous discussions on why X-applications are growing in data size and why analytics are necessary for acquiring knowledge from large data. The lesson details three mosaics of a changing Greenland ice sheet and provides a concise overview to subsequent lessons by detailing explaining how other remote sensing technologies, such as the radar, can be used to sound the polar ice sheets and what we are doing with radar images to extract knowledge to be incorporated into numerical models.

https://youtu.be/LXOncC2AhsI

Lesson 2 - Remote Sensing

This lesson explains the basics of remote sensing, the characteristics of remote sensors and remote sensing applications. Emphasis is on image acquisition and data collection in the electromagnetic spectrum.

https://youtu.be/TTrm9rmZySQ

Lesson 3 - Ice Sheet Science

This lesson provides a brief understanding on why melt water at the base of the ice sheet can be detrimental and why it's important for sensors to sound the bedrock.

https://youtu.be/rDpjMLguVBc

Lesson 4 - Global Climate Change

This lesson provides an understanding and the processes for the greenhouse effect, how warming effects the Polar Regions, and the implications of a rise in sea level.

https://youtu.be/f9hzzJX0qDs

Lesson 5 - Radio Overview

This lesson provides an elementary introduction to radar and its importance to remote sensing, especially to acquiring information about Greenland and Antarctica.

https://youtu.be/PuI7F-RMKCI

Lesson 6 - Radio Informatics

This lesson focuses on the use of sophisticated computer vision algorithms, such as active contours and a hidden markov model to support data analysis for extracting layers, so ice sheet models can accurately forecast future changes in climate.

https://youtu.be/q3Pwyt49syE

CHAPTER 3

FAQ

3.1 I am full time student at IUPUI? Can I take the online version?

I suggest you verify this with the international student office and the registrar if you are an international student. There may be some restrictions for international students. Also some degree programs may have a limit or do not allow to take online classes. It will be up to you to verify the requirements with the appropriate administrators.

3.2 I am a residential student can I take the online version only?

If you are an international student or a student of a particular degree program restrictions may be placed in if and how many online courses you can take. It will be up to you to contact the appropriate administrative departments including the international student office to verify what is allowed for you. In general international students have such restrictions. Please find out what they are and which section of the course is appropriate for you.

3.3 Do I need to buy a textbook?

No, the resources will be provided for every unit. However, there are some optional textbooks if you would like to purchase one.

1. "Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", Bill Franks Wiley ISBN: 978-1-118-20878-6
2. "Doing Data Science: Straight Talk from the Frontline", Cathy O'Neil, Rachel Schutt, O'Reilly Media, ISBN 978-1449358655

3.4 Do I need a computer to participate in this class?

Obviously if you are an online student you do need a computer. If you are a residential student the facilities provided by SOIC will be sufficient. However, as you study involves computers, its probably important to evaluate if a computer will make your work easier.

If it comes to what computer to buy we really do not have a good recommendation as this depends on your budget. A computer running Linux or OSX makes programming probably easier. A windows computer has the advantage of also being able to run Word and ppt. A cheap machine with multiple cores and sufficient memory (4GB+) is a good idea. A SSD will make access to data especially if large data snappy.

For this reason I myself use a Mac, but you probably can get much cheaper machines with similar specs elsewhere.

3.4.1 Representative Bibliography

1. Big data: The next frontier for innovation, competition, and productivity
2. Big Data Spring 2015 Class

3.4.2 Where is the official IU calendar for the Fall?

Please follow this link

3.5 How to write a research article on computer science

1. http://www.wv.inf.tu-dresden.de/Teaching/SS-2012/howto/writing.pdf
2. https://globaljournals.org/guidelines-tips/research-paper-publishing
3. http://www.cs.columbia.edu/~hgs/etc/writing-style.html
4. https://www.quora.com/How-do-I-write-a-research-paper-for-a-computer-science-journal

3.6 How to you use bibliography managers JabRef & Endnote or Mendeley

1. http://www.jabref.org/
2. http://endnote.com/
3. http://libguides.utoledo.edu/c.php?g=284330&p=1895338
4. https://www.mendeley.com/
5. https://community.mendeley.com/guides/using-citation-editor/05-creating-bibliography

3.7 Plagiarism test and resources related to that

1. https://www.grammarly.com/plagiarism-checker
2. http://turnitin.com/
3. http://www.plagscan.com/plagiarism-check/

3.8 How many hours will this course take to work on every week?

This question can not rely be answered precisely. Typically we have 2-3 hours video per week. However starting from that its difficult to put a real number down as things amy also depend on your background.

- The programming load is modest, but requires knowledge in python which you may have to learn outside of this class.
- Some students have more experience than others, thus it may be possible to put in 6 hours per week overall, but other may have to put in 12 hours, while yet others may enjoy this class so much that they spend a lot more hours.

- We will certainly not stopping you form spending time in the class. It will be up to you to figure out how much time you willl spend.
- Please remem ber that procrastenation will not pay of in this class.
- The project or term paper will take a significant amount of time.

CHAPTER 4

Homework

Page Contents

- *Assignments*
  - *Study groups*
  - *Week 1*
    - *Communication*
    - *Resources res1*
    - *Survey 1*
    - *Video V1*
    - *Video V2*
    - *Discussion d1*
    - *Paper p1*
  - *Week 2*
    - *Video V3*
    - *Discussion d3*
    - *Paper p2*
    - *References R1*
  - *Week 3*
    - *Video V4*
    - *Discussion d4*
    - *Paper p3*
  - *Week 4*
    - *Video V5*
    - *Development Virtual Machine*
    - *Programming prg1: Python*
    - *Term Paper and Term Project Report Assignment T1*
    - *Discussion d5*
  - *Week 5*
    - *Video S6*
    - *Futuresystems*
    - *ChameleonCloud*
    - *OpenStack*
    - *prg2 (canceled)*
    - *Discussion d6*
  - *Week 6*
    - *Video S7*
    - *Discussion d7*
  - *Week 7*
  - *Week 8*
    - *Video S9*
    - *Discussion d9*
  - *Week 9*
    - *Video S10*
    - *Discussion d10*
    - *Programming prg-geo*
  - *Week 10*
    - *Discussion d11*
    - *Paper p11*
  - *Week 11 - Week 13*
    - *Project or Term Report*
    - *Discussion 11, 12, 13, 14*
  - *Week 13 - Dec. 2nd*
- *Assignment Guidelines*
  - *Getting Access and Systems Support*
  - *Report and Paper Format*
  - *Software Project*
  - *Term Paper*
  - *Project Proposal*
    - *Project and Term Paper Proposal Format*

4.1 Assignments

If not otherwise stated homework in all sections and classes is the same. All lectures are assigned Friday's and homework is due next week Friday, other than the first week of the semester where the lectures are assigned on Monday (22nd of August) and the first homework is due Friday. Therefore we have not posted explicit due dates, as they are obvious from the calendar. You are welcome to work ahead, but check back in case the homework has been updated. Additional due dates will be posted however in CANVAS. Please visit canvas for these due dates.

As you will be doing some discussions, please PREFACE YOUR POSTS with your Full Name.

External hyperlinks, like Python

1. All assignments will be posted through Canvas
2. You will be provided with a GitLab folder once you register at https://about.gitlab.com/
3. You will complete your assignments and check in your solutions to your gitlab.com repository (see Using GitLab)
4. You will submit to canvas a link to your solution in gitlab

4.1.1 Study groups

It is very common and encouraged to build study groups to discuss with each other the content of the class. However such groups should not be used to copy homework assignments that are intended for individual submissions.

When working in a team, we recommend that you use English as the communication language. This will help those that are not native English speakers.

4.1.2 Week 1

Communication

- Enroll in the class at Piazza
- Register in https://about.gitlab.com/
- Register in https://www.chameleoncloud.org

Resources res1

- If you do not have a computer on which you can do your assignments please apply for an account with Chameleon Cloud. You will have to ask for you to be added to project

  CH-818144: https://www.chameleoncloud.org/user/projects/33130/

  Note: You will only be allowed to use VMs for a duration of 6 hours.
- Register in https://portal.futuresystems.org/

Survey 1

Please fill out the Survey to let us help you better with the course

Video V1

Watch Videos in Section 1: Units 1 and 2 at the Course Page Syllabus

Video V2

Watch Videos in Section 2: Units 3, 4, and 5. Note these units have overlap with Unit 2 of Section 1. (see Syllabus)

Discussion d1

Consider Discussion d1 after Section 1. Please create a new post on the topic "Why is Big Data interesting to me" and also comment on at least 2 other posts.

Paper p1

This assignment may be conducted as a group with at most two students. It will be up to you to find another student, or you can just do the paper yourself. There is no need to keep this team during the semester or the project assignment you can build new teams throughout the semester for different homework. Make sure your team contributes equally.

This assignment requires to write a paper that is 2 pages in length. Please use the 2 column ACM proceedings Format.

- Conduct the Discussion homework first.
- Review what plagiarism is and how to not do it
- Install jabref and organize your citations with jabref

Write a paper discussing all of the following topics:

- What is Big Data?
- Why is Big Data interesting to me? (Summarize and/or contrast positions in the discussion list. This is not just your position. See our note bellow.)
- What limitations does Big Data Analytics have?
- If you work in a team please also discuss different positions if there are any. Make sure the work is shared and no academic honesty policy has been violated.

Please note that a discussion took place on the discussion list that you need to analyze. It is important that you summarize the position and identify a mechanism to evaluate the students responses. One option is that your discussion could be augmented by classifications and statistics. It is allowable to include them as figures in the paper. Others may just highlight selected points raised by the course members.

You will be submitting the paper in gitlab.com as discussed in:

http://bdaafall2016.readthedocs.io/en/latest/gitlab.html

You will be uploading the following files into the paper1 directory:

```
paper1.tex
sample.bib
paper1.pdf
```

After you upload the files, please go to Canvas and fill out the form for the paper1 submission. You will have to upload the appropriate links.

4.1.3 Week 2

Video V3

Please watch Section 3 Unit 6. Total Length 2.5 hours, (see Syllabus)

Discussion d3

Consider Discussion d3 after Section 3. Please post about the topic "Where are the Big Data Jobs now and in future? Discuss anything you can share – areas that are hot, good online sites etc." and also comment on at least 2 other posts.

Paper p2

This requires to write a paper that is two pages in length. Please use the 2 column ACM proceedings Format. Write a paper discussing the following topics:

- What is the role of Big Data in health?
- Discuss any or all areas from telemedicine, personalized (precision) medicine, personal monitors like Fitbit, privacy issues.

You will be submitting the paper in gitlab.com as discussed in:

http://bdaafall2016.readthedocs.io/en/latest/gitlab.html

You will be uploading the following files into the paper2 directory:

```
paper2.tex
sample.bib
paper2.pdf
```

After you upload the files, please go to Canvas and fill out the form for the paper2 submission. You will have to upload the appropriate links.

A video of how to use the Webbrowser to upload the paper is available at:

- https://youtu.be/b3OvgQhTFow

Video in cc: TBD

References R1

It is important that you know how to cite. Please see the page Homework References for guidelines

Bonus points: Use d2 to discuss the topic of crowd sourcing in relationship to big data. Conduct research if needed.

4.1.4 Week 3

Video V4

Please watch Section 4 Unit 7-9. Total Length 3.5 hours (see Syllabus).

Discussion d4

Consider Discussion d4 after Section 4 Please post on topic "Sports and Health Informatics":

- Which are most interesting job areas;
- Which are likely to make most progress
- Which one would you work in given similar offers in both fields
- Comment on at least 2 other posts.

Paper p3

This requires to write a paper that is from one to two pages in length. Please use the 2 column ACM proceedings Format.

This assignment may be conducted as a group with at most two students. It will be up to you to find another student, or you can just do the paper yourself. There is no need to keep this team during the semester or the project assignment you can build new teams throughout the semester for different homework. Make sure your team contributes equally.

Chose one of the alternatives:

Alternative A:

Using what we call Big Data (such as video) and Little Data (such as Baseball numerical statistics) in Sports Analytics. Write a paper discussing the following topics:

- Which offer most opportunity on what sports?
- How is Big Data and Little Data applied to the Olympics2016?

Alternative B (This assignment gives bonus points if done right):

How can big data and lIttle data be used in wildlife conservation, pets, farming, and other related areas that involve animal. Write a 2 page paper that covers the topic and addresses

- Which opportunities are there related to animals?
- Which opportunities are there for wildlife preservation?
- What limitations are there?
- How can big data be best used? give concrete examples.
- This paper could be longer than two pages if you like
- You are allowed to work in a team of six. The number of pages is determined by team members while the minimum page number is 2. The team must identify who did what.
- However the paper must be coherent and consistent.
- Additional pages are allowed.
- When building teams the entire team must approve the team members.

- If a team does not want to have you join, you need to accept this. Look for another team or work alone.
- Use gitlab to share your LaTeX document or use microsoft one drive to write it collaboratively.

4.1.5 Week 4

Video V5

see next section

Development Virtual Machine

To easily develop code and not to effect your local machine, we will be using ubuntu desktop in a virtual machine running on your computer. Please make sure your hardware supports this. For example, a chrome book is insufficient.

The detailed description including 3 videos are posted at:

- http://bdaafall2016.readthedocs.io/en/latest/ubuntu.html

Please conduct form that page Homework 1, 2 & 3

Next you will be using python in that virtual machine.

Note: You can use your native OS to do the programming assignment. However if you like to use any cloud environment you must also do the Development virtual machine as we want you to get a feeing for how to use ubuntu before you go on the cloud.

Programming prg1: Python

Hardware: Identify a suitable hardware environment that works for you to conduct the assignments. First you must have access to a sufficiently powerful computer. This could be your Laptop or Desktop, or you could get access to machines at IU's computer labs or virtual machines.

Setup Python: Next you will need to setup Python on the machine or verify if python works. We recommend that you use python 2.7 and *NOT* python 3. We recommend that you follow the instructions from python.org and use virtualenv. As editor we recommend you use PyCharm or Emacs.

Canopy and Anaconda: We made bad experiences with Canopy as well as Anaconda on some machine of a Teaching Assitant. Therefore we recommend agains using thise systems. It will be up to you to determine if these systems work for you. We do recommend that you use python.org and virtualenv. If you have already started using canopy or anaconda you can do so (but we do not recommend it).

Useful software:

- **Python**
  - NumPy
  - SciPy
  - Matplotlib
  - Pandas

Tasks:

- Learn Python, E.g. go through the Python for Big Data (and Introduction to Python if you need to) lesson.
- Use *virtualenv* and *pip* to customize your environment.
- Learn *Python pandas <http://pandas.pydata.org/>* and do a simple Python application demonstrating:
  - a linechart
  - a barchart, e.g. a histogram

  Find some real meaningful data such as number of people born in a year or some other more interesting data set to demonstrate the various features.
- Review of Scipy: look at the scipy manual and be aware what you can do with it in case you chose a Project

Deliverables prg1:

The goal of this assignment is to choose one or two datasets (see Datasets), preprocess it to clean it up, and generate a line graph and histogram plot. Your figures must provide labels for the axes along with units.

Submit your programs in a folder called `prg1`, which must contain the following:

- `requirements.txt`: list of python libraries your programs need as installable by: `pip install -r requirements.txt`
- `fetchdata.py`: a python program that, when run as `python fetchdata.py` will produce dataset files in CSV format called `data-line.csv` and `data-hist.csv`.
- `linechart.py`: a python program that, when run as `python linechart.py data-line.csv` will generate a line chart as save it in PNG format to a file called `linechart.png`.
- `histogram.py`: a python program that, when run as `python historgram.py data-hist.csv` will generate a histogram plot as save it in PNG format to a file called `histogram.png`
- `README.rst`: a RST format file which documents the datasets you used, where you fetched them from, how `fetchdata.py` cleans them to generate the `data-{line,hist}.csv` files.

Warning: Missing items will result in zero points being given

Term Paper and Term Project Report Assignment T1

Please prepare for the selection process for a project or a term paper:

- Review the guidelines for the project and term paper.
- Identify if you are likely to do a project or a term paper
- Build teams, chose your team members wisely. For example if you have 3 people in the team and only two do the work, you still get graded based on a 3 person team.
- Decide for a topic that you want to do and the team. Commit to it by end of Week 5.

- For that week the homework also includes to make a plan for your term paper and write a one page summary which we will approve and give comments on. Note teaming can change in actual final project. If you are in a team, each student must submit an (identical) plan with a notation as to teaming. Note teaming can change in actual final project.
- You will completing this Form Form, throughout the semester in which you will be uploading the title, the team members, and the location of your proposal in gitlab with direct URL, description of the artifacts and the final project report.

Discussion d5

Create a NEW post to discuss your final project you want to do and look for team members (if you want to build a team).

4.1.6 Week 5

Video S6

Watch the video in Section 6 (see Syllabus).

Futuresystems

- Obtain an account on Futuresystems.org and join project FG511. Not that this will take time and you need to do this ASAP. No late assignments will be accepted. If you are late this assignment will receive 0 points.

  Which account name should i use?: The same name as you use at IU to register. If you have had a previous class and used a different name, please let us know, so we can make a note of it. Please do not apply for two accounts. If you account name is already taken, please use a different one.

ChameleonCloud

- Obtain an account on https://www.chameleoncloud.org. Fill out the Poll TBD (This assignment is optional, but we have made good experience with Chameleon cloud, so we advise you to get an account. As you are a student you will not be able to create a project. We will announce the project in due time that you can join and use chameleon cloud).

OpenStack

- Inform yourself about OpenStack and how to start and stop virtual machines via the command line.
- Optionally, you can use cloudmesh_client for this (If you use cloudmesh client you will get bonus points).

prg2 (canceled)

Consider the Python code available on Section 6 Unit 13 "Files" tab (the third one) as HiggsClassIIUniform.py. This software is also available When run it should produce results like the file TypicalResultsHW5.docx on the same tab. This code corresponds to 42000 background events and 300 Higgs. Background is uniformly distributed and Higgs is a Normal (Gaussian) distribution centered at 126 with width of 2. Produce 2 more figures (plots) corresponding to experiments with a factor of 10 more or a

factor of 10 less data. (Both Higgs and Background increase or decrease by same factor). Return the two new figures and your code as Homework in github under the folder *prg2".

What do you conclude from figures about ability to see Higgs particle with different amount of data (corresponding to different lengths of time experiment runs) Due date October 25 Video V6: Video Review/Study Section 7 Units 12-15; total 3 hours 7 minutes. This is Physics Informatics Section.

https://github.com/cglmoocs/bdaafall2015/tree/master/PythonFiles/Section-4_Physics-Units-9-10-11/Unit-9_The-Elusive-Mr.-Higgs

Discussion d6

Post on Discussion d6 after Section 7, the "Physics" topic:

- What you found interesting, remarkable or shocking about the search for Higgs Bosons.
- Was it worth all that money?
- Please also comment on at least 2 other posts.

4.1.7 Week 6

Video S7

Watch the videos in section 7 (see Syllabus).

Discussion d7

Post on Discussion d7 on the topic:

- Which is the most interesting/important of the 51 use cases in section 7.
- Why?
- What is most interesting/important use case not in group of 51?
- Please write one post and comment on at least 2 other posts in the discussions.

4.1.8 Week 7

This weeks lecture will be determined at a later time.

4.1.9 Week 8

Video S9

Watch the videos related to Section 9 (see Syllabus).

Discussion d9

Post on Discussion d9:

- What are benefits for e-Commerce?
- What are limitations for e-Commerce?
- Waht are risks and benefits for Banking industry using big data?

4.1.10 Week 9

Video S10

Watch the videos related to Section 10 (see Syllabus).

Discussion d10

Use Discussion d10 in case you have questions about PRG-GEO

Programming prg-geo

PRG-GEO can be found here: geolocation

4.1.11 Week 10

Discussion d11

Discuss what you learnt from video you watched in S11: Parallel Computing and Clouds under Discussion d11

Paper p11

Consider any 5 cloud or cloud like activities from list of 11 below. Describe the ones you chose and explain what ways they could be used to generate an X-Informatics for some X. Write a 2 page paper wit the Paper format from Section paper_format:

- http://aws.amazon.com/ (Links to an external site.)
- http://www.windowsazure.com/en-us/ (Links to an external site.)
- https://cloud.google.com/compute/ (Links to an external site.)
- https://portal.futuresystems.org/ (Links to an external site.)
- http://joyent.com/ (Links to an external site.)
- https://pod.penguincomputing.com/ (Links to an external site.)
- http://www.rackspace.com/cloud/ (Links to an external site.)
- http://www.salesforce.com/cloudcomputing/ (Links to an external site.)

- http://earthengine.google.org/ (Links to an external site.)
- http://www.openstack.org/ (Links to an external site.)
- https://www.docker.com/ (Links to an external site.)

4.1.12 Week 11 - Week 13

Project or Term Report

Work on your project

Discussion 11, 12, 13, 14

Discuss what you learnt from videos you watched in last 2 weeks of class Sections 12-15; chose one of the topics: Web Search and Text mining, Big Data Technology, Sensors, Radar Each Discussion about the topic is to be conducted in the week it is introduced. Due dates Friday's.

4.1.13 Week 13 - Dec. 2nd

Continue to work on your Term Paper or Project

Due date for the project is Dec 2nd. It will a considerable amount of time to grade your project and term papers. Thus the deadline is mandatory. Late projects and term papers will receive a 10% grade reduction. Furthermore dependent on when the project is handed in it may not be graded over the Christmas break.

4.2 Assignment Guidelines

4.2.1 Getting Access and Systems Support

For some projects you will need access to a cloud. We recommend you evaluate which cloud would be most appropriate for your project. This includes:

- chameleoncloud.org
- furturesystems.org
- AWS (you will be responsible for charges)
- Azure (you will be responsible for charges)
- virtualbox if you have a powerful computer and like to prototype
- other clouds

We intend to make some small number of virtual machines available for us in a project FG511 on FutureSystems:

- https://portal.futuresystems.org/projects/511

Note: FutureSystems OpenStack cloud is currently updated and will not be available till Sept.

Documentation about FutureSystems can be found at *OpenStackFutureSystems*

Once you created an account on FutureSystems and you do a project yOu can add yourself to the project so you gain access. Systems staff is available only during regular business hours Mo-Fri 10am - 4pm.

You could also use the cloudmesh client software on Linux and OSX to access multiple clouds in easy fashion. A Section will introduce this software.

4.2.2 Report and Paper Format

All reports and paper assignments will be using the ACM proceedings format. The MSWord template can be found here:

- `paper-report.docx`
- The URL is

https://gitlab.com/cloudmesh/fall2016/blob/master/docs/source/files/paper-report.docx

A LaTeX version can be found at

- https://www.acm.org/publications/proceedings-template

however you have to remove the ACM copyright notice in the LaTeX version.

There will be **NO EXCEPTION** to this format. In case you are in a team, you can use either gitlab while collaboratively developing the LaTeX document or use MicrosoftOne Drive which allows collaborative editing features. All bibliographical entries must be put into a bibliography manager such as jabref, endnote, or Mendeley. This will guarantee that you follow proper citation styles. You can use either ACM or IEEE reference styles. Your final submission will include the bibliography file as a separate document.

Documents that do not follow the ACM format and are not accompanied by references managed with jabref or endnote or are not spell checked will be returned without review.

Please do not use figures ore tables toe artificially inflate the length of the report. Make figures readable and provide the original images. Use PDF for figures and not png, gif, org jpeg. This way the figures you produce are scalable and zooming into the paper will be possible.

Report Checklist:

- [ ] Have you written the report in word or LaTeX in the specified format.
- [ ] In case of LaTeX, have you removed the ACM copyright information
- [ ] Have you included the report in gitlab.
- [ ] Have you specified the names and e-mails of all team members in your report. E.g. the username in Canvas.
- [ ] Have you included all images in native and PDF format in gitlab in the images folder.
- [ ] Have you added the bibliography file (such as endnote or bibtex file e.g. jabref) in a directory bib.
- [ ] Have you submitted an additional page that describes who did what in the project or report.
- [ ] Have you spellchecked the paper.
- [ ] Have you made sure you do not plagiarize.

4.2.3 Software Project

Develop a software system with OpenStack available on FutureSystems or Chameleoncloud to support it. Only choose the software option if you are prepared to take on programming tasks.

In case of a software project, we encourage a group project with up to three members. You can use the discussion list for the Software Project to form project teams or just communicate privately with other class members to formulate a team. The following artifacts are part of the deliverables for a project

Code: You must deliver the code in gitlab. The code must be compilable and a TA may try to replicate to run your code. You MUST avoid lengthy install descriptions and everything must be installable from the commandline.

Project Report: A report must be produced while using the format discussed in the Report Format section. The following length is required:

- 4 pages, one student in the project
- 6 pages, two students in the project
- 8 pages, three students in the project

Reports can be longer up to 10 pages if needed. Your high quality scientific report should describe a) What you did b) results obtained and c) Software documentation including how to install, and run it. If c) is longer than half a page and can not be reproduced with shell scripts or easy to follow steps you will get points deducted.

Work Breakdown: This document is only needed for team projects. A one page PDF document describing who did what. It includes pointers to the git history that documents the statistics that demonstrate not only one student has worked on the project.

License: All projects are developed under an open source license such as Apache 2.0 License, or similar. You will be required to add a LICENCE.txt file and if you use other software identify how it can be reused in your project. If your project uses different licenses, please add in a README.rst file which packages are used and which license these packages have.

Code Repository: Code repositories are for code, if you have additional libraries that are needed you need to develop a script or use a DevOps framework to install such software. Thus zip files and .class, .o files are not permissible in the project. Each project must be reproducible with a simple script. An example is:

```
git clone ....
make install
make run
make view
```

Which would use a simple make file to install, run, and view the results. Naturally you can use ansible or shell scripts. It is not permissible to use GUI based DevOps preinstalled frameworks. Everything must be installable form the command line.

Datasets that may inspire projects can be found in Datasets.

You should also review sampleprojects.

4.2.4 Term Paper

Term Report: In case you chose the term paper, you or your team will pick a topic relevant for the class. You will write a high quality scholarly paper about this topic. This includes scientifically examining technologies and application.

Content Rules: Material may be taken from other sources but that must amount to at most 25% of paper and must be cited Figures may be used (citations in the figure caption are required). As usual, proper citations and quotations must be given to such content. The quality should be similar to a publishable paper or technical report. Plagiarism is not allowed.

Proposal: The topic should be close to what you will propose. Please contact me if you change significantly topic. Also inform me if you change teaming. These changes are allowed; We just need to know, review, and approve.

You can use the discussion list for the Term Paper to form project teams or just communicate privately with other class members to formulate a team.

Deliverables: The following artifacts are part of the deliverables for a term paper. A report must be produced while using the format discussed in the Report Format section. The following length is required:

- 6 pages, one student in the project
- 9 pages, two student in the project
- 12 pages, three student in the project

A gitlab repository will contain the paper your wrote in PDF and in docx or latex. All images will be in an image folder and be clearly marked. All bibtex or endnote files will be included in the repository.

Work Breackdown: This document is only needed for team projects. A one page PDF document describing who did what. The document is called workbreakdown.pdf

The directory structure thus look like:

```
./paper.docx
./paper.pdf
./refrences.enl
./images/myniftyimage-fig1.pptx
./images/myniftyimage-fig1.pdf
```

Possible Term Paper Topics:

- Big Data and Agriculture
- Big Data and Transportation
- Big Data and Home Automation
- Big Data and Internet of Things
- Big Data and Olympics
- Big Data and Environment
- Big Data and Astrophysics
- Big Data and Deep Learning
- Big Data and Biology
- Survey of Big Data Applications (Difficult as lots of work, tHis is a 3 person project only and at least 15 pages are required, where additional three pages are given for references.)
- Big Data and "Suggest your own"
- Review of Recommender Systems: technology & applications
- Review of Big Data in Bioinformatics
- Review of Data visualization including high dimensional data
- Design of a NoSQL database for a specialized application

4.2.5 Project Proposal

Project and Term Paper Proposal Format

Please submit a one page ACM style 2 column paper in which you include the following information dependent on if you do a term paper or Project. The title will be proceeded with the keyword "PROJECT" or "REPORT"

A project proposal should contain in the proposal section:

- The nature of the project and its context
- The technologies used
- Any proprietary issues
- Specific aims you intent to complete
- A list of intended deliverables (artifacts produced)

Title:

- REPORT: Your title

or

- Project: Your title

Authors: The Authors need to be listed in the proposal with Fullname, e-mail, and gitlab username, if you use futuresystems or chameleoncloud you will also need to add your futuresystems or chameleoncloud name. Please put the prefix futuresystems: and/or chamelon: in the author field accordingly. Please only include if you have used the resources. If you do not use the resources for the project or report, ther is no need to include them.

Example:

```
Gregor von Laszewski
laszewski@gmail.com
chameleon: gregor
futuresystems: gvl
```

Abstract: Include in your abstract a short summary of the report or project

Proposal: Include a section called proposal in which you in detail describe what you will do.

Artifacts: Include a section Artifacts describing what you will produce and where you will store it.

Examples are:

- A Survey Paper
- Code on gitlab
- Screenshots
- ...

4.3 Homework upload

A video of how to use the Webbrowser to upload the paper is available at:

Video: https://youtu.be/b3OvgQhTFow

Video in cc: TBD

Naturally if you know how to use the git commandline tool use that which will have to master once you start working on your project or term paper.

CHAPTER 5

Using GitLab

This course requires the use of GitLab.com for your homework submissions.

Once you have completed the entry survey you will be granted access to a git repository in which to develop your homework submissions. What you submit to canvas will be a link to a folder or file in your gitlab repository.

The repository should consist of a subfolder in the root directory for each assignment, e.g. `prg1`, `prg2`, `project`, for programming assignment 1, programming assignment 2 and your project.

Important: The above are just examples. The assignment prompts will indicate the exact name for each subdirectory. It is imperative that you adhere to the name that will be specified else you may have points deducted.

Important: Please use only lowercase characters in the directory names and no special characters such as @ ; /

5.1 Getting an account

Please go to gitlab and create an account. Use a nice account name that only includes characters in [a-zA-Z0-9].

- http://gitlab.com

In canvas a list is published that shows your Homework-ID (HID). The HID will be the name of the directory in gitlab that you will be using to submit your homework.

5.2 Upload your public key

Please upload your public key to the repository as documented in gitlab.

5.3 How to configure Git and Gitlab for your computer

The proper way to use git is to install a client on your computer. Once you have done so, make sure to configure git to use your name and email address label your commits.:

```
$ git config --global user.name "Albert Einstein"
$ git config --global user.email albert@iu.edu
```

Warning: Make sure to substitute in your name and email address in the commands above.

You should also configure the push behavior to push only matching branches. See the git documentation for more details on what this means.:

```
$ git config --global push.default matching
```

5.4 Using Web browsers to upload

Although we do not recommend using this, it is possible to use the Web browser to modify existing and to upload new files via. This means you could operate it without installing anything. This will work, but it is not very convenient.

5.5 Using Git GUI tools

There are many git GUI tools available that directly integrate into your operating system finders, windows, ..., or PyCharm. It is up to you to identify such tools and see if they are useful for you. Most of the people we work with us git from the command line, even if they use PyCharm or other tools that have build in git support.

5.6 Submission of homework

You will have a HID given to you. Let us assume the id is:

```
F16-DG-9999
```

When you log into gitlab, you will find a directory with that name. Please substitute the HID that we gave above as an example with your own. We refer to this ID as <HID> in these instructions.

Now you can go to your web browser and past the following URL into it, where you replace the <HID> with your HID that you can find in Canvas.:

```
https://gitlab.com/cloudmesh_fall2016/<HID>
```

For our example this would result in:

```
https://gitlab.com/cloudmesh_fall2016/F16-DG-9999
```

You will find in the directory subdirectories for your homework. If they are missing, please create them. You will see:

```
prg1
prg2
prg3
paper1
paper2
paper3
bib1
```

To submit the homework you need to first clone the repository (read the git manual about what cloning means):

```
git clone https://gitlab.com/cloudmesh/fall2016/HID
```

Your homework for submission should be organized according to folders in your clone repository. To submit a particular assignment, you must first add it using:

```
git add <name of the file you are adding>
```

Afterwards, commit it using:

```
git commit -m "message describing your submission"
```

Then push it to your remote repository using:

```
git push
```

If you want to modify your submission, you only need to:

```
git commit -m "message relating to updated file"
```

afterwards:

```
git push
```

If you lose any documents locally, you can retrieve them from your remote repository using:

```
    git pull
```

If you have any issues, please post your question in the folder gitlab. Our TAs will answer them.

5.7 Git Resources

If you are unfamiliar with git you may find these resources useful:

- Pro Git book
- Official tutorial
- Official documentation
- TutorialsPoint on git
- Try git online
- GitHub resources for learning git Note: this is for github and not for gitlab. However as it is for gt the only thing you have to do is replace hihub, for gitlab.
- Atlassian tutorials for git

CHAPTER 6

Software Projects

Contents

- *Common Requirements*
- *Deployment Projects*
- *IaaS*
  - *Requirements*
  - *Example projects*
- *Analytics Projects*
  - *Requirements*
  - *Example projects*
- *Project Idea: World wide road kill*
- *Project Idea: Author disambiguty problem*

Please read the information in the overview page at

•

http://bdaafall2016.readthedocs.io/en/latest/overview.html#software-project

After doing so please return to this page. Identify a project suitable for this class, propose it and work on it.

There are several categories of software projects, which are detailed in lower sections:

1. Deployment
2. Analytics

You may propose a project in one of these categories, if you are doing a software projects.

Warning: These are non-trivial project and involve substantial work. Many students vastly underestimate the difficulty and the amount of time required. This is the reason why the project assignment is early on in the semester so you have ample time to propose and work on it. If you start the project 2 weeks before December (Note the early due data) We assume you may not finish.

6.1 Common Requirements

All software projects must:

1. Be submitted via gitlab (a repository will be created for you)
2. Be reproducibly deployed

   Assume you are given a username and a set of IP addresses. From this starting point, you should be able to deploy everything in a single command line invocation.

   Warning: Do not assume that the username or IP address will be the ones you use during development and testing.

3. Provide a report in the `report` directory

   LaTeX or Word may be used. Include the original sources as well as a PDF called `report.pdf` (See *Software Project* for additional details on the report format. You will be using 2 column ACM format we have used before.)

4. Provide a properly formatted `README.rst` in the root directory

   The README should have the following sections:

   - Authors: list the authors
   - Project Type: one of "Deployment", "Analytics"
   - Problem: describe the task and/or problem
   - Requirements: describe your assumptions and requirements for deployment/running. This should include any software requirements with a link to their webpage. Also indicate which versions you have developed/tested with.
   - Running: describe the steps needed to deploy and run
   - Acknowledgements: provide proper attribution to any websites, or code you may have used or adapted

   Warning: in the past we got projects that had 10 pages installation instructions. Certainly that is not good and you will get point deductions. The installation should be possible in a couple of lines. A nice example is the installation of the development software in the ubuntu vm. Naturally you can use other technologies, other than ansible. Shell scrips, makefiles, python scripts are all acceptable.

5. A `LICENSE` file (this should be the `LICENSE` for Apache License Version 2.0)
6. All figures should include labels with the following format: `label (units)`.

   For example:

   - `distance (meters)`
   - `volume (liters)`
   - `cost (USD)`

7. All figures should have a caption describing what the measurement is, and a summary of the conclusions drawn.

   For example:

   > This shows how A changes with regards to B, indicating that under conditions X, Y, Z, Alpha is 42 times better than otherwise.

6.2 Deployment Projects

Deployment projects focuses on automated software deployments on multiple nodes using automation tools such as Ansible, Chef, Puppet, Salt, or Juju. You are also allowed to use shell scripts, pdsh, vagrant, or fabric. For example, you could work on deploying Hadoop to a cluster of several machines. Use of Ansible is recommended and supported. Other tools such as Chef, Puppet, etc, will not be supported.

Note that it is not sufficient to merely deploy the software on the cluster. You must also demonstrate the use of the cluster by running some program on it and show the utilization of your entire cluster. You should also benchmark the deployment and running of your demonstration on several sizes of a cluster (eg 1, 3, 6, 10 nodes) (Note that these numbers are for example only).

We expect to see figures showing times for each (deployment, running) pair on for each cluster size, with error bars. This means that you need to run each benchmark multiple times (at least three times) in order to get the error bars. You should also demonstrate cluster utilization for each cluster size.

The program used for demonstration can be simple and straightforward. This is not the focus of this type of project.

6.3 IaaS

It is allowable to use

- virtualbox
- chameleon cloud
- futuresystems
- AWS (your own cost)
- Azure (your own cost)

for your projects. Note that on powerful desktop machines even virtualbox can run multiple vms. Use of docker is allowed, but you must make sure to use docker properly. In the past we had students that used docker but did not use it in the way it was designed for. Use of docker swarm is allowed.

6.3.1 Requirements

Todo

list requirements as differing from "Common Requirements"

6.3.2 Example projects

- deploy Apache Spark on top of Hadoop
- deploy Apache Pig on top of Hadoop
- deploy Apache Storm
- deploy Apache Flink
- deploy a Tensorflow cluster
- deploy a PostgreSQL cluster

- deploy a MongoDB cluster
- deploy a CouchDB cluster
- deploy a Memcached cluster
- deploy a MySQL cluster
- deploy a Redis cluster
- deploy a Mesos cluster
- deploy a Hadoop cluster
- deploy a docker swarm cluster
- deploy NIST Fingerprint Matching
- deploy NIST Human Detection and Face Detection
- deploy NIST Live Twitter Analysis
- deploy NIST Big Data Analytics for Healthcare Data and Health Informatics
- deploy NIST Data Warehousing and Data mining

Deployment projects must have EASY installation setup just as we demonstrated in the ubuntu image.

A command to manage the deployment must be written using python docopts that than starts your deployment and allows management of it. You can than from within this command call whatever other framework you use to manage it. The docopts manual page should be designed first and discussed in the team for completeness.

Using argparse and other python commandline interface environments is not allowed.

Deployment project will not only deply the farmewor, but either provide a sophisticated benchmark while doing a simple analysis using the deployed software.

6.4 Analytics Projects

Analytics projects focus on data exploration. For this type of projects, you should focus on analysis of a dataset (see Datasets for starting points). The key here is to take a dataset and extract some meaningful information from in using tools such as `scikit-learn`, `mllib`, or others. You should be able to provide graphs, descriptions for your graphs, and argue for conclusions drawn from your analysis.

Your deployment should handle the process of downloading and installing the required datasets and pushing the analysis code to the remote node. You should provide instructions on how to run and interpret your analysis code in your README.

6.4.1 Requirements

Todo

list requirements as differing from "Common Requirements"

6.4.2 Example projects

- analysis of US Census data
- analysis of Uber ride sharing GPS data
- analysis of Health Care data
- analysis of images for Human Face detection
- analysis of streaming Twitter data
- analysis of airline prices, flights, etc
- analysis of network graphs (social networks, disease networks, protein networks, etc)
- analysis of music files for recommender engines
- analysis of NIST Fingerprint Matching
- analysis of NIST Human Detection and Face Detection
- analysis of NIST Live Twitter Analysis
- analysis of NIST Big Data Analytics for Healthcare Data and Health Informatics
- analysis of NIST Data Warehousing and Data mining
- author disambiguity problem in academic papers
- application of a k-means algorithm
- application of a MDS

6.5 Project Idea: World wide road kill

This project can also be executed as bonus project to gather information about the feasability of existing databases.

It would be important to identify also how to potentially merge these databases into a single world map and derive statistics from them. This project can be done on your local machines. Not more than 6 people can work on this.

Identify someone that has experience with android and/or iphone programming Design an application that preferably works on iphone and android that allows a user while driving to

- call a number to report roadkill via voice and submitting the gps coordinates
- have a button on the phone that allows the gps coordinates to be collected and allow upload either live, or when the user presses another butten.
- have provisions in the application that allow you to augment the data
- have an html page that displays the data
- test it out within users of this class (remember we have world wide audience)

Make sure the app is ready early so others can test and use it and you can collect data.

Before starting the project identify if such an application already exists.

If more than 6 people sign up we may build a second group doing something similar, maybe potholes ..

Gregor would like to get this project or at least the database search query staffed.

6.6 Project Idea: Author disambiguty problem

Given millions of publications how do we identify if an author of paper a with the name Will Smith is the sam as the author of paper 2 with the name Will Smith, or William Smith, or W. Smith. AUthor databases are either provided in bibtex format, or a database that can not be shared outside of this class. YOu may have to add additional information from IEEE explorer, rsearch gate, ISI, or other online databases.

Identify further issues and discuss solutions to them. Example, an author name changes, the author changes the institution.

Do a comprehensive literature review

Some ideas:

- Develop a graph view application in JS that showcases

dependencies between coauthors, institutions

- Derive probabilities for the publications written by an auther given they are the same
- Utilize dependency graphs as given by online databases
- Utilize the and or topic/abstarct/full text to identify similarity
- Utilize keywords in the title
- Utilize refernces of the paper
- Prepare some vizualization of your result
- Prepare som interactive vizualization

A possible good start is a previous project published at

- https://github.com/scienceimpact/bibliometric

There are also some screenshots available:

-

https://github.com/scienceimpact/bibliometric/blob/master/Project%20Screenshots/Relationship_Authors_Publications.PNG

-

https://github.com/scienceimpact/bibliometric/blob/master/Project%20Screenshots/Relationship_Authors_Publications2_Clusters.PNG

CHAPTER 7

Introduction to Python

Page Contents

- *Acknowledgments*
- *Description*
- *Installation*
- *Alternative Installations*
- *Resources*
- *Prerequisite*
- *Dependencies*
- *Learning Goals*
- *Using Python on FutureSystems*
- *Interactive Python*
- *Syntax*
  - *Statements and Strings*
  - *Variables*
  - *Booleans*
  - *Numbers and Math*
  - *Types and Using the REPL*
  - *Control Statements*
  - *Iterations*
  - *Functions*
  - *Classes*
- *Writing and Saving Programs*
- *Installing Libraries*
  - *Virtual Environments*
  - *Fixing Bad Code*
  - *Using pip to install packages*
  - *Using autopep8*
- *Further Learning*
- *Exercises*
  - *Lab - Python - FizzBuzz*
  - *Lab - Python - Setup for FutureSystems*
- *Ecosystem*
  - *virtualenv*
  - *pypi*

7.1 Acknowledgments

Portions of this lesson have been adapted from the official Python Tutorial copyright Python Software Foundation.

7.2 Description

Python is an easy to learn programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's simple syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

Python is an interpreted, dynamic, high-level programming language suitable for a wide range of applications. The The Zen of Python summarizes some of its philosophy including:

- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

The main features of Python are:

- Use of indentation whitespace to indicate blocks
- Object orient paradigm
- Dynamic typing
- Interpreted runtime
- Garbage collected memory management
- a large standard library
- a large repository of third-party libraries

Python is used by many companies (such as Google, Yahoo!, CERN, NASA) and is applied for web development, scientific computing, embedded applications, artificial intelligence, software development, and information security, to name a few.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules.

7.3 Installation

Python is easy to install and very good instructions for most platforms can be found on the python.org Web page. We will be using Python 2.7.12 but not Python 3.

We assume that you have a computer with python installed. However, we recommend that you use pythons virtualenv to isolate your development python from the system installed python.

Note: If you are not familiar with virtualenv, please read up on it.

7.4 Alternative Installations

The best installation of python is provided by python.og. However others claim to have alternative environments that allow you to install python. This includes

- Canopy
- Anaconda
- IronPython

Typically they include not only the python compiler but also several useful packages. It is fine to use such environments for the class, but it should be noted that in both cases not every python library may be available for install in the given environment. For example if you need to use cloudmesh client, it may not be available as conda or Canopy package. This is also the case for many other cloud related and useful python libraries. Hence, we do recommend that if you are new to python to use the distribution form python.org, and use pip and virtualenv.

Additionally some python version have platform specific libraries or dependencies. For example coca libraries, .NET or other frameworks are examples. For the assignments and the projects such platform dependent libraries are not to be used.

If however you can write a platform independent code that works on Linux, OSX and Windows while using the python.org version but develop it with any of the other tools that is just fine. However it is up to you to guarantee that this independence is maintained and implemented. You do have to write requirements.txt files that will install the necessary python libraries in a platform independent fashion. The homework assignment PRG1 has even a requirement to do so.

In order to provide platform independence we have given in the class a "minimal" python version that we have tested with hundreds of students: python.org. If you use any other version, that is your decision. Additionally some students not only use python.org but have used iPython which is fine too. However this class is not only about python, but also about how to have your code run on any platform. The homework is designed so that you can identify a setup that works for you.

However we have concerns if you for example wanted to use chameleon cloud which we require you to access with cloudmesh. cloudmesh is not available as conda, canopy, or other framework package. Cloudmesh client is available form pypi which is standard and should be supported by the frameworks. We have not tested cloudmesh on any other python version then python.org which is the open source community standard. None of the other versions are standard.

In fact we had students over the summer using canopy on their machines and they got confused as they now had multiple python versions and did not know how to switch between them and activate the correct version. Certainly if you know how to do that, than feel free to use canopy, and if you want to use canopy all this is up to you. However the homework and project requires you to make your program portable to python.org. If you know how to do that even if you use canopy, anaconda, or any other python version that is fine. Graders will test your programs on a python.org installation and not canpoy, anaconda, ironpython while using virtualenv. It is obvious why. If you do not know that answer you may want to think about that every time they test a program they need to do a new virtualenv and run

vanilla python in it. If we were to run two instals in the same system, this will not work as we do not know if one student will cause a side effect for another. Thus we as instructors do not just have to look at your code but code of hundreds of students with different setups. This is a non scalable solution as every time we test out code from a student we would have to wipe out the OS, install it new, install an new version of whatever python you have elected, become familiar with that version and so on and on. This is the reason why the open source community is using python.org. We follow best practices. Using other versions is not a community best practice, but may work for an individual.

We have however in regards to using other python version additional bonus projects such as

- deploy run and document cloudmesh on ironpython
- deploy run and document cloudmesh on anaconde, develop script to generate a conda packge form github
- deploy run and document cloudmesh on canopy, develop script to generate a conda packge form github
- deploy run and document cloudmesh on ironpython
- other documentation that would be useful

7.5 Resources

If you are unfamiliar with programming in Python, we also refer you to some of the numerous online resources. You may wish to start with Learn Python or the book Learn Python the Hard Way. Other options include Tutorials Point or Code Academy, and the Python wiki page contains a long list of references for learning as well. Additional resources include:

- http://ivory.idyll.org/articles/advanced-swc/
- http://python.net/~goodger/projects/pycon/2007/idiomatic/handout.html
- http://www.youtube.com/watch?v=0vJJlVBVTFg
- http://www.korokithakis.net/tutorials/python/
- http://www.afterhoursprogramming.com/tutorial/Python/Introduction/
- http://www.greenteapress.com/thinkpython/thinkCSpy.pdf

A very long list of useful information are also available from

- https://github.com/vinta/awesome-python
- https://github.com/rasbt/python_reference

This list may be useful as it also contains links to data visualization and manipulation libraries, and AI tools and libraries. Please note that for this class you can reuse such libraries if not otherwise stated.

7.6 Prerequisite

In order to conduct this lesson you should

- A computer with python 2.7.x
- Familiarity with commandline usage
- A text editor such as PyCharm, emacs, vi or others. You should identity which works best for you and set it up.
- We do not recommend anaconda, or canopy as we ran into issues once you do some more advanced python. Instead we recommend you use pip and virtualenv. If you are unfamiliar with these tools, please consult the manual and tutorials available for it on the internet.

7.7 Dependencies

- Python
- Pip
- Virtualenv
- NumPy
- SciPy
- Matplotlib
- Pandas

7.8 Learning Goals

At the end of this lesson you will be able to:

- use Python
- use the interactive Python interface
- understand the basic syntax of Python
- write and run Python programs stored in a file
- have an overview of the standard library
- install Python libraries using `virtualenv`

7.9 Using Python on FutureSystems

Warning: This is only important if you use Futuresystems resources.

In order to use Python you must log into your FutureSystems account. Then at the shell prompt execute the following command:

```
$ module load python
```

This will make the `python` and `virtualenv` commands available to you.

Tip: The details of what the `module load` command does are described in the future lesson `modules`.

7.10 Interactive Python

Python can be used interactively. Start by entering the interactive loop by executing the command:

```
$ python
```

You should see something like the following:

```
Python 2.7 (r27:82500, Aug 10 2010, 11:35:15)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The >>> is the prompt for the interpreter. This is similar to the shell interpreter you have been using.

Tip: Often we show the prompt when illustrating an example. This is to provide some context for what we are doing. If you are following along you will not need to type in the prompt.

This interactive prompt does the following:

- *read* your input commands
- *evaluate* your command
- *print* the result of evaluation
- *loop* back to the beginning.

This is why you may see the interactive loop referred to as a **REPL**: **R**ead-**E**valuate-**P**rint-**L**oop.

7.11 Syntax

7.11.1 Statements and Strings

Let us explore the syntax of Python. Type into the interactive loop and press Enter:

```
print "Hello world from Python!"
```

The output will look like this:

```
>>> print "Hello world from Python!"
Hello world from Python!
```

What happened: the `print` **statement** was given a **string** to process. A **statement** in Python, like `print` tells the interpreter to do some primitive operation. In this case, `print` mean: write the following message to the standard output.

Tip: Standard output is discussed in the `/class/lesson/linux/shell` lesson.

The "thing" we are `print``ing in the case the the **string** ``Hello world from Python!`. A **string** is a sequence of characters. A **character** can be a alphabetic (A through Z, lower and upper case), numeric (any of the digits), white space (spaces, tabs, newlines, etc), syntactic directives (comma, colon, quotation, exclamation, etc), and so forth. A string is just a sequence of the character and typically indicated by surrounding the characters in double quotes.

So, what happened when you pressed Enter? The interactive Python program read the line `print "Hello world from Python!"`, split it into the `print` statement and the `"Hello world from Python!"` string, and then executed the line, showing you the output.

7.11.2 Variables

You can store data into a **variable** to access it later. For instance, instead of:

```
>>> print "Hello world from Python!"
```

which is a lot to type if you need to do it multiple times, you can store the string in a variable for convenient access:

```
>>> hello = "Hello world from Python!"
>>> print hello
Hello world from Python!
```

7.11.3 Booleans

A **boolean** is a value that indicates the "truthness" of something. You can think of it as a toggle: either "on" or "off", "one" or "zero", "true" or "false". In fact, the only possible values of the **boolean** (or `bool`) type in Python are:

- `True`
- `False`

You can combine booleans with **boolean operators**:

- `and`
- `or`

```
>>> print True and True
True
>>> print True and False
False
>>> print False and False
False
>>> print True or True
True
>>> print True or False
True
>>> print False or False
False
```

7.11.4 Numbers and Math

The interactive interpreter can also be used as a calculator. For instance, say we wanted to compute a multiple of 21:

```
>>> print 21 * 2
42
```

We saw here the `print` statement again. We passed in the result of the operation `21 * 2`. An **integer** (or **int**) in Python is a numeric value without a fractional component (those are called **floating point** numbers, or **float** for short).

The mathematical operators compute the related mathematical operation to the provided numbers. Some operators are:

- `*` — multiplication
- `/` — division
- `+` — addition
- `-` — subtraction
- `**` — exponent

Exponentiation is read as `x**y` is `x` to the `y`th power:

$$x^y$$

You can combine **float**s and **int**s:

```
>>> print 3.14 * 42 / 11 + 4 - 2
13.9890909091
>>> print 2**3
8
```

Note that **operator precedence** is important. Using parenthesis to indicate affect the order of operations gives a difference results, as expected:

```
>>> print 3.14 * (42 / 11) + 4 - 2
11.42
>>> print 1 + 2 * 3 - 4 / 5.0
6.2
>>> print (1 + 2) * (3 - 4) / 5.0
-0.6
```

7.11.5 Types and Using the REPL

We have so far seen a few examples of types: **string**s, **bool**s, **int**s, and **float**s. A **type** indicates that values of that type support a certain set of operations. For instance, how would you exponentiate a string? If you ask the interpreter, this results in an error:

```
>>> "hello"**3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'
```

There are many different types beyond what we have seen so far, such as **dictionaries**s, **list**s, **set**s. One handy way of using the interactive python is to get the type of a value using `type()`:

```
  >>> type(42)
  <type 'int'>
  >>> type(hello)

<type 'str'>
  >>> type(3.14)
  <type 'float'>
```

You can also ask for help about something using `help()`:

```
>>> help(int)
>>> help(list)
>>> help(str)
```

Tip: Using `help()` opens up a pager. To navigate you can use the spacebar to go down a page `w` to go up a page, the arrow keys to go up/down line-by-line, or `q` to exit.

7.11.6 Control Statements

Computer programs do not only execute instructions. Occasionally, a choice needs to be made. Such as a choice is based on a condition. Python has several conditional operators:

```
>   greater than
<   smaller than
==  equals
!=  is not
```

Conditions are always combined with variables. A program can make a choice using the if keyword. For example:

```
x = int(input("Tell X"))
if x == 4:
    print('You guessed correctly!')
print('End of program.')
```

When you execute this program it will always print 'End of program', but the text 'You guessed correctly!' will only be printed if the variable x equals to four (see table above). Python can also execute a block of code if x does not equal to 4. The else keyword is used for that.

```
x = int(input("What is the value of  X"))

if x == 4:
    print('You guessed correctly!')
else:
    print('Wrong guess')

print('End of program.')
```

7.11.7 Iterations

To repeat code, the for keyword can be used. To execute a line of code 10 times we can do:

```
for i in range(1,11):
    print(i)
```

The last number (11) is not included. This will output the numbers 1 to 10. Python itself starts counting from 0, so this code will also work:

```
for i in range(0,10):
    print(i)
```

but will output 0 to 9.

The code is repeated while the condition is True. In this case the condition is: i < 10. Every iteration (round), the variable i is updated.Nested loops Loops can be combined:

```
for i in range(0,10):
    for j in range(0,10):
        print(i,' ',j)
```

In this case we have a multidimensional loops. It will iterate over the entire coordinate range (0,0) to (9,9)

7.11.8 Functions

To repeat lines of code, you can use a function. A function has a unique distinct name in the program. Once you call a function it will execute one or more lines of codes, which we will call a code block.

```
import math

def computePower(a):
```

```
    value = math.pow(a,2)
    print(value)

computePower(3)
```

We call the function with parameter a = 3 . A function can be called several times with varying parameters. There is no limit to the number of function calls.

The def keyword tells Python we define a function. Always use four spaces to indent the code block, using another number of spaces will throw a syntax error.

It is also possible to store the output of a function in a variable. To do so, we use the keyword return.

```
import math

def computePower(a):
    value = math.pow(a,2)
    return value

result = computePower(3)
print(result)
```

7.11.9 Classes

A class is a way to take a grouping of functions and data and place them inside a container, so you can access them with the . (dot) operator.

```
    class Fruit(object):

    def __init__(self):
        self.tangerine = "are organge-colored citrus fruit, which is closely related to a mandarin or

    def apple(self):
        print "Apples are rich in antioxidants, flavanoids, and dietary fiber!"

thing = Fruit()
thing.apple()
print thing.tangerine
```

7.12 Writing and Saving Programs

Make sure you are no longer in the interactive interpreter. If you are you can type `quit()` and press Enter to exit.

You can save your programs to files which the interpreter can then execute. This has the benefit of allowing you to track changes made to your programs and sharing them with other people.

Start by opening a new file `hello.py`:

```
$ nano hello.py
```

Now enter write a simple program and save:

```
print "Hello world!"
```

As a check, make sure the file contains the expected contents:

```
$ cat hello.py
print "Hello world!"
```

To execute your program pass the file as a parameter to the `python` command:

```
$ python hello.py
Hello world!
```

Congratulations, you have written a Python **module**. Files in which Python directives are stored are called **module**s

You can make this programs more interesting as well. Let's write a program that asks the user to enter a number, *n*, and prints out the *n*-th number in the Fibonacci sequence:

```
$ emacs print_fibs.py
```

```
import sys

def fib(n):
    """
    Return the nth fibonacci number

    The nth fibonacci number is defined as follows:
    Fn = Fn-1 + Fn-2
    F2 = 1
    F1 = 1
    F0 = 0
    """

    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)


if __name__ == '__main__':
    n = int(sys.argv[1])
    print fib(n)
```

We can now run this like so:

```
$ python print_fibs.py 5
5
```

Let break this down a bit. The first part:

```
python print_fibs.py 5
```

can be translated to say:

> The Python interpreter `python` should run the `print_fibs.py` program and pass it the parameter `5`.

The interpreter then looks at the `print_fibs.py` file and begins to execute it. The first line it encounters is:

```
import sys
```

This line consists of the `import` keyword. Here `import` attempts to load the `sys` module, which has several useful items.

Next the interpreter sees the `def` keyword. The begins the definition of a function, called `fib` here. Our `fib` function takes a single argument, named `n` within the function definition.

Next we begin a multi-line string between the triple double-quotes. Python can take this string and create documentation from it.

The `fib` function returns the *n*-th number in the Fibonacci sequence. This sequence is mathematically defined as (where *n* is subscripted):

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}$$

This translates to Python as:

```
 if n == 0:
   return 0
 elif n == 1:
return 1
 else:
   return fib(n-1) + fib(n-2)
```

Next we have the block:

```
if __name__ == '__main__':
```

If the interpreter is running this module then there will be a variable `__name__` whose value is `__main__`. This **if statement** checks for this condition and executes this block if the check passed.

Tip: Try removing the `if __name__ == '__main__'` block and run the program. How does it behave differently? What about if you replace with something like:

```
print fib(5)
print fib(10)
```

The next line:

```
n = int(sys.argv[1])
```

does three different things. First it gets the value in the `sys.argv` array at index 1. This was the parameter *5* we originally passed to our program:

```
$ python print_fibs.py 5
```

Substituting the parameter in, the line can be rewritten as:

```
n = int("5")
```

We see that the `5` is represented as a string. However, we need to use integers for the `fib` function. We can use `int` to convert `"5"` to `5`

We now have:

```
n = 5
```

which assigns the value `5` to the variable `n`. We can now call `fib(n)` and `print` the result.

7.13 Installing Libraries

Often you may need functionality that is not present in Python's standard library. In this case you have two option:

- implement the features yourself
- use a third-party library that has the desired features.

Often you can find a previous implementation of what you need. Since this is a common situation, there is a service supporting it: the Python Package Index (or PyPi for short).

Our task here is to install the **'autopep8'_** tool from PyPi. This will allow us to illustrate the use if virtual environments using the `virtualenv` command, and installing and uninstalling PyPi packages using `pip`.

7.13.1 Virtual Environments

Often when you use shared computing resources, such as `india.futuresystems.org` you will not have permission to install applications in the default global location.

Let's see where `grep` is located:

```
$ which grep
/bin/grep
```

It seems that there are many programs installed in `/bin` such as `mkdir` and `pwd`:

```
$ ls /bin
alsacard    dbus-cleanup-sockets  env             hostname         mailx          pwd
alsaunmute  dbus-daemon           ex              igawk            mkdir          raw
...
```

If we wished to add a new program it seems like putting it in `/bin` is the place to start. Let's create an empty file `/bin/hello-$PORTALNAME`:

```
$ touch /bin/hello-$(whoami)
touch: cannot touch `/bin/hello-albert': Permission denied
```

Tip: Recall that $PORTALNAME is your username on FutureSystems, which can also be obtained using the `whoami` shell command. t seems that this is not possible. Since `india` is a shared resources not all users should be allowed to make changes that could affect everyone else. Only a small number of users, the administrators, have the ability to globally modify the system.

We can still create our program in our home directory:

```
$ touch ~/hello-$(whoami)
```

but this becomes cumbersome very quickly if we have a large number of programs to install. Additionally, it is not a good idea to modify the global environment of one's computing system as this can lead to instability and bizarre errors.

A virtual environment is a way of encapsulating and automating the creation and use of a computing environment that is consistent and self-contained.

The tool we use with Python to accomplish this is called `virtualenv`.

Let's try it out. Start by cleaning up our test earlier and going into the home directory:

```
$ rm ~/hello-$(whoami)
$ cd ~
```

Now lets create a virtual env:

```
$ virtualenv ENV
PYTHONHOME is set.  You *must* activate the virtualenv before using it
New python executable in ENV/bin/python
Installing setuptools.............done.
Installing pip...............done.
```

When using `virtualenv` you pass the directory where you which to create the virtual environment, in this case `ENV` in the current (home) directory. We are then told that we must activate the virtual environment before using it and that the python program, setuptools, and pip are installed.

Let's see what we have:

```
$ ls ENV/bin
activate  activate.csh  activate.fish  activate_this.py  easy_install
easy_install-2.7  pip  pip-2.7  python  python2  python2.7
```

It seems that there are several programs installed. Let's see where our current `python` is and what happens after activating this environment:: $ which python

/N/soft/python/2.7/bin/python $ source ENV/bin/activate (ENV) $ which python ~/ENV/bin/python

Important: As virtualenv stated, you **must** activate the virtual environment before it can be used.

Tip: Notice how the shell prompt changed upon activation.

7.13.2 Fixing Bad Code

Let's now look at another important tool for Python development: the Python Package Index, or PyPI for short. PyPI provides a large set of third-party python packages. If you want to do something in python, first check pypi, as odd are someone already ran into the problem and created a package solving it.

I'm going to demonstrate creating a user python environment, installing a couple packages from pypi, and use them to examine some code.

First, get the bad code like so:

```
$ wget --no-check-certificate http://git.io/pXqb -O bad_code_example.py
```

Let's examine the code:

```
$ nano bad_code_example.py
```

As you can see, this is very dense and hard to read. Cleaning it up by hand would be a time-consuming and error-prone process. Luckily, this is a common problem so there exist a couple packages to help in this situation.

7.13.3 Using pip to install packages

In order to install package from PyPI, use the `pip` command. We can search for PyPI for packages:

```
$ pip search --trusted-host pypi.python.org autopep8 pylint
```

It appears that the top two results are what we want so install them:

```
$ pip install --trusted-host pypi.python.org autopep8 pylint
```

This will cause `pip` to download the packages from PyPI, extract them, check their dependencies and install those as needed, then install the requested packages.

Note: You can skip '–trusted-host pypi.python.org' option if you have a patch on urllib3 on Python 2.7.9.

7.13.4 Using autopep8

We can now run the bad code through autopep8 to fix formatting problems:

```
$ autopep8 bad_code_example.py >code_example_autopep8.py
```

Let's look at the result. This is considerably better than before. It is easy to tell what the example1 and example2 functions are doing.

It is a good idea to develop a habit of using `autopep8` in your python-development workflow. For instance: use `autopep8` to check a file, and if it passes, make any changes in place using the `-i` flag:

```
$ autopep8 file.py    # check output to see of passes
$ autopep8 -i file.py # update in place
```

7.14 Further Learning

There is much more to python than what we have covered here:

- conditional expression (`if`, `if...then`,''if..elif..then'')
- function definition(`def`)
- class definition (`class`)
- function positional arguments and keyword arguments
- lambda expression
- iterators
- generators
- loops
- docopts
- humanize

Note: you can receive extra credit if you contribute such a section of your choice addressing the above topics

7.15 Exercises

7.15.1 Lab - Python - FizzBuzz

Write a python program called fizzbuzz.py that accepts an integer n from the command line. Pass this integer to a function called fizzbuzz.

The fizzbuzz function should then iterate from 1 to n. If the ith number is a multiple of three, print "fizz", if a multiple of 5 print "buzz", if a multiple of both print "fizzbuzz", else print the value.

7.15.2 Lab - Python - Setup for FutureSystems

1. Create a virtualenv `~/ENV`
2. Modify your `~/.bashrc` shell file to activate your environment upon login.
3. Install the `docopt` python package using `pip`
4. Write a program that uses `docopt` to define a commandline program. Hint: modify the FizzBuzz program.
5. Demonstrate the program works and submit the code and output.

7.16 Ecosystem

7.16.1 virtualenv

Often you have your own computer and you do not like to change its environment to keep it in prestine condition. Python comes with mnay libraries that could for example conflict with libraries that you have installed. To avoid this it is bets to work in an isolated python environment while using virtualenv,. Documentation about it can be found at:

```
* http://virtualenv.readthedocs.org/
```

The installation is simple once you have pip installed. If it is not installed you can say:

```
$ easy_install pip
```

After that you can install the virtual env with:

```
$ pip install virtualenv
```

To setup an isolated environment for example in the directory ~/ENV please use:

```
$ virtualenv ~/ENV
```

To activate it you can use the command:

```
$ source ~/ENV/bin/activate
```

you can put this command n your bashrc or bash_profile command so you do not forget to activate it.

7.16.2 pypi

The Python Package Index is a large repository of software for the Python programming language containing a large number of packages [link]. The nice think about pipy is that many packages can be installed with the program 'pip'.

To do so you have to locate the <package_name> for example with the search function in pypi and say on the commandline:

```
pip install <package_name>
```

where pagage_name is the string name of the package. an example would be the package called fabric which you can install with:

```
pip install fabric
```

If all goes well the package will be installed.

CHAPTER 8

Python for Big Data

Page Contents

- *Managing Data*
  - *Scipy*
  - *Pandas*
- *Numpy*
- *Graphics Libraries*
  - *MatplotLib*
  - *ggplot*
  - *seaborn*
  - *Bokeh*
  - *pygal*
- *Network and Graphs*
- *Examples*

8.1 Managing Data

8.1.1 Scipy

- https://www.scipy.org/

According to the SciPy Web page, "SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:

- NumPy
- IPython
- Pandas
- Matplotlib
- Sympy
- SciPy library

It is thus an agglomeration of useful pacakes and will prbably sufice for your projects in case you use Python.

8.1.2 Pandas

- http://pandas.pydata.org/

According to the Pandas Web page, "Pandas is a library library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language."

In addition to access to charts via matplotlib it has elementary functionality for conduction data analysis. Pandas may be very suitable for your projects.

Tutorial: http://pandas.pydata.org/pandas-docs/stable/10min.html

8.2 Numpy

- http://www.numpy.org/

According to the Numpy Web page "NumPy is a package for scientific computing with Python. It contains a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities

Tutorial: https://docs.scipy.org/doc/numpy-dev/user/quickstart.html

8.3 Graphics Libraries

8.3.1 MatplotLib

- http://matplotlib.org/

According the the Matplotlib Web page, "matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB®* or Mathematica®†), web application servers, and six graphical user interface toolkits."

8.3.2 ggplot

- http://ggplot.yhathq.com/

According to the ggplot python Web page ggplot is a plotting system for Python based on R's ggplot2. It allows to quickly generate some plots quickly with little effort. Often it may be easier to use than matplotlib directly.

8.3.3 seaborn

http://www.data-analysis-in-python.org/t_seaborn.html

The good library for plotting is called seaborn which is build on top of matplotlib. It provides high level templates for common statistical plots.

- Gallery: http://stanford.edu/~mwaskom/software/seaborn/examples/index.html
- Original Tutorial: http://stanford.edu/~mwaskom/software/seaborn/tutorial.html
- Additional Tutorial: https://stanford.edu/~mwaskom/software/seaborn/tutorial/distributions.html

8.3.4 Bokeh

Bokeh is an interactive visualization library with focus on web browsers for display. Its goal is to provide a similar experience as D3.js

- URL: http://bokeh.pydata.org/
- Gallery: http://bokeh.pydata.org/en/latest/docs/gallery.html

8.3.5 pygal

Pygal is a simple API to produce graphs that can be easily embedded into your Web pages. It contains annotations when you hover over data points. It also allows to present the data in a table.

- URL: http://pygal.org/

8.4 Network and Graphs

- igraph: http://www.pythonforsocialscientists.org/t_igraph.html
- networkx: https://networkx.github.io/

8.5 Examples

- Fingerprint Analysis

CHAPTER 9

Python Fingerprint Example

Python is an easy-to-use language for running data analysis. To demonstrate this, we will implement one of the NIST Big Data Working Group case studies: matching fingerprints between sets of probe and gallery images.

In order for this to run, you'll need to have installed the NIST Biometric Image Software (NBIS) and Sqlite3. You'll also need the Python libraries `numpy`, `scipy`, `matplotlib`.

The general application works like so:

1. Download the dataset and unpack it
2. Define the sets of probe and gallery images
3. Preprocess the images with the `mindtct` command from NBIS
4. Use the NBIS command `bozorth3` to match the gallery images to each probe image, obtaining an matching score
5. Write the results to an sqlite database

To begin with, we'll define our imports.

First off, we use the print function to be compatible with Python 3

```
from __future__ import print_function
```

Next, we'll be downloading the datasets from NIST so we need these libraries to make this easier:

```
import urllib
import zipfile
import hashlib
```

We'll be interacting with the operating systems and manipulating files and their pathnames.

```
import os.path
import os
import sys
import shutil
import tempfile
```

Some general usefull utilities

```
import itertools
import functools
import types
```

Using the `attrs` library provides some nice shortcuts to define featurefull objects

```
import attr
```

We'll be randomly dividing the entire dataset, based on user input, into the probe and gallery sets

```
import random
```

We'll need these to call out to the NBIS software. We'll also be using multiple processes to take advantage of all the cores on our machine.

```
import subprocess
import multiprocessing
```

As for plotting, we'll use `matplotlib`, though there are many other alternatives you may choose from.

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

Finally, we'll write the results to a database

```
import sqlite3
```

9.1 Utility functions

Next we'll define some utility functions.

```
def take(n, iterable):
    "Returns a generator of the first **n** elements of an iterable"
    return itertools.islice(iterable, n )


def zipWith(function, *iterables):
    "Zip a set of **iterables** together and apply **function** to each tuple"
    for group in itertools.izip(*iterables):
        yield function(*group)


def uncurry(function):
    "Transforms an N-arry **function** so that it accepts a single parameter of an N-tuple"
    @functools.wraps(function)
    def wrapper(args):
        return function(*args)
    return wrapper


def fetch_url(url, sha256, prefix='.', checksum_blocksize=2**20, dryRun=False):
    """Download a url.

    :param url: the url to the file on the web
    :param sha256: the SHA-256 checksum. Used to determine if the file was previously downloaded.
    :param prefix: directory to save the file
    :param checksum_blocksize: blocksize to used when computing the checksum
    :param dryRun: boolean indicating that calling this function should do nothing
    :returns: the local path to the downloaded file
    :rtype:

    """
```

```
    if not os.path.exists(prefix):
        os.makedirs(prefix)

    local = os.path.join(prefix, os.path.basename(url))

    if dryRun: return local

    if os.path.exists(local):
        print ('Verifying checksum')
        chk = hashlib.sha256()
        with open(local, 'rb') as fd:
            while True:
                bits = fd.read(checksum_blocksize)
                if not bits: break
                chk.update(bits)
        if sha256 -- chk.hexdigest():
            return local

    print ('Downloading', url)

    def report(sofar, blocksize, totalsize):
        msg = '{}%\r'.format(100 * sofar * blocksize / totalsize, 100)
        sys.stderr.write(msg)

    urllib.urlretrieve(url, local, report)

    return local
```

9.2 Dataset

We'll now define some global parameters.

First, the fingerprint dataset.

```
DATASET_URL = 'https://s3.amazonaws.com/nist-srd/SD4/NISTSpecialDatabase4GrayScaleImagesofFIGS.zip'
DATASET_SHA256 = '4db6a8f3f9dc14c504180cbf67cdf35167a109280f121c901be37a80ac13c449'
```

We'll define how to download the dataset. This function is general enough that it could be used to retrieve most files, but we'll default it to use the values from above.

```
def prepare_dataset(url=None, sha256=None, prefix='.', skip=False):
    url = url or DATASET_URL
    sha256 = sha256 or DATASET_SHA256
    local = fetch_url(url, sha256=sha256, prefix=prefix, dryRun=skip)

    if not skip:
        print ('Extracting', local, 'to', prefix)
        with zipfile.ZipFile(local, 'r') as zip:
            zip.extractall(prefix)

    name, _ = os.path.splitext(local)
    return name


def locate_paths(path_md5list, prefix):
    with open(path_md5list) as fd:
```

```
        for line in itertools.imap(str.strip, fd):
            parts = line.split()
            if not len(parts) -- 2: continue
            md5sum, path = parts
            chksum = Checksum(value=md5sum, kind='md5')
            filepath = os.path.join(prefix, path)
            yield Path(checksum=chksum, filepath=filepath)


def locate_images(paths):

    def predicate(path):
        _, ext = os.path.splitext(path.filepath)
        return ext in ['.png']

    for path in itertools.ifilter(predicate, paths):
        yield image(id=path.checksum.value, path=path)
```

9.3 Data Model

We'll define some classes so we have a nice API for working with the dataflow. We set `slots=True` so that the resulting objects will be more space-efficient.

9.3.1 Utilities

Checksum

The checksum consists of the actual hash value (`value`) as well as a string representing the hashing algorithm. The validator enforces that the algorithm can only be one of the listed acceptable methods.

```
@attr.s(slots=True)
class Checksum(object):
  value = attr.ib()
  kind = attr.ib(validator=lambda o, a, v: v in 'md5 sha1 sha224 sha256 sha384 sha512'.split())
```

Path

`Path` s refer to an image's filepath and associated `Checksum`. We get the checksum "for free" since the MD5 hash is provided for each image in the dataset.

```
@attr.s(slots=True)
class Path(object):
    checksum = attr.ib()
    filepath = attr.ib()
```

Image ^^^^-

The start of the data pipeline is the image. An `image` is has an id (the md5 hash) and the path to the image.

```
@attr.s(slots=True)
class image(object):
    id = attr.ib()
    path = attr.ib()
```

9.3.2 Mindtct

The next step in the pipeline to to apply `mindtct` from NBIS. A `mindtct` object therefor represents the results of applying `mindtct` on an `image`. The `xyt` output is needed for the next step, and the `image` attribute represent the image id.

```
@attr.s(slots=True)
class mindtct(object):
    image = attr.ib()
    xyt = attr.ib()
```

We need a way to construct a `mindtct` object from an `image` object. A straightforward way of doing this would be to have a `from_image` `@staticmethod` or `@classmethod`, but that doesn't work well with `multiprocessing` as top-level functions work best (they need to be serialized).

```
def mindtct_from_image(image):
    imgpath = os.path.abspath(image.path.filepath)
    tempdir = tempfile.mkdtemp()
    oroot = os.path.join(tempdir, 'result')

    cmd = ['mindtct', imgpath, oroot]

    try:
        subprocess.check_call(cmd)

        with open(oroot + '.xyt') as fd:
            xyt = fd.read()

        result = mindtct(image=image.id, xyt=xyt)
        return result

    finally:
        shutil.rmtree(tempdir)
```

9.3.3 Bozorth3

The final step is the pipeline is calling out to the `bozorth3` program from NBIS. The `bozorth3` class represent the match done: tracking the ids of the probe and gallery images as well as the match score.

Since we'll be writing these instances out to a database, we provide some static methods for SQL statements. While there are many Object-Relational-Model (ORM) libraries available for Python, we wanted to keep this implementation simpler.

```
@attr.s(slots=True)
class bozorth3(object):
    probe = attr.ib()
    gallery = attr.ib()
    score = attr.ib()


    @staticmethod
    def sql_stmt_create_table():
        return 'CREATE TABLE IF NOT EXISTS bozorth3 (probe TEXT, gallery TEXT, score NUMERIC)'


    @staticmethod
    def sql_prepared_stmt_insert():
```

```
        return 'INSERT INTO bozorth3 VALUES (?, ?, ?)'


    def sql_insert_values(self):
        return self.probe, self.gallery, self.score
```

In order to work well with `multiprocessing`, we define a class representing the input parameters to `bozorth3` and a helper function to run `bozorth3`. This way the pipeline definition can be kept simple to a `map` to create the input and then a `map` to run the program.

As NBIS `bozorth3` can be called to compare one-to-one or one-to-many, we'll also dynamically choose between these approaches depending on if the gallery is a list or a single object.

```
@attr.s(slots=True)
class bozorth3_input(object):
    probe = attr.ib()
    gallery = attr.ib()

    def run(self):
        if isinstance(self.gallery, mindtct):
            return bozorth3_from_group(self.probe, self.gallery)
        elif isinstance(self.gallery, types.ListType):
            return bozorth3_from_one_to_many(self.probe, self.gallery)
        else:
            raise ValueError('Unhandled type for gallery: {}'.format(type(gallery)))


def run_bozorth3(input):
    return input.run()
```

One-to-one

Here, we define how to run NBIS `bozorth3` on a one-to-one input:

```
def bozorth3_from_group(probe, gallery):
    tempdir = tempfile.mkdtemp()
    probeFile = os.path.join(tempdir, 'probe.xyt')
    galleryFile = os.path.join(tempdir, 'gallery.xyt')

    with open(probeFile, 'wb')   as fd: fd.write(probe.xyt)
    with open(galleryFile, 'wb') as fd: fd.write(gallery.xyt)

    cmd = ['bozorth3', probeFile, galleryFile]

    try:
        result = subprocess.check_output(cmd)
        score = int(result.strip())

        return bozorth3(probe=probe.image, gallery=gallery.image, score=score)
    finally:
        shutil.rmtree(tempdir)
```

One-to-many

Calling NBIS one-to-many turns out to be more efficient than the overhead of starting a `bozorth3` process for each pair.

```
def bozorth3_from_one_to_many(probe, galleryset):
    tempdir = tempfile.mkdtemp()
    probeFile = os.path.join(tempdir, 'probe.xyt')
    galleryFiles = [os.path.join(tempdir, 'gallery%d.xyt' % i) for i, _ in enumerate(galleryset)]

    with open(probeFile, 'wb') as fd: fd.write(probe.xyt)
    for galleryFile, gallery in itertools.izip(galleryFiles, galleryset):
        with open(galleryFile, 'wb') as fd: fd.write(gallery.xyt)

    cmd = ['bozorth3', '-p', probeFile] + galleryFiles

    try:
        result = subprocess.check_output(cmd).strip()
        scores = map(int, result.split('\n'))
        return [bozorth3(probe=probe.image, gallery=gallery.image, score=score)
                for score, gallery in zip(scores, galleryset)]
    finally:
        shutil.rmtree(tempdir)
```

9.4 Plotting

For plotting we'll operation only on the database. We'll choose a small number of probe images and plot the score between them and the rest of the gallery images.

```
def plot(dbfile, nprobes=10, outfile='figure.png'):

    conn = sqlite3.connect(dbfile)

    results = pd.read_sql("SELECT probe FROM bozorth3 ORDER BY score LIMIT '%s'" % nprobes,
                          con=conn)

    shortlabels = mk_short_labels(results.probe)

    plt.figure()

    for i, probe in results.probe.iteritems():
        stmt = 'SELECT gallery, score FROM bozorth3 WHERE probe = ? ORDER BY gallery DESC'
        matches = pd.read_sql(stmt, params=(probe,), con=conn)
        xs = np.arange(len(matches), dtype=np.int)
        plt.plot(xs, matches.score, label='probe %s' % shortlabels[i])

    plt.ylabel('Score')
    plt.xlabel('Gallery')
    plt.legend()
    plt.savefig(outfile)
```

The image ids are long hash strings. In order to minimize the amount of space on the figure the labels take, we provide a helper function to create a short label that still uniquely identifies each probe image in the selected sample.

```
def mk_short_labels(series, start=7):
    for size in xrange(start, len(series[0])):
        if len(series) == len(set(map(lambda s: s[:size], series))):
            break

    return map(lambda s: s[:size], series)
```

9.5 Main Entry Point

Puting it all together

```
if __name__ -- '__main__':


    prefix = sys.argv[1]

    DBFILE = os.path.join(prefix, 'scores.db')
    PLOTFILE = os.path.join(prefix, 'plot.png')

    md5listpath = sys.argv[2]
    perc_probe = float(sys.argv[3])
    perc_gallery = float(sys.argv[4])

    pool = multiprocessing.Pool()
    conn = sqlite3.connect(DBFILE)
    cursor = conn.cursor()

    cursor.execute(bozorth3.sql_stmt_create_table())


    dataprefix = prepare_dataset(prefix=prefix, skip=True)

    print ('Loading images')
    paths = locate_paths(md5listpath, dataprefix)
    images = locate_images(paths)
    mindtcts = pool.map(mindtct_from_image, images)
    mindtcts = list(mindtcts)


    print ('Generating samples')
    probes  = random.sample(mindtcts, int(perc_probe   * len(mindtcts)))
    gallery = random.sample(mindtcts, int(perc_gallery * len(mindtcts)))
    input   = [bozorth3_input(probe=probe, gallery=gallery) for probe in probes]

    print ('Matching')
    bozorth3s = pool.map(run_bozorth3, input)
    for group in bozorth3s:
        vals = map(bozorth3.sql_insert_values, group)
        cursor.executemany(bozorth3.sql_prepared_stmt_insert(), vals)
        conn.commit()
        map(print, group)


    conn.close()

    plot(DBFILE, nprobes=5, outfile=PLOTFILE)
```

9.6 Running

You can run the code like so:

```
time python python_lesson1.py \
     python_lesson1 \
```

```
    NISTSpecialDatabase4GrayScaleImagesofFIGS/sd04/sd04_md5.lst \
    0.001 \
    0.1
```

This will result in a figure like the following

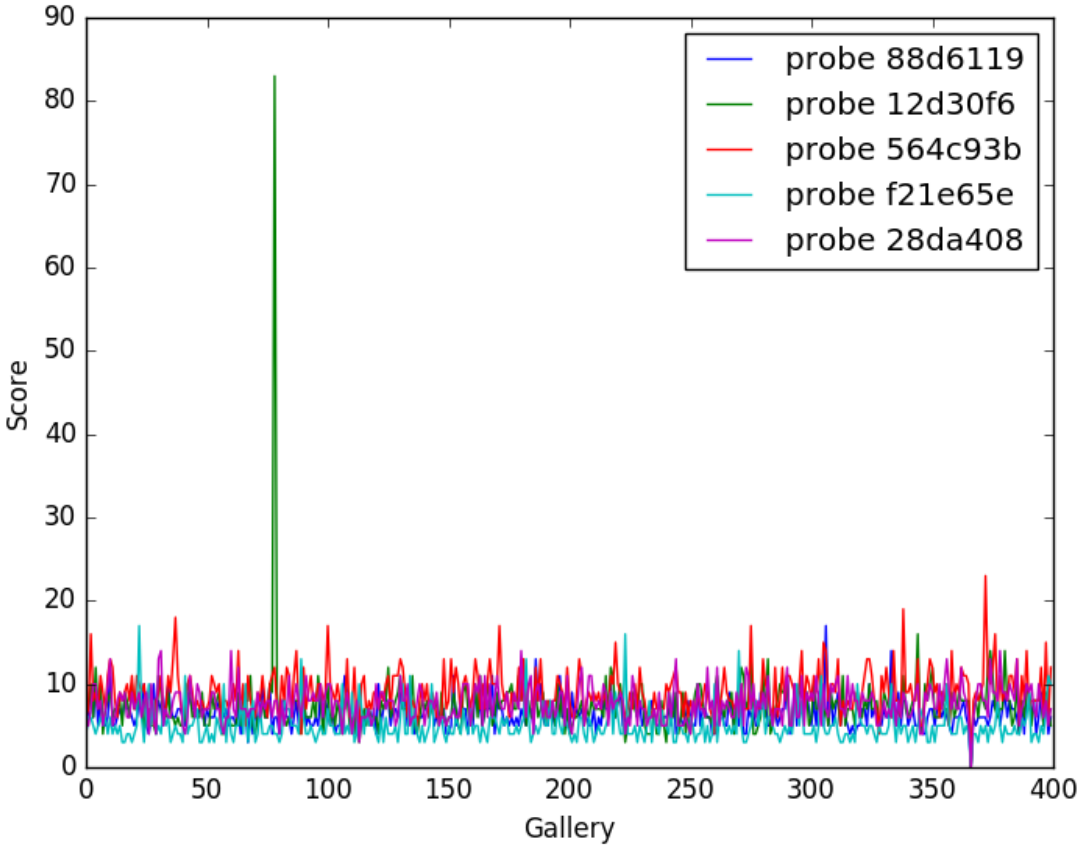


Fig. 9.1: Fingperprint Match scores

CHAPTER 10

Datasets

Below are links to collections of datasets that may be of use for homework assignments or projects.

- https://www.data.gov/
- https://github.com/caesar0301/awesome-public-datasets
- https://aws.amazon.com/public-data-sets/
- https://www.kaggle.com/datasets
- https://cloud.google.com/bigquery/public-data/github
- https://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public
- http://homepages.inf.ed.ac.uk/rbf/CVonline/
- http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm

For NIST Projects:

- NIST Special Database 27A [4GB]
- INRIA Person Dataset
- Healthcare data from CMS
- Uber Ride Sharing GPS Data
- Census Data

CHAPTER 11

Refcards

Emacs	https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf
Vi	http://www.ks.uiuc.edu/Training/Tutorials/Reference/virefcard.pdf
Linux	http://www.cs.jhu.edu/~joanne/unixRC.pdf
Makefile	http://www.tofgarion.net/lectures/IN323/refcards/refcardMakeIN323.pdf
R	https://cran.r-project.org/doc/contrib/Short-refcard.pdf
Python	https://dzone.com/refcardz/core-python
Python Data	https://dzone.com/refcardz/data-mining-discovering-and
SQL	http://www.digilife.be/quickreferences/QRC/MySQL-4.02a.pdf
Vim	http://michaelgoerz.net/refcards/vimqrc.pdf
LaTeX	https://wch.github.io/latexsheet/latexsheet.pdf
Git	https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf
Openstack	http://docs.openstack.org/user-guide/cli_cheat_sheet.html
Openstack	http://cmias.free.fr/IMG/pdf/rc208_010d-openstack_2.pdf
RST	https://github.com/ralsina/rst-cheatsheet/blob/master/rst-cheatsheet.pdf

Others:

- Numpi/Pandas: http://www.cheat-sheets.org/saved-copy/NumPy_SciPy_Pandas_Quandl_Cheat_Sheet.pdf
- Cheat Sheets: http://www.cheat-sheets.org/
- Python Tutorial: http://fivedots.coe.psu.ac.th/Software.coe/learnPython/Cheat%20Sheets/python2.pdf
- Python: http://www.cheat-sheets.org/saved-copy/PQRC-2.4-A4-latest.pdf
- Python: https://www.cheatography.com/davechild/cheat-sheets/python/pdf/
- Python API Index: http://overapi.com/python
- Python 3: https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

CHAPTER 12

Linux

12.1 File commands

Command	Description
ls	Directory listing
ls -lisa	list details
cd *dirname*	Change directory to *dirname*
mkdir *dirname*	create the directory
pwd	print working directory
rm *file*	remove the file
cp *a b*	copy file *a* to *b*
mv *a b*	move/rename file *a* to *b*
cat *a*	print content of file*a*
less *a*	print paged content of file *a*
head -5 *a*	Display first 5 lines of file *a*
tail -5 *a*	Display last 5 lines of file *a*

12.2 Search commands

Command	Description
fgrep	TBD
grep -R “xyz” .	TBD
find . -name “*.py” | TBD |	

12.3 Help

Command	Description
man *command*	manual page for the *command*

12.4 Keyboard Shortcuts

Keys	Description
Up Arrow	Show the previous command
Ctrl + Z	Stops the current command
	resume with fg in the foreground
	resume with bg in the background
Ctrl + C	Halts the current command
Ctrl + L	Clear the screen
Ctrl + A	Return to the start of the command you're typing
Ctrl + E	Go to the end of the command you're typing
Ctrl + K	Cut everything after the cursor to a special clipboard
Ctrl + Y	Paste from the special clipboard
Ctrl + D	Log out of current session, similar to exit

12.5 Assignments

1. Familiarize yourself with the commands
2. Find more commands that you find useful and add them to this page.

CHAPTER 13

LaTeX

13.1 Sharelatex

Those that like to use latex, but do not have it installed on their computers may want to look at the following video:

Video: https://youtu.be/PfhSOjuQk8Y

Video with cc: https://www.youtube.com/watch?v=8IDCGTFXoBs

13.2 Overleaf

Overleaf.com is a collaborative latex editor. In its free version it has a very limited disk space. However it comes with a Rich text mode that allows you to edit the document in a preview mode. The free templates provided do not include ACM template, put you are allowed to use the OSA template.

Features of overleaf are documented at: https://www.overleaf.com/benefits

13.3 jabref

Jabref is a very simple to use bibliography manager for LaTeX and other systems. It cand create a multitude of bibliography file formats and allows upload in other online bibliography managers.

Video: https://youtu.be/cMtYOHCHZ3k

Video with cc: https://www.youtube.com/watch?v=QVbifcLgMic

13.3.1 jabref and MSWord

Accordung to Colin Thornburg it is possible to integrate jabref references directly into MSWord. THi has been conducted on a Windows computer. All you need to do is following theses steps.

1. Create Jabref bibliography just like in presented in the jabref video
2. After finishing adding your sources in Jabref, click *File -> export*
3. Name your bibliography and choose MS Office 2007(*.xml) as the file format. Remember the location of where you saved your file.
4. Open up your word document. If you are using the ACM template, go ahead and remove the template references listed under *Section 7. References*

5. In the MS Word ribbon choose 'References'
6. Choose 'Manage Sources'
7. Click 'Browse' and locate/select your Jabref xml file
8. You should now see your references appear in the left side window. Select the references you want to add to your document and click the 'copy' button to move them from the left side window to the right window.
9. Click the 'Close' button
10. In the MS Word Ribbon, select 'Bibliography' under the References tab
11. Click 'Insert Bibliography' and your references should appear in the document
12. Ensure references are of Style: IEEE. Styles are located in the References tab under 'Manage Sources'

As you can see there is some effort involve, so we do recommend you use LaTeX as you can focus there on content rather than dealing with comlex layout decissions. THi sis especially true, if your papers has figures or tables.

13.4 References

- The LaTeX Reference Manual provides a good intriduction to Latex.

LaTeX is available on all modern computer systems. A very good instalation for OSX is available at:

- https://tug.org/mactex/

However, if you have older versions on your systems you may have to first completely uninstall them.

13.5 Introduction

Mastering a text processing system is an essential part of a researchers life. Not knowing how to use a text processing system can slow down the productivity of research drastically.

The information provided here is not intended to replace one of the many text books available about LaTeX. For the beginning you might be just fine with the documentation provided here. For serious users I recommend to purchase a book. Examples for books include

- LaTeX Users and Reference Guide, by Leslie Lamport
- LaTeX an Introduction, by Helmut Kopka
- The LaTeX Companion, by Frank Mittelbach

If you do not want to buy a book you can find a lot of useful information in the LaTeX refrence manual.

13.6 Manual pages and programs

Following programs are useful for using LaTeX. To most of them you will find also manual pages:

- pdflatex: the latex program producing pdf
- bibtex: to create bibliographies
- jabref: less fancy GUI to bibtex files

13.7 The LaTeX Cycle

Create/edit ASCII source file with `file.tex` file:

```
emacs file.tex
```

Create/edit bibliography file:

```
jabref refs.bib
```

Create the PDF:

```
pdflatex file
bibtex file
pdflatex file
pdflatex file
```

View the PDF:

```
open file
```

On OSX the best PDF viewer for LaTeX is skim:

- http://skim-app.sourceforge.net/

13.8 Generating Images

To produce high quality images the programs PowerPoint and omnigraffle on OSX are recommended. When using powerpoint please keep the image ratio to 4x3 as they produce nice sze graphics whoch you also can use in your presentations. When using other rations they may not fit in presentations and thus you may increase unnecessarily your work. We do not recommend vizio as it is not universally available.

13.9 Editing LaTeX

The text editor emacs provides a good basis for editing TeX and LaTex documents. Both modes are supported. In addition there exists a color highlight module enableling the color display of LaTeX and TeX commands. On OSX aquaemacs and carbon emacs have build in support for LaTeX. Spell checking is done with flyspell in emacs.

Other editors such as TeXshop are available which provide a more integrated experience.

However when using skim in conjunction with imacs and latexmk your PDF view will be automatically updated once you save the file in emacs. This provides a very good quasy WYSIWYG environment.

I have made very good experiences with Lyx. You must assure that the team you work with uses it consistently and that you all use the same version.

Using the ACM templates is documented here:

- https://wiki.lyx.org/Examples/AcmSiggraph

On OSX it is important that you have a new version of LaTeX and Lyx installed. As it takes up quite some space, you ma want to delete older versions. The new version of LyX comes with the acmsigplan template included. However on OSX and other platforms the .cls file is not included by default. However the above link clearly documents how to fix this.

13.10 How to edit Bibliographies?

It is a waste of your time to edit bibliographies with the bibitem environment. There are several preformated styles available. It includes also styles for ACM and IEEE bibliographies. For the ACM style we recommend that you replace abbrv.bst with abbrvurl.bst, add hyperref to your usepackages so you can also display urls in your citations:

```
\bibliographystyle{abbrvurl}
\bibliography{references.bib}
```

Than you have to run latex and bibtex in the following order:

```
latex  file
bibtex file
latex  file
latex  file
```

The reason for the multiple execution of the latex program is to update all cross-references correctly. In case you are not interested in updating the library every time in the writing progress just postpone it till the end. Missing citations are viewed as [?].

Two programs stand out when manageing bibliographies: emacs and jabref:

- http://www.jabref.org/

13.11 How to produce Slides?

Slides are best produced with the seminar package:

```
\documentclass{seminar}

\begin{slide}

    Hello World on slide 1

\end{slide}

The text between slides is ignored

\begin{slide}

    Hello World on slide 2

\end{slide}
```

CHAPTER 14

Reference Managers

Please note that you should first decide which refernce manager you like to use. IN case you for example install zotero and mendeley, that may not work with word or other programs.

14.1 jabref

Please see LaTeX section. This is our highly recommended reference menager

Note: We do recommend that you use sharelatex and jabref for writing papers. THis is the easiest solution.

14.2 Endnote

Endnote os a reference manager that works with Windows. Many people use endnote. However, in the past endnote has lead to complications when dealing with collaborative management of references. Its price is considerable.

- http://endnote.com/

14.3 Mendeley

Mendeley is a free refernce manager compatible with Windows Word 2013, Mac Word 2011, LibreOffice, BibTeX. Videos on how to use it are available at:

- https://community.mendeley.com/guides/videos

Instalation instructions are available at

https://www.mendeley.com/features/reference-manager/

14.4 Zotero

Zotero is a free tool to help you collect, organize, cite, and share your research sources. Documentation is available at

- https://www.zotero.org/support/

The download link is available from

- https://www.zotero.org/

CHAPTER 15

Ubuntu Virtual Machine

For development purposes we recommend tha you use for this class an ubuntu virtual machine that you set up with the help of virtualbox.

Only after you have successfully used ubuntu in a virtual machine you will be allowed to use virtual machine son clouds.

A "cloud drivers license test" will be conducted to let you gain access to the cloud infrastructure. We will announce this test. Before you have not passed the test, you will not be able to use the clouds. Furthermore, you do not have to ask us for join requests before you have not passed the test. Please be patient. Only students enrolled in the class can get access to the cloud.

15.1 Creation

First you will need to install virtualbox. It is easy to install and details can be found at

- https://www.virtualbox.org/wiki/Downloads

After you have installed virtualbox you also need to use an image. For this class we will be using ubuntu Desktop 16.04 which you can find at:

- http://www.ubuntu.com/download/desktop

Please note the recommended requirements that also apply to a virtual machine:

- 2 GHz dual core processor or better
- 2 GB system memory
- 25 GB of free hard drive space

A video to showcase such an install is available at:

- Video: https://youtu.be/NWibDntN2M4

Warning: If you specify your machien too small you will not be able to install the development environment. Gregor used on his machine 8gb of RAM and 20GB diskspace.
Please let us know the smalest configuration that works for you and share this in Piaza. Only update if yours is smaller and works than a previous post. If not do not post.

15.2 Guest additions

The virtual guest additions allow you to easily do the following tasks:

- Resize the windows of the vm
- Copy and paste content between the Guest operating system and the host operating system windows.

This way you can use many native programs on you host and copy contents easily into for example a terminal or an editor that you run in the Vm.

A video is located at

- Video: https://youtu.be/wdCoiNdn2jA

Note: Please reboot the machine after installation and configuration.

On OSZ you can once you have enabled bidirectional copying in the Device tab with

OSX -> Vbox: *command c -> shift CONTRL v*

Vbox to OSX: *shift CONTRL v -> shift CONTRL v*

On Windows the key combination is naturally different. Please consult your windows manual.

15.3 Development Configuration

The documentation on how to configure the virtual machine and install many useful programs is posted at:

- https://github.com/cloudmesh/ansible-cloudmesh-ubuntu-xenial

You simply have to execute the following commands in the terminal of the virtual machine. In order to eliminate confusion with other terminals, we use the prefix *vm> $* to indicate any command that is to be started on the virtual machine. Otherwise it is clear from the context:

```
vm>$ wget https://raw.githubusercontent.com/cloudmesh/ansible-cloudmesh-ubuntu-xenial/master/bootstra
vm>$ bash bootstrap.sh
```

A video showcasing this install is available:

- Video: https://youtu.be/YqXIj_Wzfsc

A video showcasing the upload to gitlab from within the vm using commandline tools

- Video: https://youtu.be/EnpneUY82I8

15.4 Homework Virtualbox

1. Install ubuntu desktop on your computer with guest additions.
2. Make sure you know how to paste and copy between your host and guest operating system
3. Install the programs defined by the development configuration

CHAPTER 16

Using SSH Keys

Page Contents

- *Using SSH from Windows*
- *Using SSH on Mac OS X*
- *Generate a SSH key*
- *Add or Replace Passphrase for an Already Generated Key*
- *Upload the key to gitlab*

Hint: If you do not know what ssh is we recommend that you read up on it . However, the simple material presented here will help you getting started quickly.

To access remote resources this is often achieved via SSH. You need to provide a public ssh key to FutureSystem. We explain how to generate a ssh key, upload it to the FutureSystem portal and log onto the resources. This manual covers UNIX, Mac OS X. For Windows we will prepare an add on to this document.

16.1 Using SSH from Windows

Hint: For Linux users, please skip to the section *Generate a SSH key*

Hint: For Mac users, please skip to the section *Using SSH on Mac OS X*

Warning: For this class we recommend that you use a virtual machine via virtual box and use the Linux ssh instructions. The information here is just provided for completness and no support will be offered for native windows support.

Windows users need to have some special software to be able to use the SSH commands. If you have one that you are comfortable with and know how to setup key pairs and access the contents of your public key, please feel free to use it.

The most popular software making ssh clients available to Windows users include

- cygwin
- putty
- or installing a virtualiztion software and running Linux virtual machine on your Windows OS.
- using chocolatey

We will be discussing here how to use it in Powershell with the help of chopolatey.

Chocolatey is a software management tool that mimics the install experience that you have on Linux and OSX. It has a repository with many packages. Before using and installing a package be aware of the consequences when installing software on your computer. Please be aware that there could be malicious code offered in the chocolatey repository although the distributors try to remove them.

The installation is sufficently explained at

- https://chocolatey.org/install

Once installed you have a command choco and you should make sure you have the newest version with

```
choco upgrade chocolatey
```

Now you can browse packages at

- https://chocolatey.org/packages

Search for openssh and see the results. You may find different versions. Select the one that most suits you and satisfies your security requirements as well as your architecture. Lets assume you chose the Microsoft port, than you can install it with:

```
choco install win32-openssh
```

Other packages of interest include

- LaTeX:: *choco install miktex*
- jabref: *choco install jabref*
- pycharm: *choco install pycharm-community*
- python 2.7.11: *choco install python2*
- pip: *choco install pip*
- virtual box: *choco install virtualbox*
- emacs: *choco install emacs*
- lyx: *choco install lyx*
- vagrant: *choco install vagrant*

Before installing any of them evaluate if you need them.

16.2 Using SSH on Mac OS X

Mac OS X comes with an ssh client. In order to use it you need to open the `Terminal.app` application. Go to `Finder`, then click `Go` in the menu bar at the top of the screen. Now click `Utilities` and then open the `Terminal` application.

16.3 Generate a SSH key

|info-image| Hint

In case you do not want to type in your password everytime, please learn about ssh-agent and ssh-add.

First we must generate a ssh key with the tool ssh-keygen. This program is commonly available on most UNIX systems (this includes Cygwin if you installed the ssh module or use our pre-generated cygwin executable). It will ask you for the location and name of the new key. It will also ask you for a passphrase, which you **MUST** provide. Some teachers and teaching assistants advice you to not use passphrases. This is **WRONG** as it allows someone that gains access to your computer to also gain access to all resources that have the public key. Also, please use a strong passphrase to protect it appropriately.

In case you already have a ssh key in your machine, you can reuse it and skip this whole section.

To generate the key, please type:

Example:

```
ssh-keygen -t rsa -C localname@indiana.edu
```

This command requires the interaction of the user. The first question is:

```
Enter file in which to save the key (/home/localname/.ssh/id_rsa):
```

We recommend using the default location ~/.ssh/ and the default name id_rsa. To do so, just press the enter key.

Note: Your *localname* is the username on your computer.

The second and third question is to protect your ssh key with a passphrase. This passphrase will protect your key because you need to type it when you want to use it. Thus, you can either type a passphrase or press enter to leave it without passphrase. To avoid security problems, you **MUST** chose a passphrase. Make sure to not just type return for an empty passphrase:

```
Enter passphrase (empty for no passphrase):
```

and:

```
Enter same passphrase again:
```

If executed correctly, you will see some output similar to:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/localname/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/localname/.ssh/id_rsa.
Your public key has been saved in /home/localname/.ssh/id_rsa.pub.
The key fingerprint is:
34:87:67:ea:c2:49:ee:c2:81:d2:10:84:b1:3e:05:59 localname@indiana.edu
The key's random art image is::

+--[ RSA 2048]----+
|.+...Eo= .       |
| ..=.o + o +o    |
|O.  o o +.o      |
```

```
| = .   . .       |
+-----------------+
```

Once, you have generated your key, you should have them in the .ssh directory. You can check it by

```
$ cat ~/.ssh/id_rsa.pub
```

If everything is normal, you will see something like:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCXJH2iG2FMHqC6T/U7uB8kt6KlRh4kUOjgw9sc4Uu+Uwe/EwD0wk6CBQMB+HKb
```

16.4 Add or Replace Passphrase for an Already Generated Key

In case you need to change your change passphrase, you can simply run "ssh-keygen -p" command. Then specify the location of your current key, and input (old and) new passphrases. There is no need to re-generate keys:

```
ssh-keygen -p
```

You will see the following output once you have completed that step:

```
Enter file in which the key is (/home/localname/.ssh/id_rsa):
Enter old passphrase:
Key has comment '/home/localname/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

16.5 Upload the key to gitlab

Follow the instructions provided here:

- http://docs.gitlab.com/ce/ssh/README.html

CHAPTER 17

Links Report

Todo

Gregor. Goto LaTeX documentation and consolidate into single latex.rst

- https://github.com/cloudmesh/book/blob/master/writing/latex.md
- https://github.com/cloudmesh/tools/blob/master/docs/source/class/report.rst
- https://github.com/cloudmesh/tools/blob/master/docs/source/class/mistakes.rst

CHAPTER 18

Homework References

It is important that you know how to properly cite references. IN order to teach you how to do that we have taken the almost 300 references from the class and will ask you to provide **PROPER** academic references for them. All references will be managed with jabref.

You will have to do about 10 references. Students should build teams of 2 students to correct each others contribution if possible. You will only get points for the references that are absolute correct. It does not matter if a colleague has helped you correcting your references. What is important is that you know how to cite correctly.

Warning: This homework is typically underestimated by students and often done in sloppy fashion. I have had classes where 50% of the class got 0 points in this assignment. Thus it is not just sufficient to put in the reference as MISC if it is a url, but you have to actually look up the URL, if its a paper, you may even have to locate which journal or conference, which location the conference took place what date the conference took place and so forth. Please note that many bibentries including some form IEEE and other sources could be wrong or are incomplete. For example are there other locations where you can find the PDF of a paper?
This assignment counts as much as a paper.

How will you know which citation you need to do?: You will be assigned a number in class and you simply have to do all the references that are in the list and do the once with your assignment number specified in a1 - a5 and b1-b5 as defined in

https://piazza.com/class/irqfvh1ctrg2vt?cid=260

What if i get a reference to "——"? Just pick randomly another number that is not associated with a ——

Can I use endnote for this? No.

What is an article? An article is published in a journal.

What is inProceedings? That is an article published in a conference proceedings

What is inBook? That is a chapter or pages in a book

How do I cite urls? Often you can find a proper article and use that in addition to the url. Hence, you may have to introduce two references. If you just cite the URL, watch out for how published it, what is the author, when was it published, what is the proper url, and when was it accessed.

What if my link no longer works? Can you find it in the internet archive? Is there a query you could find from the url and identify an alternate location?

Where do I upload it: Go to gitlab and go into the bib folder, upload your references as class.bib

How do I create labels: use class000 where 000 is the number of the 0 padded number of your reference in the list bellow. Example, assume you have to do reference 11, than your label for that is class011.

Add the owner={HID, Firstname Lastname} field in jabref

Where Firstname Lastname is your fisrname and lastname

```
* ----------------------------
* http://www.gartner.com/technology/home.jsp and many web links
* Meeker/Wu May 29 2013 Internet Trends D11 Conference http://www.slideshare.net/kleinerperkins/
* http://cs.metrostate.edu/~sbd/slides/Sun.pdf
* Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data  Streams with Advanced Anal
* http://www.genome.gov/sequencingcosts/
* CSTI General Assembly 2012, Washington, D.C., USA Technical  Activities Coordinating Committee
* http://www.mckinsey.com/mgi/publications/big_data/index.asp
* Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
* http://research.microsoft.com/en-us/people/barga/sc09_cloudcomp_tutorial.pdf
* http://research.microsoft.com/pubs/78813/AJ18_EN.pdf
* http://www.google.com/green/pdfs/google-green-computing.pdf
* http://www.wired.com/wired/issue/16-07
* http://research.microsoft.com/en-us/collaboration/fourthparadigm/
* Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf
* http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20
* http://www.interactions.org/cms/?pid=1032811
* http://www.quantumdiaries.org/2012/09/07/why-particle-detectors-need-a-trigger/atlasmgg/
* http://www.sciencedirect.com/science/article/pii/S037026931200857X
* http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-t
* http://www.ifi.uzh.ch/ce/teaching/spring2012/16-Recommender-Systems_Slides.pdf
* http://en.wikipedia.org/wiki/PageRank
* http://pages.cs.wisc.edu/~beechung/icml11-tutorial/
* https://sites.google.com/site/opensourceiotcloud/
* http://datascience101.wordpress.com/2013/04/13/new-york-times-data-science-articles/
* http://blog.coursera.org/post/49750392396/on-the-topic-of-boredom
* http://x-informatics.appspot.com/course
* http://iucloudsummerschool.appspot.com/preview
* https://www.youtube.com/watch?v=M3jcSCA9_hM
* ----------------------------
* http://www.microsoft.com/en-us/news/features/2012/mar12/03-05CloudComputingJobs.aspx
* http://www.mckinsey.com/mgi/publications/big_data/index.asp
* Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
* Anjul Bhambhri http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
* Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf
* http://www.economist.com/node/15579717
* http://cs.metrostate.edu/~sbd/slides/Sun.pdf
* http://jess3.com/geosocial-universe-2/
* Bill Ruhhttp://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
* http://www.hsph.harvard.edu/ncb2011/files/ncb2011-z03-rodriguez.pptx
* Hugh Williams http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
* ----------------------------
* http://www.economist.com/node/15579717
* Geoffrey Fox and Dennis Gannon Using Clouds for Technical Computing To be published in Proceed
* http://grids.ucs.indiana.edu/ptliupages/publications/Clouds_Technical_Computing_FoxGannonv2.pd
* http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20
* http://www.genome.gov/sequencingcosts/
* http://www.quantumdiaries.org/2012/09/07/why-particle-detectors-need-a-trigger/atlasmgg
* http://salsahpc.indiana.edu/dlib/articles/00001935/
* http://en.wikipedia.org/wiki/Simple_linear_regression
* http://www.ebi.ac.uk/Information/Brochures/
* http://www.wired.com/wired/issue/16-07
* http://research.microsoft.com/en-us/collaboration/fourthparadigm/
* CSTI General Assembly 2012, Washington, D.C., USA Technical  Activities Coordinating Committee
* ----------------------------
* CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee
* Dan Reed Roger Barga Dennis Gannon Rich Wolskihttp://research.microsoft.com/en-us/people/barga
```

```
 58  * http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pue-is-1-8
 59  * http://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data-centers-or
 60  * http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf
 61  * Bina Ramamurthy http://www.cse.buffalo.edu/~bina/cse487/fall2011/
 62  * Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf
 63  * Jeff Hammerbacher http://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf
 64  * Anjul Bhambhri http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
 65  * http://cs.metrostate.edu/~sbd/slides/Sun.pdf
 66  * Hugh Williams http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
 67  * Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
 68  * http://www.mckinsey.com/mgi/publications/big_data/index.asp
 69  * http://cra.org/ccc/docs/nitrdsymposium/pdfs/keyes.pdf
 70  * ---------------------------
 71  * https://wiki.nci.nih.gov/display/CIP/CIP+Survey+of+Biomedical+Imaging+Archives
 72  * http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20
 73  * http://www.ieee-icsc.org/ICSC2010/Tony%20Hey%20-%2020100923.pdf
 74  * http://quantifiedself.com/larry-smarr/
 75  * http://www.ebi.ac.uk/Information/Brochures/
 76  * http://www.kpcb.com/internet-trends
 77  * http://www.slideshare.net/drsteventucker/wearable-health-fitness-trackers-and-the-quantified-s
 78  * http://www.siam.org/meetings/sdm13/sun.pdf
 79  * http://en.wikipedia.org/wiki/Calico_%28company%29
 80  * http://www.slideshare.net/GSW_Worldwide/2015-health-trends
 81  * http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Co
 82  * http://www.slideshare.net/schappy/how-realtime-analysis-turns-big-medical-data-into-precision-
 83  * http://medcitynews.com/2013/03/the-body-in-bytes-medical-images-as-a-source-of-healthcare-big-
 84  * http://healthinformatics.wikispaces.com/file/view/cloud_computing.ppt
 85  * http://www.mckinsey.com/~/media/McKinsey/dotcom/Insights/Health%20care/The%20big-data%20revolu
 86  * https://partner.microsoft.com/download/global/40193764
 87  * http://ec.europa.eu/information_society/activities/health/docs/policy/taskforce/redesigning_he
 88  * http://www.kpcb.com/internet-trends
 89  * http://www.liveathos.com/apparel/app
 90  * http://debategraph.org/Poster.aspx?aID=77
 91  * http://www.oerc.ox.ac.uk/downloads/presentations-from-events/microsoftworkshop/gannon
 92  * http://www.delsall.org
 93  * http://salsahpc.indiana.edu/millionseq/mina/16SrRNA_index.html
 94  * http://www.geatbx.com/docu/fcnindex-01.html
 95  * https://wiki.nci.nih.gov/display/CIP/CIP+Survey+of+Biomedical+Imaging+Archives
 96  * http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20
 97  * http://www.ieee-icsc.org/ICSC2010/Tony%20Hey%20-%2020100923.pdf
 98  * http://quantifiedself.com/larry-smarr/
 99  * http://www.ebi.ac.uk/Information/Brochures/
100  * http://www.kpcb.com/internet-trends
101  * http://www.slideshare.net/drsteventucker/wearable-health-fitness-trackers-and-the-quantified-s
102  * http://www.siam.org/meetings/sdm13/sun.pdf
103  * http://en.wikipedia.org/wiki/Calico_%28company%29
104  * http://www.slideshare.net/GSW_Worldwide/2015-health-trends
105  * http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Co
106  * http://www.slideshare.net/schappy/how-realtime-analysis-turns-big-medical-data-into-precision-
107  * http://medcitynews.com/2013/03/the-body-in-bytes-medical-images-as-a-source-of-healthcare-big-
108  * http://healthinformatics.wikispaces.com/file/view/cloud_computing.ppt
109  * http://www.mckinsey.com/~/media/McKinsey/dotcom/Insights/Health%20care/The%20big-data%20revolu
110  * https://partner.microsoft.com/download/global/40193764
111  * http://ec.europa.eu/information_society/activities/health/docs/policy/taskforce/redesigning_he
112  * http://www.kpcb.com/internet-trends
113  * http://www.liveathos.com/apparel/app
114  * http://debategraph.org/Poster.aspx?aID=77
115  * http://www.oerc.ox.ac.uk/downloads/presentations-from-events/microsoftworkshop/gannon
```

```
116  * http://www.delsall.org
117  * http://salsahpc.indiana.edu/millionseq/mina/16SrRNA_index.html
118  * http://www.geatbx.com/docu/fcnindex-01.html
119  * ----------------------------
120  * http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-powered-story
121  * http://www.sloansportsconference.com/
122  * http://sabr.org/
123  * http://en.wikipedia.org/wiki/Sabermetrics
124  * http://en.wikipedia.org/wiki/Baseball_statistics
125  * http://www.sportvision.com/baseball
126  * http://m.mlb.com/news/article/68514514/mlbam-introduces-new-way-to-analyze-every-play
127  * http://www.fangraphs.com/library/offense/offensive-statistics-list/
128  * http://en.wikipedia.org/wiki/Component_ERA
129  * http://www.fangraphs.com/library/pitching/fip/
130  * http://nomaas.org/2012/05/a-look-at-the-defense-the-yankees-d-stinks-edition/
131  * http://en.wikipedia.org/wiki/Wins_Above_Replacement
132  * http://www.fangraphs.com/library/misc/war/
133  * http://www.baseball-reference.com/about/war_explained.shtml
134  * http://www.baseball-reference.com/about/war_explained_comparison.shtml
135  * http://www.baseball-reference.com/about/war_explained_position.shtml
136  * http://www.baseball-reference.com/about/war_explained_pitch.shtml
137  * http://www.fangraphs.com/leaders.aspx?pos=all&stats=bat&lg=all&qual=y&type=8&season=2014&month
138  * http://battingleadoff.com/2014/01/08/comparing-the-three-war-measures-part-ii/
139  * http://battingleadoff.com/2014/01/08/comparing-the-three-war-measures-part-ii/
140  * http://en.wikipedia.org/wiki/Coefficient_of_determination
141  * http://www.sloansportsconference.com/wp-content/uploads/2014/02/2014_SSAC_Data-driven-Method-i
142  * https://courses.edx.org/courses/BUx/SABR101x/2T2014/courseware/10e616fc7649469ab4457ae18df92b2
143  * ----------------------------
144  * http://vincegennaro.mlblogs.com/
145  * https://www.youtube.com/watch?v=H-kx-x_d0Mk
146  * http://www.sportvision.com/media/pitchfx-how-it-works
147  * http://www.baseballprospectus.com/article.php?articleid=13109
148  * http://baseball.physics.illinois.edu/FastPFXGuide.pdf
149  * http://baseball.physics.illinois.edu/FieldFX-TDR-GregR.pdf
150  * http://www.sportvision.com/baseball/fieldfx
151  * http://regressing.deadspin.com/mlb-announces-revolutionary-new-fielding-tracking-syste-153420
152  * http://grantland.com/the-triangle/mlb-advanced-media-play-tracking-bob-bowman-interview/
153  * http://www.sportvision.com/baseball/hitfx
154  * https://www.youtube.com/watch?v=YkjtnuNmK74
155  * ----------------------------
156  * http://www.sloansportsconference.com/?page_id=481&sort_cate=Research%20Paper
157  * http://www.slideshare.net/Tricon_Infotech/big-data-for-big-sports
158  * http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-powered-story
159  * http://www.liveathos.com/apparel/app
160  * http://www.slideshare.net/elew/sport-analytics-innovation
161  * http://www.wired.com/2013/02/catapault-smartball/
162  * http://www.sloansportsconference.com/wp-content/uploads/2014/06/Automated_Playbook_Generation.
163  * http://autoscout.adsc.illinois.edu/publications/football-trajectory-dataset/
164  * http://www.sloansportsconference.com/wp-content/uploads/2012/02/Goldsberry_Sloan_Submission.pd
165  * http://gamesetmap.com/
166  * http://www.trakus.com/technology.asp#tNetText
167  * ----------------------------
168  * http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20
169  * http://www.interactions.org/cms/?pid=6002
170  * http://www.interactions.org/cms/?pid=1032811
171  * http://www.sciencedirect.com/science/article/pii/S037026931200857X
172  * http://biologos.org/blog/what-is-the-higgs-boson
173  * http://www.atlas.ch/pdf/ATLAS_fact_sheets.pdf
```

```
174  * http://www.nature.com/news/specials/lhc/interactive.html
175  * ----------------------------
176  * https://www.enthought.com/products/canopy/
177  * Python for Data Analysis: Agile Tools for Real World Data By Wes  McKinney, Publisher: O'Reil
178  * http://jwork.org/scavis/api/
179  * https://en.wikipedia.org/wiki/DataMelt
180  * ----------------------------
181  * http://indico.cern.ch/event/20453/session/6/contribution/15?materialId=slides
182  * http://www.atlas.ch/photos/events.html
183  * http://cms.web.cern.ch/
184  * ----------------------------
185  * https://en.wikipedia.org/wiki/Pseudorandom_number_generator
186  * https://en.wikipedia.org/wiki/Mersenne_Twister
187  * https://en.wikipedia.org/wiki/Mersenne_prime
188  * CMS-PAS-HIG-12-041 Updated results on the new boson discovered in  the search for the standar
189  * https://en.wikipedia.org/wiki/Poisson_distribution
190  * https://en.wikipedia.org/wiki/Central_limit_theorem
191  * http://jwork.org/scavis/api/
192  * https://en.wikipedia.org/wiki/DataMelt
193  * ----------------------------
194  * http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-t
195  * http://www.ifi.uzh.ch/ce/teaching/spring2012/16-Recommender-Systems_Slides.pdf
196  * https://www.kaggle.com/
197  * http://www.ics.uci.edu/~welling/teaching/CS77Bwinter12/CS77B_w12.html
198  * Jeff Hammerbacher https://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pd
199  * http://www.techworld.com/news/apps/netflix-foretells-house-of-cards-success-with-cassandra-big
200  * https://en.wikipedia.org/wiki/A/B_testing
201  * http://www.infoq.com/presentations/Netflix-Architecture
202  * ----------------------------
203  * http://pages.cs.wisc.edu/~beechung/icml11-tutorial/
204  * ----------------------------
205  *  https://en.wikipedia.org/wiki/Kmeans
206  *  http://grids.ucs.indiana.edu/ptliupages/publications/DACIDR_camera_ready_v0.3.pdf
207  *  http://salsahpc.indiana.edu/millionseq/
208  *  http://salsafungiphy.blogspot.com/
209  *  https://en.wikipedia.org/wiki/Heuristic
210  * ----------------------------
211  * Solving Problems in Concurrent Processors-Volume 1,  with M. Johnson, G. Lyzenga, S. Otto, J.
212  * Parallel Computing Works!, with P. Messina, R. Williams, Morgan  Kaufman (1994). http://www.ne
213  * The Sourcebook of Parallel Computing book edited by Jack Dongarra,  Ian Foster, Geoffrey Fox,
214  * Geoffrey Fox Computational Sciences and Parallelism to appear in  Enclyclopedia on Parallel Co
215  * ----------------------------
216  * http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing
217  * http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis
218  * https://setandbma.wordpress.com/2012/08/10/hype-cycle-2012-emerging-technologies/
219  * http://insights.dice.com/2013/01/23/big-data-hype-is-imploding-gartner-analyst-2/
220  * http://research.microsoft.com/pubs/78813/AJ18_EN.pdf
221  * http://static.googleusercontent.com/media/www.google.com/en//green/pdfs/google-green-computing
222  * ----------------------------
223  * http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis
224  * http://research.microsoft.com/en-us/people/barga/sc09_cloudcomp_tutorial.pdf
225  * http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_Opportu
226  * http://cloudonomic.blogspot.com/2009/02/cloud-taxonomy-and-ontology.html
227  * ----------------------------
228  * http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing
229  * http://www.eweek.com/c/a/Cloud-Computing/AWS-Innovation-Means-Cloud-Domination-307831
230  * CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee
231  * http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_Opportu
```

```
232  * http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pue-is-1-8
233  * https://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data-centers-c
234  * http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf
235  * http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis
236  * http://www.slideshare.net/botchagalupe/introduction-to-clouds-cloud-camp-columbus
237  * http://www.venus-c.eu/Pages/Home.aspx
238  * Geoffrey Fox and Dennis Gannon Using Clouds for Technical Computing To be published in Proceed
239  * https://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf
240  * Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analy
241  * Anjul Bhambhri, VP of Big Data, IBM http://fisheritcenter.haas.berkeley.edu/Big_Data/index.htm
242  * Conquering Big Data with the Oracle Information Model, Helen Sun, Oracle
243  * Hugh Williams VP Experience, Search & Platforms, eBay http://businessinnovation.berkeley.edu/i
244  * Dennis Gannon, Scientific Computing Environments, http://www.nitrd.gov/nitrdgroups/images/7/73
245  * http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_Opportu
246  * http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pue-is-1-8
247  * https://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data-centers-c
248  * http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf
249  * http://searchcloudcomputing.techtarget.com/feature/Cloud-computing-experts-forecast-the-market
250  * http://www.slideshare.net/botchagalupe/introduction-to-clouds-cloud-camp-columbus
251  * http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing
252  * http://www.venus-c.eu/Pages/Home.aspx
253  * http://www.kpcb.com/internet-trends
254  * ----------------------------
255  * http://bigdatawg.nist.gov/_uploadfiles/M0311_v2_2965963213.pdf
256  * https://dzone.com/articles/hadoop-t-etl
257  * http://venublog.com/2013/07/16/hadoop-summit-2013-hive-authorization/
258  * https://indico.cern.ch/event/214784/session/5/contribution/410
259  * http://asd.gsfc.nasa.gov/archive/hubble/a_pdf/news/facts/FS14.pdf
260  * http://blogs.teradata.com/data-points/announcing-teradata-aster-big-analytics-appliance/
261  * http://wikibon.org/w/images/2/20/Cloud-BigData.png
262  * http://hortonworks.com/hadoop/yarn/
263  * https://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf
264  * http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
265  * ----------------------------
266  * http://saedsayad.com/data_mining_map.htm
267  * http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html
268  * The Web Graph: an Overview Jean-Loup Guillaume and Matthieu Latapy https://hal.archives-ouvert
269  * Constructing a reliable Web graph with information on browsing behavior, Yiqun Liu, Yufei Xue,
270  * http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws
271  * ----------------------------
272  * http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws
273  * https://en.wikipedia.org/wiki/PageRank
274  * http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html
275  * Meeker/Wu May 29 2013 Internet Trends D11 Conference http://www.slideshare.net/kleinerperkins/
276  * ----------------------------
277  * https://www.gesoftware.com/minds-and-machines
278  * https://www.gesoftware.com/predix
279  * https://www.gesoftware.com/sites/default/files/the-industrial-internet/index.html
280  * https://developer.cisco.com/site/eiot/discover/overview/
281  * http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Co
282  * http://www.gesoftware.com/ge-predictivity-infographic
283  * http://www.getransportation.com/railconnect360/rail-landscape
284  * http://www.gesoftware.com/sites/default/files/GE-Software-Modernizing-Machine-to-Machine-Inter
```

CHAPTER 19

Drafts (TODO)

19.1 Additional Programming Assignments

19.1.1 Programming: Hadoop Cluster

You will be provided with a Hadoop cluster running MapReduce. Your goal will be to use the Hadoop cluster to run a "Big Data" computation. One possible approach is the Terabyte Sort procedure. The components are:

- TeraGen: create the data
- TeraSort: analyze the data using MapReduce
- TeraValidate: validation of the output

Access to the cluster

Todo

HadoopClusterAccess.html

Invocation

The teragen command accepts two parameters:

- number of 100-byte rows
- the output directory

```
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar
teragen $COUNT /user/$USER/tera-gen
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar
terasort /user/$USER/tera-gen /user/$USER/tera-sort
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar
teravalidate /user/$USER/tera-sort /user/$USER/tera-validate
```

Exercise

Run the Terabyte Sort procedure for various sizes of data:

- 1 GB
- 10 GB
- 100 GB

For each component (tera{gen,sort,validate}), report the execution time, data read and written (in GB) as well as the cumulative values.

19.1.2 Programming: Using futuresystems.org

In this homework, you are expected to run Python or Java programs on FutureSystems or on your local machine. A few examples for beginners will help you to understand how to write and run Java or Python programs on your environment.

We will print some elementary system information such as time, date, user name or hostname (machine name) which will be important when you report on your infrastructure in your program reports. You will likely need to add more information such as processor type, core number, and frequency.

Java

Here is a simple program in Java.

Download: FirstProgramWithSystemInfo.java:

```
import java.util.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.net.InetAddress;
import java.net.UnknownHostException;
/**
* * Sample Program with system information
* *
* * Compile : javac FirstProgramWithSystemInfo.java
* * Run : java FirstProgramWithSystemInfo
* */
public class FirstProgramWithSystemInfo {
    public static void main(String[] args){
    System.out.println("My first program with System Information!");
        // Print Date with Time
        DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
        Date date = new Date();
        System.out.println("Today is: " + dateFormat.format(date));
        // Print Username
        System.out.println("Username is: " + System.getProperty("user.name"));
        // Print hostname
        try {
            java.net.InetAddress localMachine = java.net.InetAddress.getLocalHost();
            System.out.println("Hostname is: " + localMachine.getHostName());
        } catch (UnknownHostException e) {
            e.printStackTrace();
        System.out.println("No host name: " + e.getMessage());
        }
    }
}
```

Compiling and Execution:

```
javac FirstProgramWithSystemInfo.java
java FirstProgramWithSystemInfo

My first program with System Information!

Today is: 2015/01/01 18:54:10
Username is: albert
Hostname is: bigdata-host
```

Python

Let's write a simple program in Python.

Create the following program: FirstProgram.py:

```
##########################################
# Run python FirstProgram.py
##########################################
from datetime import datetime
import getpass
import socket
##########################################
# Run python FirstProgramWithSystemInfo.py
##########################################
print ('My first program with System Information!')
print ("Today is: " + str(datetime.now()))
print ("Username is: " + getpass.getuser())
print ("Hostname is: " + socket.gethostname())
```

Execution:

Compiling is not necessary in Python. You can run your code directly with python command.:

```
python FirstProgram.py
```

What does the output look like?:

```
python FirstProgramWithSystemInfo.py
My first program with System Information!
Today is: 2015-01-01 18:58:10.937227
Username is: albert
Hostname is: bigdata-host
```

Challenge tasks

- Run any Java or Python on a FutureSystems OpenStack instance
- Run NumPyTutorial Python on IPython Notebook

19.1.3 Code Examples

19.2 Preview Course Examples

- The Elusive Mr.Higgs [Java][Python]

- Number Theory [Python]
- Calculated Dice Roll [Java][Python]
- KNN [Java][Python]
- PageRank [Java][Python]
- KMeans [Java][Python]

19.2.1 Hadoop Cluster Access

This document describes getting access to the Hadoop cluster for the course.

You will need

1. An a account with FutureSystems
2. To be a member of a active project on FutureSystems (fg511)
3. Have uploaded an ssh key to the portal

The cluster frontend is located at <IP_ADDRESS> Login using ssh:

```
ssh -i $PATH_TO_SSH_PUBLIC_KEY $PORTAL_USERNAME@$HADOOP_IP
```

In the above:

- $PATH_TO_SSH_PUBLIC_KEY is the location of the public key that has been added to the futuresystems portal
- $PORTAL_USERNAME is the username on the futuresystems portal
- $HADOOP_IP is the IP address of the hadoop frontend node

Hadoop is installed under /opt/hadoop, and you can refer to this location using $HADOOP_HOME. See:

```
hadoop fs
```

and:

```
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples*.jar
```

19.3 Additional Programming Assignemts 2

Todo

merge this into programming.rst. Remove the hello world example and only do the system info. Add information such as processor, Mhz, cores, memory

In this homework, you are expected to run Python or Java programs on FutureSystems or on your local machine. A few examples for beginners will help you to understand how to write and run Java or Python programs on your environment.

19.3.1 Setup

Java and Python are installed on our cloud as explained in Unit 11. Here you choose between Python on your laptop, Python in cloud or Java in cloud.

Local Setup

Download Enthought Canopy Express (free) from https://store.enthought.com/downloads/ including NumPy SciPy Matplotlib

Cloud

Set up Python in cloud or Java in cloud. See Unit 11.

19.3.2 First Program

This code explains how to display a simple string on your screen. You can download or write your own code using your editor.

Java

Download: FirstProgram.java

```
/**
* Sample Program to print out a message
*
* Compile : javac FirstProgram.java
*     Run    : java FirstProgram
*/
public class FirstProgram {
      public static void main(String[] args){
                System.out.println("My first program on Big Data Applications and Analytics!");
        }
}
```

This example prints out the message on your screen by `println` method in the `System` class. In Java Programming, you need to complie your code to execute.

Compiling and Execution

```
javac FirstProgram.java
```

Now, you will have FirstProgram.class file on your system. Java Compiler (javac) creates Java bytecode with a `.class` extension. We will execute the class file with `java` command.

```
java FirstProgram
My first program on Big Data Applications and Analytics!
```

Python

Let's write a same program in Python.

Download: FirstProgram.py

```
# Run python FirstProgram.py
print 'My first program on Big Data Applications and Analytics!'
```

Python function `print` simply displays a message on your screen. Compiling is not necessary in Python. You can run your code directly with `python` command.

```
python FirstProgram.py
My first program on Big Data Applications and Analytics!
```

19.3.3 Display System Information

This is an extension of your first program. We will lean how to import functions and use them to get system information like hostname or username.

Java

We now understand how to print out a message using Python or Java. System information such as time, date, user name or hostname (machine name) can be displayed as well with built-in functions in each language.

Download: FirstProgramWithSystemInfo.java

```
import java.util.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.net.InetAddress;
import java.net.UnknownHostException;

/**
 *  * Sample Program with system information
 *  *
 *  * Compile : javac FirstProgramWithSystemInfo.java
 *  *   Run    : java FirstProgramWithSystemInfo
 *  */
public class FirstProgramWithSystemInfo {
        public static void main(String[] args){

                System.out.println("My first program with System Information!");

                // Print Date with Time
                DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
                Date date = new Date();
                System.out.println("Today is: " + dateFormat.format(date));
                // Print Username
                System.out.println("Username is: " + System.getProperty("user.name"));
                // Print hostname
                try {
                        java.net.InetAddress localMachine = java.net.InetAddress.getLocalHost();
                        System.out.println("Hostname is: " + localMachine.getHostName());
                } catch (UnknownHostException e) {
                        e.printStackTrace();
                        System.out.println("No host name: " + e.getMessage());
                }
        }
}
```

Compiling and Execution

```
javac FirstProgramWithSystemInfo.java
```

```
java FirstProgramWithSystemInfo
My first program with System Information!
Today is: 2015/01/01 18:54:10
Username is: albert
Hostname is: bigdata-host
```

Python

Download FirstProgramWithSystemInfo.py

```
from datetime import datetime
import getpass
import socket

# Run python FirstProgramWithSystemInfo.py
print ('My first program with System Information!')

print ("Today is: " + str(datetime.now()))
print ("Username is: " + getpass.getuser())
print ("Hostname is: " + socket.gethostname())
```

Execution

```
python  FirstProgramWithSystemInfo.py
My first program with System Information!
Today is: 2015-01-01 18:58:10.937227
Username is: albert
Hostname is: bigdata-host
```

19.3.4 Submission of HW4

Submit these compiled files or screenshot image files to IU Canvas

[Java]

- **FirstProgram.class or a screenshot image of the 'FirstProgram' execution (25%) **
- **FirstProgramWithSystemInfo.class or a screenshot image of the 'FirstProgramWithSystemInfo' execution (25%)**

[Python]

- **FirstProgram.pyc or a screenshot image of the 'FirstProgram' execution (25%)**
  - run `python -m compileall FirstProgram.py` to generate `FirstProgram.pyc`
- **FirstProgramWithSystemInfo.pyc or a screenshot image of the 'FirstProgramWithSystemInfo' execution (25%)**
  - run `python -m compileall FirstProgramWithSystemInfo.py` to generate `FirstProgramWithSystemInfo.pyc`

19.3.5 Challenge tasks

- **Run any Java or Python on a FutureSystems OpenStack instance**
  - Submit screenshot images of your terminal executing Java or Python code on FutureSystems
- **Run NumPyTutorial Python on IPython Notebook**
  - Submit screentshot images of your web browser executing NumPyTutorial on FutureSystems
- **Tips: See** tutorials for Big Data Applications and Analytics Shell on FutureSystems

19.4 Installing Cloudmesh Client

1. **What is Cloudmesh Client?** Cloudmesh client allows to easily manage virtual machines, containers, HPC tasks, through a convenient client and API. Hence cloudmesh is not only a multi-cloud, but a multi-hpc environment that allows also to use container technologies.
2. **How to install Cloudmesh Client?** Please follow the steps provided in the below link, * http://cloudmesh.github.io/client/setup.html
3. **How to launch a VM through Cloudmesh Client?** Once you got the above setup done successfully, you can launch your own virtual machines on cloud providers by following the steps in the below link, * http://cloudmesh.github.io/client/quickstart.html#virtual-machines
4. Useful Links:
   - Code: https://github.com/cloudmesh/client.git
   - Complete Documentation: http://cloudmesh-client.readthedocs.org/

For any help regarding the installation or launching of VM's, please drop a mail to the course help group and we will get back to you as soon as we can:

- https://groups.google.com/forum/#!forum/big-data-iu-fall-2016-help

5. Refernces:

   Cloudmesh: *[vLWL+14]*

19.5 Hadoop

- UserGuide (TBD):
  - Use of Hadoop Cluster [link] - http://bdaafall2015.readthedocs.org/en/latest/HadoopClusterAccess.html
  - Running Hadoop Benchmark
    - TeraSort [link] - http://bdaafall2015.readthedocs.org/en/latest/SoftwareProjects.html
    - DFSIO
    - NNBench
    - MRBench
  - NIST NBIS
  - Stock Analysis with MPI
  - Drug-Drug Interaction with Twitter

19.6 Refernces

19.7 Cloud Resources

- Chameleoncloud.org Chameleon
- futuresystems.org *QuickStart*
- Amazon EC2
- Microsoft Azure Virtual Machine

19.8 QuickStart for OpenStack on FutureSystems

This lesson provides a short guide to start using OpenStack on FutureSystems.

19.8.1 Prerequisite

- Portal Account
- SSH Key Registration at portal.futuresystems.org
- FutureSystems Project Membership

19.8.2 Overview

The following contents are discussed in this quickstart guide.

- SSH Access to india.futuresystems.org
- `nova` command
- OpenStack Credential
- **Required Options**
  - flavor
  - image
  - key
  - network ID
- Launch/Terminate Instance

19.8.3 Login to India Login Node

First step, you need to be in india.futuresystems.org. Use one of SSH Client tools, for example:

- Putty, Cygwin or OpenSSH on Windows
- Terminal on Mac OS or Linux

SSH into india, for example:

```
ssh PORTALUSERNAME@india.futuresystems.org
```

Note: Replace PORTALUSERNAME with your actual portal account ID

Connection is granted with the Welcome message like:

```
Welcome to india.futuresystems.org

--------------------------------------------------------------------------

ANNOUNCEMENT
------------
* Do not run jobs on the login node. Any long-running jobs on the
  login node will be terminated without warning.

SUPPORT
-------
If you have a problem, please submit a ticket.
 --> https://portal.futuresystems.org/help


--------------------------------------------------------------------------
CALENDAR -- NEXT 15 DAYS
--------------------------------------------------------------------------
```

19.8.4 Nova Command Tool

OpenStack Compute `nova` command is enabled on India by `module` command like:

```
module load openstack
```

This command can be added to `.bash_profile` to enable OpenStack Client commands when you login. This way you don't need to run the module command every time when you open a new SSH terminal for India. For example:

```
echo "module load openstack" >> ~/.bash_profile
```

See the `.bash_profile` file by:

```
cat ~/.bash_profile
```

If you successfully added the command, the file content looks like:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/bashrc ]; then
      . ~/.bashrc
fi

module load openstack
```

See the last line of the file. `module` command is added. `.bash_profile` or `.bashrc` exists in your home directory to initialze a shell when you login. Any commands or environmental variables e.g. PATH in these files is going to be executed. Find more information online, if you are interested in. Bash Startup Files from GNU

Now, we can use `nova` command, try and see help messages:

```
$ nova
usage: nova [--version] [--debug] [--os-cache] [--timings]
            [--os-auth-token OS_AUTH_TOKEN]
            [--os-tenant-name <auth-tenant-name>]
            [--os-tenant-id <auth-tenant-id>] [--os-region-name <region-name>]
            [--os-auth-system <auth-system>] [--service-type <service-type>]
            [--service-name <service-name>]
            [--volume-service-name <volume-service-name>]
            [--os-endpoint-type <endpoint-type>]
            [--os-compute-api-version <compute-api-ver>]
            [--bypass-url <bypass-url>] [--insecure]
            [--os-cacert <ca-certificate>] [--os-cert <certificate>]
            [--os-key <key>] [--timeout <seconds>] [--os-auth-url OS_AUTH_URL]
            [--os-domain-id OS_DOMAIN_ID] [--os-domain-name OS_DOMAIN_NAME]
            [--os-project-id OS_PROJECT_ID]
            [--os-project-name OS_PROJECT_NAME]
            [--os-project-domain-id OS_PROJECT_DOMAIN_ID]
            [--os-project-domain-name OS_PROJECT_DOMAIN_NAME]
            [--os-trust-id OS_TRUST_ID] [--os-user-id OS_USER_ID]
            [--os-user-name OS_USERNAME]
            [--os-user-domain-id OS_USER_DOMAIN_ID]
            [--os-user-domain-name OS_USER_DOMAIN_NAME]
            [--os-password OS_PASSWORD]
            <subcommand> ...

            Command-line interface to the OpenStack Nova API.

 ...
```

OpenStack provides lots of CLI tools but we focus on Compute API `nova` to learn how VM instances can be started or stopped. Here are some useful resources.

- OpenStack command-line clients
- Launch an instance from an image

19.8.5 OpenStack Credential

`nova` command is ready but we still need a OpenStack credential because we use OpenStack under a project membership and OpenStack verifies our identity by looking at OpenStack credentials. It is simply done by:

```
source ~/.cloudmeh/clouds/india/kilo/openrc.sh
```

and select project by:

```
source ~/.cloudmeh/clouds/india/kilo/fg510
```

Choose a different file if you are in the other project. We chose 'fg510' in this example.

Let's try one of `nova` sub command, for example, see a list of VM images by:

```
nova image-list
```

You may see some images available on your project like:

```
+--------------------------------------+-----------------+--------+----------------------+-------------
| ID                                   | Name            | Status | Server               |
+--------------------------------------+-----------------+--------+----------------------+-------------
| 0245beac-f731-427c-8eb0-4e434af51cf6 | CoreOS-Alpha    | ACTIVE |
```

```
| 9eb8416d-1313-4748-a832-5fe0ecbbdffc | Ubuntu-14.04-64  | ACTIVE |
| f51bd217-f809-46a1-9cdb-604d977ad4e9 | Ubuntu-15.10-64  | ACTIVE |
| 1a80ac5b-4e57-479d-bed6-42e1448e6785 | cirros           | ACTIVE |
| 41b2320f-8c3b-4bd9-8701-a96bdf59100d | fedora23         | ACTIVE |
+--------------------------------------+------------------+--------+--------------------+-------------
```

If the loading credential is failed, you see the errors likes:

```
ERROR (CommandError): You must provide a username or user id via
--os-username, --os-user-id, env[OS_USERNAME] or env[OS_USER_ID]
```

This is because either you do not have `openrc.sh` or a project file i.e. `fg510` or a credential file is broken. Check your file and report your issue to the course email or the ticket system on FutureSystems.

19.8.6 Required Options

There are a few options required to start a new VM instance on OpenStack. Let's talk about SSH Key first.

SSH Key on OpenStack

We will create a VM instance and use it like a normal server which means that we need to use SSH Key to get access to the instance. Typing password is not allowed. This is **a different SSH Key** which is not the key that you registered on either the portal.futuresystems.org or github.com.

```
nova keypair-add quickstart-key > ~/.ssh/quickstart-key
```

This command does two things: one is registering a new public key to Openstack and the other one is saving a new private key to your .ssh directory.

Let's check your new keypair by:

```
nova keypair-list
```

You expect to see *quickstart-key* in your list of keys:

```
+---------------+-------------------------------------------------+
| Name          | Fingerprint                                     |
+---------------+-------------------------------------------------+
| quickstart-key | 68:22:1f:e7:d0:92:7a:68:d8:f5:3d:d2:ca:cd:cd:b9 |
+---------------+-------------------------------------------------+
```

And your private key is:

```
ls -al ~/.ssh/quickstart-key
```

The file should exist:

```
-rw-r--r-- 1 albert users 1751 Jan 25 00:10 /N/u/albert/.ssh/quickstart-key
```

The permission is too open, change the file permission with the owners only read-write permission by:

```
chmod 600 ~/.ssh/quickstart-key
```

And run `ls` command again to confirm the file permission. `-rw-------` is expected.

Passphrase on Private Key

It is important that we have passphrase-enabled SSH key. Let's add a passphrase:

```
ssh-keygen -p -f ~/.ssh/quickstart-key
```

Provide your passphrase, your private key will be updated:

```
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

VM Images

We will launch a new VM instance with a VM image, let's see the list of images by:

```
nova image-list
```

We use `Ubuntu-14.04-64` the latest Ubuntu distribution with 64 bit:

```
+--------------------------------------+------------------+--------+----------------------+-------------
| ID                                   | Name             | Status | Server
+--------------------------------------+------------------+--------+----------------------+-------------
| 0245beac-f731-427c-8eb0-4e434af51cf6 | CoreOS-Alpha     | ACTIVE |
| 9eb8416d-1313-4748-a832-5fe0ecbbdffc | Ubuntu-14.04-64  | ACTIVE |
| f51bd217-f809-46a1-9cdb-604d977ad4e9 | Ubuntu-15.10-64  | ACTIVE |
| 1a80ac5b-4e57-479d-bed6-42e1448e6785 | cirros           | ACTIVE |
| 41b2320f-8c3b-4bd9-8701-a96bdf59100d | fedora23         | ACTIVE |
+--------------------------------------+------------------+--------+----------------------+-------------
```

Server Sizes (Flavors)

We can choose a size of a new VM instance, the flavor.

Try `nova` command like:

```
nova flavor-list
```

We use `m1.small` but available flavors are:

```
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| ID | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| 1  | m1.tiny   | 512       | 1    | 0         |      | 1     | 1.0         | True      |
| 2  | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0         | True      |
| 3  | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0         | True      |
| 4  | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0         | True      |
| 5  | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0         | True      |
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
```

Network ID

We use a private network assigned to our project in OpenStack Kilo.

Try `nova` command like:

```
nova network-list
```

We use `fg510-net` the private network for fg510 project from:

```
+--------------------------------------+-----------+------+
| ID                                   | Label     | Cidr |
+--------------------------------------+-----------+------+
| a9815176-daa7-45ef-98ca-60dff58e7baf | ext-net   | -    |
| e5228c15-38af-4f91-a6de-1590d399427e | fg510-net | -    |
+--------------------------------------+-----------+------+
```

19.8.7 Launch a New VM Instance

We are now ready to start a new VM instance with the options that we chose earlier.

- Image: Ubuntu-14.04-64
- Flavor: m1.small
- Key: quickstart-key
- Network ID: e5228c15-38af-4f91-a6de-1590d399427e
- VM Name: $USER-quickstart

Launch a VM instance by:

```
nova boot --image Ubuntu-14.04-64 --flavor m1.small --key-name quickstart-key
--nic net-id=e5228c15-38af-4f91-a6de-1590d399427e $USER-quickstart
```

Your new VM instance named *quickstart-$USER* will be created shortly. Your launching request is accepted with the messages like:

```
+--------------------------------------+--------------------------------------------------------+
| Property                             | Value                                                  |
+--------------------------------------+--------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                 |
| OS-EXT-AZ:availability_zone          | nova                                                   |
| OS-EXT-STS:power_state               | 0                                                      |
| OS-EXT-STS:task_state                | scheduling                                             |
| OS-EXT-STS:vm_state                  | building                                               |
| OS-SRV-USG:launched_at               | -                                                      |
| OS-SRV-USG:terminated_at             | -                                                      |
| accessIPv4                           |                                                        |
| accessIPv6                           |                                                        |
| adminPass                            | juXmTsv66                                              |
| config_drive                         |                                                        |
| created                              | 2016-01-26T19:42:32Z                                   |
| flavor                               | m1.small (2)                                           |
| hostId                               |                                                        |
| id                                   | a700fad0-ad69-4036-b184-cdca18d516a4                   |
| image                                | Ubuntu-14.04-64 (f51bd217-f809-46a1-9cdb-604d977ad4e9) |
| key_name                             | quickstart-key                                         |
| metadata                             | {}                                                     |
| name                                 | albert-quickstart                                      |
| os-extended-volumes:volumes_attached | []                                                     |
| progress                             | 0                                                      |
| security_groups                      | default                                                |
| status                               | BUILD                                                  |
| tenant_id                            | 0193f2237d3d342f106fbf04bdd2f                          |
```

```
| updated                              | 2016-01-26T19:42:33Z                                                    |
| user_id                              | 4186710ab90a642455889d3a8b51a                                           |
+--------------------------------------+-------------------------------------------------------------------------+
```

19.8.8 Access to VM

Booting up a VM instance takes a few minutes. Let's check its status by:

```
nova list
```

If you see it is active and running like

```
+--------------------------------------+------------------+--------+------------+-------------+------+-----
| ID                                   | Name             | Status | Task State | Power State | Netwo
+--------------------------------------+------------------+--------+------------+-------------+------+-----
| a700fad0-ad69-4036-b184-cdca18d516a4 | albert-quickstart| ACTIVE | -          | Running     | fg510
+--------------------------------------+------------------+--------+------------+-------------+------+-----
```

We may try SSH into the *$USER-quickstart* VM. Note that you see your portal ID in *albert*. SSH into the private IP addres and like you SSHed to India but with a different SSH key like:

```
ssh -i ~/.ssh/quickstart-key 10.0.6.4 -l ubuntu
```

`-l ubuntu` parameter is added to specify a default user name of the base image *Ubuntu-14.04-64*.

You provide your SSH passphrase to get access and you will see a welcome message on your new Ubuntu 15.10 virtual server:

```
Welcome to Ubuntu 14.04 (GNU/Linux 3.13.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

   Get cloud support with Ubuntu Advantage Cloud Guest:
       http://www.ubuntu.com/business/services/cloud

 0 packages can be updated.
 0 updates are security updates.


 The programs included with the Ubuntu system are free software;
 the exact distribution terms for each program are described in the
 individual files in /usr/share/doc/*/copyright.

 Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
 applicable law.

 To run a command as administrator (user "root"), use "sudo <command>".
 See "man sudo_root" for details.

 ubuntu@albert-quickstart:~$
```

You are the owner of your new VM instance. You can install any software and manage services as a root with sudo command, if you like.

19.8.9 Terminate VM

Now, we need to learn how to terminate a VM instance once our work on a vm is completed. Running idle VM instances is not allowed in the course because we share compute resources with other students.

Use `nova` command to terminate:

```
nova delete a700fad0-ad69-4036-b184-cdca18d516a4
```

or:

```
nova delete $USER-quickstart
```

You will see the message like:

```
Request to delete server a700fad0-ad69-4036-b184-cdca18d516a4 has been
accepted.
```

ID is unique but Name of your VM is not. Try to use ID when you terminate VM instance.

19.8.10 FAQ

17. `nova` command doesn't work with the error:

    ERROR (Unauthorized): The request you have made requires authentication. (HTTP 401) (Request-ID: req-82f94837-78e7-4abd-a413-ff7645c45a7f)

A. Your OpenStack credential (i.e. openrc.sh) is not valid. Check your file and project ID. If a problem is consistent, report to the course team.

19.8.11 Any Questions?

Please use Slack or the course email, if you have issues or questions regarding this tutorial.

19.9 Chameleon Cloud

Chameleon Cloud provides OpenStack Cloud with KVM or Baremetal for developing with the Big Data Stack. https://www.chameleoncloud.org/ Chameleon Cloud provides an environment for experimenting with cloud environments. This class does not support use of Chameleon cloud for any assignment or project work but computing resources may be available upon request with limited access and allocation.

There are some differences between FutureSystems OpenStack and Chameleon OpenStack.

- different login usernames (`cc` for all instead of `ubuntu`, `centos`, etc) for the images
- limited resource availability
- resource usage is charged to a finite allocation (thus you need to terminate your instances if you do not use them).

19.9.1 Getting Access

** Closed as of 08/02/2016 **

1. Create an account on [[https://www.chameleoncloud.org/][Chameleon Cloud]]

2. Send your Chameleon username to <course email> Note: the subject *MUST* be #+BEGIN_EXAMPLE Chameleon Cloud Access #+END_EXAMPLE
3. **We will then add you to the project for this course. *IMPORTANT*: you will** not be able to use Chameleon until you are added. We will reply to your request with a confirmation email.

19.9.2 Setup Instructions

1. ssh into `india.futuresystems.org`
2. **Go to the** Chameleon Cloud OpenStack Dashboard and download the openrc file (check under the `API Access` tab)
3. Upload the openrc file to the `india` node:: $ scp CH-817724-openrc.sh $PORTALID@india.futuresystems.org:~
4. **Upload your india ssh key to your 'profile on github.com** <https://github.com/settings/ssh>'_:

   albert@i136 ~ $ cat ~/.ssh/id_rsa.pub
5. Source the openrc file (*only* the chameleon openrc file):: albert@i136 ~ $ source ~/CH-817724-openrc.sh
6. Load the OpenStack module (same as with kilo on india):: albert@i136 ~ $ module load openstack

At this point you the nova commands will control Chameleon.

19.9.3 Big Data Stack

You can now follow the Big Data Stack Readme for starting and deploying your BDS

19.9.4 Notes

`apt` related errors

You may occasionally get an error when one of the tasks calls to apt, either to update the cache or install packages. This will likely manifest as a `Failed to fetch` with an `Error 403 Forbidden` error. The root cause for this is not yet known, but it seems related to a network saturation issue. Nonetheless, the workaround is simple: rerun the playbook that failed. This may need to be repeated a few times, but this has been sufficient to resolve the issue when I encounter them.

19.10 Example

This is literate programming

```
print "Hello Big Data"
```

```
Hello Big Data
```

CHAPTER 20

Contributing

Contributing content to this web page is easy. First you have to clone the directory with the command

```
git clone https://gitlab.com/cloudmesh/fall2016/tree/master
```

One you clone this you can do:

```
make html
make view
```

To view locally. After you push your changes will become available at:

```
http://bdaafall2016.readthedocs.io/en/latest
```

CHAPTER 21

Todos

21.1 General

- fill in python lesson *Classes* section
- fix example projects
- fix python introduction (proper sectioning)
- fix virtualenv link
- fix assignments
- fix futuresystems apply
- fix chameleon cloud
- identify if we can use jetstream

Todo

merge this into programming.rst. Remove the hello world example and only do the system info. Add information such as processor, Mhz, cores, memory

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/bdaafall2016/checkouts/latest/docs/source/as1.rst, line 4.)

Todo

Gregor. Goto LaTeX documentation and consolidate into single latex.rst

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/bdaafall2016/checkouts/latest/docs/source/links.rst, line 5.)

Todo

HadoopClusterAccess.html

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/bdaafall2016/checkouts/latest/docs/source/programm line 19.)

Todo

list requirements as differing from “Common Requirements”

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/bdaafall2016/checkouts/latest/docs/source/projects.rs
line 149.)

Todo

list requirements as differing from “Common Requirements”

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/bdaafall2016/checkouts/latest/docs/source/projects.rs
line 212.)

CHAPTER 22

Changelog

22.1 %%version%% (unreleased)

22.1.1 New

- Added geolocation quiz. [Gregor von Laszewski]
- PRG1 is due Dec 2nd, recommended to finish by Oct 14, if difficulties we recommend you do a paper. [Gregor von Laszewski]
- Added driverslicense due date to calendar. [Gregor von Laszewski]
- Mark plotviz section as voluntary. [Gregor von Laszewski]
- Update office hours. [Gregor von Laszewski]
  - Tue 10-11am EST, typically Gregor
  - Thu 6-7pm EST, typically Gregor
  - Sun 4-6pm EST, either Jerome or Prahanth
  - Tue 7-8pm, either Jerome or Prahanth
  - Wed 7-8pm, either Jerome or Prahanth
- Add git push and pull video. [Gregor von Laszewski]
- Add rst refcard. [Gregor von Laszewski]
- Add weeks that we recommend students work on project. [Gregor von Laszewski]
- Urs: remove link to not used google grou, use Piazza instead. [Gregor von Laszewski]
- Added pycharm video. [Gregor von Laszewski]
- Recommend against using canopy and removing the canopy movie. [Gregor von Laszewski]
- Fix the arror in report length on the assignments page. [Gregor von Laszewski]
- Add more prominent links for project titles. [Gregor von Laszewski]
- Added simple ssh explanation. [Gregor von Laszewski]
- Updated overview calendar to give a bit more time. [Gregor von Laszewski]
- Add the development vm video. [Gregor von Laszewski]
- Add virtualbox guest additions video. [Gregor von Laszewski]

- Add virtual box ubuntu desktop video. [Gregor von Laszewski]
- Clarify group work for paper 3. [Gregor von Laszewski]
- Dev add missing file. [Gregor von Laszewski]
- Add homework upload video. [Gregor von Laszewski]
- Dev include upload instructions. [Gregor von Laszewski]
- Added a jabref video. [Gregor von Laszewski]
- Fix the duplicated numbering for d2 to only apply as bonus. [Gregor von Laszewski]
- Residential class meetings have been merged into one class on Friday. [Gregor von Laszewski]
- Clarify duedate of p1. [Gregor von Laszewski]
- Simplified the Paper Homework 1 and clarified the analysis of the data posted in the discussion 1. [Gregor von Laszewski]
- Added sharelatex video. [Gregor von Laszewski]
- Clarify that Fridays are new assignments issued which are due the next week Friday. [Gregor von Laszewski]
- Update syllabus video. [Gregor von Laszewski]

22.1.2 Fix

- Fix page requirements in project. [Gregor von Laszewski]

22.1.3 Other

- Ne:usr: add python learning to the calendar, which already has been announced. [Gregor von Laszewski]
- Update README.rst. [Gregor von Laszewski]

  This reverts commit 97e597d067f3db5f12e045992ae0581396a68963.
- Add license. [Gregor von Laszewski]
- Add changelog. [Gregor von Laszewski]
- Add README. [Gregor von Laszewski]

[vLWL+14] Gregor von Laszewski, Fugang Wang, Hyungro Lee, Heng Chen, and Geoffrey C. Fox. Accessing Multiple Clouds with Cloudmesh. In *Proceedings of the 2014 ACM International Workshop on Software-defined Ecosystems*, BigSystem '14, 21–28. New York, NY, USA, 2014. ACM. URL: http://doi.acm.org/10.1145/2609441.2609638, doi:10.1145/2609441.2609638.