Big Data Applications
e534

Geoffrey C. Fox
Gregor von Laszewski
Editor

laszewski@gmail.com

https://cloudmesh-community.github.io/book/vonlaszewski-big-data-applications.epub

August 11, 2019 - 04:22 AM

Created by Cloudmesh & Cyberaide Bookmanager, https://github.com/cyberaide/bookmanager

Big Data Applications

Geoffrey C. Fox Gregor von Laszewski

Big Data Applications

1 PREFACE

Sun Aug 11 04:22:30 EDT 2019

1.1 CORRECTIONS

The material collected in this document is managed in

- https://github.com/cloudmesh-community/book/chapters

In case you see an error or like to make a contribution of your own section or chapter, you can do so in github via pull requests.

The easiest way to fix an error is to read the ePub and click on the cloud  symbol in a heading where you see the error. This will bring you to an editable document in github. You can directly fix the error in the web browser and create there a pull request. Naturally, you need to be signed into github before you can edit and create a pull request.

As a result contributors and authors will be integrated automatically next time we compile the material. Thus even if you corrected a single spelling error, you will be acknowledged.

1.2 CONTRIBUTORS

Contributors are sorted by the first letter of their combined Firstname and Lastname and if not available by their github ID. Please, note that the authors are identified through git logs in addition to some contributors added by hand. The git repository from which this document is derived contains more than the documents included in this document. Thus not everyone in this list may have directly contributed to this document. However if you find someone missing that has contributed (they may not have used this particular git) please let us know. We will add you. The contributors that we are aware of include:

> *Anand Sriramulu, Ankita Rajendra Alshi, Anthony Duer, Arnav,*

Averill Cate, Jr, Bertolt Sobolik, Bo Feng, Brad Pope, Dave DeMeulenaere, De'Angelo Rutledge, Eliyah Ben Zayin, Eric Bower, Fugang Wang, Geoffrey C. Fox, Gerald Manipon, Gregor von Laszewski, Hyungro Lee, Ian Sims, IzoldaIU, Javier Diaz, Jeevan Reddy Rachepalli, Jonathan Branam, Juliette Zerick, Keith Hickman, Keli Fine, Mallik Challa, Mani Kagita, Miao Jiang, Mihir Shanishchara, Min Chen, Murali Cheruvu, Orly Esteban, Pulasthi Supun, Pulasthi Supun Wickramasinghe, Pulkit Maloo, Qianqian Tang, Ravinder Lambadi, Richa Rastogi, Ritesh Tandon, Saber Sheybani, Sachith Withana, Sandeep Kumar Khandelwal, Silvia Karim, Swarnima H. Sowani, Tharak Vangalapat, Tim Whitson, Tyler Balson, Vafa Andalibi, Vibhatha Abeykoon, Vineet Barshikar, Yu Luo, ahilgenkamp, aralshi, bfeng, brandonfischer99, btpope, garbeandy, harshadpitkar, himanshu3jul, hrbahramian, isims1, janumudvari, joshish-iu, juaco77, karankotz, keithhickman08, mallik3006, manjunathsivan, qianqian tang, rajni-cs, rirasto, shilpasingh21, swsachith, trawat87, tvangalapat, varunjoshi01, vineetb-gh, xianghang mi, zhengyili4321

1.3 CREATING THE EPUBS FROM SOURCE

In case you wish to create the ePub from source, we have included this section.

However the easiest way is to use our docker container as described in Section 1.3.1.

1.3.1 Docker

We recommend the docker creation method for

- Ubuntu
- Windows 10
- macOSX

1.3.1.1 Using OSX

The easiest way to create a system that can compile the book on macOS, is to

use a docker container. To do so you will need to first install docker on macOS while following the simple instructions at

- https://docs.docker.com/docker-for-mac/install/

Once you have docker installed, you can follow the instructions in Section 1.3.1.

1.3.1.2 Using the Docker Image

In case you have docker installed on your computer you can create ePubs with our docker image. To create that image by hand, we have included a simple makefile. Alternatively you can use our image from dockerhub if you like, it is based on ubuntu and uses our Dockerfile.

First, you need to download the repository:

```
$ git clone https://github.com/cloudmesh-community/book.git
cd book
```

To open an interactive shell into the image you say

```
$ make shell
```

Now you can skip to Section 1.3.4 and compile the book just as documented there.

Please note that we have not integrated pandoc-mermaid and pandoc-index at this time in our docker image. If you like to contribute them, please try it and make a pull request once you got them to work.

In case you want to create or recreate the image from our Dockerfile (which is likely not necessary, you can use the command

```
$ make image
```

1.3.2 Using the Native System

In case you like to use your native environment (which is typically faster than the container) you need to make sure you have an up to date environment.

Please note, that you must have at least Pandoc version 2.5 installed as earlier

versions will not work. We recommend that you use Python version 3.7.4 to run the scripts needed to assamble the document. However eralier version of Python 3 may also work, but are not tested. You can check the versions with

```
$ pandoc --version
$ python --version
```

1.3.3 Using Vagrant

In case you have installed vagrant on your computer which is available for macOS, Linux, and Windows 10, you can use our vagrant file to start up a virtual machine that has all software installed to create the ePub.

First, you need to download the repository:

```
$ git clone https://github.com/cloudmesh-community/book.git
$ cd book
```

Next you have to create the virtual machine with

```
$ vagrant up
```

You can loginto the VM with

```
$ vagrant ssh
```

The book folder will be mounted in the VM and you can can follow the instructions in Section 1.3.1.

1.3.4 Creating a Book

Oncw you have decided for one of the methods, you can create a book.

To create a book, you have to first check out the book source from github with if you have not yet done so (for example if you were to use the docker container method):

```
git clone git@github.com:cloudmesh-community/book.git
```

Books are organized in directories. We currently have created the following directories

```
./book/cloud/
./book/big-data-applications/
```

```
./book/pi
./book/writing
./book/222
./book/516
```

To compile a book go to the directory and make it. Let us assume you like to create the cloud book for cloud

```
$ git clone https://github.com/cloudmesh-community/book.git
$ cd cloud
$ make new
```

To view it you say

```
$ make view
```

After you have done modifications, you need to do one of two things. In case you add new images you need to use

```
$ make new
```

otherwise you can just use

```
$ make
```

The structure of the books is maintained in the yaml file `chapters.yaml`. You can add this chapter to the yaml file, but discuss this first with Gregor. In case you add a new chapter, you have to say

```
$ make clean
$ make update
$ make
$ make view
```

1.3.5 Publishing the Book to GitHub

> ⚠ *This task is only to be done by Gregor von Laszewski. You will not have to do this step.*

To publish the book say

```
$ make publish
```

1.3.6 Creating Unpublished Drafts

Developers of the manual can modify the `Makefile` and locate the variable `DRAFT=` to add additional sections and chapters they work on, but should not yet been

distributed with the main publication. Simply add them to the list and say

```
$ make draft
$ make view
```

to create the draft sections only and view them.

To conveniently call them in a lazy fashion in a terminal you could use the following two aliases.

```
alias m='make; make view'
alias d='make draft; make view'
```

This allows you to typ `m` for the main volume and `q` for the draft. Please note that all artifacts are written into the dest folder.

1.3.7 Creating a New Book

Let us assume you like to create a new book. The easiest way to start is to copy from an existing book. However, make sure not to copy old files in dest. Let us assume you like to call the book gregor and you coppy from the 222 directory.

You have to do the following

```
$ cd 222
$ make clean
$ cd ..
$ cp -r 222 gregor
```

Now edit the file chapters.yaml and copy the section with `BOOK_222=` to `BOOK_gregor=`. Make modifications to the outline as you see fit.

Now you can create the book with

```
$ cd gregor
$ make update
$ make new
```

1.3.8 Managing Images

In case you have added images to the book, they must be on the same level as your contribution, but in a directory called images. E.g.

```
./chapters/cloud/mydocument.md
./chapters/cloud/images/myimage.md
```

In the document the image is than refered to as

```
![My imaage caption](images/myimage.md){#fig:cloud-myimage}
```

The label `#fig:cloud-myimage` must be unique in all of the documents. While adding the directory cloud before the image name this is the case in our example.

1.4 ePub Readers

This document is distributed in ePub format. Every OS has a suitable ePub reader to view the document. Such readers can also be integrated into a Web browser so that when you click on an ePub it is automatically openedin your browser. As we use ePubs the document can be scaled based on the users prefernc If you ever see a content that does not fit on a page we recommend you zoom out to make sure you can see the entire content.

We have made good experiences with the following readers:

- **macOSX**: Books, which is a build in ebook reader
- **Windows 10**: Microsoft edge, but it must be the newest version, as older versions have bugs. Alternatively use calibre
- **Linux**: calibre

If your have an iPad or Tablet with enough memory, you may also be able to use them.

1.5 Notation

The material here uses the following notation. This is especially helpful, if you contribute content, so we keep the content consistent.

if you like to see the details on how to create them in the markdown documents, you will have to look at the file source while clicking on the cloud in the heading of the Notation section (Section 1.5). This will bring you to the markdown tex, but you will still have to look at the raw content to see the details.

`![Github](images/github.png)`

> *If you click on the*  *in a heading, you can go directly to the > document in github that contains the next content. This is > convenient to fix errors or make additions to the content. The cloud will be automatically added upon inclusion of a new markdown file that includes in its first line a section header.*

$

> *Content in bash is marked with verbatim text and a dollar sign*

```
$ This is a bash text
```

[@las14cloudmeshmultiple]

> *References are indicated with a number and are included in the > reference chapter [@las14cloudmeshmultiple]. Use it in markdown with* `[@las14cloudmeshmultiple]`*. References must be added to the* `refernces.bib` *file in BibTex format.*

> *Chapters marked with this emoji are not yet complete or have some issue that we know about. These chapters need to be fixed. If you like to help us fixing this section, please let us know. Use it in markdown with* `![No](images/no.png)`**.**

 REST 36:02

> *Example for a video with the* `![Video](images/video.png)` *emoji. Use it in markdown with* `[![Video](images/video.png) REST 36:02](https://youtu.be/xjFuA6q5N_U)`

 Slides 10

> *Example for slides with the* `![Presentation](images/presentation.png)` *emoji. These slides may or may not include audio.*

 Slides 10

Slides without any audio. They may be faster to download. Use it in markdown with `[![Presentation](images/presentation.png) Slides 10](TBD)`.

A set of learning objectives with the `![Learning](images/learning.png)` *emoji.*

A section is release when it is marked with this emoji in the syllabus. Usie it in markdown with `![Ok](images/ok.png)`.

Indicates opportunities for contributions. Usie it in markdown with `![Question](images/question.png)`.

Indicates sections that are worked on by contributors. Use it in markdown with `![Construction](images/construction.png)`.

Sections marked by the contributor with this emoji `![Smiley](images/smile.png)` *when they are ready to be reviewed.*

Sections that need modifications are indicated with this emoji `![Comment](images/comment.png)`.

A warning that we need to look at in more detail `![Warning](images/warning.png)`

> *Notes are indicated with a bulb* `![Idea](images/idea.png)`

Other emojis

> *Other emojis can be found at https://gist.github.com/rxaviers/7360908. However, note that emojis may not be viewable in other formats or on all platforms. We know that some emojis do not show in calibre, but they do show in macOS iBooks and MS Edge*

1.5.1 Figures

Figures have a caption and can be refereed to in the ePub simple with a number. We show such a refernce pointer while referring to Figure 1.

Figure 1: Figure example

Figures must be written in the md as

```
![Figure example](images/code.png){#fig:code-example width=1in}
```

Note that the text must be in one line and must not be broken up even if it is longer than 80 characters. You can refer to them with `@fig:code-example`. Please note in order for numbering to work figure references must include the `#fig:` followed by a unique identifier. Please note that identifiers must be really unique and that identifies such as `#fig:cloud` or similar simple identifiers are a poor choice and will likely not work. To check, please list all lines with an identifier such as.

```
$ grep -R "#fig:" chapters
```

and see if your identifier is truly unique.

1.5.2 Hyperlinks in the document

To create hyperlinks in the document other than images, we need to use proper

markdown syntax in the source. This is achieved with a refernce for example in sections headers. Let us discuss the refernce header for this section, e.g. Notation. We have augmented the section header as follows:

```
# Notation {#sec:notation}
```

Now we can use the refernce in the text as follows:

```
In @sec:notation we explain ...
```

It will be rendered as: In Section 1.5 we explain …

1.5.3 Equations

Equations can be written as

```
$$a^2+b^2=c^2$${#eq:pythagoras}
```

and used in text:

$$a^2 + b^2 = c^2 \qquad (1)$$

It will render as: As we see in Equation 1.

The equation number is optional. Inline equations just use one dollar sign and do not need an equation number:

```
This is the Pythagoras theorem: $a^2+b^2=c^2$
```

Which renders as:

This is the Pythagoras theorem: $a^2 + b^2 = c^2$.

1.5.4 Tables

Tables can be placed in text as follows:

```
: Sample Data Table {#tbl:sample-table}

x   y   z
--- --- ---
1   2   3
4   5   42
```

As usual make sure the label is unique. When compiling it will result in an error

if labels are not unique. Additionally there are several md table generators available on the internet and make creating table more efficient.

2 ORGANIZATION

2.1 ORGANIZATION

This class is an online class. Online classes require you to be very disciplined in order to execute the tasks necessary for the class in time. It is your responsibility to organize the lessons so that you can complete them not only by the end of the semester, but also in time for conducting your assignments. This is a great opportunity for you to structure the class based on your availability. The classes are attended by two different set of students. One set are remote online students, while to other are residential students. For the residential students we have a mandatory in person meeting that takes place at the posted location and hours once a week. For pure online students we have weekly online hours that we will identify based on our availability and a doodle poll.

Figure *Components of the Class i523, i423, e534* showcases the different parts of the class. If you have taken a previous class with us you are able to continue your previous project upon approval. It must however be a significant improvement. Please note that the in i523 and i524 the project and it's report can be substituted by a longer term paper that does not require programming. As this is a significant reduction in work and goals, for that class, the maximum grade in this case for the entire class can only be an A-.

There will not be any bonus projects or tasks to improve grades. Instead make sure your deliverables of the few assignments are truly outstanding.

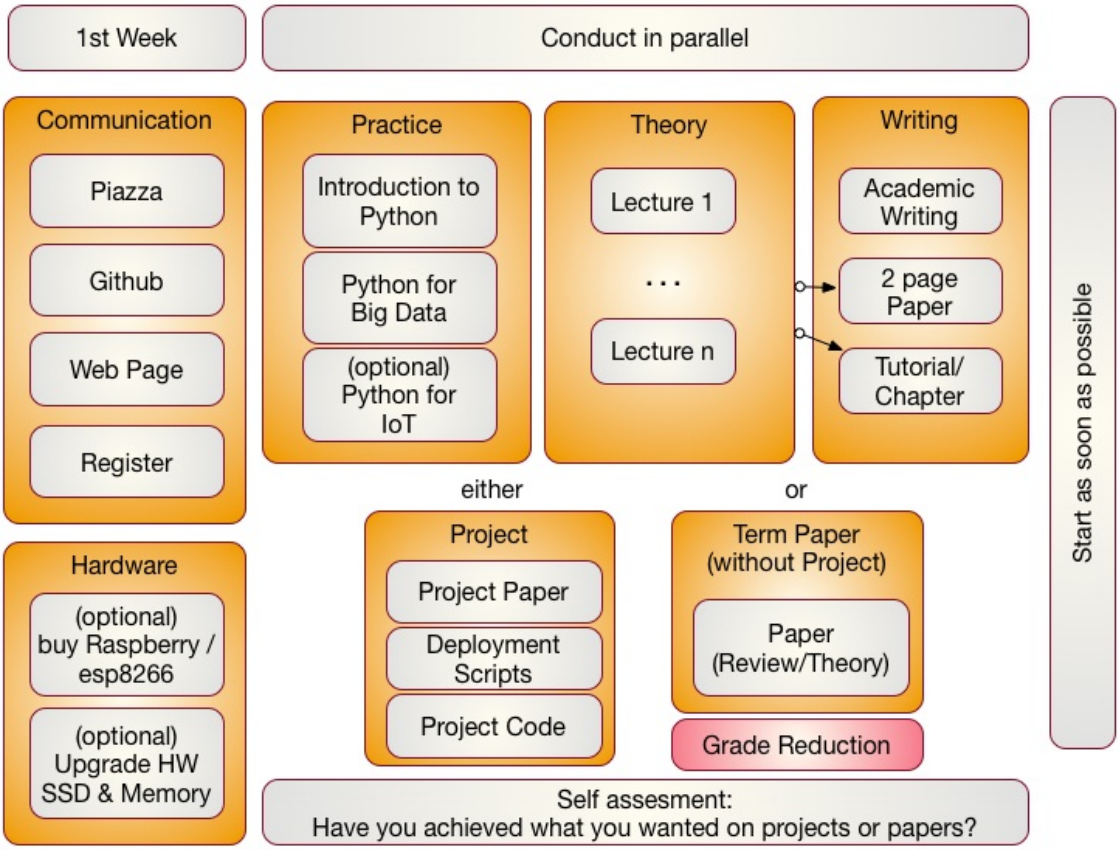


Figure: Components of the Class i523, i423, e534

The content for this class will be available through a series of documents that will be regularly updated and are linked from this document. All communication is done with Piazza. issues.

2.1.1 First Week

In the first week we will be introducing you how we communicate to you. Naturally you need to register for the class. Once you register you need to set up a number of services.

2.1.2 Access to Clouds

As part of the course you will also need access to a cloud. We will try our best to provide you with access to suitable computers for the class, but do be reminded that the amount of time and access to supercomputers and clouds we offer is limited. Our class policy is to use the compute resources only when you really need them. Thus you **must** shut down your VMs when they are not in use. It would be a violation of class policy if we would find out through an analysis of the cloud logs that you unnecessarily keep your VMs running. Thus we will implement a **strict policy** that you must record yourself how many hours you run VM's and provide this information to us. We will than compare that time with the time recorded by the computer system as well as with your target application and will deduct points form your project if you can not justify why you have not shut down your VMs. A resource section needs to be added to your report justifying the used resources.

Why is this such a big deal you may ask? For example we estimate if every student in class violates this policy it would cost about $200000 to rent the time for this on a public cloud. Due to this high cost, we no longer tolerate deliberate violations of the policy and will terminate your account. Furthermore, violators will have to find alternative resources to conduct their projects while not using our resources. In our case the problem is even beyond the issue of cost as our allocation on the clouds would be terminated due to abuse and **no student**, including those that follow policies, could use the cloud. It may take weeks to reestablish cloud access and would effect every student in class.

We will provide clarification for accessing cloud resources and teach you how to avoid getting in such a situation. I am sure that a future employer of yours will be real happy if you have a deep understanding of resource vs. cost estimate.

Listing the used computer time for your project is part of your report.

2.1.3 Using Your Own Computer

In many cases however you could and are recommended to use your own personal computer, but make sure the computer is up-to-date. We also like to make sure that you do not use a work computer as you need to avoid that when you develop a cloud program you do not by accident introduce a security risk on your machine. This does not mean that you need to buy a new computer, or need to upgrade it. However, if you consider an upgrade of an older machine please

consider the following.

These days we recommend that your computer has a solid-state drive and fast memory (put as much memory in your machine as is supported). We recommend 16 GB off main memory which gives you enough space to run containers, virtual machines and naturally the main operating system. We fond that students with only 8GB could do the work but it was slow. In some cases the memory to conduct their projects was not sufficient. Make sure you follow your upgrade guide to your computer and by suitable memory chips. In most cases you have to buy them in **pairs** and make sure all chips in your computer are the same. When it comes to buying a solid-state drive, make sure that you buy one that is compatible with motherboards bus speed. As you may want to reuse your solid-state drive at a later time I suggest to get a 6GB/s SSD and not a 3GB/s.

In case of Windows, your could also get yourself a UBS stick or external SSD drive and place ubuntu on it. You could than use your bios to boot of from that drive. This way you do not have to modify anything on your computer. This method works very well for most computers and allows you to use the maximum memory while for example using ubuntu.

Students that only had a chromebook and took this class gave us the feedback that they are too inconvenient as they do not allow you to program directly in python on them and the ssh terminals to login to other computer although working are not supporting the GUI tools.

Another option is (if money is an issue) you can buy a Raspberry Pi and edit your programs there and when satisfied run them on a cloud. However a PI is small and has only very limited memory and processing power.

We also like to remind you that this course does not require you to purchase expensive text books, thus the money you safe on this could be used in upgrading your hardware or renting yourself from your own money time on AWS. However, be careful with the cloud its easy to spend lots of money there if you are not careful.

2.1.3.1 Self Discipline

As this class has no graded tests and only few graded homework, we like that

you deliver an **exceptional** project report or paper. Instead of focusing on preparing for tests we provide you with the opportunity to **explore** without the pressure of grades. However you should not give up or take the easy way out or it will effect you in your project execution. Also, to achieve your best do not just say: *We do not have a test, so let me not do this weeks assignment, let me do it next week.* After a couple of times with this attitude you will be in big trouble. All this requires discipline. For example, if you believe you are so good that you can do a project within one week before deadline, you will **certainly fail**. To avoid this and to introduce discipline, you will also be monitored on progress and we check your github for activities which will be part of the participation grade.

It will be up to you to assess what you want to deliver before handing it in to us. Self assessment or a check with other students is a real good way to do that. You should not expect to get an A if you yourself are not convinced about your project or are unsure about it. Common sense prevails.

2.1.3.2 Fun

I hope you have fun and are able to integrate in the projects your own thoughts and interests.

We have quotes form students such as

> *"This is the best class I have taken …"*

or

> *"I really enjoyed taking this class and having maximum flexibility to schedule the lectures."*

or

> *"The lessons learned from this class were adopted within my company."*

Furthermore you should know that the way we teach the class has also been adopted in STEM classes. As a result a team coached by Gregor von Laszewski

won an award at the FLL Robotics World Championship. The certainly had lots of fun and integrated their own ideas into the project that won the award.

2.1.3.3 Uniqueness

We will try to have every project or paper to be non overlapping with another topic, If there are overlaps we may ask you to modify your focus.

2.1.3.4 Continuation

If you like to put additional effort in the project, the report could be made to a conference or workshop paper. Dr. von Laszewski is happy to help as co-author.

2.1.4 Parallel Tracks

In this class we have three parallel tracks.

2.1.4.1 Track 1: Practice

Track 1 introduces you to using python for Big Data. We recommend that you do know a programming language for any of our courses. Learning a programming language is not part of the hours you spend for this class. It is an additional time requirement that you must plan for. Maybe you want to take for example a python programming language class at the same time. This can also be done in self study. Although you do not need to know any programming language, it is certainly useful as it will make this course much easier for you. We had students that had no prior programming knowledge and successfully completed the course. So we know it can be done. The course is designed in such a fashion, that there is enough time to learn programming and do a project.

We provide you with a general introduction to Python. This includes enough knowledge so you can conduct a project with it. We will build on these technologies to introduce you to python libraries that can be used for big data. Not every section in the Python chapter will need to be used in this class. At minimum you must understand python classes, and the map reduce function.

2.1.4.2 Track 2: Theory

The theory track includes a number of online lectures that introduces you to a variety of topics related to Big Data. You have especially the opportunity to become part of a project that would contribute to the understanding and the development of a Big Data Architecture developed in collaboration with NIST. Other topics that are covered include IoT, Health Care, Physics, Science, Biology, Genomics, and so forth. We will update the Theory track and will release lectures in the specified areas. Some lectures may be used in multiple classes.

2.1.4.3 Track 3: Writing

You have a choice in this class between writing a two page review paper about a big data technology or application (area), or contribute a chapter to this document. We explain next the difference:

Review Paper: A review paper will introduce your into how to write an academic paper and conduct proper bibliography management. Knowing how to write is a preparation for your term project.

You will be writing a paper that is 2 pages long (in a particular format, typically ACM) possibly within a team. In case you work in a team you have to produce as many papers as you have team members. We like to avoid that all students take the same topic. We will use github to avoid that everyone chooses same topic. Knowing how to write is a preparation fo your project or term paper.

We noticed a curious observation in previous classes. Any paper written in MSWord was inferior. Thus we no longer provide the choice to write papers in MSWord in order for you to achieve your best. Papers and Project reports must be written in LaTeX or markdown. For the classes starting in 2018 we do prefer markdown and may restrict all document to this format.

Chapter: A is chapter to a review paper, but is written in markdown and can be added to the lecture notes. A chapter should be formulated in a consistent form and is equivalent in length (number of words) to those of the 2 page paper. Bibliography management is conducted in bibtex and can be used in the

markdown document.

Important in both cases is that you stay focused. You can assume that if you write a document about “Big Data in Baseball”, you do not spend 1.5 pages to describe what big data is and only half a page where baseball fits in. What you should do is focus on the topic. A chapter could also include some practical lessons with real programming lessons.

2.1.4.4 Track 4: Term Paper/Project

The major deliverable of the course is a term project or paper. The exact details will be posted on the Web page in this document. The important part is that you start on this project once you are sufficiently familiar with Track 1-3. However you can also use the project to for example learn python and engage in a goal oriented learning activity while working towards implementing your project and integrating the python lessons that you encounter. The same is valid for the theory.

It is **expected** that you identify a suitable analysis and data set for the project and that you learn how to apply this analysis as well as justify it. It is part of the learning outcome that you determine this instead of us giving you a topic.

Furthermore, it is also important to note that if you do not do a project (this is your option) the maximum grade for the entire class is limited to an A-. This is achieved simply by reducing the grade of your term report by a full grade due to the distribution of the grade this will result in a fractional grade reduction and limits the maximum grade to an A-.

Starting in 2018 the paper format will be Markdown.

2.1.5 Plagiarism

In the first week(s) of class you will need to read the information about plagiarism. If there are any questions about plagiarism we require you to take a course offered from the IU educational department.

Warning:

> *If we find cheating or plagiarism, your assignment will be receiving an F. This especially includes copying text without proper attribution. We are required to follow IU policy and report your case to the dean of students who may elect to expel you form the university. Please understand that it is your doing and the instructors have no choice as to follow university policies. Thus, please do not blame the instructors for your actions. Excuses such as "I missed the lecture on plagiarism", "I forgot to include the original reference as I ran out of time", "I did not understand what plagiarism is" do not apply as we explicitly make the policies clear. This applies to all material prepared for class including assignments, exercises, code, sections, tutorials, papers, and projects. If there is no time, do not submit and instead of an F ask for an incomplete. In fact if you know you have plagiarized, do not even have us review your paper.*

For more information on this topic please see:

- https://studentaffairs.indiana.edu/student-conduct/misconduct-charges/academic-misconduct.shtml

Furthermore you are supposed to review our lecture material on plagiarism and take the plagiarism test. The information is located at:

- Scientific Writing with Markdown

In Piazza a form will be posted that will ask you for your passing ID. If the form is not yet posted, please be patient till it is.

2.2 Course Policies

We describe briefly some class policies.

2.2.1 Discussion via Piazza

1. All communication is done in Piazza. It is an IU approved communication tool and superior to CANVAS discussion list
2. CANVAS is only used with students that are not in Piazza. The only

messages they will get is to activate Piazza and use the class Piazza
3. You are allowed to use whatever calendar system you like.
4. We will not use CANVAS calendar, however you can manage that yourself. CANVASS allows you to add events such as assignment deadlines.
5. Piazza is FERPA compliant https://piazza.com/legal/ferpa

2.2.2 Managing Your Own Calendar

From time to time we get the question from a very small number of students why we are not using or uploading the assignment deadlines and the assignment descriptions to CANVAS. The reason for this is manifold. First, our class has different deadlines for different students within the same class. This is not supported by CANVAS and if we would use CANVAS leads to confusion and clearly shows the limitation of CANVAS. Second, we are teaching cloud computing. CANVAS is not a tool that you likely will use after graduating. Thus we are providing you the ability to explore industry standard tools such as github to maintaining your own tasks and deadlines, while for example using github issues (see the section about github). We highly recommend that you explore this as part of this class and you will see that managing the assignments in github is **superior** to CANVAS. Naturally you can not make that assessment if you are not trying it. Thus we like you to do so and it is part of any assignment in your class to use github issues to manage your assignments for this class.

However, if you still want to manage your tasks in CANVAS, you can do so. CANVAS allows you to create custom events, so if you see an assignment in piazza or the handbook, you are more than welcome to add that task yourself to your own CANVAS tasks. As we have only a very small number of assignments this will not pose a problem either for graduate or undergraduate. Being able to organize your deadlines and assignment with industry accepted tools is part of your general learning experience at IU.

Obviously, this makes it also possible to use any other task or calendar system that you may use such as google calendar, jira, microsoft project, and others.

As you can see through this strategy we provide the most flexible system for any student of the class, while giving each student the ability to chose the system they prefer for managing their assignment deadlines. It is obvious that this

strategy is superior to CANVAS as it is much more general.

2.2.3 Online and Office Hours

To support you we have established an open policy of sharing information not only as part of the class material, but also as part of how we conduct support. We establish the following principals:

- in case of doubt how to communicate address this early in class and attend online hours;
- all office hours if not of personal nature are open office hours meaning that any student in class can be joined by other students of the class and all meeting times are posted publicly. This includes in person office hours with TAs. Other students are allowed to listen in and participate.
- it is in your responsibility to attend in person classes and online hours as we found that those that do get better grades. For residential students participation in the residential classes may be mandatory. International students may need to check university policies.
- instructors of this class will attempt within reason to find suitable times for you to attend an online hour in case you are an online student.

2.2.3.1 Office Hour Calendar

Online Students:

- Online hours are prioritized for online students, residential students should attend the residential meetings.

Residential Students:

- Residential students participate in the official meeting times. If additional times are required, they have to be done by appointment. Office hours will be announced publicly. All technical office hours are public and can be attended by any student. Online hours are not an excuse not to come to the residential class.

However Residential students can in addition to the residential class use the online student meeting times. However, in that case online students will be served first. It is probably good to check into the zoom meeting and identify if the TA has time. They will be in zoom.

Meeting times and phone numbers are posted in your piazza in the `Resources` section

2.2.4 Class Material

As the class material will evolve during the semester it is obvious that some content will be improved and material will be added. This benefits everyone. To stay up to date, please, revisit this document on weekly basis. This is practice in any class.

2.2.5 HID

You will be assigned an hid (Homework IDentifier) which allows us to easily communicate with you and does allow us to not use your university ID to communicate with you.

You will receive the HID within the first couple of weeks of the semester by the TA's.

2.2.6 Class Directory

You will get a class directory on github.com and not the iu github. For that reason you will be asked to give us a github id so we can create a openly accessible directory for you in which you can collaborate with the students of this class. The directories are only used to store the artifacts of the class. As all artifacts are supposed to be open source github.com provides us with the service that millions of professionals and researchers use for their work.

2.2.7 Notebook

All students are required to maintain a *class notebook* in github in which they summarize their weekly activities for this course in bullet form. This includes a self maintained list of which lecture material they viewed and what they worked

on in each week of the class.

The notebook is maintained in the class github.com in your hid project folder. It is a file called `notebook.md` that uses markdown as format. Notebooks are expected to be set up as soon as the git repository was created.

You will be responsible to set up and maintain the `notebook.md` and update it accordingly. We suggest that you prepare sections such as: Logistic, Theory, Practice, Writing and put in bullet form what you have done into these sections during the week. We can see from the github logs when you changed the notebook.md file to monitor progress. The management of the notebook will be part of your discussion grade.

The format of the notebook is very simple markdown format and must follow these rules:

- use headings with the # character and have a space after the # Use `# Week X: mm/dd/yyyy - mm/dd/yyyy` as the subject line for each week
- use bullets in each topic.
- Do not refer to section numbers form the ePub in your notebook as they can change. Instead use the section name or headline and possibly a URL. When using URLS in md format they must be enclosed in `<>` or `[text](URL)`

Please examine carefully the sample note book is available at:

- https://github.com/cloudmesh-community/hid-sample/blob/master/notebook.md

The notebook.md is not a blog and should only contain a summary of what you have done.

2.2.8 Blog

You can maintain your own optional blog. However, the blog will not be used for grading. Do not include sensitive information in either the blog or the notebook. A blog is not a replacement for the notebook. If something does not go so well, do not focus on the negative things, but focus on how that experience

can be overcome and how you turn it to a positive experience. Be positive in general.

2.2.9 Waitlist

The waitlist contains students that are unable to enroll in a section of a course. Students choose to add themselves to the waitlist. They are not automatically added, but choose to do so intentionally based on the status of the course. There are two reasons for students to be on the waitlist. The first, and primary, reason is that the class is already at the scheduled, maximum capacity. Since there are no seats available, the student can elect to add themselves to the waitlist. The second reason is that the students' own schedule has a time conflict. This occurs when they are trying to enroll in a class that overlaps with the time of a class they are already enrolled in.

Students are moved from the waitlist to the regular section during a daily batch process, and not in real time. The process is not in realtime because the registrar receives many requests to increase capacity, decrease capacity, and change rooms. If the process were real time there would be a catastrophe of conflicts.

Students are moved from the waitlist in chronological order that they added themselves to the waitlist. If you are still on the waitlist there are no spaces free, the batch process has not run for the day, or the student in question has a schedule conflict.

Faculty are not able to selectively choose students from the waitlist.

How long does the waitlist process stay active?: The automated processing of the waitlist ends on Thursday of the first week of class At this time the waitlist will no longer be processed. As the residential class starts on Friday, this may cause issues. Either talk to the department on Thursday or show up on Friday. Most likely there will be spaces left. Students on the waitlist at that time will remain on the waitlist, but remain there until the student decides to change their registration. Students may not do that, because they get assessed a change schedule fee.

Students tell me they still want to enroll after the first week of classes. How do they do this?

Beginning Monday, after the first week of class students begin to use the eAdd process to do a late addition of the course. The request is routed to the professor of record on an eDoc and the faculty will be notified via email. Faculty can deny or approve based on whatever criteria they wish to apply. If the faculty member approves, the eDoc is electronically forwarded to the Academic Operations office and we will approve the late add **if the room capacity** allows the addition, otherwise we must deny the addition because of fire marshal regulations. Many times, there are seats in a classroom/discussion/lab, but because other students have not *officially* dropped, enrollment is still at capacity.

After everything, a student that was unable to enroll in the class attended all year and completed all course work as if they had enrolled. Can the student get credit and can I give the student a grade?

Yes. There is a provision for a late registration - contact our office if this occurs. Students will be assessed a tuition fee at the time of late or retroactive registration.

2.2.10 Registration

The Executive Associate Dean for Academic Affairs requires starting Spring 2018 that students that are not officially enrolled, can not register at the end of the class if they in-officially took the class. Please make sure that within the first month you have enrolled. If we do not see in CANVAS, you are not in the class. In case you are on a waitlist it is your responsibility to work with the administration after the waitlist is over to be added to the class by getting permission from the School.

2.2.11 Auditing the class

We no longer allow students to audit the class because:

- Seating in the lecture room is limited and we want foster students that enroll full time first.

- The best way to take the class is to conduct a project. As this can not be achieved without taking the class full time and as auditing the class does not provide the full value of the class, e.g. not more than 10% of the class.

Hence, we do not think it is useful to audit the class.

- Accounts and services have to be set up and require considerable resources that are not accessible to students that audit the class.

2.2.12 Resource restrictions

- It is not allowed to use our services we offer as part of the class for profit (e.g. just enrolling in the class to use our clouds).

- In case of abuse of available compute time on our clouds the student is aware that we will terminate the computer account on our clouds and the student may have to conduct the project on a public cloud or his own computer under own cost. There will be no guarantee that cloud services we offer will be available after the semester is over. Projects can be conducted as part of the class that do not require access to the cloud.

2.2.13 Incomplete

Incompletes have to be explicitly requested in piazza through a private mail to *instructors*. All incompletes have to be filed by DATE TO BE ANNOUNCED.

Incomplete's will receive a fractional Grade reduction: A will become A-, A- will become B+, and so forth. There is enough time in the course to complete all assignments without getting an incomplete.

Why do we have such a policy? As we teach state-of-the-art software this software is subject to change, not only within the course, but also after the course. As we may offer some services and only have access to the TA's during the semester it is obvious that we like all class projects and homework assignments to be completed within a semester. Services that were offered during the semester may no longer be available after the semester is over and could adversely effect your planing. It will be in the students responsibility to identify such services and provide alternatives if they become unavailable. We try hard to avoid this but we can not guarantee it.

Furthermore, once an incomplete is requested, you will have 10 month to complete it. We will need 2 month to grade. No grading will be conducted over

breaks including the summer. This may effect those that require student loans. Please plan ahead and avoid an incomplete.

The incomplete request needs to be off the following format in piazza:

```
Subject:
    INCOMPLETE REQUEST: HID000: Lastname, Firstname

Body:
    Firstname: TBD
    Lastname: TBD
    HID: TBD
    Semester: TBD
    Course: TBD
    Online: yes/no

    URL notebook: TBD
    URL assignment1:
    URL assignment2: TBD
    ....
    URL paper1: TBD
    URL project: TBD

    URL other1: TBD
```

Please make sure that the links are clickable in piazza. Also as classes have different assignments, make sure to include whatever is relevant for that class and add the appropriate artifacts.

In case of an incomplete you may be asked to do additional assignments and assignments that have been adapted based on experience from the class. Please note also that we could reject an assignment if it is identified to no longer reflect the state-of-the-art. All previously submitted assignments such as papers, sections, and so on will be reviewed on this criterion. For example, let us assume you developed a tutorial on technology visit version x. Let us assume that since you completed this task a version x+1 comes out. It will be your obligation to update the deliverable. This is also true if the tutorial has been graded previously. The incomplete and the change of the software have at this time negated the originally assigned grade. In most cases the changes may be small. In other cases the changes could be substantial. Hence avoid an incomplete.

Here is the process for how to deal with incompletes at IU are documented:

- http://registrar.indiana.edu/grades/grade-values/grade-of-incomplete.shtml

2.2.13.1 Exercises

E.Policy.1

> *Take the Plagiarism test, See the Scientific Writing I ePub for more details.*

2.3 Course Description

2.3.1 Big Data Applications and Big Data Applications Analytics

- Indiana University
- Fall 2018
- Course Numbers: E534, I523, I423
- Faculty: Dr. Geoffrey C. Fox
- Credits: 3
- Prerequisite(s): Knowledge of a programming language, the ability to pick up other programming languages as needed, willingness to enhance your knowledge from online resources and additional literature. You will need access to a "modern" computer that allows using virtual machines and/or containers. Knowledge of material taught by e516 is desirable and will make project execution easier. e516 and this class can be taken in parallel.
- This page is maintained and updated at e534-i523: Big Data Applications and Big Data Applications Analytics
- Course Description URL: https://github.com/cloudmesh-community/book/blob/master/chapters/class/e534-i523.md
- Registrar information and Other related classes

2.3.2 Course Objectives

This class investigates the use of clouds running data analytics collaboratively for processing Big Data to solve problems in Big Data Applications and Analytics. Case studies such as Netflix recommender systems, Genomic data, Sports, Health, and more will be discussed.

The course has the following objectives:

- Provide an introduction to Big Data
- Provide an introduction to Big Data Analytics
- Provide overviews of different Big Data Application areas
- Explore state-of-the-art big data and cloud technologies and services while

providing a write up about it and exploring it practically with a section that you develop
- Enforce the theoretical knowledge with a project that you conduct in one of the application areas.

2.3.3 Learning Outcomes

- Be able to explain the concepts of the big data paradigm including its paradigm shift, its characteristics, and the advantages. Contrast them with the challenges and disadvantages.
- Be able to identify bigdata applications and analytical methods needed to support real world applications.
- Be able to implement a real world application in big data.
- Be able to conduct sophisticated analysis of big data.
- Be able to communicate the results through sections, chapters, manuals, and reports.
- Be able to work in a team to develop collaboratively software or contribute collaboratively to develop sections using clouds.

2.3.4 Course Syllabus

- Release classes are marked with ✓
- Classes that will be improved are marked with ⊗

	Date[a]	Unit	Title	Description
✓	08/24	1	Fundamentals	Introduction to Big data
✓	08/24	1	Introduction	Intoduction to Clouds ⊗
✓	08/24	1	Introduction	Overview of Big Data ⊗
✓	09/03			Big Data Use Cases
✓	10/15	2	Basic Math	Minimal Statistics
✓	09/17		Applications	Physics and Astronomy

✓	09/24	3		Lifestyle and e-Commerce
✓	10/15			kNN and Clustering
✓	10/15			k-means
✓	10/08			Web Search
✓	10/08			Sports
✓	10/15			Health
✓	10/22			Sensor
✓	10/29			Radar
✗	11/05	4	Basic Cloud	Cloud Computing for Big Data I ✗ (outdated)
✗	11/19			Cloud Computing for Big Data II ✗
✗	12/03	5	Applications	Edge Computing and the Cloud ✗
✓	09/03-11/02*	5	Technology Summaries	Contribute your assigned technology summaries while not plagiarizing.
✓	09/03-11/02*	5	Example	Contribute a significant technical example related to Big Data technologies. Do not develop redundant or duplicated content.
✓	09/03-11/02*	5	Chapter	Contribute a significant chapter while not plagiarizsing that may use your example to the class documentation. Do not develop redundant or duplicated content.
✓	09/03-11/26*	6	Project Type A	Build a Big Data Application Project (maximum grade for class A or A+)

	09/03-11/26*	Project Type B	Write a comprehensive Report without programming (maximum grade for entire class A-)

Students need only to do one project. The project is conducted thought the entire semester.

a. Dates may change as the semester evolves

(*) The project is a long term assignment (and are ideally worked on weekly by residential students). It is the major part of the course grade.

(*) Sections prepare you for documenting a technical aspect related to cloud computing. It is a preparation for a document that explains how to execute your project in a reproducible manner to others.

- all times are in EST

Additional Lectures will be added that allow easy management of the project. These lectures can be taken any time when needed

2.3.5 Assessment

This course is focusing on the principal *Learning by Doing* which is assessed by simple graded and non-graded activities. The assessment may include comprehension of the material taught, programming assignments, participation in online discussion forums, or the contribution of additional material to the class showcasing your comprehension.

The comprehension is also measured by the development of one or more sections in markdown that can be distributed and replicated to other students. This is done in preparation for the project that must include a simple deployment and runtime instruction set.

The main deliverable of the class is a project. The project is assessed through the following artifacts:

- a. Deployment and install instructions,
- b. Project report (typically 2-3 pages per month, sections and chapters can be reused if possible),
- c. Working project code that can be installed and executed in reproducible manner by a third party
- d. Code developed by the project team distributed in github.com
- e. Project progress notes checked into github throughout the semester. Each week the project progress is reported and will be integrated into the final grade.
- f. three discussions or progress reports with the instructors about your project

The grade distribution is as follows

- 10% Comprehension Activities
- 10% Section
- 10% Chapter
- 70% Project

As the project is the main deliverable of the course it is obvious that those starting a week before the deadline will not succeed in this class. The project will take a significant amount of time and fosters the principal of “Learning by Doing” at all stages throughout the semester.

The class will not have a regular midterm, but it is expected that you have worked on your project and can provide a snapshot of the progress outlining the goals of the project and how you will achieve these goals till the end of the semester.

The final Project is due Dec. 1st. Issues with your project ought to have been discussed before this deadline with the TA’s The TAs will in the next 14 days go over the projects and evaluate major and minor issues that you may be able to fix

without penalty. Larger changes will receive a grade penalty. The last fix (upon approval) possible will be Dec 7th.

2.3.5.1 Incomplete

Please see the university regulations for getting an incomplete. However, as this class uses state-of-the-art technology that changes frequently, you must expect that an incomplete may result in significant additional work on your behalf as your project may need significant updates on infrastructure, technology, or even programming models used. It is best to complete the course within one semester.

2.3.5.2 Calendar

Assignment #	**Event**		**Date**
	Full Term		16 Weeks
	Begins		Mon 08/20
1, 2	**Bio, Notebook**	assigned	Mon 08/20
1, 2	**Bio, Notebook**	due	Mon 08/27 9am
3	**Section**	assigned	Mon 09/03
4	**Chapter**	assigned	Mon 09/03
5	**Project**	selection or proposal	Mon 09/03
	Labor Day		Mon 09/03
5	**Project**	update	10/05 9am EST
	Fall Break		10/05 - 10/07
	Auto W		Sun 10/21
5	**Project**	update	11/02 9am EST
3	**Section**	due	11/02 9am EST
4	**Chapter**	due	11/02 9am EST
	Thanksgiving		11/18 - 11/25
5	**Project**	due	11/26 9am EST

5	**Project** (no penalty)	improvements	11/26 - 12/03 9am EST
5	**Project** (with penalty)	improvements	12/04 - 12/07 9am EST
	Final Deliverables due		12/07 9am EST
	Grading		12/01 - 12/14
	Ends		Fri 12/14
	grade submission to school		Fri 12/14
	grade posting by registrar		12/31

- TA's must be available till all grades have been submitted.
- Bio: a formal 3 paragraph Bio
- Notebook: a markdown in which you record your progress of this class in bullet form
- All times are in EST
- Dependent on class progress Comprehension Assignments may be added

2.4 Example Artifacts

Learning Objectives

- Identify what other students have done previously.
- Look at previous chapters, which are collected as technology reviews.
- Look at previous project reports.
- Looking at the documents provides you with an initial overview of the scope for the artifacts.

As part of this class you will be delivering some artifacts that are being graded. Some of them include writing a *chapter* that can be contributed to the class lecture and a project. To showcase you some example artifacts take a closer look

at the documents listed in this section. Please also note that you can not duplicate or replicate a students previous work without significant improvements. All material listed here is available online, including all source code.

2.4.1 Technology Summaries

We are maintaining a large list of technologies related to clouds and Big Data at

- https://github.com/cloudmesh/technologies

This repository generates the following epub

- https://github.com/cloudmesh/technologies/blob/master/vonLaszewski-cloud-technologies.epub

Students that have to contribute as part of their class Technology summaries are asked to produce meaningful, advertisement free summaries of the technology and indicate in some cases if not obvious show they relate to cloud or big data. The length of such summaries is about 300 words. Students of E516 do not have to contribute to this and will instead focus on programming. Students of I423, I523 and other sections must contribute to it and will get an assignment related to it. We post here the existence of this document also for 516 students. They can voluntarily improve or add sections if they like which will go into their discussion credit.

Please use the following indicators to mark the progress of summaries that you are working on.

 ready for review

 selected by student so others do not select it and we know what is worked on

 needs revision (only assigned by ta, after smiley)

The signs are put as follows. You can view an example at https://github.com/cloudmesh/technologies/blob/master/chapters/tech/bioconduct

Ex - Title Of Summary  fa18-xxx-xx

2.4.2 Chapters

Previously we asked students to write a 2 page paper on a topic related to bigdata analytics or cloud technologies (dependent on the course). Example papers are listed bellow

- Use Cases in Big Data Software and Analytics Vol. 1, Gregor von Laszewski, Fall 2017, http://cyberaide.org/papers/vonLaszewski-i523-v1.pdf
- Use Cases in Big Data Software and Analytics Vol. 2, Gregor von Laszewski, Fall 2017, http://cyberaide.org/papers/vonLaszewski-i523-v2.pdf
- Big Data Software Vol 1., Gregor von Laszewski, Spring 2017, https://github.com/cloudmesh/sp17-i524/blob/master/paper1/proceedings.pdf
- Big Data Software Vol 2., Gregor von Laszewski, Spring 2017, https://github.com/cloudmesh/sp17-i524/blob/master/paper2/proceedings.pdf
- Vol 8, Gregor von Laszewski, Spring 2018, http://cyberaide.org/papers/vonLaszewski-cloud-vol-8.pdf

This has however resulted in a large number of duplicated material especially in the introductions and motivations. Thus we like this year to have you more focused on the topic and do not write a large introduction on what big data or cloud computing is. Therefore we renamed the 2 paper to a chapter, while you could assume certain things that have already been taught to you and you do not have to repeat it.

2.4.3 Project Reports

The goal of the class is to use open source technology to also write your

technical reports. As a beneficial side product, we are able to distribute all previous reports from students to you. In your reports you will be doing a similar report, but will not use the same topic, without a significant improvement from a report already delivered in that area. For big data we have more than 1000 data sets we point to. I am sure you can do a unique project. For engineering cloud there are recently so many new technologies that there is not much chance of an overlap. TA's will review your project proposal, but it is your responsibility to make sure they are unique.

Please note that we do not make any quality assumptions to the published papers. It is up to you to identify outstanding papers.

- Use Cases in Big Data Software and Analytics Vol. 3, Gregor von Laszewski, Fall 2017, http://cyberaide.org/papers/vonLaszewski-i523-v3.pdf
- Big Data Projects, Gregor von Laszewski, Spring 2017, https://github.com/cloudmesh/sp17-i524/blob/master/project/projects.pdf
- Vol 9, Gregor von Laszewski, Spring 2018, http://cyberaide.org/papers/vonLaszewski-cloud-vol-9.pdf

2.5 DATASETS

Given next 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

For NIST Projects:

- NIST Special Database 27A - 4GB
- INRIA Person Dataset
- Healthcare data from CMS
- Uber Ride Sharing GPS Data
- Census Data

2.6 Assignments

2.6.1 Due dates

For due dates see the calendar section.

2.6.2 Terminology

Dependent on the class you need to do different assignments. The assignments will be clearly posted. In case of questions, we will update this document to provide clarifications if needed. We use the following terminology:

License:

> All projects are developed under an open source license such as Apache 2.0 License. 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.md file which packages are used and which license these packages have.

Sections:

> Sections are written in markdown and include information on a particular technical issue that is in general helpful for other students. Sections must be about a substantial topic and include an introduction a section that teaches a reader a significant issue, as well as practical code examples. Multiple

small sections can lead to a substantial contribution. We expect that the sections are of high quality and can be included in our handbooks. Please be careful of plagiarism and do not just copy the sections from tutorials or code or from elsewhere.

Technology or Review Paper :

A technology paper is a summary paper about a technology, application, or topic that is not yet covered in other technology papers delivered by previous students of this class. A review paper is a paper that reviews a specific topic related to this class.

In either case includes useful information that provides an overview of what you are trying to describe and analyses its relationship to the class topic. Be mindful about plagiarism. The paper is written in LaTeX or Markdown and uses bibtex for bibliography management. It uses the same format as your report paper. The format is discussed in the Section Report Format.

A technology paper is 2 pages long. This will make it between 2000-2400 words.

Note: that for the 2018 we decided to just us Markdown and not LaTeX. We will calculate the exact number of words needed.

Project:

We refer with the term project to the major activity that you chose as part of your class. The default case is an implementation project that requires a *project report* and project code. In case you have issues with code development you can also chose a *term paper* as project.

Term Paper:

A term paper is an enhanced topic paper (only available for I523). The difference is in length and depth of coverage. Comparative or review papers can also become term papers. In case you chose the term paper, you or your team will pick a topic relevant for the class. Term papers should have the quality to be publishable either in a workshop or as part of the handbook. Not all classes allow you to do a term paper, but require you to do a project.

Please confirm with your class. For the classes listed here the term paper wil result in a quarter reduction in grade for the entire class not just the paper. Remember tables and figures do not count towards the paper length. A term paper has the following length.

- 8 pages, one student in the project
- 10 pages, two student in the project
- 12 pages, three student in the project

We estimate that a single page is between 1000-1200 words. Please note that for 2018 the format will be markdown, so the word count will be used instead.

Project Report:

A project report is an enhanced topic paper that includes not just the analysis of a topic, but an actual code, with **benchmark** and demonstrated application use. Obviously it is longer than a term paper and includes descriptions about reproducibility of the application. A README is provided that describes in a section how others can reproduce your project and run it. Term papers should have the quality to be publishable either in a workshop or as part of the handbook. The format is discussed in the Section Report Format. Remember tables and figures do not count towards the paper length. The following length is required:

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

We estimate that a single page is between 1000-1200 words. Please note that for 2018 the format will be markdown, so the word count will be used instead.

Project Code:

This is the **documented** and **reproducible** code and scripts that allows a TA do replicate the project. In case you use images they must be created from scratch locally and may not be uploaded to services such as dockerhub. You can however reuse vendor uploaded images such as from ubuntu or centos. All code, scripts, and documentation must be uploaded to

github.com under the class specific github directory.

Data:

Data is to be hosted on IUs google drive if needed. If you have larger data, it should be downloaded from the internet. It is in your responsibility to develop a download program. The data **must** not be stored in github. You will be expected to write a python program that downloads the data.

Work Breakdown:

This is an appendix to the document that describes in detail who did what in the project. This section comes in a new page after the references. It does not count towards the page length of the document. It also includes explicit URLs to the git history that documents the statistics to demonstrate not only one student has worked on the project. If you can not provide such a statistic or all check-ins have been made by a single student, the project has shown that they have not properly used git. Thus points will be deducted from the project. Furthermore, if we detect that a student has not contributed to a project we may invite the student to give a detailed presentation of the project.

Bibliography:

All bibliography has to be provided in a jabref/bibtex file. This is regardless if you use LaTeX or Word. There is **NO EXCEPTION** to this rule. Please be advised doing references right takes some time so you want to do this early. Please note that exports of Endnote or other bibliography management tools do not lead to properly formatted bibtex files, despite they claiming to do so. You will have to clean them up and we recommend to do it the other way around. Manage your bibliography with jabref, and if you like to use it import them to endnote or other tools. Naturally you may have to do some cleanup to. If you use LaTeX and jabref, you have naturally much less work to do. What you chose is up to you.

2.6.2.1 Project Deliverables

The objective of the project is to define a clear problem statement and create a

framework to address that problem as it relates to big data your project must address the reproducibility of the deployment and the application. A dataset must be chosen and you can analyze the data. YOu must make sure your project can be deployed on the TAs computer through scripts that make your project reproducible.

You have plenty of time to make this choice and if you find you struggle with programming you may want to consider a term paper instead of a project.

In case you chose a project your maximum grade for the entire class 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 or workshop/

In case you chose a term paper your maximum grade for the *entire* class will be an A-.

2.6.2.1.0.1 Deliverables

- Find a data set with reasonable size (this may depend on your resources and needs to include a benchmark in your paper for justification).
- Clean up the data set or make it smaller or find a bigger data set
- Identify existing algorithms and tools and technologies that you can use to analyze your data
- Provide benchmarks.
- Take results in two different cloud services and your local PC (ex: Chameleon Cloud, echo kubernetes). Make sure your system can be created and deployed based on your documentation.
- Create a Makefile with the tags deploy, run, kill, view, clean that deploys your environment, runs application, kills it, views the result and cleans up after wards. You are allowed to have different makefiles for the different clouds and different directories. Keep the code and directory structure clean and document how to reproduce your results.

- For python use a requirements.txt file also
- For docker use a Dockerfile also
- Write a report that includes the following sections
    - Abstract
    - Introduction
    - Architecture
    - Implementation
        - Technologies Used
    - Design
    - Implementation
    - Results
        - Deployment Benchmarks
        - Application Benchmarks
    - (Limitations)
    - Conclusion
    - (Work Breakdown)
- Your paper will not have a future work section as this implies that you will do work in future, instead you can use an optional limitations section.

2.6.2.2 Group work

You are allowed to work on any assignment in class in groups up to 3 team members. We will not allow more team members as previous examples showed that more team members do not result in better projects than thise delivered with 3 team members.

The assignment is only to be added into github by one team member. Please make sure that you do pull requests to the repository of that team member. If your team likes direct access to the repo form the lead, please communicate this in a private post to piazza with the github user names and we will add the team members. The lea should be aware that in this case all team members have access to all files from the team leader, not only that assignment. If the team leader does not like this, the team should stick with pull requests that the team lead coordinates and integrates.

Groups are expected to have significantly better artifacts than a single student. It is not the goal of the group to deliver a project or paper that is done by n people but could have been done by a single person. Therefore the requirements of all group projects are increased accordingly. Typically this is not an issue. Make sure all team members contribute.

Group requirements Technology summaries:

> If you have n team members together you need to have at least 4 * n technologies. The technologies that have been assigned to each team member will have to be completed. YOu can work in collaboration on the technologies, but you need to place all hids that worked on the technology in the headline.

Group requirements 2-page Technology or Review paper:

> Option multiple papers. If you have n team members you need to write n different papers each of which has 2 pages. The n team members can be authoring the n papers jointly. Put your hids and names in the paper
>
> Option one large paper. If you have n team members you need to write one large paper with 2 * n pages. The paper must be well written and integrated and not just the concatenation of 2 pages from each author.

Group requirements project paper:

> The requirements are clearly stated in another section of the ePub.

3 DETAILS

3.1 Introduction to Big Data Applications

This is an overview course of Big Data Applications covering a broad range of problems and solutions. It covers cloud computing technologies and includes a project. Also, algorithms are introduced and illustrated.

3.1.1 General Remarks Including Hype cycles

This is Part 1 of the introduction. We start with some general remarks and take a closer look at the emerging technology hype cycles.

1.a Gartner's Hypecycles and especially those for emerging technologies between 2016 and 2018

- .

1.b Gartner's Hypecycles with Emerging technologies hypecycles and the priority matrix at selected times 2008-2015

- .

1.a + 1.b:

-

- Technology trends
- Industry reports

3.1.2 Data Deluge

This is Part 2 of the introduction.

2.a Business usage patterns from NIST

- 

2.b Cyberinfrastructure and AI

- 

2.a + 2.b

- 

- Several examples of rapid data and information growth in different areas
- Value of data and analytics

3.1.3 Jobs

This is Part 3 of the introduction.

- 

- Jobs opportunities in the areas: data science, clouds and computer science and computer engineering
- Jobs demands in different countries and companies.
- Trends and forecast of jobs demands in the future.

3.1.4 Industry Trends

This is Part 4 of the introduction.

4a. Industry Trends: Technology Trends by 2014

- 

4b. Industry Trends: 2015 onwards

- 

An older set of trend slides is available from:

4a. Industry Trends: Technology Trends by 2014

- 

A current set is available at:

4b. Industry Trends: 2015 onwards

-  

4c. Industry Trends: Voice and HCI, cars,Deep learning

- 

- Many technology trends through end of 2014 and 2015 onwards, examples in different fields
- Voice and HCI, Cars Evolving and Deep learning

3.1.5 Digital Disruption and Transformation

This is Part 5 of the introduction.

5. Digital Disruption and Transformation

-   

- The past displaced by digital disruption

3.1.6 Computing Model

This is Part 6 of the introduction.

6a. Computing Model: earlier discussion by 2014:

- .

6b. Computing Model: developments after 2014 including Blockchain:

- .

- Industry adopted clouds which are attractive for data analytics, including big companies, examples are Google, Amazon, Microsoft and so on.
- Some examples of development: AWS quarterly revenue, critical capabilities public cloud infrastructure as a service.

- Blockchain: ledgers redone, blockchain consortia.

3.1.7 Research Model

This is Part 7 of the introduction.

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

- .

- The 4 paradigm of scientific research: Theory,Experiment and observation,Simulation of theory or model,Data-driven.

3.1.8 Data Science Pipeline

This is Part 8 of the introduction. 8. Data Science Pipeline

- .

- DIKW process:Data, Information, Knowledge, Wisdom and Decision.
- Example of Google Maps/navigation.

- Criteria for Data Science platform.

3.1.9 Physics as an Application Example

This is Part 9 of the introduction.

- 9. Physics as an Application Example
- Physics as an application example.

3.1.10 Technology Example

This is Part 10 of the introduction.

- 10. Technology Example: Recommender Systems I
- Overview of many informatics areas, recommender systems in detail.
- NETFLIX on personalization, recommendation, datascience.

3.1.11 Exploring Data Bags and Spaces

This is Part 11 of the introduction.

11. Exploring data bags and spaces: Recommender Systems II

- .
- Distances in funny spaces, about “real” spaces and how to use distances.

3.1.12 Another Example: Web Search Information Retrieval

This is Part 12 of the introduction. 12. Another Example: Web Search Information Retrieval

- .

3.1.13 Cloud Application in Research

This is Part 13 of the introduction discussing cloud applications in research.

13. Cloud Applications in Research: Science Clouds and Internet of Things

- 

3.1.14 Software Ecosystems: Parallel Computing and MapReduce

This is Part 14 of the introduction discussing the software ecosystem

14. Software Ecosystems: Parallel Computing and MapReduce

- 

3.1.15 Conclusions

This is Part 15 of the introduction with some concluding remarks. 15. Conclusions

- 

3.2 Overview of Data Science

What is Big Data, Data Analytics and X-Informatics?

We start 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.

3.2.1 Data Science generics and Commercial Data Deluge

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.

 Commercial Data Deluge (45)

3.2.1.1 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 discuss how clouds running Data Analytics Collaboratively processing Big Data can solve problems in X-Informatics. We list many values of X you can defined in various activities across the world.

-  X Informatics (9:49)

3.2.1.2 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.

-  Jobs (2:58)

3.2.1.3 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.

-  Data Deluge (13:04)

3.2.1.4 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.

-  Data Science Process (4:27)

3.2.1.5 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.

- Internet (3:42)

3.2.1.6 Data Deluge: Business

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.

- Business I (6:00)
- Business II (7:34)
- Business III (9:37)

3.2.1.7 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
- Anjul Bhambhri
- Jeff Hammerbacher
- http://www.economist.com/node/15579717
- http://cs.metrostate.edu/~sbd/slides/Sun.pdf
- http://jess3.com/geosocial-universe-2/
- Bill Ruh
- http://www.hsph.harvard.edu/ncb2011/files/ncb2011-z03-rodriguez.pptx
- Hugh Williams

3.2.2 Data Deluge and Scientific Applications and Methodology

3.2.2.1 Overview of Data Science

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.

 Methodology (22)

3.2.2.2 Science and Research

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.

-  Science and Research (11:27)
-  Science and Research (11:49)

3.2.2.3 Implications for Scientific Method

We discuss the emergencies 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.

-  Scientific Methods (5:07)

3.2.2.4 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 do not generate a lot of data each but together they add up to Big data.

-  Long Tail of Science (2:10)

3.2.2.5 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.

-  Internet of Things (5:45)

3.2.2.6 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_Com
- http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%
- 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

3.2.3 Clouds and Big Data Processing; Data Science Process and Analytics

3.2.3.1 Overview of Data Science

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.

 Clouds (35)

3.2.3.2 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.

-  Clouds (16:04){MP4}

3.2.3.3 Aspect of Data Deluge

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

-  Data Deluge (8:02)
-  Data Deluge (6:24)

3.2.3.4 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.

- Scientific Process (11:28)

3.2.3.5 Data Analytics

Data Analytics (30) We stress the importance of data analytics givi ng examples from several fields. We note that better analytics is as important as better computing and storage capability. In the second video we look at High Performance Computing in Science and Engineering: the Tree and the Fruit.

- Data Analytics (7:28)
- Data Analytics (6:51)

3.2.3.6 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 Wolski http://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/koomeydatacenterelectuse2
- Bina Ramamurthy
- Jeff Hammerbacher
- Jeff Hammerbacher

- Anjul Bhambhri
- http://cs.metrostate.edu/~sbd/slides/Sun.pdf
- Hugh Williams
- Tom Davenport
- http://www.mckinsey.com/mgi/publications/big_data/index.asp
- http://cra.org/ccc/docs/nitrdsymposium/pdfs/keyes.pdf

3.3 PHYSICS

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.

3.3.1 Looking for Higgs Particles

3.3.1.1 Bumps in Histograms, Experiments and Accelerators

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.

-  Higgs (20)
- <{gitcode}/physics/mr-higgs/higgs-classI-sloping.py>

3.3.1.2 Particle Counting

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.

-  Discovery of Higgs Particle (13:49)

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.

-  Looking for Higgs Particle and Counting Introduction II (7:38)

3.3.1.3 Experimental Facilities

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.

-  Looking for Higgs Particle Experiments (9:29)

3.3.1.4 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.

-  Accelerator Picture Gallery of Big Science (11:21)

3.3.1.5 Resources

- http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%
  [@fox2011does]
- http://www.sciencedirect.com/science/article/pii/S037026931200857X

[@aad2012observation]
- http://www.nature.com/news/specials/lhc/interactive.html

Looking for Higgs Particles: Python Event Counting for Signal and Background (Part 2)

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.

-  Higgs II (29)

Files:

- <{gitcode}/physics/mr-higgs/higgs-classI-sloping.py>
- <{gitcode}/physics/number-theory/higgs-classIII.py>
- <{gitcode}/physics/mr-higgs/higgs-classII-uniform.py>

3.3.1.6 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).

-  Event Counting (7:02)

3.3.1.7 Monte Carlo

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.

-  With Python examples of Signal plus Background (7:33) 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.

-  Change shape of background & num of Higgs Particles (7:01)

3.3.1.8 Resources

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

3.3.1.9 Random Variables, Physics and Normal Distributions

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.

-  Higgs (39)
- <{gitcode}/physics/number-theory/higgs-classIII.py>

3.3.1.10 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.

-  Random variables and normal distributions (8:19)

3.3.1.11 Physics and Random Variables

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.

- Physics and Random Variables I (8:34)
- Physics and Random Variables II (5:50)

3.3.1.12 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.

- Statistics of Events with Normal Distributions (11:25)

3.3.1.13 Gaussian Distributions

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

- Gaussian Distributions (9:08)

3.3.1.14 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.

- Using Statistics (14:02)

3.3.1.15 Resources

- http://indico.cern.ch/event/20453/session/6/contribution/15?

materialId=slides
- ~~http://www.atlas.ch/photos/events.html~~ (this link is outdated)
- https://cms.cern/ [@cms]

3.3.1.16 Random Numbers, Distributions and Central Limit Theorem

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.

-  Higgs III (44)

Files:

- <{gitcode}/physics/calculated-dice-roll/higgs-classIV-seeds.py>

3.3.1.16.1 Generators and Seeds

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.

-  Higgs Particle Counting Errors (6:28)
-  Generators and Seeds II (7:10)

3.3.1.16.2 Binomial Distribution

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

-  Binomial Distribution: (12:38)

3.3.1.16.3 Accept-Reject

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

-  Accept-Reject (5:54)

3.3.1.16.4 Monte Carlo Method

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

-  Monte Carlo Method (2:23)

3.3.1.16.5 Poisson Distribution

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

-  Poisson Distribution (4:37)

3.3.1.16.6 Central Limit Theorem

We introduce Central Limit Theorem and give examples from Wikipedia.

-  Central Limit Theorem (4:47)

3.3.1.16.7 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.

-  Interpretation of Probability (12:39)

3.3.1.16.8 Resources

3.3.2 SKA – Square Kilometer Array

Professor Diamond, accompanied by Dr. Rosie Bolton from the SKA Regional Centre Project gave a presentation at SC17 “into the deepest reaches of the observable universe as they describe the SKA’s international partnership that will map and study the entire sky in greater detail than ever before.”

- http://sc17.supercomputing.org/presentation/?id=inspkr101&sess=sess263

A summary article about this effort is available at:

- https://www.hpcwire.com/2017/11/17/sc17-keynote-hpc-powers-ska-efforts-peer-deep-cosmos/ The video is hosted at
- http://sc17.supercomputing.org/presentation/?id=inspkr101&sess=sess263 Start at about 1:03:00 (e.g. the one hour mark)

3.4 E-COMMERCE AND LIFESTYLE

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.

3.4.1 Recommender Systems

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.

 Lifestyle Recommender (45)

3.4.1.1 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”.

 Recommender Systems I (8:06)

3.4.1.2 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 are not 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.

 Recommender Systems Introduction (12:56)

3.4.1.3 Kaggle Competitions

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

 Kaggle Competitions: (3:36)

> ⚠ *Please not that we typically do not accept any projects using kaggle data for this classes. This class is not about winning a kaggle competition and if done wrong it does not fullfill the minimum requiremnt for this class. Please consult with the instructor.*

3.4.1.4 Examples of Recommender Systems

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

 Examples of Recommender Systems (1:00)

3.4.1.5 Netflix on Recommender Systems

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.

 Netflix on Recommender Systems (14:20)

Next 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.

 Consumer Data Science (13:04)

3.4.1.6 Other Examples of Recommender Systems

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.

 Lifestyle Recommender (49)

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

 Recap and Examples of Recommender Systems (5:48)

3.4.1.6.1 Examples of Recommender Systems

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

 Examples of Recommender Systems (8:34)

3.4.1.6.2 Recommender Systems in Yahoo Use Case Example

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.

 Recap of Recommender Systems II (8:46)

 Recap of Recommender Systems III (10:48)

 Case Study of Recommender systems (3:21)

3.4.1.6.3 User-based nearest-neighbor collaborative filtering

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.

 User-based nearest-neighbor collaborative filtering I (7:20)

 User-based nearest-neighbor collaborative filtering II (7:29)

3.4.1.6.4 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.

 Vector Space Formulation of Recommender Systems new (9:06)

3.4.1.7 Resources

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

3.4.2 Item-based Collaborative Filtering and its Technologies

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.

 Lifestyle Filtering (18)

3.4.2.1 Item-based Collaborative Filtering

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.

 Item Based Filtering (11:18)

 k Nearest Neighbors and High Dimensional Spaces (7:16)

3.4.2.2 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.

 k Nearest Neighbors and High Dimensional Spaces (10:03)

3.4.2.2.1 Recommender Systems - K-Neighbors

Next we provide some sample Python code for the k Nearest Neighbor 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. This lesson is adapted from 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.

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.

3.4.2.2.2 Plotviz

The 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.

3.4.2.2.3 Files

- https://github.com/cloudmesh-community/book/blob/master/examples/python/knn/kNN.py
- https://github.com/cloudmesh-community/book/blob/master/examples/python/knn/kNN_Driver.py
- https://github.com/cloudmesh-community/book/blob/master/examples/python/knn/dating_test_set2.txt
- https://github.com/cloudmesh-community/book/blob/master/examples/python/knn/clusterFinal-M3-C3Dating-ReClustered.pviz
- https://github.com/cloudmesh-community/book/blob/master/examples/python/knn/dating_rating_original_
- https://github.com/cloudmesh-community/book/blob/master/examples/python/knn/clusterFinal-M30-C28.pviz
- https://github.com/cloudmesh-community/book/blob/master/examples/python/plotviz/clusterfinal_m3_c3d
- https://github.com/cloudmesh-community/book/blob/master/examples/python/plotviz/fungi_lsu_3_15_to_3

3.4.2.3 Resources k-means

- http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-tutorial [@www-slideshare-building]

- http://www.ifi.uzh.ch/ce/teaching/spring2012/16-Recommender-Systems_Slides.pdf [@www-ifi-teaching]
- https://www.kaggle.com/ [@www-kaggle]
- http://www.ics.uci.edu/~welling/teaching/CS77Bwinter12/CS77B_w12.htm [@www-ics-uci-welling]
- Jeff Hammerbacher[@20120117berkeley1]
- http://www.techworld.com/news/apps/netflix-foretells-house-of-cards-success-with-cassandra-big-data-engine-3437514/ [@www-techworld-netflix]
- https://en.wikipedia.org/wiki/A/B_testing [@wikipedia-ABtesting]
- http://www.infoq.com/presentations/Netflix-Architecture [@www-infoq-architec]

3.5 Sports

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.

3.5.1 Basic Sabermetrics

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.

Overview (40)

3.5.1.1 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.

 Introduction and Sabermetrics (Baseball Informatics) Lesson (31:4)

3.5.1.2 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.

 Basic Sabermetrics (26:53)

3.5.1.3 Wins Above Replacement

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

 Wins Above Replacement (30:43)

3.5.2 Advanced Sabermetrics

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

 Sporta II (41)

3.5.2.1 Pitching Clustering

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

Pitching Clustering (20:59)

3.5.2.2 Pitcher Quality

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

Pitcher Quality (10:02)

3.5.3 PITCHf/X

Examples of use of PITCHf/X.

PITCHf/X (10:39)

3.5.3.1 Other Video Data Gathering in Baseball

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

Other Video Data Gathering in Baseball (18:5) Other Sports

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.


Sport Sports III (44)

3.5.3.2 Wearables

Consumer Sports, Stake Holders, and Multiple Factors.

Wearables (22:2)

3.5.3.3 Soccer and the Olympics

Soccer, Tracking Players and Balls, Olympics.

 Soccer and the Olympics (8:28)

3.5.3.4 Spatial Visualization in NFL and NBA

NFL, NBA, and Spatial Visualization.

 Spatial Visualization in NFL and NBA (15:19)

3.5.3.5 Tennis and Horse Racing

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

 Tennis and Horse Racing (8:52)

3.5.3.6 Resources

- http://www.slideshare.net/Tricon_Infotech/big-data-for-big-sports [@www-slideshare-tricon-infotech]
- http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-powered-storytelling [@www-slideshare-sports]
- http://www.slideshare.net/elew/sport-analytics-innovation [@www-slideshare-elew-sport-analytics]
- http://www.wired.com/2013/02/catapault-smartball/ [@www-wired-smartball]
- http://www.sloansportsconference.com/wp-content/uploads/2014/06/Automated_Playbook_Generation.pdf [@www-sloansportsconference-automated-playbook]
- http://autoscout.adsc.illinois.edu/publications/football-trajectory-dataset/ [@www-autoscout-illinois-football-trajectory]
- http://www.sloansportsconference.com/wp-content/uploads/2012/02/Goldsberry_Sloan_Submission.pdf [@sloansportconference-goldsberry]
- http://gamesetmap.com/ [@gamesetmap]

- http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-powered-storytelling [@www-slideshare-sports-datapowered]
- http://www.sloansportsconference.com/ [@www-sloansportsconferences]
- http://sabr.org/ [@www-sabr]
- http://en.wikipedia.org/wiki/Sabermetrics [@wikipedia-Sabermetrics]
- http://en.wikipedia.org/wiki/Baseball_statistics [@www-wikipedia-baseball-statistics]
- http://m.mlb.com/news/article/68514514/mlbam-introduces-new-way-to-analyze-every-play [@www-mlb-mlbam-new-way-play]
- http://www.fangraphs.com/library/offense/offensive-statistics-list/ [@www-fangraphs-offensive-statistics]
- http://en.wikipedia.org/wiki/Component_ERA [@www-wiki-component-era]
- http://www.fangraphs.com/library/pitching/fip/ [@www-fangraphs-pitching-fip]
- http://en.wikipedia.org/wiki/Wins_Above_Replacement [@www-wiki-wins-above-replacement]
- http://www.fangraphs.com/library/misc/war/ [@www-fangraphs-library-war]
- http://www.baseball-reference.com/about/war_explained.shtml [@www-baseball-references-war-explained]
- http://www.baseball-reference.com/about/war_explained_comparison.shtml [@www-baseball-references-war-explained-comparison]
- http://www.baseball-reference.com/about/war_explained_position.shtml [@www-baseball-reference-war-explained-position]
- http://www.baseball-reference.com/about/war_explained_pitch.shtml [@www-baseball-reference-war-explained-pitch]
- http://www.fangraphs.com/leaders.aspx?pos=all&stats=bat&lg=all&qual=y&type=8&season=2014&month=0&seas [@www-fangraphs-leaders-pose-qual]
- http://battingleadoff.com/2014/01/08/comparing-the-three-war-measures-part-ii/ [@battingleadoff-baseball-player]
- http://en.wikipedia.org/wiki/Coefficient_of_determination [@www-wiki-coefficient-of-determination]
- http://www.sloansportsconference.com/wp-content/uploads/2014/02/2014_SSAC_Data-driven-Method-for-In-game-Decision-Making.pdf [@ganeshapillai2014data]

- https://courses.edx.org/courses/BUx/SABR101x/2T2014/courseware/10e61(
- http://vincegennaro.mlblogs.com/ [@www-vincegennaro-mlblogs]
- https://www.youtube.com/watch?v=H-kx-x_d0Mk [@www-youtube-watch]
- http://www.baseballprospectus.com/article.php?articleid=13109 [@www-baseball-prospectus-spinning-yarn]
- http://baseball.physics.illinois.edu/FastPFXGuide.pdf [@baseball-physics-PITCHf]
- http://baseball.physics.illinois.edu/FieldFX-TDR-GregR.pdf [@baseball-physics-fieldfx]
- http://regressing.deadspin.com/mlb-announces-revolutionary-new-fielding-tracking-syste-1534200504 [@www-deadspin-field-tracking-syste]
- http://grantland.com/the-triangle/mlb-advanced-media-play-tracking-bob-bowman-interview/ [@grantland-mlb-bob-bowman]

- https://www.youtube.com/watch?v=YkjtnuNmK74 [@www-youtube-science-home-run]

These resources do not exsit:

- ~~http://www.sportvision.com/baseball~~
- ~~http://www.sportvision.com/media/pitchfx-how-it-works~~
- ~~http://www.sportvision.com/baseball/fieldfx~~
- ~~http://www.sportvision.com/baseball/hitfx~~
- ~~http://www.trakus.com/technology.asp#tNetText~~
- ~~http://www.sloansportsconference.com/?page_id=481&sort_cate=Research%20Paper~~

- ~~http://www.liveathos.com/apparel/app~~

3.6 Cloud Computing

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.

3.6.1 Parallel Computing (Outdated)

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.

 Parallel Computing (33)

3.6.1.1 Decomposition

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.

-  Decomposition (8:51)
-  Examples of Application (13:22)
-  Decomposition Strategies (9:22)

3.6.1.2 Parallel Computing in Society

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.

-  Parallel Computing in Society I (8:24)
-  Parallel Computing in Society II (8:01)

3.6.1.3 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.

- Processing for Hadrian's Wall (9:24)

3.6.1.4 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).
- 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.

3.6.2 Introduction

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.

 Introduction (45)

3.6.2.1 Cyberinfrastructure for E-Applications

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.

- Cloud Computing Introduction Part1 (13:34)

3.6.2.2 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.

- What is Cloud Computing Intro (12:01)

3.6.2.3 What and Why is Cloud Computing: Other Views I

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

- Other Views I (5:25)
- Other Views II (6:41)
- Other Views III (7:27)

3.6.2.4 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.

-  Gartners Emerging Technology Landscape (11:26)

3.6.2.5 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.

-  Examples (3:26)

3.6.2.6 Value of Cloud Computing

Some comments on fiscal value of cloud computing.

-  Value of Cloud Computing (4:20)

3.6.2.7 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/green-computing.pdf

3.6.3 Software and Systems

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.

 Software and Systems (32)

3.6.3.1 What is Cloud Computing

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

-  What is Cloud Computing (6:20)

3.6.3.2 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. 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.

-  Intro to IaaS and PaaS I (7:42)
-  Intro to IaaS and PaaS II (6:42)

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 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.

-  Software Architecture: (7:42)
-  IaaS and PaaS II: (6:43)

3.6.3.3 Using the HPC-ABDS Software Stack

Using the HPC-ABDS Software Stack.

-  ABDS (27:50)

3.6.3.4 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_OpportunitiesAn
- http://cloudonomic.blogspot.com/2009/02/cloud-taxonomy-and-ontology.html

3.6.4 Architectures, Applications and Systems

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.

scroll: Architectures (64)

3.6.4.1 Cloud (Data Center) Architectures

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.

- Coud Architecture (8:38)
- Cloud Data Center Architecture (9:59)

3.6.4.2 Analysis of Major Cloud Providers

Gartner 2014 Analysis of leading cloud providers.

- Analysis of Major Cloud Providers (21:40)

3.6.4.3 Commercial Cloud Storage Trends

Use of Dropbox, iCloud, Box etc.

- Commercial Storage Trends (3:07)

3.6.4.4 Cloud Applications I

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.

- Cloud Applications I (7:57)
- Cloud Applications II (7:44)

3.6.4.5 Science Clouds

Science Applications and Internet of Things.

-  Science Clouds (19:26)

3.6.4.6 Security

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

-  Security (2:34)

3.6.4.7 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.

-  Comments on Fault Tolerance and Synchronicity Constraints (8:55)

3.6.4.8 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_OpportunitiesAn
- 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/koomeydatacenterelectuse2
- 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
- https://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.
- 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
- Conquering Big Data with the Oracle Information Model, Helen Sun, Oracle
- Hugh Williams VP Experience, Search & Platforms, eBay
- Dennis Gannon, Scientific Computing Environments
- http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_OpportunitiesAn
- 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/koomeydatacenterelectuse2
- 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

3.6.5 Data Systems

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.

 Data Systems (49)

3.6.5.1 The 10 Interaction scenarios (access patterns) I

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

-  The 10 Interaction scenarios I (10:26)

3.6.5.2 The 10 Interaction scenarios. Science Examples

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

-  The 10 Interaction scenarios. Science Examples (16:34)

3.6.5.3 Remaining general access patterns

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

 Access Patterns (11:36)

3.6.5.4 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.

 Data in the Cloud (10:24)

3.6.5.5 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.

 Processing Big Data (8:45)

3.6.6 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
- http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html

3.7 Big Data Use Cases Survey

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.

3.7.1 NIST Big Data Public Working Group

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.

 Overview (45)

3.7.1.1 Introduction to NIST Big Data Public Working

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.

 Introduction (13:02)

3.7.1.2 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.

 Taxonomies (7:42)

3.7.1.3 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.

 Architecture (10:05)

3.7.1.4 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.

 Security (9:51)

3.7.1.5 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.

 Technology (4:14)

3.7.1.6 Interfaces Subgroup

This subgroup is working on the following document: *NIST Big Data Interoperability Framework: Volume 8, Reference Architecture Interface*.

This document summarizes interfaces that are instrumental for the interaction with Clouds, Containers, and HPC systems to manage virtual clusters to support the NIST Big Data Reference Architecture (NBDRA). The Representational State Transfer (REST) paradigm is used to define these interfaces allowing easy integration and adoption by a wide variety of frameworks. . This volume, Volume 8, uses the work performed by the NBD-PWG to identify objects instrumental for the NIST Big Data Reference Architecture (NBDRA) which is introduced in the NBDIF: Volume 6, Reference Architecture.

This presentation was given at the *2nd NIST Big Data Public Working Group (NBD-PWG) Workshop* in Washington DC in June 2017. It explains our thoughts on deriving automatically a reference architecture form the Reference Architecture Interface specifications directly from the document.

The workshop Web page is located at

- https://bigdatawg.nist.gov/workshop2.php

The agenda of the workshop is as follows:

- https://bigdatawg.nist.gov/2017_NIST_Big_Data_PWG_WorkshopAgenda_

The Web cas of the presentation is given bellow, while you need to fast forward to a particular time

- Webcast: Interface subgroup: https://www.nist.gov/news-events/events/2017/06/2nd-nist-big-data-public-working-group-nbd-pwg-workshop
    - see: Big Data Working Group Day 1, part 2 Time start: 21:00 min, Time end: 44:00
- Slides: https://github.com/cloudmesh/cloudmesh.rest/blob/master/docs/NBDPWG-vol8.pptx?raw=true
- Document: https://github.com/cloudmesh/cloudmesh.rest/raw/master/docs/NIST.SP.1508-draft.pdf

You are welcome to view other presentations if you are interested.

3.7.1.7 Requirements and Use Case Subgroup

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.

Requirements (27:28)

3.7.2 51 Big Data Use Cases

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.


51 Use Cases (100)

3.7.2.1 Government Use Cases

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).

Government Use Cases (17:43)

3.7.2.2 Commercial Use Cases

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.

Commercial Use Cases (17:43)

3.7.2.3 Defense Use Cases

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.

 Defense Use Cases (15:43)

3.7.2.4 Healthcare and Life Science Use Cases

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.

 Healthcare and Life Science Use Cases (30:11)

3.7.2.5 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.

 Deep Learning and Social Networks Use Cases (14:19)

3.7.2.6 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.

 Research Ecosystem Use Cases (9:09)

3.7.2.7 Astronomy and Physics Use Cases

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.

 Astronomy and Physics Use Cases (17:33)

3.7.2.8 Environment, Earth and Polar Science Use Cases

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.

 Environment, Earth and Polar Science Use Cases (25:29)

3.7.2.9 Energy Use Case

This covers Consumption forecasting in Smart Grids.

 Energy Use Case (4:01)

3.7.3 Features of 51 Big Data Use Cases

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.

 Features (43)

3.7.3.1 Summary of Use Case Classification

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.

 Summary of Use Case Classification (23:39)

3.7.3.2 Database(SQL) Use Case Classification

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

 Database (SQL) Use Case Classification (11:13)

3.7.3.3 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.

 NoSQL Use Case Classification (11:20)

3.7.3.4 Other Use Case Classifications

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

 Use Case Classifications I (12:42) This discusses a subset of use case features: Pleasingly parallel, MRStat, Data Assimilation, Crowd sourcing, Agents, data fusion and agents, EGO and security.

 Use Case Classifications II (20:18)

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.

 Use Case Classifications III (17:25)

3.7.3.5 Resources

- NIST Big Data Public Working Group (NBD-PWG) Process
- Big Data Definitions
- Big Data Taxonomies
- Big Data Use Cases and Requirements
- Big Data Security and Privacy
- Big Data Architecture White Paper Survey
- Big Data Reference Architecture

- Big Data Standards Roadmap

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
- On line 51 Use Cases

- Summary of Requirements Subgroup

- ~~Use Case 6 Mendeley~~ **(this link does not exist any longer)**
- ~~Use Case 7 Netflix~~
- 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)
- Use Case 11 and Use Case 12 Simulation driven Materials Genomics
- 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_COLWell
  - http://stids.c4i.gmu.edu/STIDS2011/papers/STIDS2011_CR_T1_Salm
  - ~~http://stids.c4i.gmu.edu/papers/STIDSPapers/STIDS2012/_T14/_Smith~~
  - ~~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/HadoopGIS
- Use Case 19 Genome in a Bottle Consortium:
  - ~~www.genomeinabottle.org~~
- Use Case 20 Comparative analysis for metagenomes and genomes
- Use Case 25
  - Biodiversity
  - 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~~
  - 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
- Use Case 28
    - Truthy: Information diffusion research from Twitter Data
    - 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~~
- Use Case 31 NIST Information Access Division analytic technology performance measurement, evaluations, and standards
- Use Case 32
    - DataNet Federation Consortium DFC: The DataNet Federation Consortium,
    - iRODS
- Use Case 33 The 'Discinnet process', big data global experiment
- 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
- 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%2(
    - 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~~ **(old link does not exist, new link: https://www.belle2.org)**
- Use Case 41 EISCAT 3D incoherent scatter radar system
- 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
- 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/~~ **(old link does not exsit, new link: http://dslab.usc.edu/smartgrid.php)**
    - 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=r&_afrWindowId=r#%40%3F_afrWindowId%3Dr%26_afrLoop%3D157401916661989%26_afrWindowMode%3D0%26_adf.ctrl-state%3Db7yulr4rl_17
    - http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6475927

3.8 Sensors

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.

Sensor I (31)

Sensor II (44)

3.8.1 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.

 Internet of Things (12:36)

3.8.2 Robotics and IoT

Examples of Robots and Drones.

 Robotics and IoT Expectations (8:05)

3.8.3 Industrial Internet of Things

We summarize surveys and expectations Industry wide.

 Industrial Internet of Things (24:02)

3.8.4 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.

 Sensor Clouds (4:40)

3.8.5 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.

Earth/Environment/Polar Science data gathered by Sensors (4:58)

3.8.6 Ubiquitous/Smart Cities

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

Ubiquitous/Smart Cities (1:44)

3.8.7 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.

U-Korea (U=Ubiquitous) (2:49)

3.8.8 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.

Smart Grid (6:04)

3.8.9 Resources

- http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Competitive-Landscape-Industries.pdf

[@www-accenture-insight-industrial]

- http://www.gesoftware.com/ge-predictivity-infographic [@www-predix-ge-Industrial]
- http://www.getransportation.com/railconnect360/rail-landscape [@www-getransportation-digital]

- http://www.gesoftware.com/sites/default/files/GE-Software-Modernizing-Machine-to-Machine-Interactions.pdf [@www-ge-digital-software]

These resources do not exsit:

- ~~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/~~

3.9 RADAR

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.

 Radar (58)

3.9.1 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.

- Radar Informatics (3:31)

3.9.2 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.

- Remote Sensing (6:43)

3.9.3 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.

- Ice Sheet Science (1:00)

3.9.4 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.

- Global Climate Change (2:51)

3.9.5 Radio Overview

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

-  Radio Overview (4:16)

3.9.6 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.

-  Radio Informatics (3:35)

3.10 Web Search and Text Mining

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.

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 the section.

3.10.1 Web Search and Text Mining

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.

 Web Search and Text Mining (56)

3.10.1.1 The Problem

 Text Mining (9:56)

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.

3.10.1.2 Information Retrieval

 Information Retrieval (6:06)

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.

3.10.1.3 History

 Web Search History (5:48)

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

3.10.1.4 Key Fundamental Principles

 Principles (9:30)

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.

3.10.1.5 Information Retrieval (Web Search) Components

 Fundamental Principals of Web Search (5:06)

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

3.10.2 Search Engines

 Search Engines (3:08)

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.

3.10.2.1 Boolean and Vector Space Models

 Boolean and Vector Space Model (6:17)

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

3.10.2.2 Web crawling and Document Preparation

 Web crawling and Document Preparation (4:55)

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

3.10.2.3 Indices

 Indices (5:44)

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.

3.10.2.4 TF-IDF and Probabilistic Models

 TF-IDF and Probabilistic Models (3:57)

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.

3.10.3 Topics in Web Search and Text Mining

 Text Mining (33)

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.

3.10.3.1 Data Analytics for Web Search

 Web Search and Text Mining II (6:11)

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.

3.10.3.2 Link Structure Analysis including PageRank

 Related Applications (17:24)

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.

3.10.3.3 Web Advertising and Search

 Web Advertising and Search (9:02)

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.

3.10.3.4 Clustering and Topic Models

 Clustering and Topic Models (6:21)

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.

3.10.3.5 Resources

All resources accessed March 2018.

- http://saedsayad.com/data_mining_map.htm
- http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html
- The Web Graph: an Overviews
- Jean-Loup Guillaume and Matthieu Latapy
- 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.ifis.cs.tu-bs.de/teaching/ss-11/irws
- https://en.wikipedia.org/wiki/PageRank
- Meeker/Wu May 29 2013 Internet Trends D11 Conference

3.11 Health Informatics

 Health Informatics (131)

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.

3.11.1 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.

 Big Data and Health (10:02)

3.11.2 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.

 Status of Healthcare Today (16:09)

3.11.3 Telemedicine (Virtual Health)

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

 Telemedicine (8:21)

3.11.4 Medical Big Data in the Clouds

An impressive exemplar Cloud-based medicine system from Potsdam.

 Medical Big Data in the Clouds (15:02)

3.11.4.1 Medical image Big Data

 Medical Image Big Data (6:33)

3.11.4.2 Clouds and Health

 Clouds and Health (4:35)

3.11.4.3 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.

 McKinsey Report (14:53)

3.11.4.4 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.

 Microsoft Report on Big Data in Health (2:26)

3.11.4.5 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.

 EU Report on Redesigning health in Europe for 2020 (5:00)

3.11.4.6 Medicine and the Internet of Things

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

 Medicine and the Internet of Things (8:17)

3.11.4.7 Extrapolating to 2032

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

 Extrapolating to 2032 (15:13)

3.11.4.8 Genomics, Proteomics and Information Visualization

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.

 Genomics, Proteomics and Information Visualization (6:56)

Next we continue 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.

 Genomics, Proteomics and Information Visualization I (10:33)

 Genomics, Proteomics and Information Visualization: II (7:41)

 Proteomics and Information Visualization (131)

3.11.4.9 Resources

- https://wiki.nci.nih.gov/display/CIP/CIP+Survey+of+Biomedical+Imaging+ [@wiki-nih-cip-survey]
- http://grids.ucs.indiana.edu/ptliupages/publications/Where\%20does\%20all\ [@fox2011does]
- ~~http://www.ieee-icsc.org/ICSC2010/Tony\%20Hey\%20-\%2020100923.pdf~~(this link does not exist any longer)

- http://quantifiedself.com/larry-smarr/ [@smarr13self]
- http://www.ebi.ac.uk/Information/Brochures/ [@www-ebi-aboutus]
- http://www.kpcb.com/internet-trends [@www-kleinerperkins-internet-trends]
- http://www.slideshare.net/drsteventucker/wearable-health-fitness-trackers-and-the-quantified-self [@www-slideshare-wearable-quantified-self]
- http://www.siam.org/meetings/sdm13/sun.pdf [@archive –big-data-analytics-healthcare]
- http://en.wikipedia.org/wiki/Calico_\%28company\%29 [@www-wiki-calico]
- http://www.slideshare.net/GSW_Worldwide/2015-health-trends [@www-slideshare-2015-health trends]
- http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Changing-Competitive-Landscape-Industries.pdf [@www-accenture-insight-industrial-internet]
- http://www.slideshare.net/schappy/how-realtime-analysis-turns-big-medical-data-into-precision-medicine [@www-slideshare-big-medical-data-medicine]
- http://medcitynews.com/2013/03/the-body-in-bytes-medical-images-as-a-source-of-healthcare-big-data-infographic/ [@medcitynews-bytes-medical-images]
- ~~http://healthinformatics.wikispaces.com/file/view/cloud_computing.ppt~~ (this link does not exist any longer)
- https://www.mckinsey.com/~/media/mckinsey/industries/healthcare%20syst [@www-mckinsey-industries-healthcare]
- ~~https://partner.microsoft.com/download/global/40193764~~ (this link does not exist any longer)
- https://ec.europa.eu/eip/ageing/file/353/download_en?token=8gECi1RO
- ~~http://www.liveathos.com/apparel/app~~
- http://debategraph.org/Poster.aspx?aID=77 [@debategraph-poster]
- ~~http://www.oerc.ox.ac.uk/downloads/presentations-from-events/microsoftworkshop/gannon~~(this link does not exist any longer)
- ~~http://www.delsall.org~~ (this link does not exist any longer)
- http://salsahpc.indiana.edu/millionseq/mina/16SrRNA_index.html [@www-salsahpc-millionseq]
- http://www.geatbx.com/docu/fcnindex-01.html [@www-geatbx-parametric-optimization]

3.12 TECHNOLOGIES

3.12.1 Statistics

We assume that you are familiar with elementary statistics including

- mean, minimum, maximum
- standard deviation
- probability
- distribution
- frequency distribution
- Gaussian distribution
- bell curve
- standard normal probabilities
- tables (z table)
- Regression
- Correlation

Some of these terms are explained in various sections throughout our application discussion. This includes especially the Physics section. However these terms are so elementary that any undergraduate or highschool book will provide you with a good introduction.

It is expected from you to identify these terms and you can contribute to this section with non plagiarized subsections explaining these topics for credit.

Topics identified by a :?: can be contributed by students. If you are interested, use piazza for announcing your willingness to do so.

Mean, minimum, maximum:

Standard deviation:

Probability:

Distribution:

Frequency distribution:

Gaussian distribution:

Bell curve:

Standard normal probabilities:

Tables (z-table):

Regression:

Correlation:

3.12.1.1 Exercise

E.Statistics.1:

> *Pick a term from the previous list and define it while not plagiarizing. Create a pull request. Coordinate on piazza as to not duplicate someone else's contribution. Also look into outstanding pull requests.*

E.Statistics.2:

> *Pick a term from the previous list and develop a python program demonstrating it and create a pull request for a contribution into the examples directory. Make links to the github location. Coordinate on piazza as to not duplicate someone else's contribution. Also look into outstanding pull requests.*

3.12.2 Practical K-Means, Map Reduce, and Page Rank for Big Data Applications and Analytics

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.

 K-Means I (11:42)

 K-Means II (11:54)

3.12.2.1 K-means in Practice

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.

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.

Files:

- https://github.com/cloudmesh-community/book/blob/master/examples/python/kmeans/xmean.py
- https://github.com/cloudmesh-community/book/blob/master/examples/python/kmeans/sample.csv
- https://github.com/cloudmesh-community/book/blob/master/examples/python/kmeans/parallel-kmeans.py
- https://github.com/cloudmesh-community/book/blob/master/examples/python/kmeans/kmeans-extra.py

3.12.2.1.1 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.

3.12.2.1.2 Analysis of 4 Artificial Clusters

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.

3.12.2.2 Parallel K-means

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.

Files:

- https://github.com/cloudmesh-community/book/blob/master/examples/python/kmeans/parallel-kmeans.py

3.12.2.3 PageRank in Practice

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.

Files:

- https://github.com/cloudmesh-community/book/blob/master/examples/python/page-rank/pagerank1.py
- https://github.com/cloudmesh-community/book/blob/master/examples/python/page-rank/pagerank2.py

3.12.2.4 Resources

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

3.12.3 Plotviz

NOTE: This an legacy application this has now been replaced by WebPlotViz which is a web browser based visualization tool which provides added functionality's.

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.

3.12.3.1 Using Plotviz Software for Displaying Point Distributions in 3D

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.

 Plotviz (34)

Files:

- https://github.com/cloudmesh-community/book/blob/master/examples/python/plotviz/fungi-lsu-3-15-to-3-26-zeroidx.pviz
- https://github.com/cloudmesh-community/book/blob/master/examples/python/plotviz/datingrating-originallabels.pviz
- https://github.com/cloudmesh-community/book/blob/master/examples/python/plotviz/clusterFinal-M30-C28.pviz
- https://github.com/cloudmesh-community/book/blob/master/examples/python/plotviz/clusterfinal-m3-c3dating-reclustered.pviz

3.12.3.1.1 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.

 Motivation (7:58)

3.12.3.1.2 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.

 Example I (9:45)

3.12.3.1.3 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.

 Proteomics and Synchronized Rotation (9:14)

3.12.3.1.4 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.

 Larger Proteomics Sample (8:37)

3.12.3.1.5 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.

 Plotviz I (10:17)

3.12.3.1.6 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.

 Plotviz II (14:58)

3.12.3.2 Resources

Download

4 DEVTOOLS

4.1 REFCARDS

Learning Objectives

- Obtain quickly information about technical aspects with the help of reference cards.

We present you with a list of useful short reference cards. This cards can be extremely useful to remind yourself about some important commands and features. Having them could simplify your interaction with the systems, We not only collected here some refcards about Linux, but also about other useful tools and services.

If you like to add new topics, let us know via your contribution (see the contribution section).

CheatSheets

- CheatSheets

Editors

- Emacs
- Vi
- Vim

Documentation

- LaTeX
- RST

Linux

- Linux
- Makefile
- Git

Cloud/Virtualization

- Openstack
- Openstack
- vagrant

SQL

- SQL

Languages

- R

Python

- Python
- PythonData
- Numpy/Pandas
- PythonTutorial
- Python
- Python
- PythonAPIIndex
- Python3

4.2 Virtual Box

For development purposes we recommend that you use for this class an Ubuntu virtual machine that you set up with the help of virtualbox. We recommend that you use the current version of ubuntu and do not install or reuse a version that you have set up years ago.

As access to cloud resources requires some basic knowledge of linux and security we will restrict access to our cloud services to those that have

demonstrated responsible use on their own computers. Naturally as it is your own computer you must make sure you follow proper security. We have seen in the past students carelessly working with virtual machines and introducing security vulnerabilities on our clouds just because “it was not their computer.” Hence, we will allow using of cloud resources only if you have demonstrated that you responsibly use a linux virtual machine on your own computer. Only after you have successfully used ubuntu in a virtual machine you will be allowed to use virtual machines on clouds.

A *cloud drivers license test* will be conducted. Only after you pass it we wil 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 to cloud projects before you have not passed the test. Please be patient. Only students enrolled in the class can get access to the cloud.

If you however have access to other clouds yourself you are welcome to use the, However, be reminded that projects need to be reproducible, on our cloud. This will require you to make sure a TA can replicate it.

Let us now focus on using virtual box.

4.2.1 Installation

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 some hardware you may have may be too old or has too little resources to be useful. We have heard from students that the following is a minimal setup for the desktop machine:

- multi core processor or better allowing to run hypervisors
- 8 GB system memory
- 50 GB of free hard drive space

For virtual machines you may need multiple, while the minimal configuration may not work for all cases.

As configuration we often use

minimal

1 core, 2GB Memory, 5 GB disk

latex

2 core, 4GB Memory, 25 GB disk

A video to showcase such an install is available at:

 Using Ubuntu in Virtualbox (8:08)

> ⚠ *Please note that the video shows the version 16.04. You should however use the newest version which at this time is 18.04.*

If you specify your machine too small you will not be able to install the development environment. Gregor used on his machine 8GB RAM and 25GB diskspace.

Please let us know the smallest configuration that works.

4.2.2 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

 Virtualbox (4:46)

Please reboot the machine after installation and configuration.

On OSX you can once you have enabled bidirectional copying in the Device tab with

OSX to 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. If you let us know TAs will add the information here.

4.2.3 Exercises

E.Virtualbox.1:

> *Install ubuntu desktop on your computer with guest additions.*

E.Virtualbox.2:

> *Make sure you know how to paste and copy between your host and guest operating system.*

E.Virtualbox.3:

> *Install the programs defined by the development configuration.*

E.Virtualbox.4:

> *Provide us with the key combination to copy and paste between Windows and Vbox.*

4.3 VAGRANT

Learning Objectives

- Be able to experiment with virtual machines on your computer before you go on a cloud.
- Simulate a virtual cluster with multiple VMs running on your computer if it is big enough.

A convenient tool to interface with Virtual Box is vagrant.Vagrant allows us to manage virtual machines directly from the commandline. It support also other providers and can be used to start virtual machines and even containers. The latest version of vagrant includes the ability to automatically fetch a virtual machine image and start it on your local computer. It assumes that you have virtual box installed. Some key concepts and advertisement are located at

- https://www.vagrantup.com/intro/index.html:

Detailed documentation for it is located

- https://www.vagrantup.com/docs/index.html

A list of *boxes* is available from

- https://app.vagrantup.com/boxes/search

One image we will typically use is Ubuntu 18.04. Please note that older version may not be suitable for class and we will not support any questions about them. This image is located at

- https://app.vagrantup.com/ubuntu/boxes/bionic64

4.3.1 Installation

Vagrant is easy to install. You can go to the download page and download and install the appropriate version:

- https://www.vagrantup.com/downloads.html

4.3.1.1 macOS

On MacOS, download the dmg image, and click on it. You will find a pkg in it that you double click. After installation vagrant is installed in

- `/usr/local/bin/vagrant`

Make sure `/usr/local/bin` is in your `PATH` Start a new terminal to verify this.

Check it with

```
echo $PATH
```

If it is not in the path put

export PATH=/usr/local/bin:$PATH

in the terminal command or in your `~/.bash_profile`

4.3.1.2 Windows 

 students contribute

4.3.1.3 Linux 

 students contribute

4.3.2 Usage

To download, start and login into install the 18.04:

```
host$ vagrant init ubuntu/bionic64
host$ vagrant up
host$ vagrant ssh
```

Once you are logged in you can test the version of python with

```
vagrant@ubuntu-bionic:~$ sudo apt-get update
vagrant@ubuntu-bionic:~$ python3 --version
Python 3.6.5
```

To install a newer version of python, and pip you can use

```
vagrant@ubuntu-bionic:~$ sudo apt-get install python3.7
vagrant@ubuntu-bionic:~$ sudo apt-get install python3-pip
```

To install the light weight idle development environment in case you do not want o use pyCharm, please use

```
vagrant@ubuntu-bionic:~$ sudo apt-get install idle-python
```

So that you do not have to always use the number 3, you can also set an alias with

```
alias python=python3
```

When you exit the virtual machine with the

```
exit command
```

It does not terminate the VM. You can use from your host system the commands such as

```
host$ vagrant status
host$ vagrant destroy
host$ vagrant suspend
host$ vagrant resume
```

to manage the vm.

4.4 Packer 

Packer is an open source tool for creating identical machine images for multiple platforms from a single source configuration. Packer runs on every major

operating system, and creates machine images for multiple platforms in parallel form configuration specufications.

Some key concepts are located at

- https://www.packer.io/intro/index.html

Detailed documentation is located at

- https://www.packer.io/docs/index.html

Use cases for packer is located at

- https://www.packer.io/intro/use-cases.html

4.4.1 Installation

Installation instructions for all platforms is located at

- https://www.packer.io/intro/getting-started/install.html

4.4.2 Usage

In the Section vagrant we use vagrant to start up an Ubuntu 18.04 virtual machine. Once the VM was up and running, vagrant allowed the user to log in and setup the VM according to the user's requirements. In that example, the user ran commands to install and upgrade software dependencies:

1. upgrade from Python 3.6.5 to Python 3.7
2. installing python3-pip and idle-python
3. alias `python` to `python3`

Let us assume that the VM is now in a desirable state for the purpose of doing development on a large number of virtual machines and you want to distribute it to the rest of your team or community so that all are using the same environment. You could simply send your team members a copy of your Ubuntu 18.04 VirtualBox VM assuming they will be developing on VMs using VirtualBox. However, let us assume one community member wants to develop on Google Cloud Platform, another on AWS and another on OpenStack. In this

case, they will each need to figure out how to import a VirtualBox VM into the respective cloud vendor they're utilizing. Packer can help this situation by codifying the state of the development environment with a single configuration file which can then be used to create images in different cloud environments.

Assuming packer has been installed, let's create a packer JSON file that will build an Ubuntu 18.04 image and provision it as we did manually using Vagrant. In this example, we will create the image in Google Compute Platform.

First download your Google Cloud credentials according to the documentation at

- https://www.packer.io/docs/builders/googlecompute.html#running-without-a-compute-engine-service-account

Save the credential file as `accounts.json`. Also, determine the project ID you will use in your Google Cloud Platform account. In this example, we will use `my_project_id` for our project ID.

Next save the following JSON to a file named `e516.json`:

```
{
  "variables": {
    "google_project_id": null
  },
  "builders": [
    {
      "type":         "googlecompute",
      "account_file": "account.json",
      "project_id":   "{{ user `google_project_id` }}",
      "image_name":   "ubuntu-1804-dev-e516",
      "source_image": "ubuntu-1804-bionic-v20180911",
      "ssh_username": "packer",
      "zone":         "us-central1-a"
    }
  ],
  "provisioners": [
    {
      "type": "shell",
      "expect_disconnect": true,
      "inline": [
        "sudo apt-get update -y",
        "sudo apt-get install -y python3.7 python3-pip idle-python3.7",
        "echo \"alias python='python3'\" > .bash_aliases"
      ]
    }
  ]
}
```

The packer file format specifies 3 sections, `variables`, `builders` and `provisioners`. The `variables` section allows you to declare variables that are to be used in the rest of the document. By declaring a variable in this section, for example `google_project_id`, it allows the user to pass in the value of that variable via the packer command line.

The `builders` section allows you to declare the builders for any cloud vendor supported by packer. The list of supported vendors can be found here:

- https://www.packer.io/docs/builders/index.html

In our example, we define the builder for Google Cloud Platform which requires our credential file (account.json), our project ID, base image name, ssh username and zone.

Finally, the `provisioners` section allows the user to customize the base image defined in the `builders` section. In our example, we simply use the `shell` provisioner which allows us to type in shell commands to provision the image as we want it. Here we install python3.7, python3-pip and idle-python3.7. We also write out an aliases file so that upon login, the user can access python3.7 using the python alias.

To build the image, we now run packer:

```
$ packer build -var 'google_project_id=my_project_id' e516.json
```

You will see output that shows the progress of packer as it starts up and provisions the instance. Upon success, packer will create an image from the instance and clean up after itself:

```
$ googlecompute output will be in this color.

==> googlecompute: Checking image does not exist...
==> googlecompute: Creating temporary SSH key for instance...
==> googlecompute: Using image: ubuntu-1804-bionic-v20180911
==> googlecompute: Creating instance...
    googlecompute: Loading zone: us-central1-a
    googlecompute: Loading machine type: n1-standard-1
    googlecompute: Requesting instance creation...
    googlecompute: Waiting for creation operation to complete...
    googlecompute: Instance has been created!
==> googlecompute: Waiting for the instance to become running...
    googlecompute: IP: 104.154.21.240
==> googlecompute: Waiting for SSH to become available...
==> googlecompute: Connected to SSH!
==> googlecompute: Provisioning with shell script: /var/folders/rm/g1h4bhf54x750jzjyryckmnc000lxd/T/packer-shell210916201
    googlecompute: Get:1 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
...
    googlecompute: Setting up idle-python3.7 (3.7.0-1~18.04) ...
    googlecompute: Processing triggers for libc-bin (2.27-3ubuntu1) ...
    googlecompute: Processing triggers for ureadahead (0.100.0-20) ...
    googlecompute: Processing triggers for systemd (237-3ubuntu10.3) ...
==> googlecompute: Deleting instance...
    googlecompute: Instance has been deleted!
==> googlecompute: Creating image...
==> googlecompute: Deleting disk...
    googlecompute: Disk has been deleted!
Build 'googlecompute' finished.

==> Builds finished. The artifacts of successful builds are:
--> googlecompute: A disk image was created: ubuntu-1804-dev-e516
```

You can now click on the list of images in the Google Compute Platform console to see your new image. The new image is ready to use for development.

Next, let's add a builder for an AWS AMI. Before we do that, setup your AWS credentials using the AWS CLI according to the documentation here:

- https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html

Ensure your `default` profile is saved under `~/.aws/credentials`.

Update the `e516.json` so that the contents is as follows:

```
{
  "variables": {
    "google_project_id": null,
    "image_name":   "ubuntu-1804-dev-e516",
    "ssh_username": "packer"
  },
  "builders": [
    {
      "type":         "googlecompute",
      "account_file": "account.json",
      "ssh_username": "{{ user `ssh_username` }}",
      "project_id":   "{{ user `google_project_id` }}",
      "image_name":   "{{ user `image_name` }}",
      "source_image": "ubuntu-1804-bionic-v20180911",
      "zone":         "us-central1-a"
    },
    {
      "type":         "amazon-ebs",
      "ssh_username": "{{ user `ssh_username` }}",
      "profile":      "default",
      "ami_name":   "{{ user `image_name` }}",
      "source_ami":   "ami-0bbe6b35405ecebdb",
      "instance_type":"t2.micro",
      "region":       "us-west-2"
    }
  ],
  "provisioners": [
    {
      "type": "shell",
      "expect_disconnect": true,
      "inline": [
        "sudo apt-get update -y",
        "sudo apt-get install -y python3.7 python3-pip idle-python3.7",
        "echo \"alias python='python3'\" > .bash_aliases"
      ]
    }
  ]
}
```

Note that we've added the AWS builder in the `builders` section and that we've refactored the `ssh_username` and `image_name` to the `variables` section since those variable hold values that can be reused in both the Google Compute and AWS builders.

Let's rerun packer:

```
packer build -var 'google_project_id=my_project_id' e516.json
```

You will see output that states the image already exists in your Google Compute account and so packer smartly skips building that image. The output also shows the progress of packer as it starts up and provisions the instance in AWS. Upon success, packer will create an AMI from the instance and clean up after itself:

```
amazon-ebs output will be in this color.
googlecompute output will be in this color.

==> googlecompute: Checking image does not exist...
==> amazon-ebs: Prevalidating AMI Name: ubuntu-1804-dev-e516
==> googlecompute: Image ubuntu-1804-dev-e516 already exists.
==> googlecompute: Use the force flag to delete it prior to building.
Build 'googlecompute' errored: Image ubuntu-1804-dev-e516 already exists.
Use the force flag to delete it prior to building.
    amazon-ebs: Found Image ID: ami-0bbe6b35405ecebdb
==> amazon-ebs: Creating temporary keypair:
    packer_5bad9d99-f631-1778-1e83-afd19ad0d5cc
==> amazon-ebs: Creating temporary security group for this instance:
    packer_5bad9d9b-38c5-252d-0368-74aa75bfb286
==> amazon-ebs: Authorizing access to port 22 from 0.0.0.0/0
    in the temporary security group...
==> amazon-ebs: Launching a source AWS instance...
==> amazon-ebs: Adding tags to source instance
    amazon-ebs: Adding tag: "Name": "Packer Builder"
    amazon-ebs: Instance ID: i-0d0383f9f84b54051
```

You can now click on the list of images in the AWS EC2 console to see your new AMI. The new AMI is ready to use for development.

4.5 Ubuntu on an USB stick 

In case you cannot install any programs on your development computer most often the easiest way is to use the hardware but boot the OS from a USB stick. Make sure you have access to the Bios or your system to actually boot from a USB device before you start this activity.

4.5.1 Ubuntu on an USB stick for macOS via Command Line

The easiest way to create an ubuntu distribution that can be booted from an USB stick is done via command line. The original Web page for this method is available at this [link].

We have copied some of the information from this Web page but made enhancements to it. Currently all images are copied form that Web page.

> ⚠ *Please test it out and improve if it does not work.*

Our goal is to create a USB stick that has either Ubuntu 18.04 LTS that can be

downloaded from this [link]. You will need a USB stick/flash drive. We recommend a 8GB or larger. Please let us know if it works for you on larger than 8GB drives.

We assume that you downloaded the iso from ubuntu to a folder called */iso. Next we open a terminal and cd into the folder /iso*. Now we need to convert the is to an image file. This is done as follows and you need to execute the command for the version of ubuntu you like to use.

Your folder will look something like this

```
$ ls -1

    ubuntu-18.04-desktop-amd64.iso
```

You will need to generate an image with the following command

```
$ hdiutil convert ubuntu-18.04-desktop-amd64.iso -format UDRW -o ubuntu-18.04-desktop-amd64.img
```

macOS will append a .dmg behind the name. At this time **do not** plug in your usb stick. Just issue the command

```
$ diskutil list
```

Observe the output. Now plug in the USB stick. Wait till the USB stick registers in the Finder. If this does not work find a new USB stick or format it. Execute the command

```
$ diskutil list
```

and observer the output again. Another device will register and you will see something like

```
/dev/disk2 (external, physical):
#:     TYPE NAME               SIZE       IDENTIFIER
0:     FDisk_partition_scheme *8.2 GB     disk2
1:     DOS_FAT_32 NO NAME      8.2 GB     disk2s1
```

Please note in this example the device path and number is recognized as

```
/dev/disk2
```

It also says external, which is a good sign as the USB stick is external. Next, we need to unmount the device with

```
$ diskutil unmountDisk /dev/diskN
```

where you replace the number N with the disk number that you found for the device. In our example it would be 2. If you see the error “Unmount of diskN failed: at least one volume could not be unmounted”, start Disk Utility.app and unmount the volume (do not eject). If it was successful, you will see

```
Unmount of all volumes on disk2 was successful
```

The next step is dangerous and you need to make sure you follow it. So please do not copy and paste, but read first, reflect and only if you understand it execute it. We know we say this all the time, but better saying it again instead of you destroying your system. This command also requires sudo access so you will either have to be in the sudo group, or use

```
$ su <your administrator name>
```

login and than execute the command under root.

```
$ sudo dd if=ubuntu-18.04-desktop-amd64.img.dmg of=/dev/diskN bs=1m
```

(Not tested: Using /dev/rdisk instead of /dev/disk may be faster according to the ubuntu documentation)

Ubuntu’s Web page also gives the following tips:

- “If you see the error dd: Invalid number ‘1m’, you are using GNU dd. Use the same command but replace bs=1m with bs=1M.”
- “If you see the error dd: /dev/diskN: Resource busy, make sure the disk is not in use. Start Disk Utility.app and unmount the volume (do not eject).”

You will see an error window popping up telling you: **The disk inserted was not readable by this compute**. Please, leave the window as is and instead type in on the terminal.

```
$ diskutil eject /dev/diskN
```

Now remove the flash drive, and press in the error window **Ignore**
Now you have a flash drive with ubuntu installed and you can boot from it. To do so, please

restart your Mac and press option key

while the Mac is restarting to choose the USB-Stick

You will need a plug for USB keyboard, USB mouse, and network cable.

There are some issue from this point on.

```
$ sudo apt-get update
```

Add universe to the window for application updates

see https://help.ubuntu.com/community/Repositories/Ubuntu

```
$ sudo apt-get install vnc4server
```

Start the server and set up a password

```
$ vncserver
```

The next section is untested and needs verification.

4.5.1.1 Boot from the USB Stick

To boot from the USB stick, you need to restart or power-on the Mac with the USB stick inserted while you press the Option/alt key.

The launch *Startup Manager* will be started showing a list of bootable devices connected to the machine. Your USB stick should appear as gold/yellow and labelled *EFI Boot*. Use your cursor keys to move to the most right EFI boot device in that list (likely the USB stick) and press ENTER. YOu can also use the mouse.

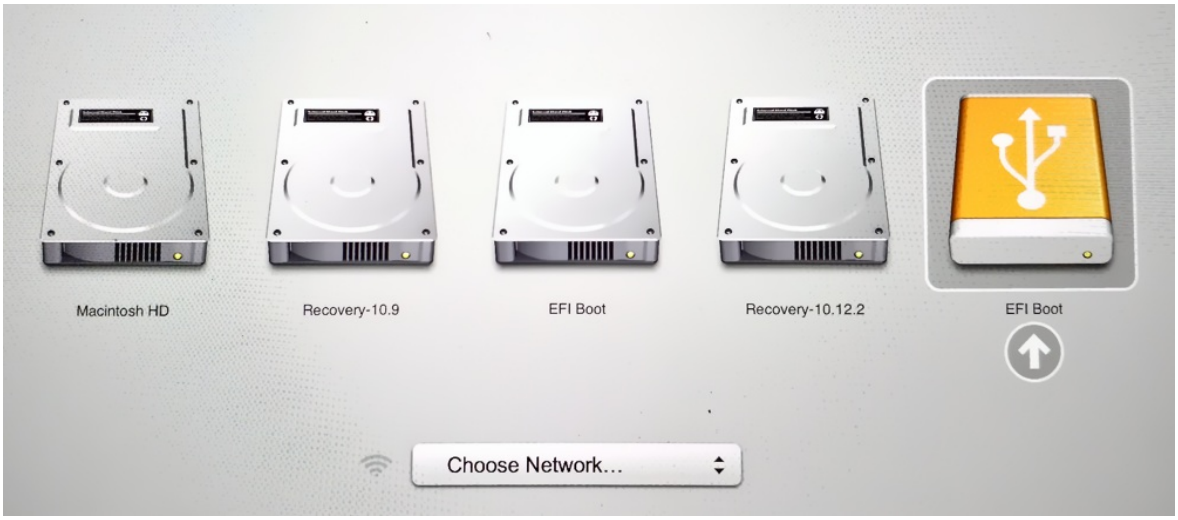


Figure: Boot Screen

A boot menu will shortly start up and after you press again ENTER your machine will boot into Ubuntu.

For more information on how to setup ubuntu see:

- https://tutorials.ubuntu.com/tutorial/tutorial-install-ubuntu-desktop#0

After you have booted and logged in, you need to update the distribution. We recommend that you switch on Universe in the applications settings.

Next you need to issue in the command terminal

```
$ sudo apt-get update
```

You will likely see some warnings with number 95 which you can ignore. Please report your experience and we update this page based on your feedback.

4.5.2 Ubuntu on an USB stick for macOS via GUI

An alternative to the Command Line solution to create an USB stick with bootable Ubuntu on is to use the macOS GUI. This method is more complex than the command line solution. In addition as we are learning about cloud computing in this book, it is of advantage to learn how to do this from commandline as the replication of the approach via commandline is easier and more scalable. However for completeness, we have also included here the GUI-based method.

The material in this section was copied and modified from

- https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-macos

You will need a USB stick/flash drive. We recommend a 8GB or larger. Please let us know if it works for you on larger than 8GB drives.

4.5.2.1 Install Etcher

Etcher is a tool that allows you to easily write an ISO onto a USB stick. Etcher is integrated in the macOS GUI environment and allows to drag the iso into it for burning. Etcher can be found at

- https://etcher.io/

As this is an application from unidentified developers (not registered in the apple store), you need to enable it after downloading. To do so, you can enable the *App Store and identified developers* in the *Security and Privacy* pane in the System Preferences. IN case you get a warning about running the application, click *Open Anyway* in the same pane.

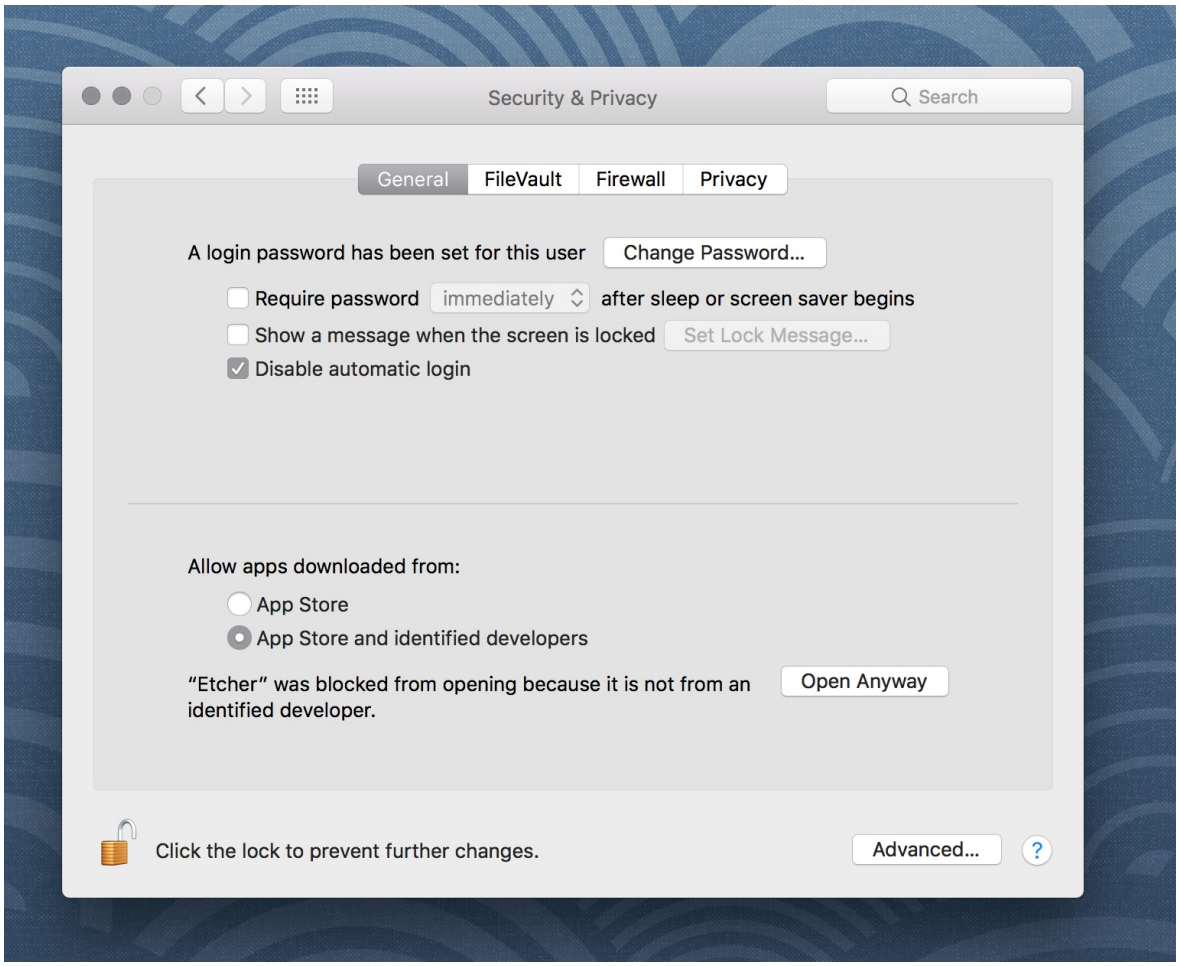


Figure: Setting

4.5.2.2 Prepare the USB stick

The Disk Utility needs to be used with caution as selecting the wrong device or partition can result in data loss.

Next you need to conduct the following steps which we copied from the Ubuntu Web page:

- Launch Disk Utility from Applications>Utilities or Spotlight search
- Insert your USB stick and observe the new device added to Disk Utility
- Select the USB stick device and select Erase from the tool bar (or right-click menu)
- Set the format to MS-DOS (FAT) and the scheme to GUID Partition Map Check you've chosen the correct device and click Erase

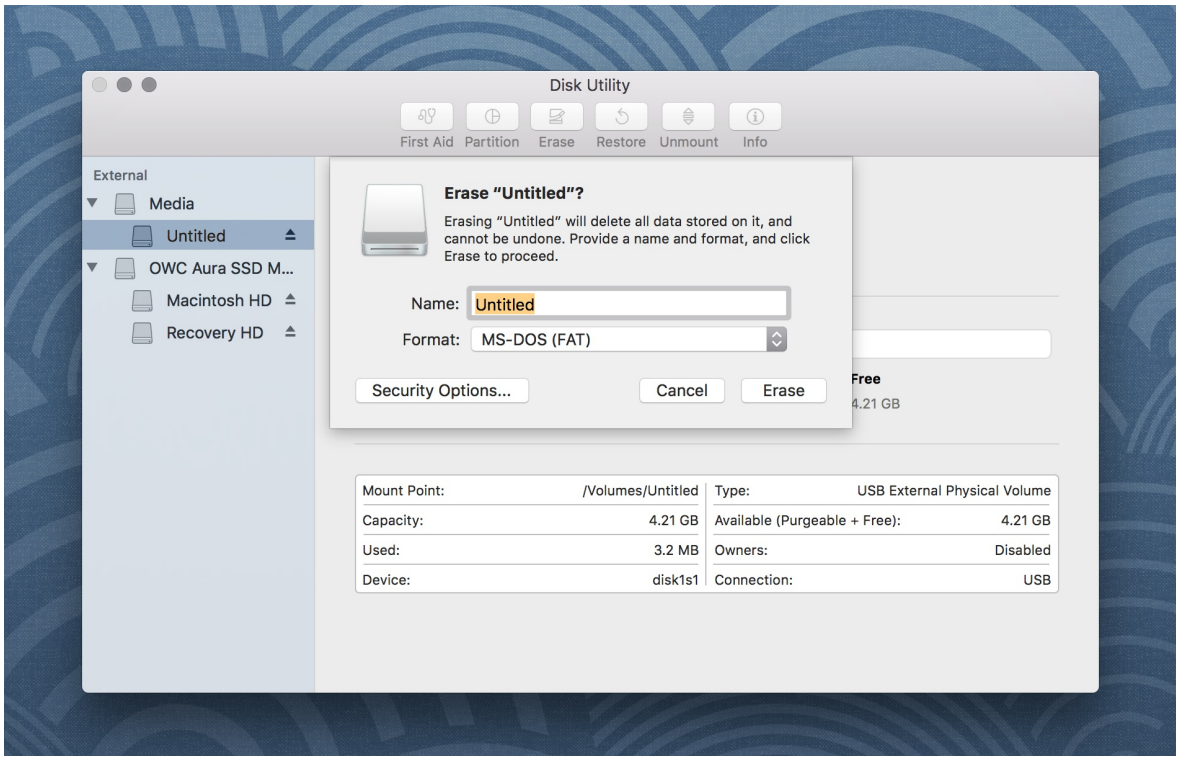


Figure: Diskutil

4.5.2.3 Etcher configuration

Next we use Etcher to configure and write to your USB device as follows (copied form the Ubuntu Web page):

- Select image will open a file requester from which should navigate to and select the ISO file downloaded previously. By default, the ISO file will be in your Downloads folder.
- Select drive, replaced by the name of your USB device if one is already attached, lets you select your target device. You will be warned if the storage space is too small for your selected ISO.
- Flash! will activate when both the image and the drive have been selected. As with Disk Utility, Etcher needs low-level access to your storage hardware and will ask for your password after selection.

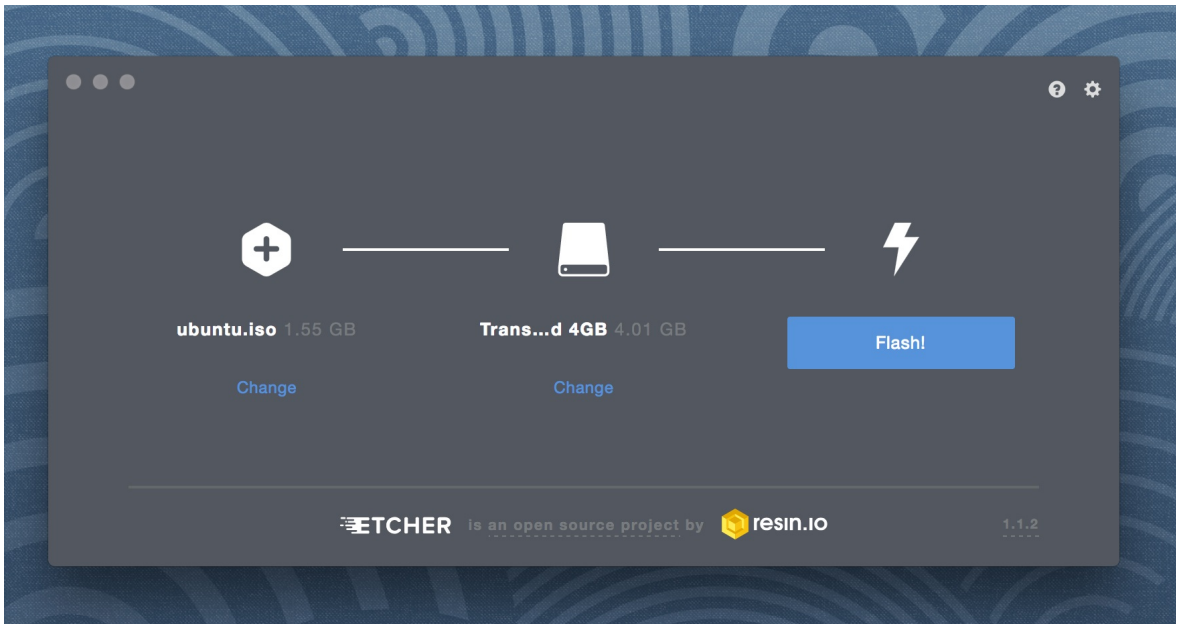


Figure: Etcher complete message

4.5.2.4 Write to the USB stick

When writing to the USB, Etcher will ask you for your password. It will write the ISO file, once you confirmed the password.

You will see the progress reported to the Etcher window. Once it has finished, Etcher will report on the successful process.

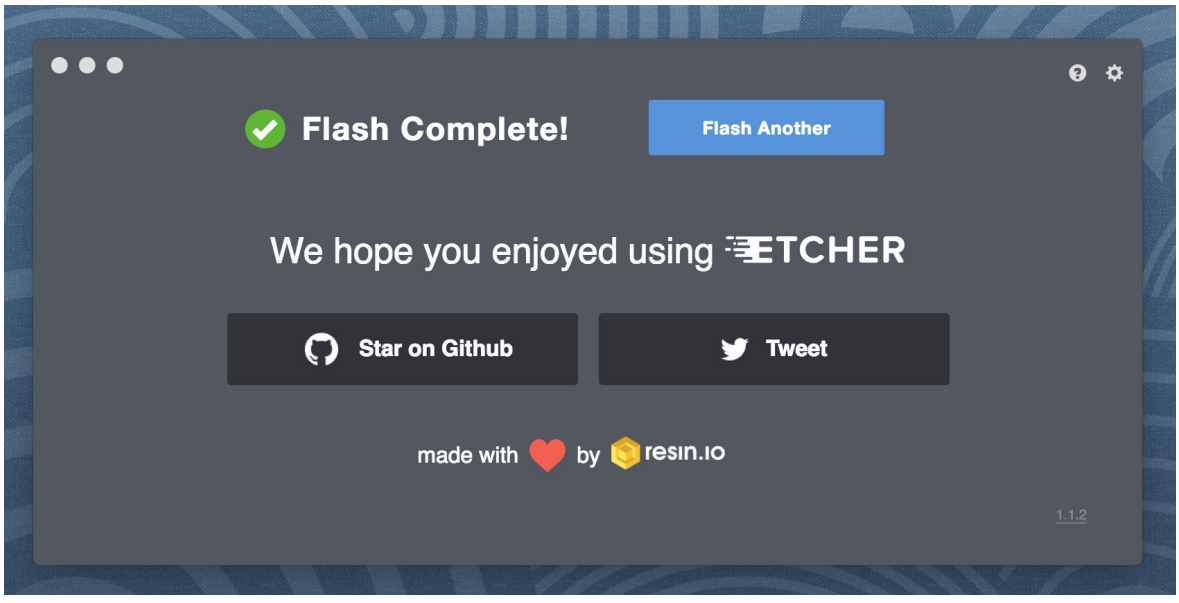


Figure: Etcher

After the write process has completed, macOS may inform you that *The disk you inserted was not readable by this computer*. Do not select Initialise. Instead, select Eject and remove the USB device.

4.5.3 Ubuntu on an USB stick for Windows 10

See exercise Development.Server.1

Material for this directions were taken from a detailed tutorial [link]

First you will need to install Rufus, which is a free program to create bootable USB drives on windows. Rufus is available at

- https://rufus.akeo.ie/

Next you need to launch Rufus, insert the USB stick, and observe that it is added to Rufus. Select the Device on which you like to place ubuntu. Be careful that you do not bya accident use a wrong device.

Select the partition scheme and target system type set as MBR partition scheme for UEFI. (in case you have older hardware try MBR Partition Scheme for BIOS or UEFI instead).

Select the ubuntu iso file.

Next press the `Start` button so we activate the write process. This will take quite a while. Select `Write in ISO Image mode (Recommended)`

Once the process is completed, try booting from it. How to activate the boot in your system depends on your hardware and vendor. Please consult with your documentation.

4.5.4 Exercise

Development.Server.1

> *If you are in need to but from a USB stick in Windows, please verify and expand on our section similar to the one provided by macOS. It does not matter if you chose a GUI or a commandline option via*

gitbash.

4.6 GITHUB

4.6.1 Github

Learning Objectives

- Be able to use the github cloud sevices to collaborately develop contents and programs.
- Be able to use github as part of an open source project.

In some classes the material may be openly shared in code repositories. This includes class material, papers and project. Hence, we need some mechanism to share content with a large number of students.

First, we like to introduce you to git and github.com (Section 1.1). Next, we provide you with the basic commands to interact with git from the commandline (Section 1.12). Than we will introduce you how you can contribute to this set of documentations with pull requests.

4.6.1.1 Overview

Github is a code repository that allows the development of code and documents with many contributors in a distributed fashion. There are many good tutorials about github. Some of them can be found on the github Web page. An interactive tutorial is for example available at

- https://try.github.io/

However, although these tutorials are helpful in many cases they do not address some cases. For example, you have already a repository set up by your organization and you do not have to completely initialize it. Thus do not just replicate the commands in the tutorial, or the once we present here before not evaluating their consequences. In general make sure you verify if the command

does what you expect **before** you execute it.

A more extensive list of tutorials can be found at

- https://help.github.com/articles/what-are-other-good-resources-for-learning-git-and-github

The github foundation has a number of excellent videos about git. If you are unfamiliar with git and you like to watch videos in addition to reading the documentation we recommend these videos

- https://www.youtube.com/user/GitHubGuides/videos

Next, we introduce some important concepts used in github.

4.6.1.2 Upload Key

Before you can work with a repository in an easy fashion you need to upload a public key in order to access your repository. Naturally, you need to generate a key first which is explained in the section about ssh key generation (TODO: lessons-ssh-generate-key include link) before you upload one. Copy the contents of your `.ssh/id_rsa.pub` file and add them to your github keys.

More information on this topic can be found on the github Web page.

4.6.1.3 Fork

Forking is the first step to contributing to projects on GitHub. Forking allows you to copy a repository and work on it under your own account. Next, creating a branch, making some changes, and offering a pull request to the original repository, rounds out your contribution to the open source project.

Git 1:41 Fork

4.6.1.4 Rebase

When you start editing your project, you diverge from the original version. During your developing, the original version may be updated, or other developers may have some of their branches implementing good features that you would like to include in your current work. That is when *Rebase* becomes useful. When you *Rebase* to certain points, could be a newer Master or other custom branch, consider you graft all your on-going work right to that point.

Rebase may fail, because some times it is impossible to achieve what we just described as conflicts may exist. For example, you and the to-be-rebased copy both edited some common text section. Once this happens, human intervention needs to take place to resolve the conflict.

 Git 4:20 Rebase

4.6.1.5 Remote

Collaborating with others involves managing the remote repositories and pushing and pulling data to and from them when you need to share work. Managing remote repositories includes knowing how to add remote repositories, remove remotes that are no longer valid, manage various remote branches and define them as being tracked or not, and more.

Though out this semester, you will typically work on two *remote* repos. One is the office class repo, and another is the repo you forked from the class repo. The class repo is used as the centralized, authority and final version of all student submissions. The repo under your own Github account is for your personal storage. To show progress on a weekly basis you need to commit your changes on a weekly basis. However make sure that things in the master branch are working. If not, just use another branch to conduct your changes and merge at a later time. We like you to call your development branch dev.

- https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes

4.6.1.6 Pull Request

Pull requests are a means of starting a conversation about a proposed change back into a project. We will be taking a look at the strength of conversation,

integration options for fuller information about a change, and cleanup strategy for when a pull request is finished.

 Git 4:26 Pull Request

4.6.1.7 Branch

Branches are an excellent way to not only work safely on features or experiments, but they are also the key element in creating Pull Requests on GitHub. Lets take a look at why we want branches, how to create and delete branches, and how to switch branches in this episode.

 Git 2:25 Branch

4.6.1.8 Checkout

Change where and what you are working on with the checkout command. Whether we are switching branches, wanting to look at the working tree at a specific commit in history, or discarding edits we want to throw away, all of these can be done with the checkout command.

 Git 3:11 Checkout

4.6.1.9 Merge

Once you know branches, merging that work into master is the natural next step. Find out how to merge branches, identify and clean up merge conflicts or avoid conflicts until a later date. Lastly, we will look at combining the merged feature branch into a single commit and cleaning up your feature branch after merges.

 Git 3:11 Merge

4.6.1.10 GUI

Using Graphical User Interfaces can supplement your use of the command line

to get the best of both worlds. GitHub for Windows and GitHub for Mac allow for switching to command line, ease of grabbing repositories from GitHub, and participating in a particular pull request. We will also see the auto-updating functionality helps us stay up to date with stable versions of Git on the command line.

 Git 3:47 GUI

There are many other 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, eclipse, or other tools that have build in git support. You can identify a tool that works best for you.

4.6.1.11 Windows

This is a quick tour of GitHub for Windows. It offers GitHub newcomers a brief overview of what this feature-loaded version control tool and an equally powerful web application can do for developers, designers, and managers using Windows in both the open source and commercial software worlds. More: http://windows.github.com

 Git 1:25 Windows

4.6.1.12 Git from the Commandline

Although github.com provides a powerful GUI and other GUI tools are available to interface with github.com, the use of git from the commandline can often be faster and in many cases may be simpler.

Git commandline tools can be easily installed on a variety of operating systems including Linux, macOS, and Windows. Many great tutorials exist that will allow you to complete this task easily. We found the following two tutorials sufficient to get the task accomplished:

- https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

- https://www.atlassian.com/git/tutorials/install-git

Although the later is provided by an alternate repository to github. The installation instructions are very nice and are not impacted by it. Once you have installed git you need to configure it.

4.6.1.13 Configuration

Once you installed Git, you can need to configure it properly. This includes setting up your username, email address, line endings, and color, along with the settings' associated configuration scopes.

 Git 2:47 Configuration

It is important that make sure that use the `git config` command to initialize git for the first time on each new computer system or virtual machine you use. This will ensure that you use on all resources the same name and e-mail so that git history and log will show consistently your checkins across all devices and computers you use. If you do not do this, your checkins in git do not show up in a consistent fashion as a single user. Thus on each computer execute the following commands:

```
$ git config --global user.name "Albert Zweistein"
$ git config --global user.email albert.zweistein@gmail.com
```

where you replace the information with the information related to you. You can set the editor to emacs with:

```
$ git config --global core.editor emacs
```

Naturally if you happen to want to use other editors you can configure them by specifying the command that starts them up. You will also need to decide if you want to push branches individually or all branches at the same time. It will be up to you to make what will work for you best. We found that the following seems to work best:

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

More information about a first time setup is documented at:

```
* http://git-scm.com/book/en/Getting-Started-First-Time-Git-Setup
```

To check your setup you can say:

```
$ git config --list
```

One problem we observed is that students often simply copy and paste instructions, but do not read carefully the error that is reported back and do not fix it. Overlooking the proper set of the push.default is often overlooked. Thus we remind you: **Please read the information on the screen when you set up**.

4.6.1.14 Upload your public key

Please upload your public key to the repository as documented in github, while going to your account and find it in settings. There you will find a panel SSH key that you can click on which brings you to the window allowing you to add a new key. If you have difficulties with this find a video from the github foundation that explains this.

4.6.1.15 Working with a directory that will be provided for you

In case your course provided you with a github directory, starting and working in it is going to be real simple. Please wait till an announcement to the class is send before you ask us questions about it.

If you are the only student working on this you still need to make sure that papers or programs you manage in the repository work and do not interfere with scripts that instructors may use to check your assignments. Thus it is god to still create a branch, work in the branch and than merge the branch into the master once you verified things work. After you merged you can push the content to the github repository.

Tip: Please use only **lowercase** characters in the directory names and no special characters such as `@ ; / _` and spaces. In general we recommend that you avoid using directory names with capital letters spaces and _ in them. This will simplify your documentation efforts and make the URLs from git more readable. Also while on some OS's the directories *MyDirectory* is different from *mydirectory* on macOS it is considered the same and thus renaming from capital to lower case can not be done without first renaming it to another directory.

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
```

4.6.1.16 README.yml and notebook.md

In case you take classes e516 and e616 with us you will have to create a README.yaml and notebook.md file in the top most directory of your repository. It serves the purpose of identifying your submission for homework and information about yourself.

It is important to follow the format precisely. As it is yaml it is an easy homework to write a 4 line python script that validates if the README.yaml file is valid. In addition you can use programs such as `yamllint` which is documented at

- https://yamllint.readthedocs.io/en/latest/

This file is used to integrate your assignments into a proceedings. An example is provided at

- https://github.com/cloudmesh-community/hid-

sample/blob/master/README.yml

Any derivation from this format will not allow us to see your homework as our automated scripts will use the README.yml to detect them. Make sure the file does not contain ay TABs. Please also mind that all filenames of all homework and the main directory must be **lowercase** and do not include spaces. This will simplify your task of managing the files across different operating systems.

In case you work in a team, on a submission, the document will only be submitted in the author and hid that is listed first. All other readme files, will have for that particular artifact a `duplicate: yes` entry to indicate that this submission is managed elsewhere. The team will be responsible to manage their own pull requests, but if the team desires we can grant access for all members to a repository by a user. Please be aware that you must make sure you coordinate with your team.

We will not accept submission of homework as pdf documents or tar files. All assignments must be submitted as code and the reports in native latex and in github. We have a script that will automatically create the PDF and include it in a proceedings. There is no exception from this rule and all reports not compilable will be returned without review and if not submitted within the deadline receive a penalty.

Please check with your instructor on the format of the README.yaml file as it could be different for your class.

To see an example for the notebook.md file, you can visit our sample hid, and browse to the notebook.md file. Alternatively you can visit the following link

- https://github.com/cloudmesh-community/hid-sample/blob/master/notebook.md

The purpose of the notebook md file is to record what you did in the class to us. We will use this file at the end of the class to make sure you have recorded on a weekly basis what you did for the class. Inactivity is a valid response. Not updating the notebook, is not.

The sample directory contains other useful directories and samples, that you may

want to investigate in more detail. One of the most important samples is the github issues (see Section 1.19). There is even a video in that section about this and showcases you how to organize your tasks within this class, while copying the assignments from piazza into one or more github issues. As we are about cloud computing, using the services offered by a prominent cloud computing service such as github is part of the learning experience of this course.

4.6.1.17 Contributing to the Document

It is relatively easy to contribute to the document if you understand how to use github. The first thing you will need to do is to create a fork of the repository. The easiest way to do this is to visit the URL

- https://github.com/cloudmesh-community/book

Towards the upper right corner you will find a link called **Fork**. Click on it and chose into which account you like to fork the original repository. Next you will create a colne from your corked directory. You will see in your fork a green clone button. You will see a URL that you can copy into your terminal. If the links does not include your username, it is the wrong link.

In your terminal you now say

```
git colne  https://github.com/<yourusername>/book
```

Now cd into this directory and make your changes.

```
$ cd book
```

Use the usual git commands such as `git add`, `git commit`, `git push`

Note you will push into your local directory.

4.6.1.17.1 Stay up to date with the original repo

Form time to time you will see that others are contributing to the original repo. To stay up to date you want to not only sync from your local copy, but also from the original repo. To link your repo with what is called the upstream you need to do the following once, so you can issue `git pull` tha also pulls from the upstream

Make sure you have upstream repo defined:

```
$ git remote add upstream \
      https://github.com/cloudmesh-community/book
```

Now Get latest from upstream:

```
$ git rebase upstream/master
```

In this step, the conflicting file shows up (in my case it was refs.bib):

```
$ git status
```

should show the name of the conflicting file:

```
$ git diff <file name>
```

should show the actual differences. May be in some cases, It is easy to simply take latest version from upstream and reapply your changes.

So you can decide to checkout one version earlier of the specific file. At this stage, the re-base should be complete. So, you need to commit and push the changes to your fork:

```
$ git commit
$ git rebase origin/master
$ git push
```

Then reapply your changes to refs.bib - simply use the backed up version and use the editor to redo the changes.

At this stage, only refs.bib is changed:

```
$ git status
```

should show the changes only in refs.bib. Commit this change using:

```
$ git commit -a -m "new:usr: <message>"
```

And finally push the last committed change:

```
$ git push
```

The changes in the file to resolve merge conflict automatically goes to the original pull request and the pull request can be merged automatically.

You still have to issue the pull request from the Github Web page so it is registered with the upstream repository.

4.6.1.17.2 Resources

- 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 github, for gitlab.
- Atlassian tutorials for git

In addition the tutorials from atlassian are a good source. However remember that you may not use bitbucket as the repository, so ignore those tutorials. We found the following useful

- What is git: https://www.atlassian.com/git/tutorials/what-is-git
- Installing git: https://www.atlassian.com/git/tutorials/install-git
- git config: https://www.atlassian.com/git/tutorials/setting-up-a-repository#git-config
- git clone: https://www.atlassian.com/git/tutorials/setting-up-a-repository#git-clone
- saving changes: https://www.atlassian.com/git/tutorials/saving-changes
- collaborating with git: https://www.atlassian.com/git/tutorials/syncing

4.6.1.18 Exercises

E.Github.1:

> *How do you set your favorite editor as a default with github config*

E.Github.2:

> *What is the differencebetween merge and rebase?*

E.Github.3:

> *Assume you have made a change in your local fork, however other users have since committed to the master branch, how can you make sure your commit works off from the latest information in the master branch?*

E.Github.4:

> *Find a spelling error in the Web page or a contribution and create a pull request for it.*

E.Gitlab.5:

> *Create a README.yml in your github account directory provided for you for class.*

4.6.1.19 Github Issues

 Github 8:29 Issues

When we work in teams or even if we work by ourselves, it is prudent to identify a system to coordinate your work. While conduction projects that use a variety of cloud services, it is important to have a system that enables us to have a cloud service that enables us to facilitate this coordination. Github provides such a feature through its *issue* service that is embedded in each repository.

Issues allow for the coordination of tasks, enhancements, bugs, as well as self defined labeled activities. Issues are shared within your team that has access to your repository. Furthermore, in an open source project the issues are visible to the community, allowing to easily communicate the status, as well as a roadmap to new features.

This enables the community to participate also in reporting of bugs. Using such a system transforms the development of software from the traditional closed shop development toa truly open source development encouraging contributions from others. Furthermore it is also used as bug tracker in which not only you, but the community can communicate bugs to the project.

A good resource for learning more about issues is provided at

- https://guides.github.com/features/issues/

4.6.1.19.1 Git Issue Features

A git issue has the following features:

title

: – a short description of what the issue is about

description

: a more detailed description. Descriptions allow also to conveniently add check-boxed todo’s.

label

: a color enhanced label that can be used to easily categorize the issue. You can define your own labels.

milestone

: a milestone so you can identify categorical groups issues as well as their due date. You can for example group all tasks for a week in a milestone, or you could for example put all tasks for a topic such as developing a paper in a milestone and provide a deadline for it.

assignee

: an assignee is the person that is responsible for making sure the task is executed or on track if a team works on it. Often projects allow only one assignee, but in certain cases it is useful to assign a group, and the group identifies if the task can be split up and assigns them through check-boxed todo’s.

comments

allow anyone with access to provide feedback via comments.

4.6.1.19.2 Github Markdown

Github uses markdown which we introduce you in Section [S:markdown].

As github has its own flavor of markdown we however also point you to

as a reference. We like to mention the special enhancements fo github's markdown that integrate well to support project management.

4.6.1.19.2.1 Task lists

Taks lists can be added to any description or comment in github issues To create a task list you can add to any item `[ ]`. This includes a task to be done. To make it as complete simple change it to `[x]`. Whoever the great feature of tasks is that you do not even have to open the editor but you can simply check the task on and of via a mouse click. An example of a task list could be

```
Post Bios

* [x] Post bio on piazza
* [ ] Post bio on google docs
* [ ] Post bio on github
* [ ] \(optional) integrate image in google docs bio
```

In case you need to use a `(`have at the beginning ot the task text, you need to escape it with a `\`

4.6.1.19.2.2 Team integration

A person or team on GitHub can be mentioned by typing the username proceeded by the `@` sign. When posting the text in the issue, it will trigger a notification to them and allow them to react to it. It is even possible to notify entire teams, which are described in more detail at

- https://help.github.com/articles/about-teams/

4.6.1.19.2.3 Referencing Issues and Pull requests

Each issue has a number. If you use the `#` followed by the issue number you can refer to it in the text which will also automatically include a hyperlink to the

task. The same is valid for pull requests.

4.6.1.19.2.4 Emojis

Although github supports emojis such as `:+1:` we do not use them typically in our class.

4.6.1.19.3 Notifications

Github allows you to set preferences on how you lik to receive notifications. You can receive them either via e-mail or the Web. This is controlled by configuring it in *your settings*, where you can set the preferences for participating projects as well as projects you decide to watch. To access the notifications you can simply look at them in the *notification* screen. In this screen when you press the `?` you will see a number of commands that allow you to control the notification when pressing on one of them.

4.6.1.19.4 cc

To carbon copy users in your issue text, simply use `/cc` followed by the `@` sign and their github user name.

4.6.1.19.5 Interacting with issues

Github has the ability to search issues with a search query and a search language that you can find out more about it at

https://guides.github.com/features/issues/#search

A dashboard gives convenient overviews of the issues including a *pulse* that lists todo's status if you use them in the issue description.

4.6.1.20 Glossary

The Glossary is copied from

- https://cdcvs.fnal.gov/redmine/projects/cet-is-public/wiki/GitTipsAndTricks#A-suggested-work-flow-for-distributed-

projects-NoSY

Add
: put a file (or particular changes thereto) into the index ready for a commit operation. Optional for modifications to tracked files; mandatory for hitherto un-tracked files.

Branch
: a divergent change tree (eg a patch branch) which can me merged either wholesale or piecemeal with the master tree.

Commit
: save the current state of the index and/or other specified files to the local repository.

Commit object
: an object which contains the information about a particular revision, such as parents, committer, author, date and the tree object which corresponds to the top directory of the stored revision.

Fast-forward
: an update operation consisting only of the application of a linear part of the change tree in sequence.

Fetch
: update your local repository database (not your working area) with the latest changes from a remote.

HEAD
: the latest state of the current branch.

Index
: a collection of files with stat information, whose contents are stored as objects. The index is a stored version of your working tree. Files may be staged to an index prior to committing.

Master
: the main branch: known as the trunk in other SCM systems.

Merge
: join two trees. A commit is made if this is not a fast-forward operations (or one is requested explicitly.

Object
: the unit of storage in git. It is uniquely identified by the SHA1 hash of its contents. Consequently, an object can not be changed.

Origin

the default remote, usually the source for the clone operation that created the local repository.

Pull
: shorthand for a fetch followed by a merge (or rebase if –rebase option is used).

Push
: transfer the state of the current branch to a remote tracking branch. This must be a fast-forward operation (see merge).

Rebase
: a merge-like operation in which the change tree is rewritten (see Rebasing below). Used to turn non-trivial merges into fast-forward operations.

Remote
: another repository known to this one. If the local repository was created with “clone” then there is at least one remote, usually called, “origin.”

Stage
: to add a file or selected changes therefrom to the index in preparation for a commit.

Stash
: a stack onto which the current set of uncommitted changes can be put (eg in order to switch to or synchronize with another branch) as a patch for retrieval later. Also the act of putting changes onto this stack.

Tag
: human-readable label for a particular state of the tree. Tags may be simple (in which case they are actually branches) or annotated (analogous to a CVS tag), with an associated SHA1 hash and message. Annotated tags are preferable in general.

Tracking branch
: a branch on a remote which is the default source / sink for pull / push operations respectively for the current branch. For instance, origin/master is the tracking branch for the local master in a local repository.

Un-tracked
: not known currently to git.

4.6.1.21 Example commands

To work in your local directory you can use the following commands. Please note that these commands do not upload your work to github, but only introduce

version control within your local files.

The command list is copied from

- https://cdcvs.fnal.gov/redmine/projects/cet-is-public/wiki/GitTipsAndTricks#A-suggested-work-flow-for-distributed-projects-NoSY

4.6.1.21.1 Local commands to version contril your files

Obtain differences with

```
$ git status
```

Move files from one part of your directory tree to another:

```
$ git mv <old-path> <new-path>
```

Delete unwanted tracked files:

```
$ git rm <path>
```

Add un-tracked files:

```
$ git add <un-tracked-file>
```

Stage a modified file for commit:

```
$ git add <file>
```

Commit currently-staged files:

```
$ git commit -m <log-message>
```

Commit only specific files (regardless of what is staged):

```
$ git commit -m <log-message>
```

Commit all modified files:

```
$ git commit -a -m <log-message>
```

Un-stage a previously staged (but not yet committed) file:

```
$ git reset HEAD <file>
```

Get differences with respect to the committed (or staged) version of a file:

```
$ git diff <file>
```

Get differences between local file and committed version:

```
$ git diff --cached <file>
```

Create (but do not switch to) a new local branch based on the current branch:

```
$ git branch <new-branch>
```

Change to an existing local branch:

```
$ git checkout <branch>
```

Merge another branch into the current one:

```
$ git merge <branch>
```

4.6.1.21.2 Interacting with the remote

Get the current list of remotes (including URIs) with

```
$ git remote -v
```

Get the current list of defined branches with

```
$ git branch -a
```

Change to (creating if necessary) a local branch tracking an existing remote branch of the same name:

```
$ git checkout <branch>
```

Update your local repository ref database without altering the current working area:

```
$ git fetch <remote>
```

Update your current local branch with respect to your repository's current idea of a remote branch's status:

```
$ git merge <branch>
```

Pull remote ref information from all remotes and merge local branches with their

remote tracking branches (if applicable):

```
$ git pull
```

Examine changes to the current local branch with respect to its tracking branch:

```
$ git cherry -v
```

Push changes to the remote tracking branch:

```
$ git push
```

Push all changes to all tracking branches:

```
$ git push --all
```

4.6.2 Git Pull Request

4.6.2.1 Introduction

Git pull requests allow developers to submit work or changes they have done to a repository, The developers can then check the changes that have been proposed in the pull request, discuss and make changes if needed. After the content off the pull request has been agreed upon it can be merged to the repository to add the information or changes in the pull request into the repository.

4.6.2.2 How to create a pull request

In this document we will see how we can create a pull request for the Cloudmesh technologies repo that is located at

- https://github.com/cloudmesh/technologies

However if you do pull request on other directories, you just have to replace the url with that of the repository you like to use. A common one four our classes is also

- https://github.com/cloudmesh-community/book

Which contains this book.

You can either create a pull request through a branch or through a fork. In this document we will be looking at how we can create a pull request through a fork.

4.6.2.3 Fork the original repository

First you need to create a fork of the original repository. A fork is your own copy of the repository to which you can make changes to. To fork the Cloudmesh technologies goto Cloudmesh technologies repo and click on the Fork button on the top right corner. Now you can notice that instead of `cloudmesh/technologies` the name of the repo says `YOURGITUSERNAME/technologies`, where `YOURGITUSERANAME` is indeed your github user name. That is because you are now in your own copy of the `cloudmesh/technologies` repository. n our case the user name will be `pulashti`.

4.6.2.4 Clone your copy

Now that you have your fork created, we can go ahead and clone it into our machine. Instructions on how to clone a repository can be found in the Github documentation - Cloning a repository. Make sure that you clone your version of the technologies repo.

4.6.2.5 Adding an upstream

Before we can start working on our copy of the git repo it is good to add an upstream (a link to the original repo) so that we can get all the latest changes in the original repository into our copy. Use the following commands to add an upstream to `cloudmesh/technologies`. First go into the folder which contains your git repo that you cloned and execute the following command.

```
$ git remote add upstream https://github.com/cloudmesh/technologies.git'
```

To make sure you have added it correctly execute the following command

```
$ git remote -v
```

You should see something similar to the following as the output

```
origin  https://github.com/pulasthi/technologies.git (fetch)
origin  https://github.com/pulasthi/technologies.git (push)
upstream    https://github.com/cloudmesh/technologies.git (fetch)
upstream    https://github.com/cloudmesh/technologies.git (push)
```

4.6.2.6 Making changes

Now you can make changes to your repo as with any normal git repository. However to make sure you have the latest copy from the original execute the following command before you start making changes. This will pull the latest changes from the original `cloudmesh/technologies` into your local copy

```
$ git pull upstream master
```

Now make the needed changes commit and push, the changes will be pushed to your copy of the repo i Github, not the `cloudmesh/technologies` repo.

4.6.2.7 Creating a pull request

Once we have changes pushed, you can go into your repository in Github to create a pull request. As seen in @#fig:button-pullrequest, you have an button named `Pull request`

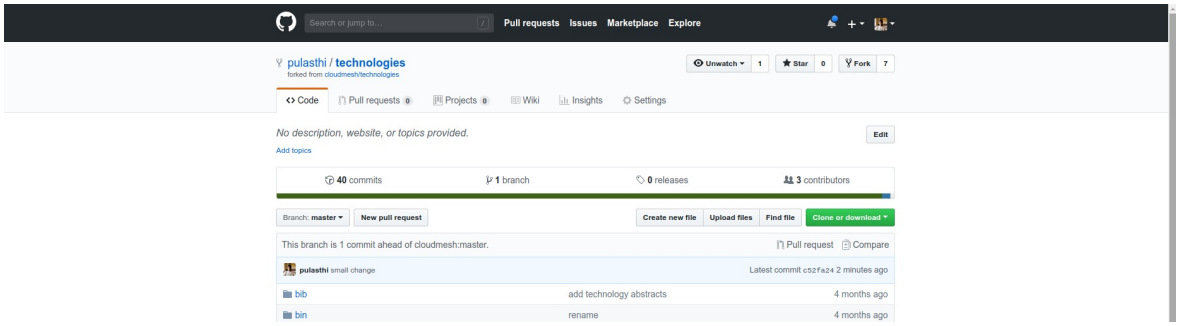


Figure 2: Button Pull request

Once you click on that button you will be taken to a page to create the pull request, which will look similar to Figure 3.

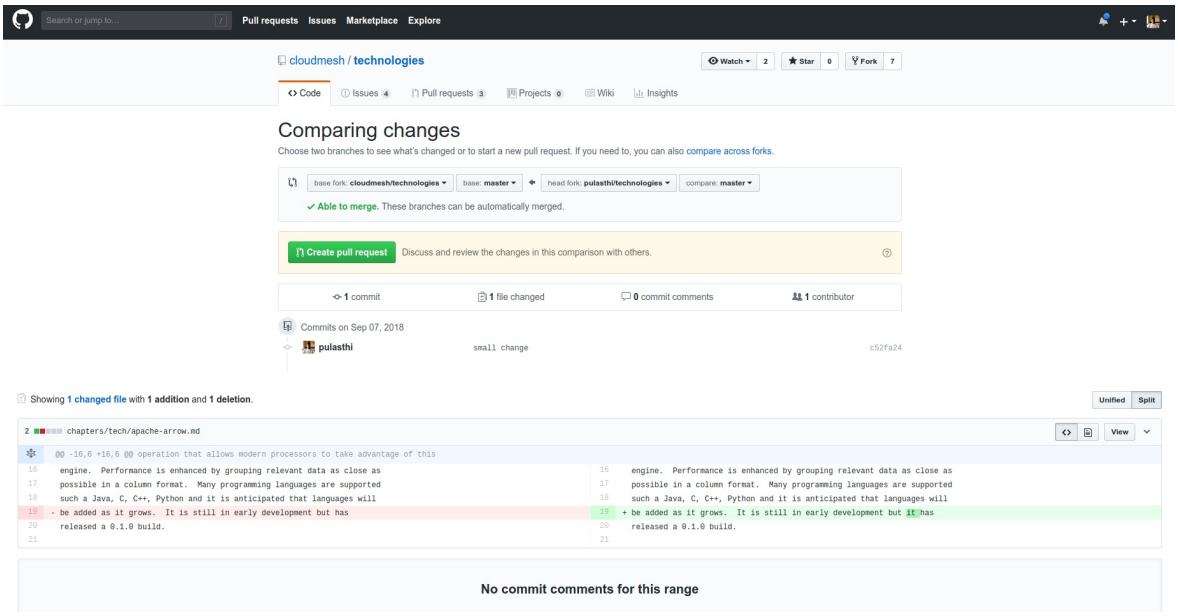


Figure 3: Create a pull request

Once you click on the `Create pull request` button you will be given an option to add a title and a comment for the pull request. Once you complete the details and submit the pull request will appear in the original `cloudmesh/technologies` repo.

Note: Make sure you see the `Able to merge` sign before you submit the pull request, otherwise your pull will not be able to directly merged to the original repo. If you do not see this that means you have not properly done the `git pull upstream master` command before you made the changes

 git example on CL 10:09

4.6.3 Tig

Many browsers exist to gain insight into git repositories. In case you have Linux or Ubuntu a tool to display information in a terminal is available.

- https://jonas.github.io/tig/

On OSX it can be installed with:

```
$ brew install tig
```

Tig has many different views including views for main, log, diff, tree, blob,

blame, refs, status, stage. stash, grep, and pager .

A screenshot shows some if its basic functionality is shown in Figure 4

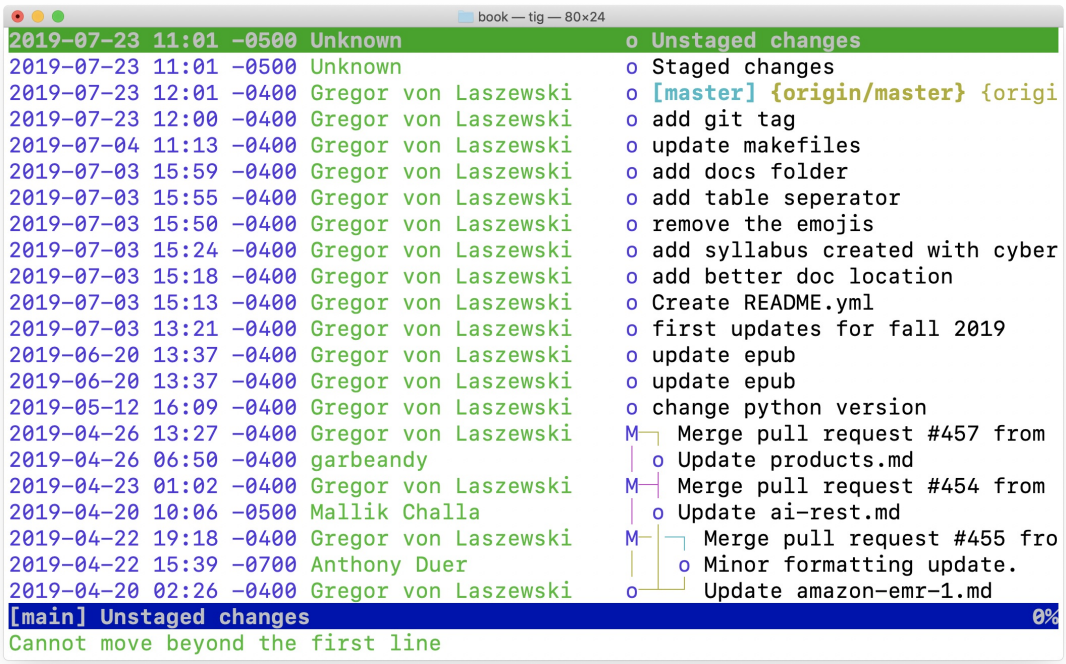


Figure 4: Git tig main vie

Example infocations are

```
$ tig
$ git show | tig
$ git log | tig
```

4.7 LINUX

4.7.1 Linux

Learning Objectives

- Be able to know the basic commands to work in a Linux terminal.
- Get familiar with Linux Commands

Now that you have Linux or a Linux like environment (such as `gitbash`) on your computer it is time to learn a number of useful commands to interact with the system.

In order for this task to enhance your knowledge you are encouraged to find additional material and are required to complete the table of useful Linux commands. You will do this as team and create pull requests improving and completing this documentation. The TAs will provide a mapping between students and commands to be documented. If you find additional commands that aught to be listed here, please add.

4.7.1.1 History

LINUX is a reimplementation by the community of UNIX which was developed in 1969 by Ken Thompson and Dennis Ritchie of Bell Laboratories and rewritten in C. An important part of UNIX is what is called the *kernel* which allows the software to talk to the hardware and utilize it.

In 1991 Linus Torvalds started developing a Linux Kernel that was initially targeted for PC's. This made it possible to run it on Laptops and was later on further developed by making it a full Operating system replacement for UNIX.

4.7.1.2 Shell

One of the most important features for us will be to access the computer with the help of a *shell*. The shell is typically run in what is called a terminal and allows interaction to the computer with commandline programs.

There are many good tutorials out there that explain why one needs a linux shell and not just a GUI. Randomly we picked the first one that came up with a google query. This is not an endorsement for the material we point to, but could be a worth while read for someone that has no experience in Shell programming:

http://linuxcommand.org/lc3_learning_the_shell.php

Certainly you are welcome to use other resources that may suite you best. We will however summarize in table form a number of useful commands that you may als find even as a RefCard.

http://www.cheat-sheets.org/#Linux

We provide in the next table a number of useful commands that you want to

explore. For more information simply type man and the name of the command.

.

Command	**Description**
man *command*	manual page for the *command*
apropos *text*	list all commands that have text in it
ls	Directory listing
ls -lisa	list details
tree	list the directories in graphical form
cd *dirname*	Change directory to *dirname*
mkdir *dirname*	create the directory
rmdir *dirname*	delete 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*
cat -n *filename*	print content of file*a* with
	line numbers
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*
du -hs .	show in human readable form the space used by the current directory
df -h	show the details of the disk file system
wc *filename*	counts the word in a file
sort *filename*	sorts the file
uniq *filename*	displays only uniq entries in the file
	tars up a compressed version of the

tar -xvf *dir*	directory
rsync	faster, flexible replacement for rcp
gzip *filename*	compresses the file
gunzip *filename*	compresses the file
bzip2 *filename*	compresses the file with
	block-sorting
bunzip2 *filename*	uncompresses the file with block-sorting
clear	clears the terminal screen
touch *filename*	change file access and modification times or if file does not exist creates file
who	displays a list of users that are currently logged on, for each user the login name, date and time of login, tty name, and hostname if not local are displayed
whoami	displays the users effective id see also id
echo -n *string*	write specified arguments to standard output
date	displays or sets date & time, when invoked without arguments the current date and time are displayed
logout	exit a given session
exit	when issued at the shell prompt the shell will exit and terminate any running jobs within the shell
kill	terminate or signal a process by sending a signal to the specified process usually by the pid
	displays a header line followed by

ps	all processes that have controlling terminals
sleep	suspends execution for an interval of time specified in seconds
uptime	displays how long the system has been running
time *command*	times the command execution in seconds
find */ [-name] file-name.txt*	searches a specified path or directory with a given expression that tells the find utility what to find, if used as shown the find utility would search the entire drive for a file named file-name.txt
diff	compares files line by line
hostname	prints the name of the current host system
which	locates a program file in the users path
tail	displays the last part of the file
head	displays the first lines of a file
top	displays a sorted list of system processes
locate *filename*	finds the path of a file
grep *'word' filename*	finds all lines with the word in it
grep -v *'word' filename*	finds all lines without the word in it
chmod ug+rw *filename*	change file modes or Access Control Lists. In this example user and group are changed to read and write
chown	change file owner and group
history	a build-in command to list the past commands
sudo	execute a command as another user

su	substitute user identity
uname	print the operating system name
set -o emacs	tells the shell to use Emacs commands.
chmod go-rwx *file*	changes the permission of the file
chown *username file*	changes the ownership of the file
chgrp *group file*	changes the group of a file
fgrep *text filename*	searches the text in the given file
grep -R *text* .	recursively searches for xyz in all files
find . -name *.py	find all files with `.py` at the end
ps	list the running processes
kill -9 1234	kill the process with the id 1234
at	que commands for later execution
cron	daemon to execute scheduled commands
crontab	manage the time table for execution commands with cron
mount /dev/cdrom /mnt/cdrom	mount a filesystem from a cd rom to /mnt/cdrom
users	list the logged in users
who	display who is logged in
whoami	print the user id
dmesg	display the system message buffer
last	indicate last logins of users and ttys
uname	print operating system name
date	prints the current date and time
time *command*	prints the sys, real and user time
shutdown -h “shut down”	shutdown the computer
ping	ping a host

netstat	show network status
hostname	print name of current host system
traceroute	print the route packets take to network host
ifconfig	configure network interface parameters
host	DNS lookup utility
whois	Internet domain name and network number directory service
dig	DNS lookup utility
wget	non-interactive network downloader
curl	transfer a URL
ssh	remote login program
scp	remote file copy program
sftp	secure file transfer program
watch *command*	run any designated command at regular intervals
awk	program that you can use to select particular records in a file and perform operations on them
sed	stream editor used to perform basic text transformations
xargs	program that can be used to build and execute commands from STDIN
cat *some_file.json* | python -m json.tool	quick and easy JSON validator

4.7.1.3 Multi-command execution

One of the important features is that one can execute multiple commands in the shell.

To execute command 2 once command 1 has finished use

```
command1; command2
```

To execute command 2 as soon as command 1 forwards output to stdout use

```
command1; command2
```

To execute command 1 in the background use

```
command1 &
```

4.7.1.4 Keyboard Shortcuts

These shortcuts will come in handy. Note that many overlap with emacs short cuts.

.

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 line
Ctrl + e	Go to the end of the line
Ctrl + k	Cut everything after the cursor to a special clipboard
Ctrl + y	Paste from the special clipboard
Ctrl + d	Logout of current session, similar to exit

4.7.1.5 bashrc and bash_profile

Usage of a particular command and all the attributes associated with it, use `man` command. Avoid using `rm -r` command to delete files recursively. A good way to

avoid accidental deletion is to include the following in your `.bash_profile` file:

```
alias e=open_emacs
alias rm='rm -i'
alias mv='mv -i'
alias h='history'
```

More Information

https://cloudmesh.github.io/classes/lesson/linux/refcards.html

4.7.1.6 Makefile

Makefiles allow developers to coordinate the execution of code compilations. This not only includes C or C++ code, but any translation from source to a final format. For us this could include the creation of PDF files from latex sources, creation of docker images, and the creation of cloud services and their deployment through simple workflows represented in makefiles, or the coordination of execution targets.

As makefiles include a simple syntax allowing structural dependencies they can easily adapted to fulfill simple activities to be executed in repeated fashion by developers.

An example of how to use Makefiles for docker is provided at http://jmkhael.io/makefiles-for-your-dockerfiles/.

An example on how to use Makefiles for LaTeX is provided at https://github.com/cloudmesh/book/blob/master/Makefile.

Makefiles include a number of rules that are defined by a target name. Let us define a target called hello that prints out the string “Hello World”.

```
    hello:
        @echo "Hello World"
```

Important to remember is that the commands after a target are not indented just by spaces, but actually by a single TAB character. Editors such as emacs will be ideal to edit such Makefiles, while allowing syntax highlighting and easy manipulation of TABs. Naturally other editors will do that also. Please chose your editor of choice. One of the best features of targets is that they can depend on other targets. Thus, iw we define

```
hallo: hello
    @echo "Hallo World"
```

our makefile will first execute hello and than all commands in hallo. As you can see this can be very useful for defining simple dependencies.

In addition we can define variables in a makefile such as

```
HELLO="Hello World"

hello:
    @echo $(HELLO)
```

and can use them in our text with $ invocations.

Moreover, in sophisticated Makefiles, we could even make the targets dependent on files and a target rules could be defined that only compiles those files that have changed since our last invocation of the Makefile, saving potentially a lot of time. However, for our work here we just use the most elementary makefiles.

For more information we recommend you to find out about it on the internet. A convenient reference card sis available at http://www.cs.jhu.edu/~joanne/unixRC.pdf.

4.7.1.6.1 Makefiles on Windows

Makefiles can easily be accessed also on windows while installing gitbash. Please reed to the internet or search in this handbook for more information about gitbash.

4.7.1.7 chmod

The chmod command stand for *change mode* and changes the access permissions for a given file system object(s). It uses the following syntax: `chmod [options] mode[,mode] file1 [file2…]`. The option parameters modify how the process runs, including what information is outputted to the shell:

Option:	Description:
`-f`, `--silent`, `--quiet`	Forces process to continue even if errors occur
`-v`, `--verbose`	Outputs for every file that is processed

`-c`, `--changes`	Outputs when a file is changed
`--reference=RFile`	Uses RFile instead of Mode values
`-R`, `--recursive`	Make changes to objects in subdirectories as well
`--help`	Show help
`--version`	Show version information

Modes specify which rights to give to which users. Potential users include the user who owns the file, users in the file's Group, other users not in the file's Group, and all, and are abbreviated as `u`, `g`, `o`, and `a` respectively. More than one user can be specified in the same command, such as `chmod –v ug(operator)(permissions) file.txt`. If no user is specified, the command defaults to `a`. Next, a `+` or `-` indicates whether permissions should be added or removed for the selected user(s). The permissions are as follows:

Permission:	**Description:**
`r`	Read
`w`	Write
`x`	Execute file or access directory
`X`	Execute only if the object is a directory
`s`	Set the user or group ID when running
`t`	Restricted deletion flag or sticky mode
`u`	Specifies the permissions the user who owns the file has
`g`	Specifies the permissions of the group
`o`	Specifies the permissions of users not in the group

More than one permission can be also be used in the same command as follows:

```
$ chmod –v o+rw file.txt
```

Multiple files can also be specified:

```
$ chmod a-x,o+r file1.txt file2.txt
```

4.7.1.8 Exercises

E.Linux.1

> *Familiarize yourself with the commands*

E.Linux.2

> *Find more commands that you find useful and add them to this page.*

E.Linux.3

> *Use the sort command to sort all lines of a file while removing duplicates.*

E.Linux.4

> *Should there be other commands listed in the table with the Linux commands If so which? Create a pull request for them.*

E.Linux.5

> *Write a section explaining chmod. Use letters not numbers*

E.Linux.6

> *Write a section explaining chown. Use letters not numbers*

E.Linux.7

> *Write a section explaining su and sudo*

E.Linux.8

> *Write a section explaining cron, at, and crontab*

4.7.2 Secure Shell

Learning Objectives

- This is one of the most important sections of the book, studdy it carefully.
- learn how to use SSH keys
- Learn how to use ssh-add and ssh-keycahin so you only have to type in your password once
- Understand that each computer needs its own ssh key

Secure Shell is a network protocol allowing users to securely connect to remote resources over the internet. In many services we need to use SSH to assure that we protect he messages send between the communicating entities. Secure Shell is based on public key technology requiring to generate a public-private key pair on the computer. The public key will than be uploaded to the remote machine and when a connection is established during authentication the public private key pair is tested. If they match authentication is granted. As many users may have to share a computer it is possible to add a list of public keys so that a number of computers can connect to a server that hosts such a list. This mechanism builds the basis for networked computers.

In this section we will introduce you to some of the commands to utilize secure shell. We will reuse this technology in other sections to for example create a network of workstations to which we can log in from your laptop. For more information please also consult with the SSH Manual.

> ⚠ *Whatever others tell you, the private key should never be copied to another machine. You almost always want to have a passphrase protecting your key.*

4.7.2.1 ssh-keygen

The first thing you will need to do is to create a public private key pair. Before you do this check whether there are already keys on the computer you are using:

```
ls ~/.ssh
```

If there are files named id_rsa.pub or id_dsa.pub, then the keys are set up already, and we can skip the generating keys step. However you must know the

passphrase of the key. If you forgot it you will need to recreate the key. However you will lose any ability to connect with the old key to the resources to which you uploaded the public key. So be careful.

To generate a key pair use the command ssh-keygen. This program is commonly available on most UNIX systems and most recently even Windows 10.

To generate the key, please type:

```
$ ssh-keygen -t rsa -C <comment>
```

The comment will remind you where the key has been created, you could for example use the hostname on which you created the key.

In the following text we will use *localname* to indicate the username on your computer on which you execute the command.

The 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.

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.

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. Please use a strong passphrase to protect it appropriately. Some may advise you (including teachers and TA's) 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. Only for some system related services you may create passwordless keys, but such systems need to be properly protected.

 *Not using passphrases poses a security risk!*

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

+--[ RSA 2048]----+
|.+...Eo= .       |
| ..=.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/U7uB8kt
6KlRh4kUOjgw9sc4Uu+Uwe/kshuispauhfsjhfm,anf6787sjgdkjsgl+EwD0
thkoamyi0VvhTVZhj61pTdhyl1t8hlkoL19JVnVBPP5kIN3wVyNAJjYBrAUNW
4dXKXtmfkXp98T3OW4mxAtTH434MaT+QcPTcxims/hwsUeDAVKZY7UgZhEbiE
xxkejtnRBHTipi0W03W05TOUGRW7EuKf/4ftNVPilCO4DpfY44NFG1xPwHeim
Uk+t9h48pBQj16FrUCp0rS02Pj+4/9dNeS1kmNJu5ZYS8HVRhvuoTXuAY/UVc
ynEPUegkp+qYnR user@myemail.edu
```

The directory `~/.ssh` will also contain the private key `id_rsa` which you must not share or copy to another computer.

> ⚠ *Never, copy your private key to another machine or check it into a repository!*

To see what is in the .ssh directory, please use

```
$ ls ~/.ssh
```

Typically you will se a list of files such as

```
authorized_keys
id_rsa
id_rsa.pub
known_hosts
```

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.
```

4.7.2.2 ssh-add

Often you wil find wrong information about passphrases on the internet and people recommending you not to use one. However it is in almost all cases better to create a key pair and use `ssh-add` to add the key to the current session so it can be used in behalf of you. This is accomplished with an agent.

The `ssh-add` command adds SSH private keys into the SSH authentication agent for implementing single sign-on with SSH. ssh-add allows the user to use any number of servers that are spread across any number of organizations, without having to type in a password every time when connecting between servers. This is commonly used by system administrators to login to multiple server.

`ssh-add` can be run without arguments. When run without arguments, it adds the following default files if they do exist:

- `~/.ssh/identity` - Contains the protocol version 1 RSA authentication identity of the user.
- `~/.ssh/id_rsa` - Contains the protocol version 1 RSA authentication identity of the user.
- `~/.ssh/id_dsa` - Contains the protocol version 2 DSA authentication identity of the user.

- `~/.ssh/id_ecdsa` - Contains the protocol version 2 ECDSA authentication identity of the user.

To add a key you can provide the path of the key file as an argument to ssh-add. For example,

```
ssh-add ~/.ssh/id_rsa
```

would add the file `~/.ssh/id_rsa`

If the key being added has a passphrase, `ssh-add` will run the `ssh-askpass` program to obtain the passphrase from the user. If the `SSH_ASKPASS` environment variable is set, the program given by that environment variable is used instead.

Some people use the `SSH_ASKPASS` environment variable in scripts to provide a passphrase for a key. The passphrase might then be hard-coded into the script, or the script might fetch it from a password vault.

The command line options of `ssh-add` are as follows:

Option	Description
`-c`	Causes a confirmation to be requested from the user every time the added identities are used for authentication. The confirmation is requested using ssh-askpass.
`-D`	Deletes all identities from the agent.
`-d`	Deletes the given identities from the agent. The private key files for the identities to be deleted should be listed on the command line.
`-e` pkcs11	Remove key provided by pkcs11
`-L`	Lists public key parameters of all identities currently represented by the agent.
`-l`	Lists fingerprints of all identities currently represented by the agent.
`-s` pkcs11	Add key provided by pkcs11.
`-t` life	Sets the maximum time the agent will keep the given key. After the timeout expires, the key will be automatically removed from the agent. The default value is in seconds, but

	can be suffixed for m for minutes, h for hours, d for days, or w for weeks.
`-X`	Unlocks the agent. This asks for a password to unlock.
`-x`	Locks the agent. This asks for a password; the password is required for unlocking the agent. When the agent is locked, it cannot be used for authentication.

4.7.2.3 SSH Add and Agent

To not always type in your password, you can use `ssh-add` as previously discussed

It prompts the user for a private key passphrase and add it to a list of keys managed by the ssh-agent. Once it is in this list, you will not be asked for the passphrase as long as the agent is running.with your public key. To use the key across terminal shells you can start an ssh agent.

To start the agent please use the following command:

```
eval `ssh-agent`
```

or use

```
eval "$(ssh-agent -s)"
```

It is important that you use the backquote, located under the tilde (US keyboard), rather than the single quote. Once the agent is started it will print a PID that you can use to interact with later

To add the key use the command

```
ssh-add
```

To remove the agent use the command

```
kill $SSH_AGENT_PID
```

To execute the command upon logout, place it in your `.bash_logout` (assuming you use bash).

On OSX you can also add the key permanently to the keychain if you do toe

following:

```
ssh-add -K ~/.ssh/id_rsa
```

Modify the file `.ssh/config` and add the following lines:

```
Host *
  UseKeychain yes
  AddKeysToAgent yes
  IdentityFile ~/.ssh/id_rsa
```

4.7.2.3.1 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.

4.7.2.3.2 Using SSH on Linux

All Linux versions come with ssh and can be used right from the terminal.

4.7.2.3.3 Using SSH on Raspberry Pi 3

SSH is available on Raspbian. However, to ssh into the PI you have to activate it via the configuration menu. For a more automated configuration, we will provide more information in the Raspberry PI section.

4.7.2.3.4 SSH on Windows 

> ⚠ *This section is outdated and should be replaced with information from SSH in powershell and the new ubuntu running in windows.*

- https://www.howtogeek.com/336775/how-to-enable-and-use-windows-10s-built-in-ssh-commands/

In case you need access to ssh Microsoft has fortunately updated their software to be able to run it directly from the Windows commandline including PowerShell.

However it is as far as we know not activated by default so you need to follow

some setup scripts. Also this software is considered beta and its development and issues can be found at

https://github.com/PowerShell/Win32-OpenSSH
https://github.com/PowerShell/Win32-OpenSSH/issues What you have to do is to install it by going to

```
Settings > Apps
```

and click

```
Manage optional features
```

under

```
Apps & features
```

Next, Click on the `Add feature`. You will be presented with a list in which you scroll down, till you find `OpenSSH Client (Beta)`. Click on it and invoke `Install`.

After the install has completed, you can use the `ssh` command. Just type it in the commandshell or PowerShell

```
PS C:\Users\gregor> ssh
```

Naturally you can now use it just as on Linux or OSX. and use it to login to other resources

```
PS C:\Users\gregor> ssh myname@example.com
```

4.7.2.4 SSH and putty

We no longer recommend the use of putty and instead you should be using SSH over Powershell for this class.

4.7.2.4.1 Access a Remote Machine

Once the key pair is generated, you can use it to access a remote machine. To dod so the public key needs to be added to the `authorized_keys` file on the remote machine.

The easiest way to do tis is to use the command `ssh-copy-id`.

```
$ ssh-copy-id user@host
```

Note that the first time you will have to authenticate with your password.

Alternatively, if the ssh-copy-id is not available on your system, you can copy the file manually over SSH:

```
$ cat ~/.ssh/id_rsa.pub | ssh user@host 'cat >> .ssh/authorized_keys'
```

Now try:

```
$ ssh user@host
```

and you will not be prompted for a password. However, if you set a passphrase when creating your SSH key, you will be asked to enter the passphrase at that time (and whenever else you log in in the future). To avoid typing in the password all the time we use the ssh-add command that we described earlier.

```
$ ssh-add
```

4.7.2.5 SSH Port Forwarding

> ⚠ *this section has not been vetted yet*

TODO: Add images to illustrate the concepts

SSH Port forwarding (SSH tunneling) creates an encrypted secure connection between a local computer and a remote computer through which services can be relayed. Because the connection is encrypted, SSH tunneling is useful for transmitting information that uses an unencrypted protocol.

4.7.2.5.1 Prerequisites

- Before you begin, you need to check if forwarding is allowed on the SSH server you will connect to.
- You also need to have a SSH client on the computer you are working on.

If you are using the OpenSSH server:

```
$ vi /etc/ssh/sshd_config
```

and look and change the following:

```
AllowTcpForwarding = Yes
GatewayPorts = Yes
```

Set the `GatewaysPorts` variable only if you are going to use remote port forwarding (discussed later in this tutorial). Then, you need to restart the server for the change to take effect.

4.7.2.5.2 How to Restart the Server

If you are on:

- Linux, depending upon the init system used by your distribution, run:

  ```
  $ sudo systemctl restart sshd
  $ sudo service sshd restart
  ```

  Note that depending on your distribution, you may have to change the service to ssh instead of sshd.

- Mac, you can restart the server using:

  ```
  $ sudo launchctl unload /System/Library/LaunchDaemons/ssh.plist
  $ sudo launchctl load -w /System/Library/LaunchDaemons/ssh.plist
  ```

- Windows and want to set up a SSH server, have a look at MSYS2 or Cygwin.

4.7.2.5.3 Types of Port Forwarding

There are three types of SSH Port forwarding:

4.7.2.5.4 Local Port Forwarding

Local port forwarding lets you connect from your local computer to another server. It allows you to forward traffic on a port of your local computer to the SSH server, which is forwarded to a destination server. To use local port forwarding, you need to know your destination server, and two port numbers.

Example 1:

```
$ ssh -L 8080:www.cloudcomputing.org:80 <host>
```

Where `<host>` should be replaced by the name of your laptop. The -L option specifies local port forwarding. For the duration of the SSH session, pointing your browser at `http://localhost:8080/` would send you to `http://cloudcomputing.com`

Example 2:

This example opens a connection to the www.cloudcomputing.com jump server, and forwards any connection to port 80 on the local machine to port 80 on `intra.example.com`.

```
$ ssh -L 80:intra.example.com:80 www.cloudcomputing.com
```

Example 3:

By default, anyone (even on different machines) can connect to the specified port on the SSH client machine. However, this can be restricted to programs on the same host by supplying a bind address:

```
$ ssh -L 127.0.0.1:80:intra.example.com:80 www.cloudcomputing.com
```

Example 4:

```
$ ssh -L 8080:www.Cloudcomputing.com:80 -L 12345:cloud.com:80 <host>
```

This would forward two connections, one to `www.cloudcomputing.com`, the other to `www.cloud.com`. Pointing your browser at `http://localhost:8080/` would download pages from www.cloudcomputing.com, and pointing your browser to `http://localhost:12345/` would download pages from www.cloud.com.

Example 5:

The destination server can even be the same as the SSH server.

```
$ ssh -L 5900:localhost:5900 <host>
```

The LocalForward option in the OpenSSH client configuration file can be used to configure forwarding without having to specify it on command line.

4.7.2.5.5 Remote Port Forwarding

Remote port forwarding is the exact opposite of local port forwarding. It

forwards traffic coming to a port on your server to your local computer, and then it is sent to a destination. The first argument should be the remote port where traffic will be directed on the remote system. The second argument should be the address and port to point the traffic to when it arrives on the local system.

```
$ ssh -R 9000:localhost:3000 user@clodcomputing.com
```

SSH does not by default allow remote hosts to forwarded ports. To enable remote forwarding add the following to: `/etc/ssh/sshd_config`

```
GatewayPorts yes
```

```
$ sudo vim /etc/ssh/sshd_config
```

and restart SSH

```
$ sudo service ssh restart
```

After completing the previous steps you should be able to connect to the server remotely, even from your local machine. `ssh -R` first creates an SSH tunnel that forwards traffic from the server on port 9000 to your local machine on port 3000.

4.7.2.5.6 Dynamic Port Forwarding

Dynamic port forwarding turns your SSH client into a SOCKS proxy server. SOCKS is a little-known but widely-implemented protocol for programs to request any Internet connection through a proxy server. Each program that uses the proxy server needs to be configured specifically, and reconfigured when you stop using the proxy server.

```
$ ssh -D 5000 user@clodcomputing.com
```

The SSH client creates a SOCKS proxy at port 5000 on your local computer. Any traffic sent to this port is sent to its destination through the SSH server.

Next, you'll need to configure your applications to use this server. The *Settings* section of most web browsers allow you to use a SOCKS proxy.

4.7.2.5.7 ssh config

Defaults and other configurations can be added to a configuration file that is

placed in the system. The ssh program on a host receives its configuration from

- the command line options
- a user-specific configuration file: `~/.ssh/config`
- a system-wide configuration file: `/etc/ssh/ssh_config`

Next we provide an example on how to use a config file

4.7.2.5.8 Tips

Use SSH keys

- You will need to use ssh keys to access remote machines

No blank passphrases

- In most cases you must use a passphrase with your key. In fact if we find that you use passwordless keys to futuresystems and to chameleon cloud resources, we may elect to give you an *F* for the assignment in question. There are some exceptions, but they will be clearly communicated to you in class. You will as part of your cloud drivers license test explain how you gain access to futuresystems and chameleon to explicitly explain this point and provide us with reasons what you can not do.

A key for each server

- Under no circumstances copy the same private key on multiple servers. This violates security best practices. Create for each server a new private key and use their public keys to gain access to the appropriate server.

Use SSH agent

- So as to not to type in all the time the passphrase for a key, we recommend using ssh-agent to manage the login. This will be part of your cloud drivers license.

  But shut down the ssh-agent if not in use

keep an offline backup, put encrypt the drive

- You may for some of our projects need to make backups of private keys on other servers you set up. If you like to make a backup you can do so on a USB stick, but make sure that access to the stick is encrypted. Do not store anything else on that key and look it in a safe place. If you lose the stick, recreate all keys on all machines.

4.7.2.5.9 References

- The Secure Shell: The Definitive Guide, 2 Ed (O'Reilly and Associates)

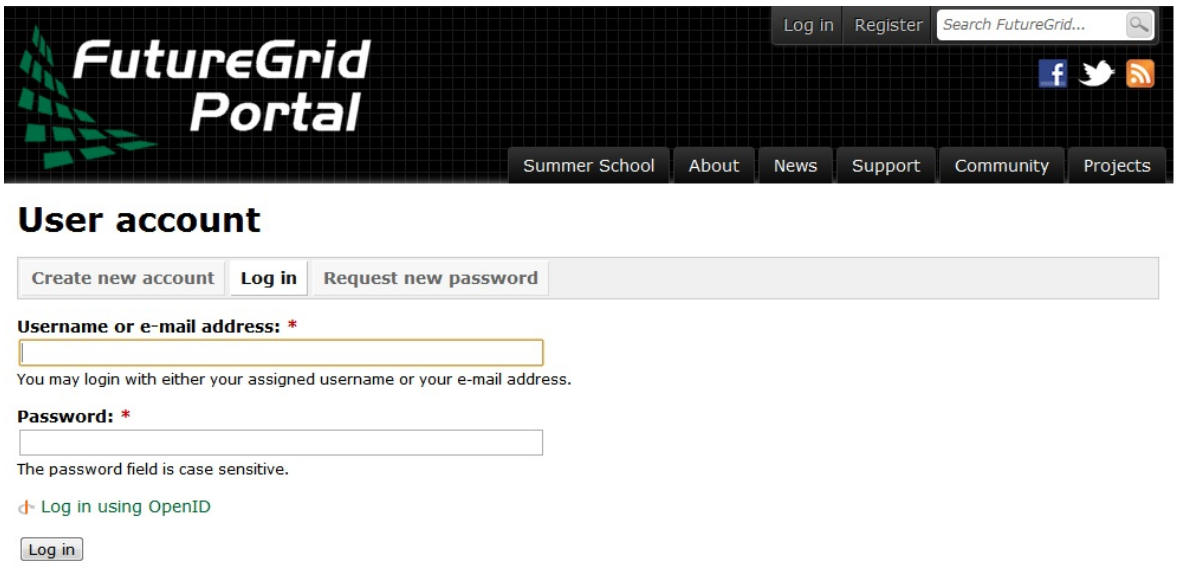
4.7.2.6 SSH to FutureSystems Resources

Learning Objectives

- Obtain a Future system account so you can use kubernetes or dockerswarm or other services offered by FutureSystems.
- Note that we no longer support OpenStack in FutureSystems.

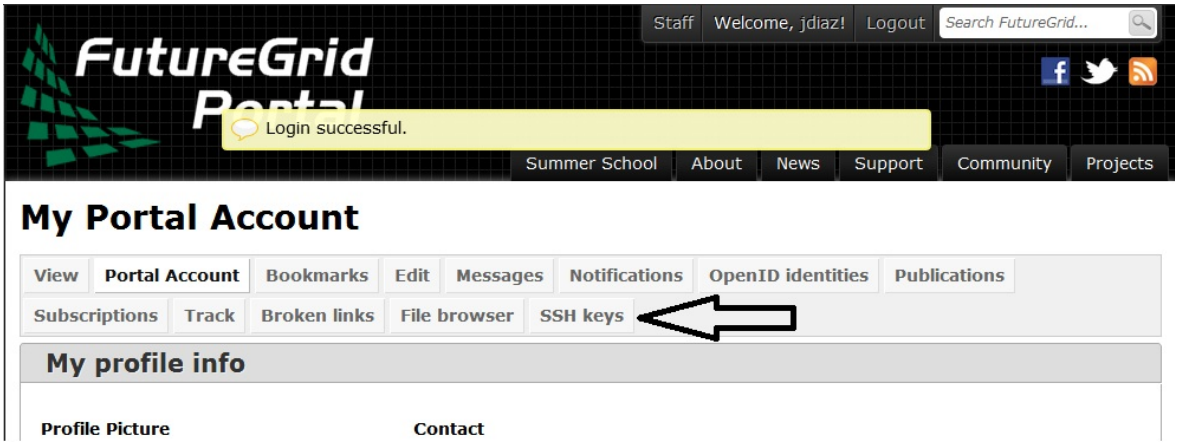
Next, you need to upload the key to the portal. You must be logged into the portal to do so.

Step 1: Log into the portal



image

Step 2: Click in the “ssh key” button or go directly to https://portal.futuresystems.org/my/ssh-keys



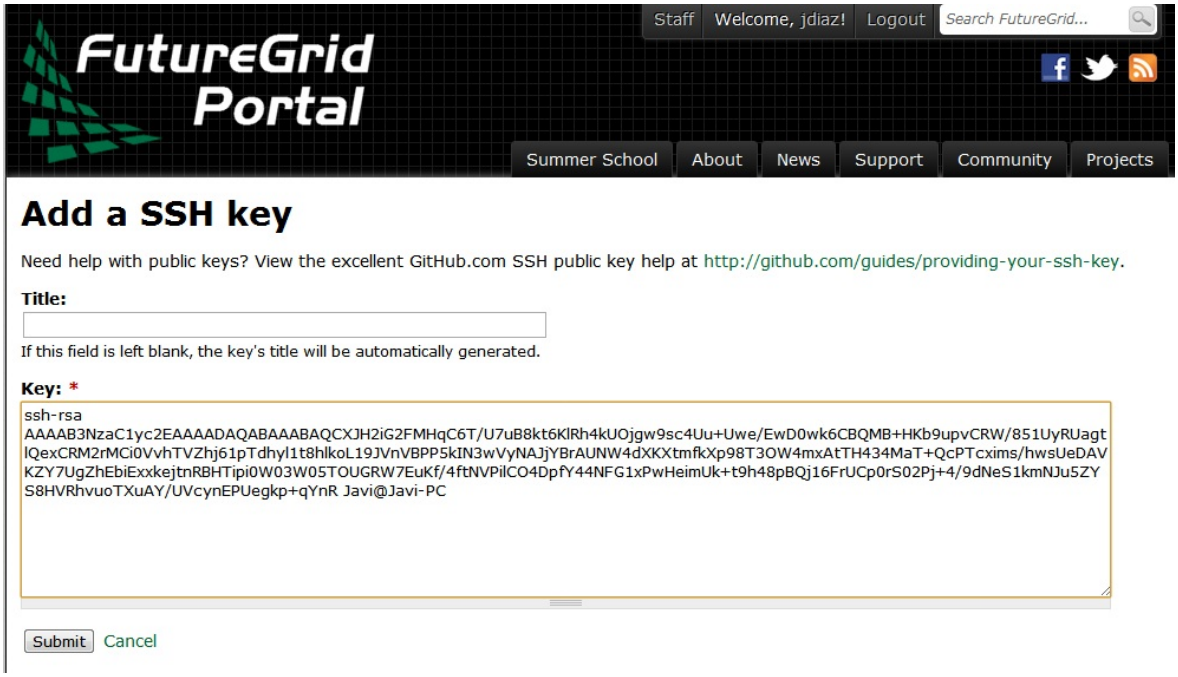
image

Step 3: Click in the “add a public key” link.

image

Step 4: Paste your ssh key into the box marked Key. Use a text editor to open the `id_rsa.pub`. Copy the entire contents of this file into the ssh key field as part of your profile information. Many errors are introduced by users in this step as they do

not paste and copy correctly.



image

Step 5: Click the submit button. **IMPORTANT**: Leave the Title field blank. Make sure that when you paste your key, it does not contain newlines or carriage returns that may have been introduced by incorrect pasting and copying. If so, please remove them.

At this point, you have uploaded your key. However, you will still need to wait till all accounts have been set up to use the key, or if you did not have an account till it has been created by an administrator. Please, check your email for further updates. You can also refresh this page and see if the boxes in your account status information are all green. Then you can continue.

4.7.2.6.1 Testing your FutureSystems ssh key

If you have had no FutureSystem account before, you need to wait for up to two business days so we can verify your identity and create the account. So please wait. Otherwise, testing your new key is almost instantaneous on india. For other clusters like it can take around 30 minutes to update the ssh keys.

To log into india simply type the usual ssh command such as:

```
$ ssh portalname@india.futuresystems.org
```

The first time you ssh into a machine you will see a message like this:

```
The authenticity of host 'india.futuresystems.org (192.165.148.5)' cannot be established.
RSA key fingerprint is 11:96:de:b7:21:eb:64:92:ab:de:e0:79:f3:fb:86:dd.
Are you sure you want to continue connecting (yes/no)? yes
```

You have to type yes and press enter. Then you will be logging into india. Other FutureSystem machines can be reached in the same fashion. Just replace the name india, with the appropriate FutureSystems resource name.

4.7.2.7 Exercises

E.SSH.1:

> *Create an SSH key pair*

E.SSH.2:

> *Upload the public key to git repository you use. Create a fork in git and use your ssh key to clone and commit to it section/ssh.tex*

E.SSH.3:

> *The images in the futuresystems ssh section are a bit outdated. Please update them. Make sure to blend out your username and fingerprints in the images. or invent some …*

E.SSH.4

> Get an account on futuresystems.org (if you are authorized to do so). Upload your key to https://futuresystems.org. Login to india.futuresystems.org. Note that this could take some time as administrators need to approve you. Be patient.

4.8 PYTHON

4.8.1 Python

Please see the Python book:

- *Introduction to Python for Cloud Computing, Gregor von Laszewski, Aug. 2019*

5 FAQ

5.1 FAQ: General

In this section TA's and students can add FAQ's from the piazza. As the material especially the programming related one is so useful that it is shared by now in multiple classes, however they use different piazza's sharing the information in an FAQ in the handbook allows us to quickly disseminate the relevant information between classes. If an FAQ is only for one class we will be especially mark it.

5.1.1 Can I assume that all information is in the FAQ to do the class?

No. The class book will be our main source of information not just a collection of FAQ's.

5.1.2 Piazza

5.1.2.1 Why are some FAQs that are on piazza not here?

Two reason:

1. some of them need not to be in this FAQ.
2. The TAs will evaluate the FAQs every day at the end of the day and integrate those that need to be in this list at that time. Hence it may take up to 24 hours for FAQs to appear here.

Once an FAQ is in the book answered (it may actually be part of another section, TA's will mark the FAQ in piazza, so you can make sure which FAQs are already in the book. We recommend to look in the book as there could be information in it that you otherwise missed.

5.1.3 How do I find all FAQ's in Piazza?

Two ways exist

a. Please visit your class piazza. You will find a “faq” tag in your piazza window. Click on it and all posts marked with FAQ will show up,

b. In the search field type in FAQ. All posts with the text FAQ in it will be listed.

5.1.4 Has SOIC computers I can use remotely?

See: https://uisapp2.iu.edu/confluence-prd/pages/viewpage.action?pageId=114491559

5.1.5 When contributing to the book my name is not listed properly or not at all

The following reasons exist:

1. if its not listed at all your contribution may be in a different repository, please contact Gregor

2. if it does not show up correctly and only shows your github name, which you can see in the contributor section or with

```
$ git shortlog -s -e
2  laszewsk <laszewski@gmail.com>
...
```

You need to do two things.

First, add your name to the file

- https://github.com/cloudmesh-community/book/blob/master/.mailmap

Second, complete the set up your git on the machine you work with in case you use a commandline tool with git init (see our notes on this)

If you use the GUI you may need to go to the account settings and associate a first name lastname, I however do not know ho to do that, so if you kwon reply ti this

5.1.6 How to read the technical sections of the lecture notes

We will add throughout the semester some technical lecture notes. These notes contain information on how to install and run certain programs on a computer. What we have seen in the past with some students is that they do not read the text between the sections. Instead they just execute things without reading or understanding assuming that they can juste copy and paste. These sections include valuable information that you **must** read before you execute any code in them.

Here is the workflow on how to read such technical sections

1. Do not execute anything yet
2. Read the entire section including the lines between the gray boxes
3. Step back and reflect on what you read
4. Reread the section, if a section needs more information google for it (things could be overnight updated on the internet, please remember we are just presenting a snapshot in time here)
5. Once you have obtained knowledge, decide if the section is relevant for you (e.g. windows sections may not be relevant for MacOS users)
6. Carefully execute the relevant portions for you

> ⚠ *AS ALWAYS THERE IS NO GUARANTEE THAT WHAT THE CODE WORKS OR COULD NOT DESTROY SOMETHING. MAKE SURE TO HAVE A BACKUP. IF IN DOUBT RUN IN A VIRTUAL MACHINE IF YOU CAN.*

5.1.7 How to check if a yaml file is valid?

In case you need to check an open source public YAML file you can use the following

The easiest is to use yamllint:

```
$ pip install yamllint
$ yamllint README.yml
```

Using yamllint is our preferred method.

A python script to check it is available at

- https://github.com/cloudmesh-community/book/tree/master/examples/yaml-validation/validate_yml.py

This python script depends on `ruamel.yaml` package. We can install it using following command:

```
$ pip install ruamel.yaml
```

It accepts file path as an argument. This script will load `YAML` file and dump its content on console. For invalid syntax it will throw an error.

To execute python script you need to run following command after you clone *book* repository.

```
$ cd examples/yaml-validation
$ chmod +x validate_yml.py
$ ./validate_yml.py <path to yaml file to validate>
```

Online checkers are available at

- https://codebeautify.org/yaml-validator

A ruby script can do

```
$ ruby-e "require 'yaml';puts YAML.load_file('./README.yml')"
```

YAML validation in visual studio can be achieved also

- https://marketplace.visualstudio.com/items?itemName=redhat.vscode-yaml

5.1.8 Download the ePub ferquently

Please be reminded that the ePub is updated frequently and we recommend that you download it before you read.

I myself have integrated an ePub reader in my Web browser so that every time I

click on the View Raw in github, I get the most up to date version.

I use ibooks on OSX, calibre is a good system on Windows and Linux, MS also has Microsoft Edge. However on Microsoft edge you will need the lates version which starts with 42

5.1.9 Spelling of filenames in github

Most of our scripts require proper spellingg including proper capitalization. The spelling of `notebook.md` is `notebook.md`

not

`Notebook.md` or `NOTEBOOK.md` or other spelling

Please, correct if you did not use lower case

The spelling of `README.yml` is`README.yml` and not

`README.md` (which needs to be removed) or `readme.yaml`

please correct if needed. We will not grade any assignments if your README.yaml or notebook.md is misspelled or missing or is not following our simple format.

5.1.10 How to open the ePub from Github?

If you see thw `view Raw`, you need to click on it. It will download the file. Than you can open that.

However, If you use edge or integrated your ePub viewer in your browser and clicking on it will automatically open your ePub browser.

5.1.11 Assignment Summary

 outdated

a. The assignment is discussed in Chapter 1 of the lecture notes

b. Examples of what other students have done are in the Example Artifacts section

Please look at both sections

In this class we addressed 3 assignment that related to your grade

Tech summaries - they have been assigned to you in https://piazza.com/class/jl6rxey6w413gi?cid=89 to show to the TAs that you work on them use the nomenclature that is discussed in the preface of the technology handbook. Put yours hid in the “headline” and a smiley when done, If you work on it put in a hand. Project - look at examples in the example artifact sections A paper has typically the following sections

Theory Implementation (e.g. Python) Benchmark

A more detailed outline is * Paper * Title * Abstract * Introductions * Requirements * Architecture * Implementation * Benchmark * Conclusions * Bibliography * Work breakdown

5.1.12 Auto 80 char

 outdated

Those that use emcas could experiment with the following. I do not know if this works well yet.

The following will autoformat an entire file to 80 chars. The reason i put it in test.md is that I do not know if it reliably works on all md files, just inspect the output and decide for yourself. some md files you may not want to manipulate with this though

```
cp file.md test.md
emacs -batch test.md --eval '(fill-region (point-min) (point-max))' -f save-buffer
```

5.1.13 Useful FAQs for residential and online students

 this is outdated.

You will know if this post applies to you.

This class does not have a high volume on posts via Piazza

What we find is that some students create a high overhead on themselves by not following our FAQs or documentation on the technology summaries. When we observe something we just post it in an FAQ in piazza that we expect you look at. Yet we find some students that keep on resubmitting their technology summaries while not integrating our tips from the FAQs that cost less then a minute to do. Those that do not read and follow the FAQ make their work unnecessary complicated. We even start now noticing students that remove bibliography entries instead of just fixing them. Also, we saw recently students that had perfectly geat entries, other than the authors (see FAQ) and instead of fixing the authors in case it was a company or organization, to fix random fields such as the titles and thus creating even more work on themselves.

We have lots of online hours during the week There are 4 hours you can attend, Mo, We Thu, so if you do have something you do not understand, I recommend that you use these hours. In case you are a residential student you also have Fridays. To start, I would review our FAQs

Interestingly we see these issues more with residential students than with online students. This may indicate that the residential students in question forget to read the posts in piazza?

5.1.14 What if i committed a wrong file to github, a.g. a private key?

The answer to this question is more complicated than you think. Thus the best way to deal with it is to

AVOID IT:

a. first do github adds file by fill with git add. Avoid using adds on AND DO NOT USE

```
git add .                # <<<<<<< DO NOT USE
```

b. only use ssh keys in ~/.ssh **NEVER** place keys in directories that are

managed by git

YOU CAN NOT EASILY DELETE FILES FROM GIT:

c. as you may already know despite you deleting a file from git it is still in the git history. Also there are bad characters out there so if you checked in you ssh private key just for a second

you must assume your private key is now compromised and all machines that use it are compromised.

d. although GIt allows you to delete the file, it is still in the githistory, which can be mined so despite you pressing delete its still there and can be found. This is not a bug in git but this is you having git not used right.

e. There are ways to purge such files, but it would imply that everyone that did a fork needs to do a new fork which is naturally a big issue, so we do not do this during the semester.

NOW WHAT?

f. every machine on which you used the public key of this private key is to be considered now compromised.

g. put them off from the network while plugging out the network plug

h. if the machines are not owned by you but for example, IU, notify the people that own the machine to ask for help with mitigation.

i. if you are lucky, replace the key, this is the case for example for services such as github. Make sure to inspect the configurations and see if your account has not been hijacked.

j. We will immediately remove you from services such as future systems and chameleon cloud as a precaution or deactivate your membership in our cloud accounts.

k. if you used the keys on other services, including IU, it is up to you to identify how to deal with this,

l. definitely create a new key and use that from now on.

m. you can call Gregors office number or use piazza to set up a call to identify what the impact is as this is typically an emergency use 812 856 1311. Do not leave a msg, but instead send e-mail.with your phone number so we can call you back to assess the situation.

n. if you use them on public clouds that cost money, shut down all machines that use them. I would not start them again but instead use new once. It may be time to drop everything and do this first. Sorry for making you now panic.

5.2 FAQ: 423/523 AND OTHERS COLOCATED WITH THEM

This section contains FAQs relevant for 523.

5.2.1 Bibtex tips for consitency across contributors

Congratulations, the majority of the bitex entries were done correctly. However, there are some few and small issues you could improve. This helps consistency across all contributors

a. use camel case in all titles consistently

b. see the FAQ on authors and keys

c. allowed are

howpublished={Web page},

howpublished={Blog},

howpublished={Presentation},

howpublished={Github},

howpublished={Bitbucket},

in miscs are allowed

not allowed are: webpage Webpage website Website,

any author or organization name

in case of wikipedia, the author ie author = {{Wikipedia}},

all misc labels ought to have a www- in it if they are online resources

c. we do not use [@Online] which is specific to some publishers and not universal

d. if a online citation has a publishing date we use month and year not date, you can not mix date with month and year, use month and year instead, If the publication date is not known use month and date for access and put note={Accessed}, in the entry

e. Use camel case for authors. Note that some authors have strange last name such as mine, so my author name is von Laszewski, Gregor

f. do not uses utf-8 chars such as “u”a and so on use instead {"u} {"a} and so on, see LaTeX bibtex manuals for details

Please remember this is not a lot to change for you.

A TA is assigned to help on this.

5.2.2 Misc entries require an author or key

Misc entries do require either an author or a key. The author lastname or the key is used for sorting. An entry without either is invalid.

You have time to fix this for a month. Most of you added an author.

In case the author is a company it must be in double brackets, example: author= {{My Company}}, keys do naturally do not need double brackets.

5.2.3 TODO list location

I still haven’t received any feedback from my earlier question this morning about locating the todo list in the ePub so that I may see what is wrong with my technology summaries. I’ve also checked that I do not have any outstanding pull requests. Could someone please help me find this so I may fix my summaries by tonight?

The Todo is in the ePub of the technologies its a section header

5.2.4 Video on how to find the error reports for Technology Summaries

Those with pull request errors may want to look at

https://www.youtube.com/watch?v=FDqlKtQcy1U

5.2.5 only one url in url=

There can only be one url in the bibtex url filed, IF you need multiple, each one must have its own citation entry.

5.2.6 Incomplete analysis of your technologies

While reviewing some of your technologies we found that some students checked on their technologies with a smiley so we started looking at them. The good news is that many are good improvements.

However, we have a couple of suggestions so you can achieve your best.

a. we see that some students have missing bibtex entries or use labels wrong

b. it is in the student’s responsibility to fix all duplicated bibtex entries in all technologies. For example we only need one www-google bibtex and not Google-web page Google, and other labels, all labels in all technologies should be changed to a single entry

c. We see that when we assigned you a technology you do not cross check if other entries use your technology. Naturally, if we assign you a technology and the entry is duplicated you nee dto discuss with us what to do. In most

cases you also have to fix the other entry for which you get also credit for.

d. students do not use linux tools such as grep because they have not yet switched to using command line tools for git.

As an example I like to provide what you what I would do if i were to improve the entry “Flume”

1. I would grep for it:

```
grep -n -R -i flume chapters bib
```

As a result I get

```
chapters/tech/flume.md:## Flume

chapters/tech/flume.md:| title    | Flume          |

chapters/tech/flume.md:Flume is distributed, reliable and available service for efficiently

chapters/tech/flume.md:data [@apache-flume]. Flume was created to allow you to flow data

chapters/tech/flume.md:from a source into your Hadoop environment.  In Flume, the entities

chapters/tech/flume.md:be any data source, and Flume has many predefined source adapters. A

chapters/tech/flume.md:or removing pieces of information, and more [@ibm-flume].

chapters/tech/google-flumejava.md:## Google FlumeJava ![Construction](images/construction.png) fa18-523-83

chapters/tech/google-flumejava.md:| title    | Google FlumeJava       |

chapters/tech/google-flumejava.md:FlumeJava is a Java library that is built based on the concepts of MapReduce to simplif

chapters/tech/google-flumejava.md:FlumeJava is an easier-to-use version of MapReduce, make it simplier to build operatior

chapters/tech/google-flumejava.md:FlumeJava was able to optimize MapReduce tasks and decrease execution time of MapReduce

bib/references.bib:@Misc{apache-flume,

bib/references.bib:  Title =        {Apache Flume},

bib/references.bib:  Url =          {https://flume.apache.org/index.html}

bib/references.bib:@Misc{ibm-flume,

bib/references.bib:  Title =        {What is Flume?},

bib/references.bib:                  {https://www-01.ibm.com/software/data/infosphere/hadoop/flume/}

bib/references.bib:@Inproceedings{flumejava-paper,

bib/references.bib:  Title =        {FlumeJava: Easy, Efficient Data-Parallel Pipelines},

bib/references.bib:@Misc{www-flumejava-google,

bib/references.bib:  Key =          {FlumeJava Google research},

bib/references.bib:@Misc{apache-flume,

bib/references.bib:  Title =        {Apache Flume},
```

```
bib/references.bib:  Url =          {https://flume.apache.org/index.html}

bib/references.bib:@Misc{ibm-flume,

bib/references.bib:  Title =        {What is Flume?},

bib/references.bib:                  {https://www-01.ibm.com/software/data/infosphere/hadoop/flume/}

bib/references.bib:@Inproceedings{flumejava-paper,

bib/references.bib:  Title =        {FlumeJava: Easy, Efficient Data-Parallel Pipelines},

bib/references.bib:@Misc{www-flumejava-google,

bib/references.bib:  Key =          {FlumeJava Google research},
```

Now I see the following

a. I get two entries that relate to fume. So I need to look at both of them and potentially fix both of them or if it makes sense merging them

b. I see a howl bunch of bib tex entries. In fact it looks like that many are duplicated. Thus I need to make sure that I reduce the number of bibtex entries, but I must be careful, as the once that I would be deleting could be used elsewhere and if I delete them or change the label I need to change them elsewhere also. This includes files that may use the bibtex that have nothing to do with my technology. It is easy, I just check with grep for all entries label that I remove or rename and change the corresponding time.

This all takes a minute in the command line. I am unaware that this task even can be done in the GUI and if it would take hours.

So if you spend hours on doing things in the GUI you do something wrong and must attend the online hour so TAs can teach you how to do it right. Naturally, we have all that information in the handbook also and many students do it right.

c. due to the nature of students may needing to change labels staying in sync with upstream is much easier on the commandline. as documented in the handbook.

So what would I do for Flume in summary

1. merge the entries while flume java becomes a subsection on the 3rd level (assuming the flume we talk about are the same)

2. remove all duplicated entries and use new labels I define
3. for an old entry i just leave the bibtex label and use that
4. in case I need to use new citations I prepend my hid
5. all www resources have www- in it. if not I rename the label

Based on this analysis the Flume entry does not pass our review

I have made significant changes that require a new fork or an update to the fork. Please do so. This could even be done from the GUI, however, the commandline is easier, so we do not teach you how to do it in the GUI.

5.2.7 The pull requests of technology summaries

Hi professor,

I have updated four summaries in the github and got the reply that “use in addition to” also >“, I have no idea about the meaning of that, could you please clarify it? My hid is fa18-523-85, the four technologies are”blaze“,”daal (intel)“,”lxc“,”OSGi".

Looking forward to hearing from you.

look at markdown documentation for >

look at ePub and find a technology report from a student with a gray bar

5.2.8 REMINDER: quotes for technologies

In the technologies, we like you to make the quoting style consistent among all entries that are assigned to you. Please fix them in your entries. please see the following example:

> *"This is a quote over multiple lines*
>
> *for the technologies" [@label].*

Please be reminded to use straight quotes not left and right quotes and use the greater sign at the beginning. Please remember that you are only allowed to use 30% of quotes and that the technologies typically have a 300 word minimum.

5.2.9 Headings

a. please do not use all caps for heading

wrong:

```
## HEADING
```

correct

```
## Heading
```

The title is the only heading that has only one #

5.2.10 Quote characters in markdown

Some editors such as word try to be extra smart and do replace the quotes with a left and a right quote. However, markdown is designed to just use straight quotes.

use proper quotes which are " not left and right quote. Markdown however as we use it uses only one quote and that is “quote” if your editor puts left and right quotes in automatically, find a different editor such as emacs or pycharm

5.2.11 Tech Summaries. Puntuation, citations. Please read.

We see several check-ins that have good content but do not follow the rules for citations.

a. a technology section does not have a *References* section at the end. All technologies just use bibtex. The bibtex is automatically inserted where the label is, so you do not have to worry about managing a references section

b. a citation must be in the same sentence before the sentence ends.

```
This is wrong. [@label]
```

```
"This is also wrong." [@label]



This is the right way to cite [@label].
```

As citations are an important placement please check all your entries that you are responsible for as you only have 4 that should be an issue of minutes not hours.

IF we find such punctuation errors, your entry will be downgraded to a B if nothing else is wrong. We will also keep it marked with a red circle. The same rules apply to your 2 page paper and the project report.

Why so strict? Citation rules are strict and must be done correctly I had professional editors that would reject a submission with citation errors such as this.

5.2.12 use of underscore for em and bf

As we do some translations of the markdown, we noticed that whan you use _ instead of * in your markdown this may lead to issues. please use

```
*italic*

and

**bold**
```

6 GLOSSARY

6.1 GLOSSARY

6.1.1 VM and Container

Dom0

TBD

Hypervisor

TBD

KVM

TBD

Virtual Machine

TBD

Virtual Machine Manager

TBD

XEN

TBD

cgroups

TBD

chroot

TBD

container

TBD

Kernel namespace

TBD

6.1.2 Network

Bridged Networking

TBD

External Network

TBD

Internal Network

TBD

Local Bridge

TBD

Network Address Translation (NAT)

TBD

6.1.3 Storage

Block Device

TBD

Virtual Disk

TBD

Raw Disk

TBD