Fall 2017 - W205 – Storing and Retrieving Data
Week 2 Live Class Session Agenda
Kevin R. Crook

- Schedule
  - 1 page printable PDF on KevinCrook.com
  - Spending class time going over the schedule?
  - For next week, please note we are covering “Unit 7 – Querying Data” out of order to give students some exposure to SQL and ERDs earlier so they will be prepared for Exercise 1 which is coming up.

- Revisit Agenda for Week 1
  - Safari Books – couple that are not available in Berkeley library
  - Exercise 1 – list of items to come up to speed
  - Live Class Sessions – updated - can change if it doesn’t work out

- Was everyone able to get access up and running?
  - Berkeley Library
    - Access electronic book remotely (VPN setup or proxy setup)
  - Slack.com
    - Join the W205 channel
    - Join the section channel and post a message with you first and last name if it isn’t obvious from you slack name
  - Amazon Web Services (AWS)
    - $100 credit for students after verification
  - SSH Terminal for Linux Command Line
    - Windows – PuTTY
    - Mac – Terminal
    - 3rd party – on your own
  - GitHub
    - Sign up
    - Academic discount
    - Create w205_2017_fall repo
      - private (Berkeley Honor Code requires this)
      - grant collaborator access to kevin-crook-ucb
      - create directories exercise_1, exercise_2, project
      - get command line synch up from Linux working
  - Tableau
    - Academic version – get it working before we need it late in the semester

Break Out Exercise
"Soft Skills" – applying theoretical material from the asynchronous curriculum to real world application

(All Groups)

Using Data Science to make the world a better place

Data Science is usually associated with making more money for commercial companies.

Discuss an example of a non-profit area where Data Science can make the world a better place that isn't related to business or making money.

(next page)

(Group 1 start here – if you finish read through the other groups' topics)

Data-Driven Organizations

When discussing data driven organizations, most data science material tend to focus on and highlight high technology companies. However, pretty much every Fortune 500 "old school" company now has a data science department and is trying to morph itself into a data-driven organization. Some "old school" organizations have always been data driven, even before the era of data science. A few seemingly have no interest in data.

Discuss an example of an "old school" Fortune 500 company that by the nature of its business has always been data driven.

Discuss an example of an "old school" Fortune 500 company that seemingly is run by intuition with no use of data. Is it successful? If so, why? If no, why not? Are there some businesses inherently better with intuition than data?

Discuss an example of an "old school" Fortune 500 company that was previous run by intuition without use of data, is currently transforming itself, or has transformed itself to a data driven organization.

Sink / Source Latency

Give some real world examples of sink latency.

Give some real world examples of source latency.

GPU processing – how to classify it

The asynchronous material classifies various type of processing: single node, parallel, distributed, cluster, grid, etc.

How would you classify GPU processing? What GPU processing reference architecture is of interest to data science? What company is renting GPUs in the cloud?

(Group 2 start here – if you finish read through the other groups' topics)

Talking in terms of Reference Architecture

Talking in terms of reference architecture is a soft skill that sounds much more professional than the usual way people discuss hardware and software architectures and decisions to use them. In data science, this is an even more crucial soft skill to have than normal because we are bringing in new and unfamiliar technologies. As we will see later in this semester, a lot of technologies that work for small data sets, simply won't work for large data sets, and vice versa. So, we have a bit of an uphill battle in some "old school" companies to overcome resistance to new technology.

In a lot of companies, people tend to have their favorite architectures for both hardware and software. Unfortunately, a lot of people tend to use "trashing" or "bashing" products as their main technique to discuss these items. A much more professional and rational approach is to discuss these in terms of a reference architecture, the advantages and disadvantages of this reference architecture, and appropriate uses of the reference architectures.

Discuss several examples of technologies framing the discussion in terms of reference architecture. See how different this type of discussion is from the usual manner of discussing new technologies.

Computational Complexity – why do we need it in the real world?

Discuss computational complexity and the Big O notation and why we need it in the real world.

Why do most Machine Learning algorithms exist? (or generalize that to why do most algorithms exist?)

CPU Bound versus I/O Bound

Discuss real world examples of CPU bound processes.

Discuss real world examples of I/O bound processes.

(Group 3 start here – if you finish read through the other groups' topics)

T-Shirt Sizing – applying reference architecture to the data dimension – developing an intuition for sizing

In the asynchronous material, in the discussion of data dimension, they mentioned sizing things into Small, Medium, Large, X-Large, etc. They created a table of sizing to fit these categories. In industry, this is often called T-shirt sizing.

Discuss an aspect of technology (hardware, software, etc.) and create a T-Shirt sizing chart for it. What are the advantages of T-shirt sizing? Talking to non-technical people? Getting technical people to commit to numbers?

Throwing hardware at a problem won't always fix it

In the asynchronous material, it goes over measuring performance in terms of IOPS which is similar to the T-shirt sizing previously given. IOPS allow us to gauge performance at a high level, figure out where bottle necks are, and what hardware will help.

People tend to think in terms of throwing hardware at a problem to fix it. Often this works. If we can double memory and increase our performance ten fold, why not, it's money well spent?

However, as we discovered, often throwing the wrong hardware at a problem won't help us.

Discuss some real world example of performance bottle necks, what hardware would help, and what hardware would not help.

Scale Up / Scale Out – when can they help?

Discuss the difference between Scale Up and Scale Out.

Discuss some real world examples:

- Scale Up
  - will help
  - will not help
- Scale Out
  - will help
  - will not help

Instructor Led Exercise
"Hard Skills" – Linux Command Line

(we will go as far as we can and pick it up next class if we don't get through this)

Users
root
w205
login shells versus a change user

Groups
Users have primary group, plus other groups

Buffered I/O
File systems have to be in synch before you can stop an instance
Hard killing a process with open files can easily corrupt a file system

Processes
Listing processes
Killing processes (soft kill versus hard kill)

Files
Listing files
Creating files
Copying files
Renaming files
Deleting files
Dots: before, inside, multiple, after
Wild cards aka regular expressions
Linking to files (hard links and soft links)
Checking permissions on files
Changing permissions on files

Directories
Absolute paths
Relative paths
Moving around in directories
Dots: single, double
Creating directories
Copying directories
Deleting directories

Renaming directories
Checking permissions on directories
Changing permission on directories

Standard I/O
stdin
stdout
stderr
redirecting

Processes
Listing processes
Sequential execution
Pipelining
Background and Foreground
Killing (soft kill versus hard kill)
nohup

Tar, Tarball, Zip, Unzip

/tmp