Overview of UC Berkeley Demography Computing Resources

Joshua Quan, Demography Computing Director

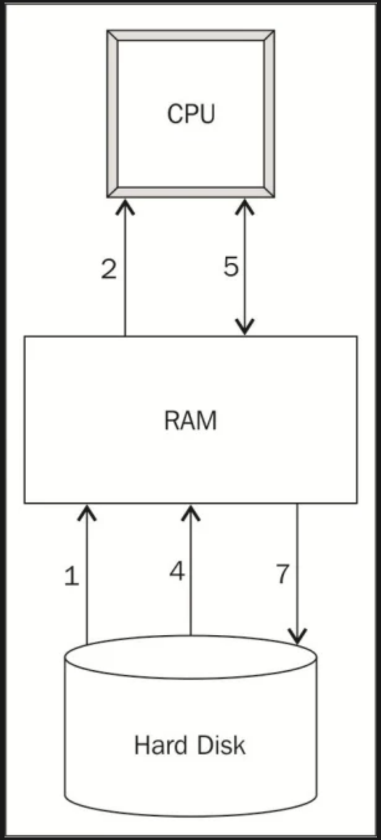
2024-09-12

Goals for today

- Savio (for high performance computing)
- Datahub (for instruction in Python and R)
- Pop Science Lab Documentation

Some jargon

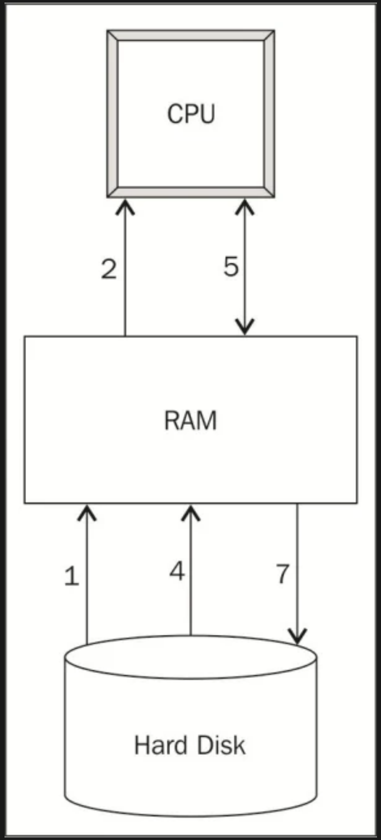
A mental model of how a computer works



```
data <- read.csv("mydata.csv")
totals <- colSums(data)
write.csv(totals, "totals.csv")
```

- 1. When we load and run an R program, the R code is first loaded into RAM.
- 2. The R interpreter then translates the R code into machine code and loads the machine code into the CPU.
- 3. The CPU executes the program.
- 4. The program loads the data to be processed from the hard disk into RAM (read.csv() in the example).
- 5. The data is loaded in small chunks into the CPU for processing.

A mental model of how a computer works

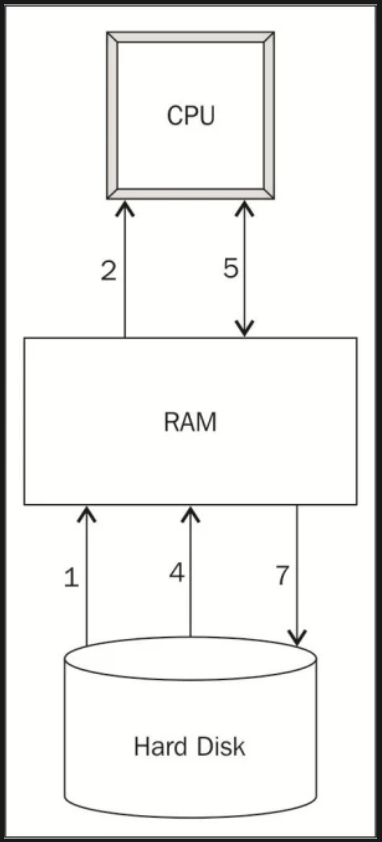


```r
data <- read.csv("mydata.csv")
totals <- colSums(data)
write.csv(totals, "totals.csv")
```

- 6. The CPU processes the data one chunk at a time, and exchanges chunks of data with RAM until all the data has been processed (in the example, the CPU executes the instructions of the colSums() function to compute the column sums on the data set).
- 7. Sometimes, the processed data is stored back onto the hard drive (write.csv() in the example).

A mental model of how a computer works

Performance Bottlenecks



```r
data <- read.csv("mydata.csv")
totals <- colSums(data)
write.csv(totals, "totals.csv")
```

- The speed and performance of the CPU determines how quickly computing instructions, such as colSums() in the example, are executed. This includes the interpretation of the R code into the machine code and the actual execution of the machine code to process the data. *CPU Bound*
- The size of RAM available on the computer limits the amount of data that can be processed at any given time. In this example, if the mydata.csv file contains more data than can be held in the RAM, the call to read.csv() will fail. *Memory Bound*
- The speed at which the data can be read from or written to the hard disk (read.csv() and write.csv() in the example), that is, the speed of the disk input/output (I/O) affects how quickly the data can be loaded into the memory and stored back onto the hard disk. *I/O Bound*

Research Computing

Savio High Performance Computing (HPC)

- Berkeley Lawrence National Labs and Berkeley Research Computing
- 600 nodes, 15,000 cores
- Fast CPUs & Memory, networking, I/O, Petabytes of Disk
- Much more powerful than a laptop/workstation
- Secure for up to P3 data with some extra configurations.

Savio High Performance Computing (HPC)

Savio High Performance Computing (HPC)

- Different partitions/nodes with modern hardware for different computing applications
  - Simplest partitions boast at least 64 GB RAM, ~20 CPU cores.
  - Specialised partitions have lots of memory - savio_bigmem has 512 GB RAM
  - GPUs for machine learning (nVIDIA Tesla V100 GPU;nVIDIA GTX 2080ti GPU, nVIDIA GTX 2080ti GPU)

Savio High Performance Computing (HPC)

Savio High Performance Computing (HPC)

Savio High Performance Computing (HPC)

```
Job_priority =

    (PriorityWeightAge * age_factor) +

    (PriorityWeightQOS * QOS_factor) +

    (PriorityWeightPartition * partition_factor) +

    (PriorityWeightJobSize * job_size_factor) +

    (PriorityWeightFairshare * fair-share_factor) +

    (PriorityWeightAssoc * assoc_factor) +

    SUM(TRES_weight_<type> * TRES_factor_<type>, ...)

    - nice_factor + site_factor
```

- A lot more information about the math behind it in the SLURM documentation

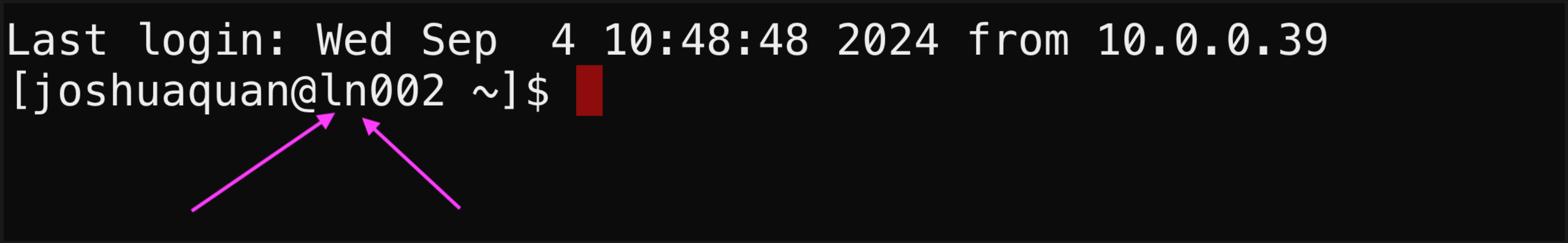
Savio High Performance Computing (HPC)

- Demography account is `fc_demog`
- 3 ways to compute: batch shell, interactive shell, interactive web-browser: Open OnDemand

-
  ```
  ssh joshuaquan@hpc.brc.berkeley.edu
  ```

-
  ```
  srun -A fc_demog -p savio3  -t 01:00:00 --pty bash
  ```
- Username is calnet cred without punctuation. Password is four digit pin + OTP
- Let’s: Sign up

Savio High Performance Computing (HPC)

- 

```
Last login: Wed Sep  4 10:48:48 2024 from 10.0.0.39
[joshuaquan@ln002 ~]$ 
```

- log-in nodes are meant for exactly that, logging in. Do not run code here!
- 💀 !! ⚠️🙅🏻‍♂️

Instructional Computing

Datahub (Jupyter)

- Data/Jupyterhub allows for scalable Jupyterlab, Jupyter Notebooks and Rstudio environments that connect to BCourses for teaching.
- Works well for classes and workhops of most any participant size where everyone needs the exact same environment. Good example is Economic Demography (c175)
- Works not as well for compute intensive applications
- Kubernetes under the hood to flexibly provision docker containers, “pods”, for an individual user
- At Berkeley we connect Datahub to Bcourses. The Jupyter environment performs a git pull on a course github repository. Here’s ours: https://github.com/berkeley-demography/demog-213-f24

Your Assignment

- Describe the interaction of CPU, RAM, and Disk.
- With your neighbor discuss what a “node” is in the context of High Performance Computing. What is a “log-in node”? Should you ever run an analysis on a “log-in node”?

Before next class

- Sign up for a Savio account.
- Set up your one-time pin
- Log-in to open ondemand, schedule an Rstudio or Jupyter session.