Linux on HPC

Harry Mangalam
<harry.mangalam@uci.edu>

Good Judgement
comes from Experience

Experience comes from
Bad Judgement

Good Judgement comes from Experience

Experience comes from Bad Judgement

Some comments

- ~~You know Linux at at least user level~~
- You're bright, can Google, and can read further by yourself.
- You know how to tell useful info from pure fantasy.
- I speak too fast; let me know when I do.
- The Unknown Unknowns problem.
- Questions, please, but I may not answer them immediately.

Philosophy about computing

- Be lazy.
- Copy others.
- Don't invent anything you don't have to.
- Re-use, re-cycle, DON'T re-invent.
- Don't be afraid to ask others.
- Resort to new code only when absolutely necessary.
- New code should be useful to others; beyond 100 lines of code should at least have a usage() stanza.

Useful Concepts

- LEARN HOW TO GOOGLE (see ***Fix IT Yourself with Google*** in the resources).
- Listservs, forums, IRCs are VERY useful for more involved questions
- The HPC HOWTO <goo.gl/kzlqI>
- The HPC IRC (irc.freenode.net, #uci-hpc)
- Software Carpentry
- Showmedo.com
- Us – Adam, Harry, Joseph.
- BUT!! Unless you ask questions intelligently, you will get nothing but grief.

How to Ask Questions

- Reverse the situation: if you were answering the question, what information would you need?
- Not Science, but it is Logic.
- Include enough info to recreate the problem.
- Exclude what's not helpful or ginormous (use <pastie.org> or <tny.cz>)
- Use text, not screenshots if possible.

This is a bad question:

Why doesn't 'X' work?

A good question

I tried running the new podunk/2.8.3 module this morning and it looks like I can't get it to launch on the Free64 Q. My output files aren't helping me figure out what is wrong.

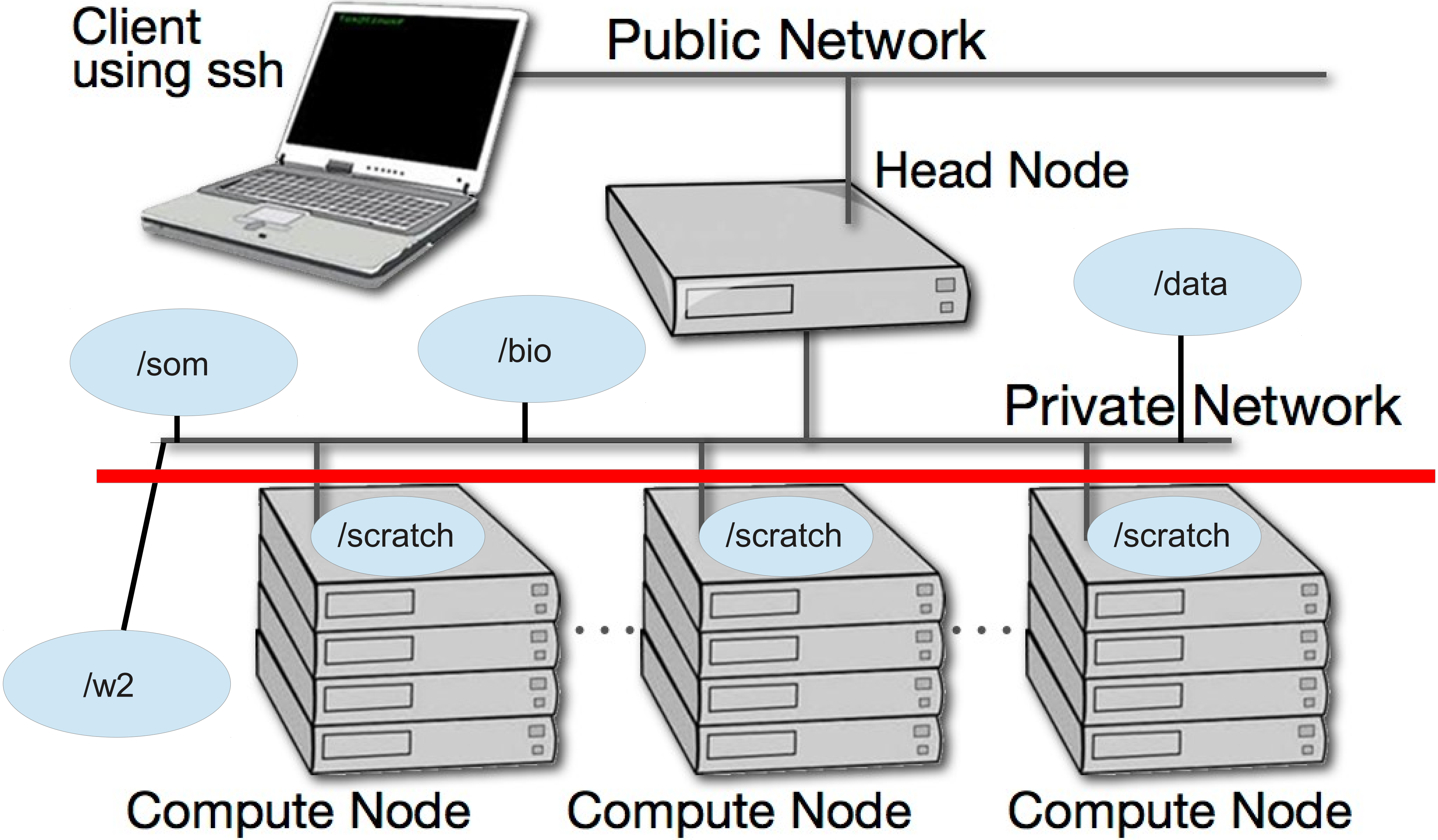
I am working out of
/bio/joeuser/RNA_Seq_Data/M_sexta_RNAseq
and the qsub script is '~/scripts/job12.sh'

When I submit the job, it appears to go thru the scheduler but then dies immediately when it hits the execution node.

I can't find any output to tell me what's wrong.

What is a cluster?

- bunch of big general purpose computers
- running the Linux Operating System
- linked by some form of networking
- have access to networked storage
- that can work in concert to address large problems
- by scheduling jobs very efficiently


Client
using ssh
Public Network
Head Node
/data
/som
/bio
Private Network
/scratch
/scratch
/scratch
/w2
Compute Node
Compute Node
Compute Node

HPC Specifically

- ~ 2500 64bit compute cores
- ~14TB aggregate RAM (fast silicon memory)
- 3/4 PB of storage (1000x slower than RAM)
- connected by 1Gb ethernet (100MB/s), DDR (400MB/s), QDR Infiniband (800MB/s) (per channel)
- uses the Grid Engine scheduler to handle Queueing
- >400 users, of whom 20-100 are online at any time

HPC is NOT

- Your personal machine.
- It's a shared resource
- Pretty well protected against mischief and disaster
- But don't take that as a challenge
- !! → Data is NOT backed up. ← !!

Data Sizes

- Especially with NGS techniques, you'll be crossing the line into BigData.
- BigData is somewhat 'dangerous' due to its bigness.
- Think before you start. You can't predict everything, but you can predict a lot of things.
- We will be discussing BigData in a separate section.

NO BACKUPS

- Data on HPC is not backed up. Agitate to your advisors to get us more $ if you want this.
- Most data is stored on RAID6 storage.
- BUT! Any of that data can disappear at any moment.
- So if it's valuable to you, back it up elsewhere.
- Really, REALLY not kidding.

Logging in with ssh

- Linux is case-SensTiVe, for passwords, filenames, commands, etc.
- Ssh is an encrypted protocol so that info over the connection can't be deciphered by others.
- Underlies 'scp' secure copy
- Also 'sshfs' which allows you to attach your filesystem to HPC (or vice versa).
- You MUST use ssh to connect to HPC.
- Use 'logout' or '^D' to logout.

Commandline Cons

- The tyranny of the blank page
- No visual clues
- Type vs click
- Have to know what to type

Commandline Pros

- It doesn't get much worse than this.
- When you do learn it, you'll know it and it probably won't change for the rest of your life, unless they perfect mind control..
- It's an efficient way of interacting with the computer (which is why it's survived for 50yrs).
- You can use it to create simple, but very effective pipelines and workflows.

Screen & Byobu

- If you need to maintain a live connection for some reason, use 'byobu or screen'.
- It allows you to multiplex and maintain connections in a single terminal window.
- Somewhat unintuitive interface but very powerful.
- You know about cheatsheets

x2go

- Linux uses X11 for graphics
- X11 is very chatty, high bandwidth, sensitive to network hops/latency.
- If you need graphics programs on HPC, use x2go vs native X11.
- x2go is described in the Tutorial & HOWTO.

The Shell

- Program that intercepts and translates what you type, to tell the computer what to do.
- What you will be interacting with mostly.
- HPC shell is '***bash***'.
- A ***qsub script*** is just a series of bash commands that sets up your resource requirements, PATHs, executes your jobs, and does the post-processing.

Know the shell, Embrace the shell

- If you don't get along with the shell, life will be hard.
- Before you submit anything to the cluster via qsub, get it going in your login shell.
- You're welcome to start jobs in on the login node, but don't let them run long (<15m should be enough).

Commandline Editing

- Since you'll be spending a lot of time fighting with the cmdline, make it easy on yourself.
- Learn cmdline editing to edit previous cmds
- Up/Down arrow keys scroll thru cmd history
- L/R arrow keys scroll by 1 char
- ^ makes L/R arrow jump by a word
- Home, End, Insert, Delete keys work (except Macs lack 'Delete' keys
- ^u kills from cursor left; ^k kills from cursor to right

STDIN, STDOUT, STDERR

- STDIN is usually the keyboard, but...
- STDOUT is usually the screen, but...
- STDERR is *also* usually the screen, but...
- All can be redirected all over the place
- to files, to pipes, combined, split (by 'tee'), etc
- More on this later.

Files & Directories

- Files & folders much like on Mac & Win
- Except...
- Names are case-sensitive, 256 char long
- 'Folders' → 'Directories' , separated by '/'
- No spaces in names(*)
- . means 'in this dir'
- ~ means 'home dir'
- A leading '/' means 'from the root dir'

Foreground & background jobs

- Foreground (fg) jobs are connected to the terminal. You kill a fg job with '^C'.
- Background (bg) jobs have been disconnected from the terminal.
- Send a job to the bg by appending '&'
- Recall a job to the fg with 'fg'.
- Send a fg job to the bg with '^z' (suspend), then 'bg'.

Pipe = |

- Works with STDIN/OUT/ERR to create 'pipelines'
- Very similar to plumbing; can add 'tee's to introduce splits
- STDOUT of one program goes to the STDIN of another command whose STDOUT goes to the STDIN of another program ad infinitum.
- Sooooo......

Pipe Example

```
$ w |cut -f1 -d ' ' | sort | egrep -v "(^$|USER)" | uniq -c | wc
```

w spits out who is on the system right now

cut -f1 -d ' ' chops out the 1st field (the user), based on the space token

sort sorts the usernames alphabetically

egrep -v "(^$|USER)" filters out both blank lines and lines with 'USER'

uniq -c counts the unique lines

wc word-counts that output.

How to use commands

- 'cmd -h'
- 'cmd --help'
- 'man cmd'
- 'info cmd' (but you hope not)
- And ….. Google...

Finally, commands

- ls [many options] = list fil<tab><tab>
- cd [up or down] = change directory
- find [from] -name [name] = find files
- locate [name] = where is this file?
- tree [options] = show the dir tree
- file [name(s)] = what is this?
- du = disk usage
- df = disk free
- less [names] = view files
- cols [file] = view file in columns

Creative/destructive commands

- mkdir [name] – make a dir
- rmdir [name] – remove a dir
- mv [from] [to] = move or rename
- cp [from] [to] = copy file(s)
- rm [file] = delete file(s)
- wget [URL] = get a file from the Internet
- curl -O [URL] = ditto, but on steroids

More informational cmds

- mc = Midnight Commander
- [ah]top = top CPU using processes
- time [command] = how long does it take?
- [aef]grep [regex] [files] = find regex in files
- cat [files] = print the files to STDOUT
- head/tail [files] = dump the top / bottom of files

More informational cmds

- mc = Midnight Commander
- [ah]top = top CPU using processes
- time [command] = how long does it take?
- [aef]grep [regex] [files] = find regex in files
- cat [files] = print the files to STDOUT
- head/tail [files] = dump the top / bottom of files

Regular Expressions

- Among the most powerful concepts in pattern matching
- Simple in concept, NASTY in implementation
- Among the ugliest / most confusing things to learn well
- But pretty easy to learn the simple parts.
- But you NEED to learn it – it's central to computers and especially biology

Regexes

- Simplest form is called globbing (a*)
- Mix it up (a*.txt)
- A bit more (a*th.txt)
- Can be MUCH more complex:
- [aeiou] = any of 'aeiou'
- F{3,5} = 3-5 'F's
- H+ = 1 or more 'H's
- . = any character
- Also classes of characters (#s, alphabetic, words)

Archiving/Compression

- tar = std archive format for Linux
- zip = common archive format, from Windows
- gzip/unzip = common compressed format
- bzip2/bunzip2 = another compressed format
- pigz = parallel gzip (for large files)
- pbzip – parallel bzip2 (ditto)

Editors: simple → complex

Text-based:

nano, joe, vi/vim, emacs

GUI-based:

gedit, nedit, kate, jedit, emacs

(choose one and learn it)

Customize Your Environment

- Change your prompt to something useful to you (and to us):
- `PS1="\n\t \u@\h:\w\n\! \$ "`
- Set aliases (`alias nu=”ls -lt |head -22”`)
- Set Environment Variables (`export EDITOR=joe`)

Disk Quotas

- Unlike BDUC, HPC enforces disk quotas
- You can only have so much space.
- 20GB for most users
- More for Condo owners or groups who have bought extra disk space.
- AGAIN: the fact that you are allowed 20 GB or 200GB does not mean that it's SAFE. It is not.

Moving Data to / from HPC

- Covered in detail in HPC USER HOWTO, which references: goo.gl/XKFEp
- scp, bbcp, netcat/tar on Mac, Linux.
- WinSCP, Filezilla, CyberDuck,FDT on Win
- Everyone should know how to use rsync. Not the easiest to learn, but very powerful.
- rsync GUIs for Linux, Windows, MacOSX

Programs, finally

- 3 main sets of programs
- Your personal set (later)
- The default system utilities
- The module system programs

How to find them

- locate <partial search term>
- apropos <search term>
- na<tab><tab> → name
- yum search <search term>   # CentOS
- module avail (will dump all modules)
- Google
- Ask us.

The scheduler (GE)

- Just another program that juggles requests for resources
- Make sure a program is working on a small set of test data.
- Need a short bash script (aka ***qsub script***) to tell the GE what your program needs to run.
- Can improve the performance of your program in a variety of ways (staging data, running in parallel, using array jobs, etc)

Solving Problems

- Reduce the scope of the problem
- What in particular is failing?
- Debug in the ***login shell*** rather in qsub shell as long as possible.
- Things will start faster and fail faster in the login shell.
- (almost) anything in a qsub script can be pasted into a bash shell and have the same effect.
- Think of your login shell as your home and the cluster as a slightly sketchy bar.

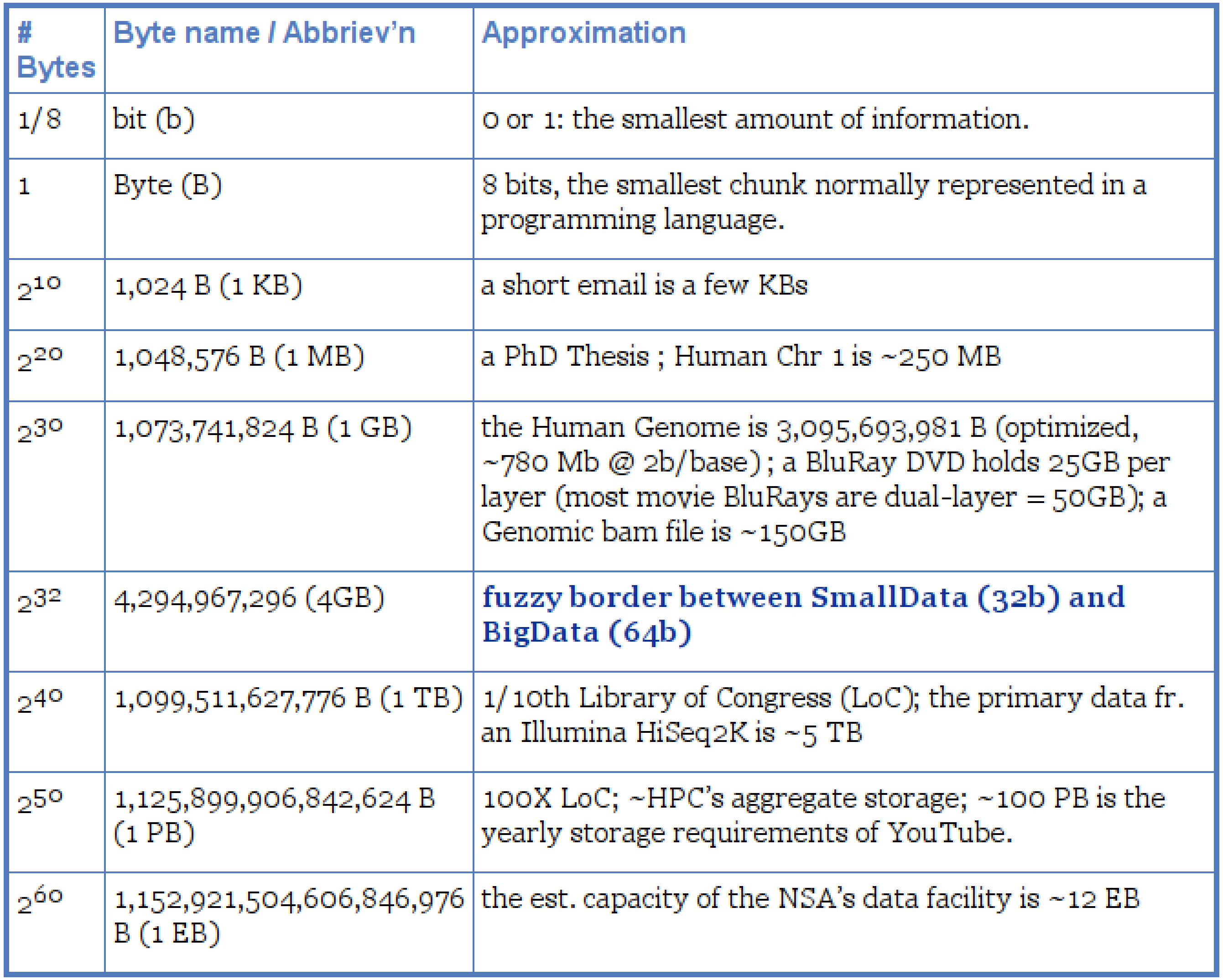
A simple qsub script

```
#!/bin/bash
# Usage: sleeper.sh [seconds]
#        default for time is 60 seconds
#$ -N Sleeper1
#$ -S /bin/bash
# Make sure that the .e and .o file arrive in the working directory
#$ -cwd
#Merge the standard out and standard error to one file
#$ -j y
/bin/echo Here I am: `hostname`. Sleeping now at: `date`
/bin/echo Running on host: `hostname`.
/bin/echo In directory: `pwd`
/bin/echo Starting on: `date`
#$ -m be
#$ -M hmangala@uci.edu
time=60
if [ $# -ge 1 ]; then
   time=$1
fi
sleep $time
echo Now it is: `date`
```

BigData

- Volume
  - Scary sizes, and getting bigger
- Velocity
  - Special approaches to speed analysis
- Variety
  - Domain-specific standards (HDF5, bam/sam, FITS), but often aggregations of unstructured data
- BigData Hints for Newbies

How Big?



# Bytes	Byte name / Abbriev'n	Approximation
1/8	bit (b)	0 or 1: the smallest amount of information.
1	Byte (B)	8 bits, the smallest chunk normally represented in a programming language.
2^{10}	1,024 B (1 KB)	a short email is a few KBs
2^{20}	1,048,576 B (1 MB)	a PhD Thesis ; Human Chr 1 is ~250 MB
2^{30}	1,073,741,824 B (1 GB)	the Human Genome is 3,095,693,981 B (optimized, ~780 Mb @ 2b/base) ; a BluRay DVD holds 25GB per layer (most movie BluRays are dual-layer = 50GB); a Genomic bam file is ~150GB
2^{32}	4,294,967,296 (4GB)	**fuzzy border between SmallData (32b) and BigData (64b)**
2^{40}	1,099,511,627,776 B (1 TB)	1/10th Library of Congress (LoC); the primary data fr. an Illumina HiSeq2K is ~5 TB
2^{50}	1,125,899,906,842,624 B (1 PB)	100X LoC; ~HPC's aggregate storage; ~100 PB is the yearly storage requirements of YouTube.
2^{60}	1,152,921,504,606,846,976 B (1 EB)	the est. capacity of the NSA's data facility is ~12 EB

Inodes and ZOTfiles

- Inodes contain the metadata for files and dirs
- Inodes are pointers to the data
- Regardless of size, a file needs at least one inode to locate it.
- A file of 1 byte takes up the same minimum inode count as a file of 1TB
- DO NOT USE ZOTFILES!!

Streaming Reads & Writes

- Think of a pack of cards....

Pointless Data Replication

- How informative is this?
- How informative is this?
- How informative is this?
- How informative is this?
- How informative is this?
- vs
- How informative is this? [5x]
- __LINE__

Editing BigData

- Use format-specific utilities to view such files and hash values to check if they’re identical to what they should be.
- Try not to be the member of your class who tried to open a 200GB compressed data file with nano.

[De]Compression

- If your applications can deal with compressed data, KEEP IT COMPRESSED.
- If they can't, try to use pipes (|) to decompress in memory and feed the decompressed stream to the app.
- Use native utilities to examine the compressed data (zcat/unzip/gunzip, grep, archivemount, Vitables, ncview, etc.

How to Move BigData

- Don't.
- Otherwise, plan where your data will live for the life of the analysis, have it land there, and don't move it across filesystems.
- Don't DUPLICATE DUPLICATE DUPLICATE BigData
- See: How to Move Data
  - rsync for modified data
  - bbcp for new transfers of large single files, regardless of network
  - tar/netcat for deep/large dir structures over LANs
  - tar/gzip/bbcp to copy deep/large dir structures over WANs

Checksums

- They work. Choose one and use it.
- md5sum / jacksum
- Use MANIFEST files & copy them along with the data files.
- See Checksum example

Processing Approaches

- Files (HDF5, bam/sam) and specialized utilities (nco/ncview, [Py/Vi]tables, R, Matlab)
- Relational Dbs (SQLite, Postgres, MySQL)
- NoSQLs (MongoDB, CouchDB)
- Binary Dumps (Perl's Data::Dumper, Python's pickle)
- Non-Storage (pipes, named pipes/FIFOs, sockets)
- Keep it RAM-resident.

Big, but not forever

- HPC is not backed-up.
- Cannot tolerate old, unused BigData.
- RobinHood is looking for your old BigData.
- Please help us by doing your own data triage.
- Ask your PIs to bug our boss to provide more resources so we can provide more resources.