CMS Computing Model Simulation

Stephen Gowdy/FNAL

Overview

- Want to look at different computing models for HL-LHC
  - To use caching (eg CDN, NDN)
  - Where to place caches
  - How large they need to be
- Discussion with others to possibly collaborate
- Writing a basic Python simulation
  - Can consider to change to C++ if better performance is needed

Simulation

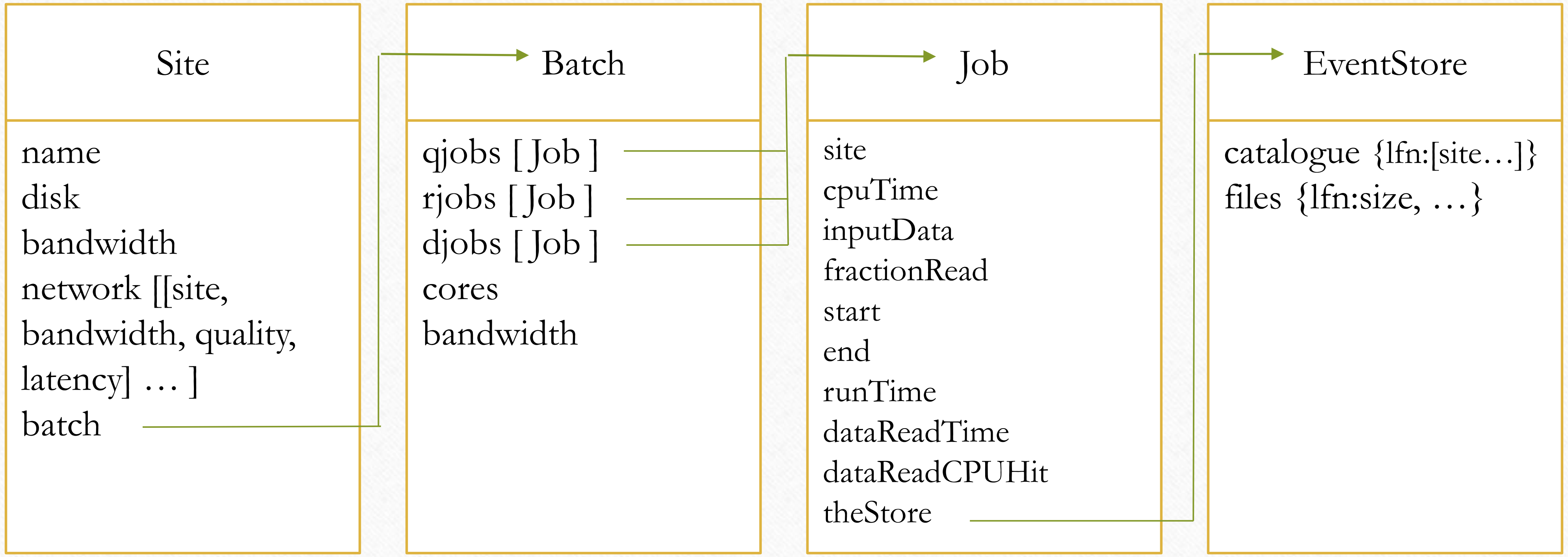
- Time driven discrete simulation
  - 100 seconds used as time slices currently
- Takes account of slots in sites
- Allows for transfers between sites
- Code is in https://github.com/gowdy/sitesim

Methodology

- Flat files read to load in site, network, job and file information
- Setup sites and links
- Next setup catalogue of data
- Read in simulation parameters for CPU efficiency, remote read penalty and file transfer rates
- Start processing jobs in sequence
  - Use list of jobs from dashboard to feed simulation
  - See how it performs to process current jobs

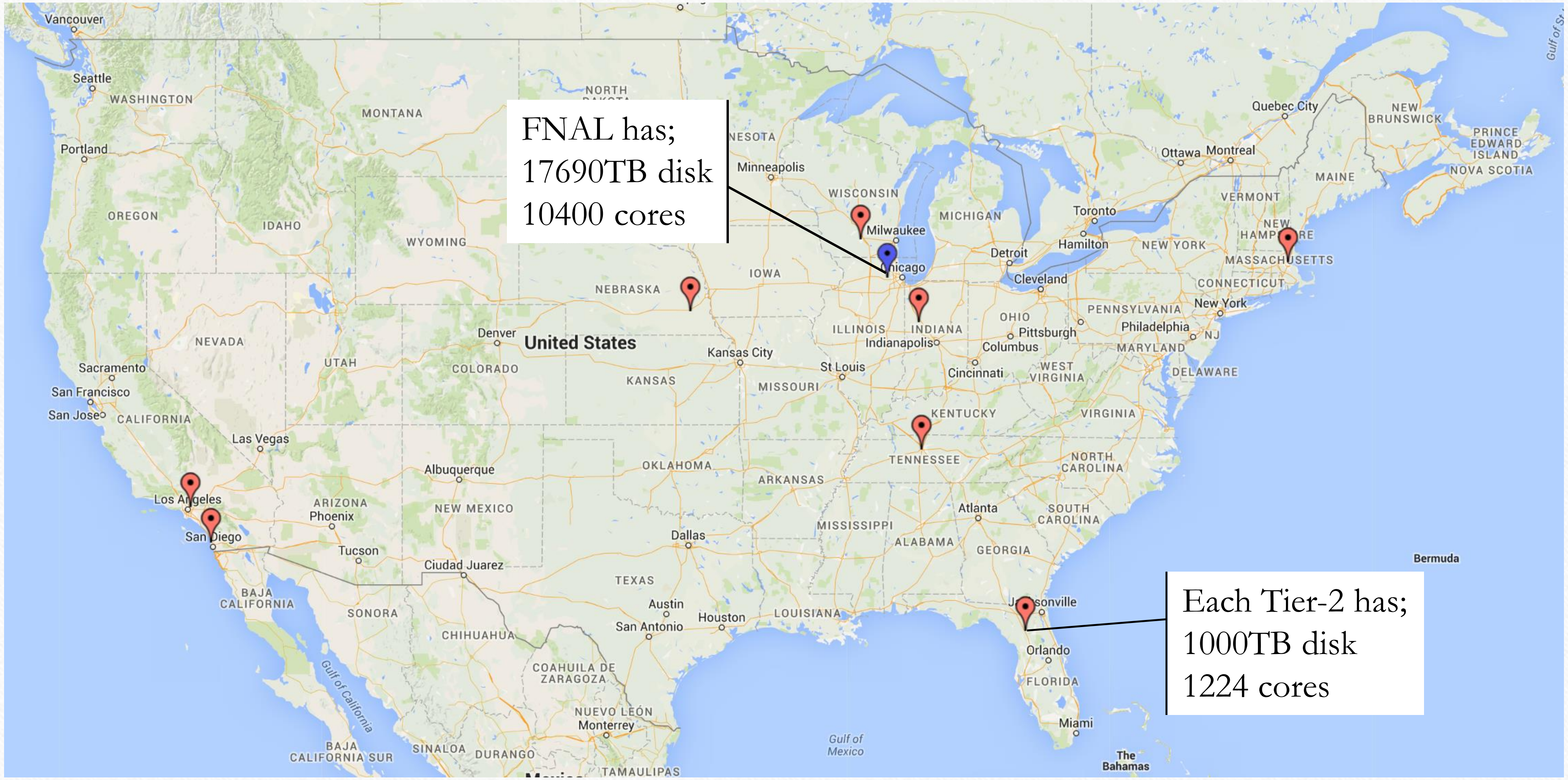
Simulation Parameters

- CPU Efficiency derived from actual jobs
- Latency between sites guessed at the moment
- CPU Efficiency penalty when reading remotely
  - 0ms: 0, >=1ms 5%, >=50ms 20%
- Single file transfer maximum speed
  - 0ms: 10Gbps, >=1ms 1Gbps, >=50ms 100Mbps, >=100ms 50Mbps


Site
name
disk
bandwidth
network [[site, bandwidth, quality, latency] …]
batch
Batch
qjobs [Job]
rjobs [Job]
djobs [Job]
cores
bandwidth
Job
site
cpuTime
inputData
fractionRead
start
end
runTime
dataReadTime
dataReadCPUHit
theStore
EventStore
catalogue {lfn:[site…]}
files {lfn:size, …}

Site Information

- Extracted from SiteDB pledge database
  - Use information for 2014, most recent update
  - If site has no pledge just assume 10TB and 100 slots
    - Tier-2s default is larger, should probably update
  - No internal bandwidth information so assume 20GB/s at all sites
- Recently only considering US Tier-1 and Tier-2 sites
  - Sizes taken by hand from REBUS (could probably automate also)
  - Vanderbilt assumed to be the same as others


FNAL has;
17690TB disk
10400 cores

Each Tier-2 has;
1000TB disk
1224 cores

Job Information

- Site, Start Time, Wall Clock, CPU time, files read
- Extracted job information from dashboard
  - A week from 15th to 22nd February
  - About 5% of jobs have no site information (discarded)
  - About 33% have no CPU time (derived from wall clock)
  - <<1% have no start time (use CPU time before end time)
  - <<1% have no input file defined (discarded)
- Will compare wall clock in simulation with actual for quality of simulation check
- Compare overall simulated wall clock time to compare different scenarios

File Information

- Extract network mesh from PhEDEx
  - Using the links interface
  - Also get reliability information
    - If not present assumed 99%
  - No actual transfer rate information available for links
    - Use what is available to get a number between 1GB/s and 10GB/s, not at all accurate. Default 1GB/s.
- Extract file location and size information from PhEDEx
  - No historical information is available
  - When updating job information need to get an update for file locations
  - Only get information on files used by jobs
    - Some of jobs read a file outside the US, place copy at FNAL to allow job to work when considering only US sites

Startup output when only using US T1 and T2 sites;

```
$ python python/Simulation.py
Read in 9 sites.
Read in 72 network links.
Read in 99266 files.
Read in 279178 locations.
Read in 3 latency bins.
Read in 4 transfer bins.
Read in 10 job efficiency slots.
About to read and simulate 113899 jobs...
…
```

Caching

- Need to add different caching strategy later
  - Cache hierarchy
- Including cache cleaning if getting full
- Currently simulation allows no transfers, or transfers. Also can discard transfers.
- Won’t transfer if there is no space available at a site
- Implement different models
  - With new version of xrootd can read while still transferring

Scenarios Considered

- Run standard set of 56949 US jobs
  - Each job ran twice to spread load across all Tier-2 sites more evenly

1. Run with a similar situation to today
   - Vast majority of data already placed at execution site
   - Small number of jobs will transfer data from another site (usually FNAL)
2. Only use a local cache, data initially at FNAL
3. Only read data from FNAL, no local copy (no local disk needed)

Vary Input Parameters

- Total wall clock time used in billions of seconds
- Each box has three values: Preplaced Data/Transfer File/Remote Read

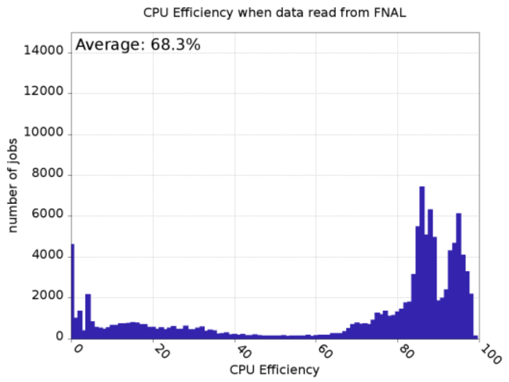
	Half CPU Hit	Normal CPU Hit	Double CPU Hit
Half Tran. Speed	2.77/3.32/3.78	2.77/3.32/3.94	2.77/3.32/4.25
Normal Tran. Speed	2.77/3.32/3.78	2.77/3.32/3.94	2.77/3.32/4.25
Double Tran. Speed	2.77/3.32/3.78	2.77/3.32/3.94	2.77/3.32/4.25

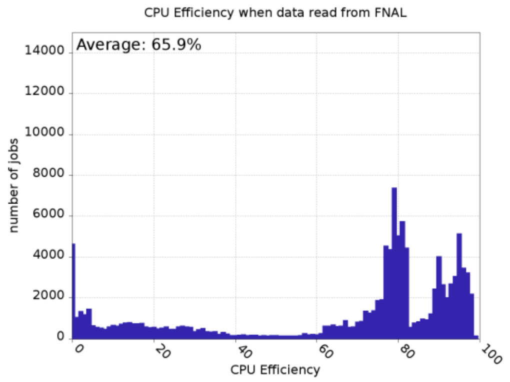
- There is a very small difference in the Transfer File time with the change in transfer speed

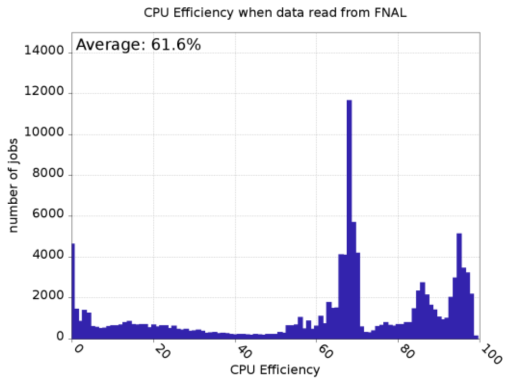
Plots from running with different parameters

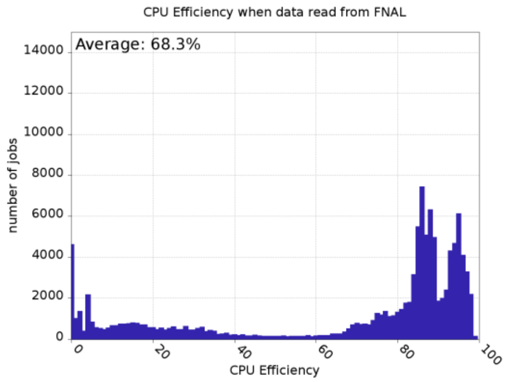
- Grid of three by three graphs, similar to previous table
  - Left to right vary Remote Read penalty
    - Half: 0ms: 0, >=1ms 2.5%, >=50ms 10%
    - Normal: 0ms: 0, >=1ms 5%, >=50ms 20%
    - Double: 0ms: 0, >=1ms 10%, >=50ms 40%
  - Top to bottom vary maximum single file transfer rate
    - Half: 0ms 5Gbps, >=1ms 500Mbps, >=50ms 50Mbps, >=100ms 25Mbps
    - Normal: 0ms 10Gbps, >=1ms 1Gbps, >=50ms 100Mbps, >=100ms 50Mbps
    - Double: 0ms 20Gbps, >=1ms 2Gbps, >=50ms 200Mbps, >=100ms 100Mbps

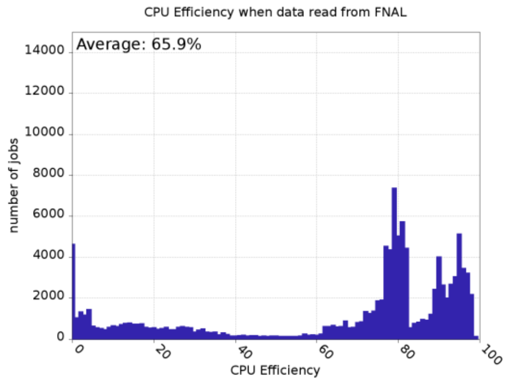
CPU Efficiency when Data Read from Fermilab

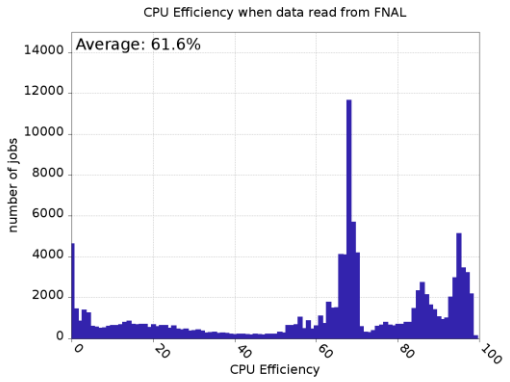


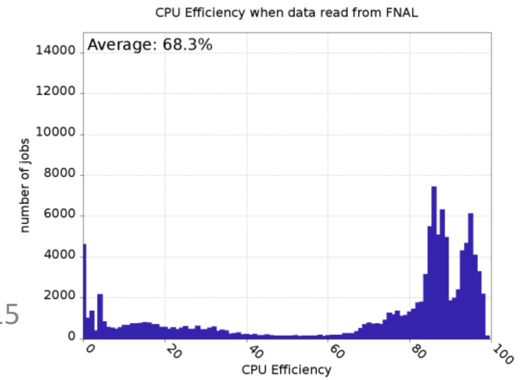


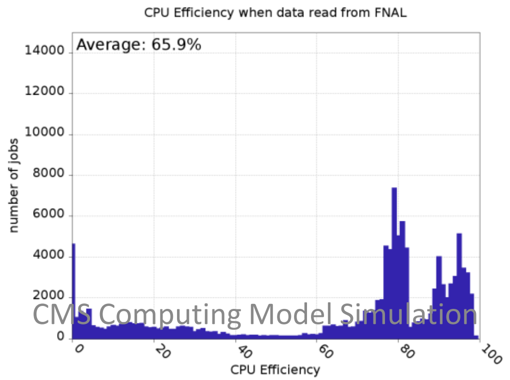


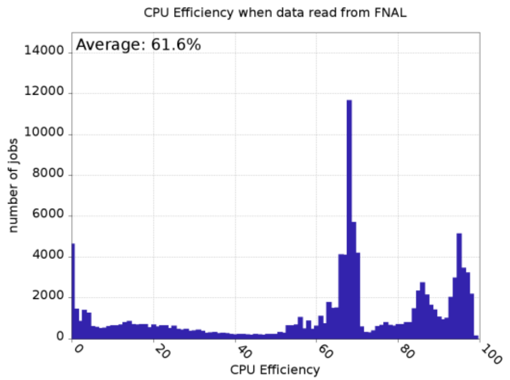




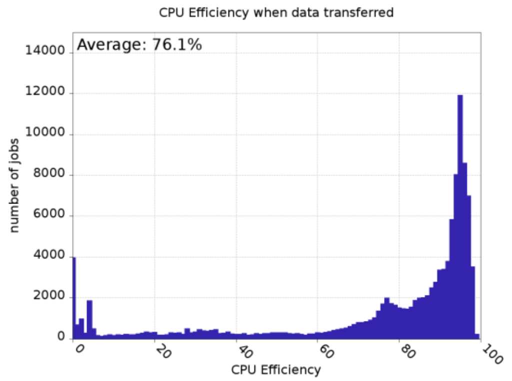


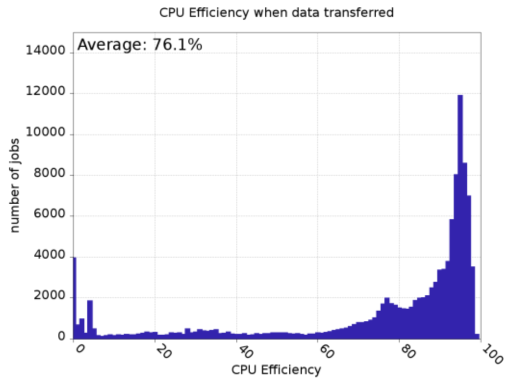


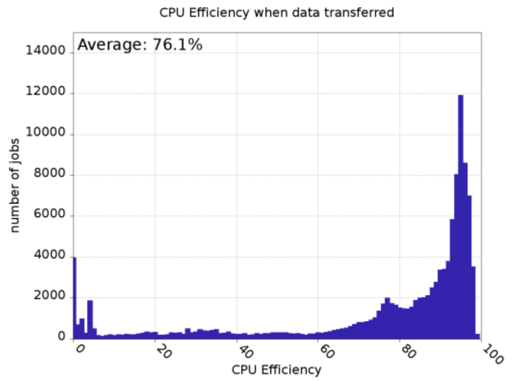


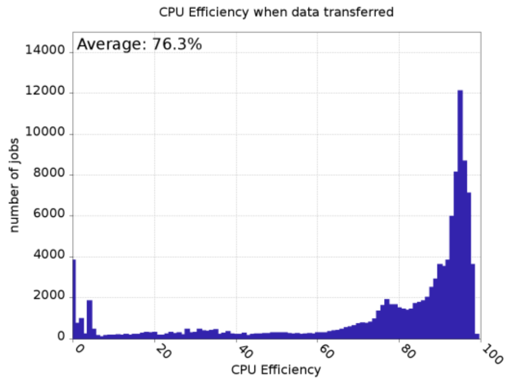


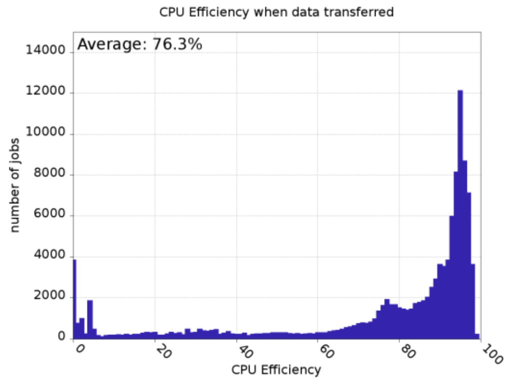
CPU Eff. when data copied from Fermilab

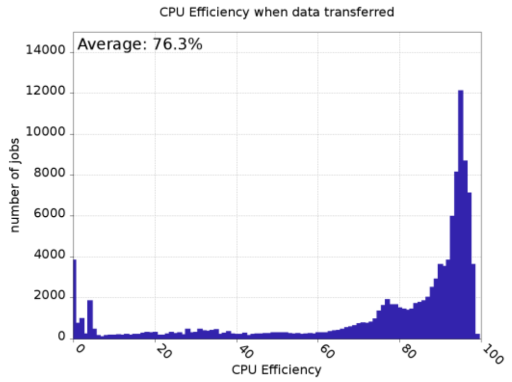


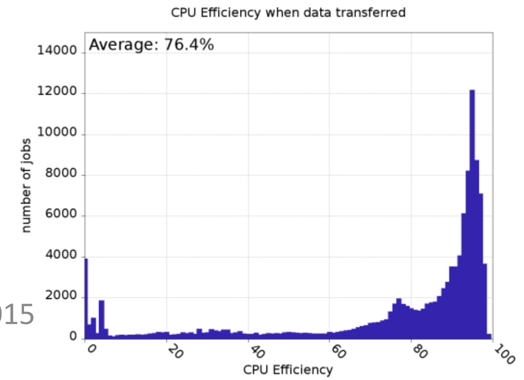


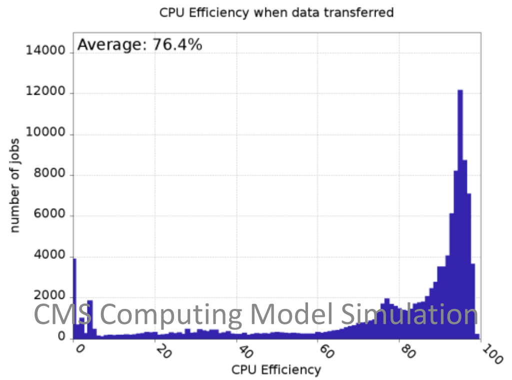


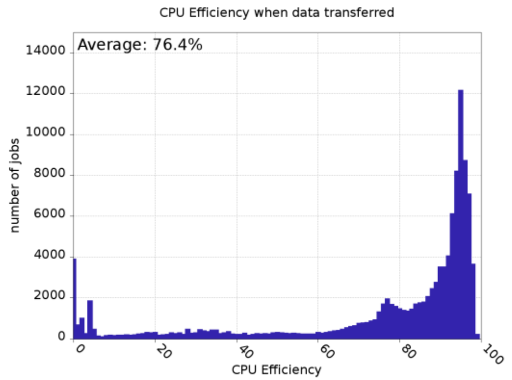




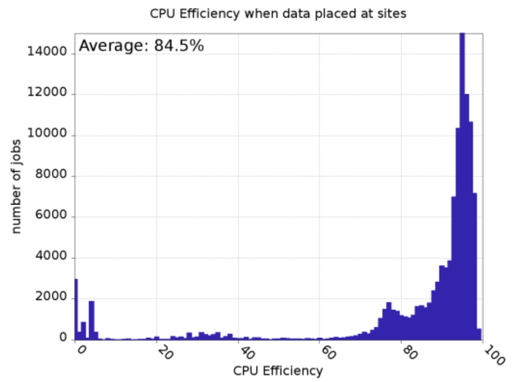


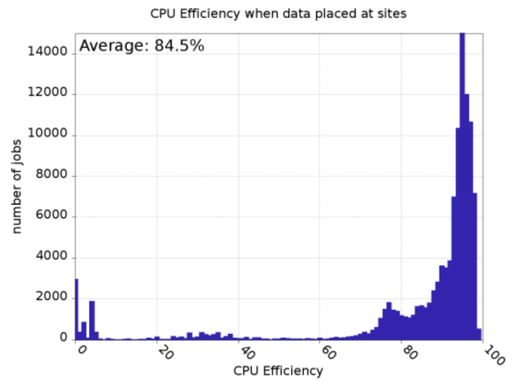


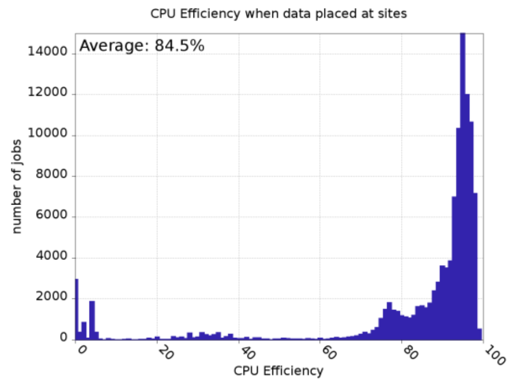


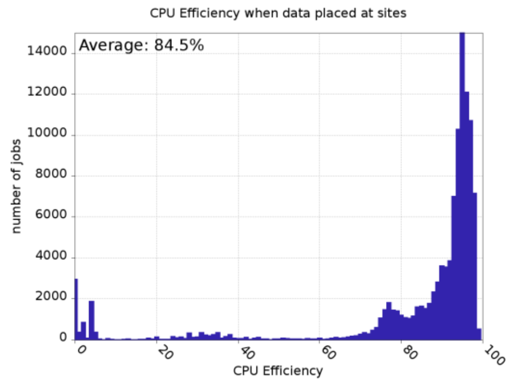


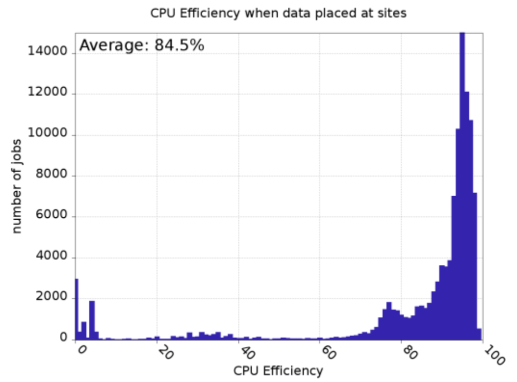
CPU Efficiency when data mostly preplaced

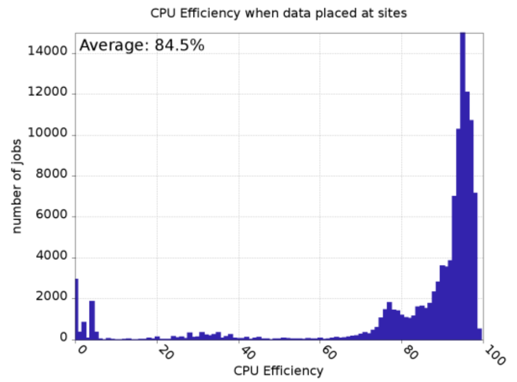


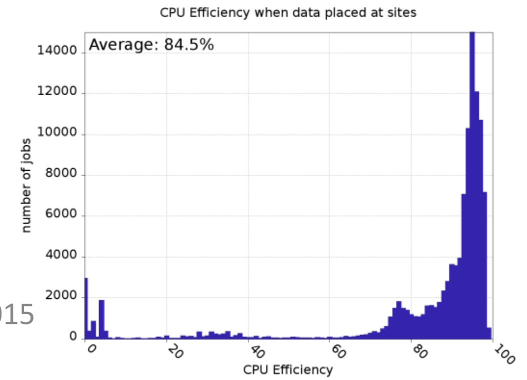


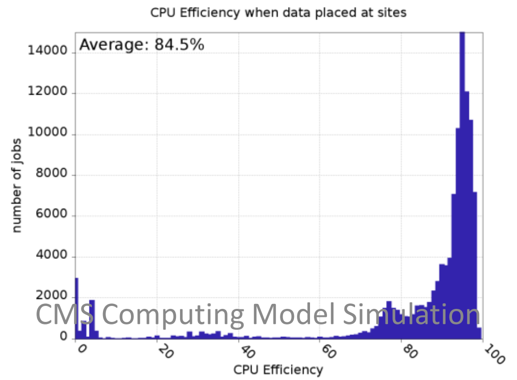


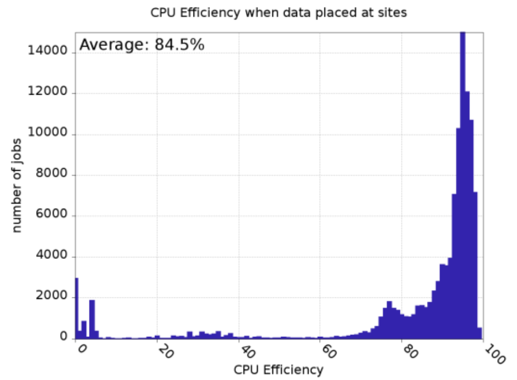




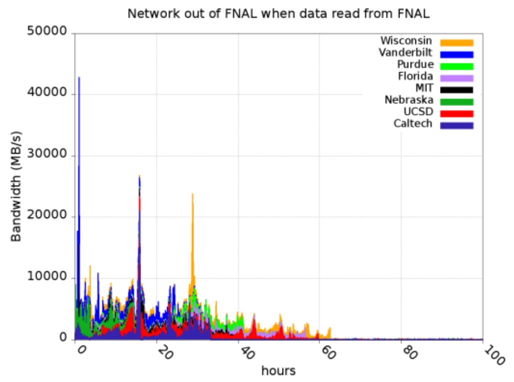


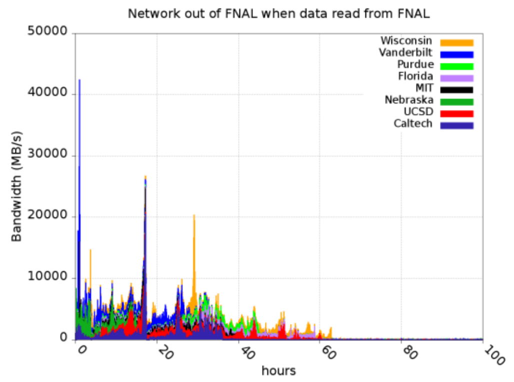


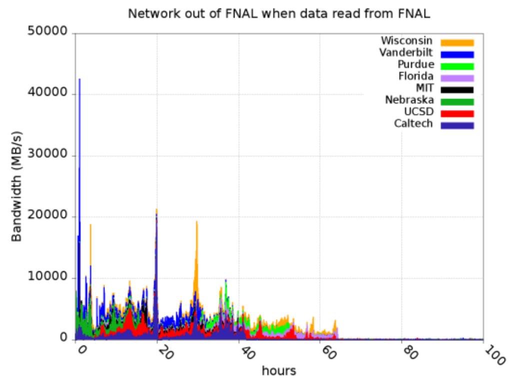


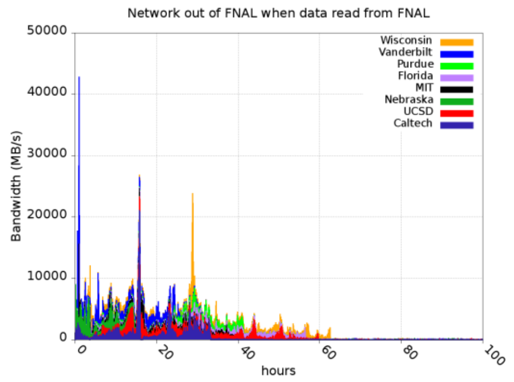


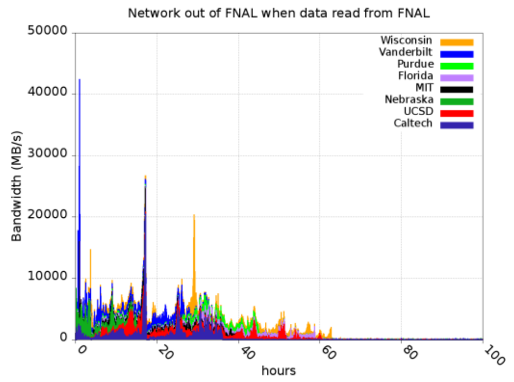
Rate from FNAL when data read from FNAL

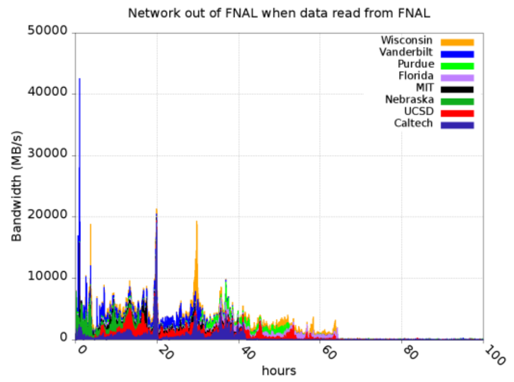


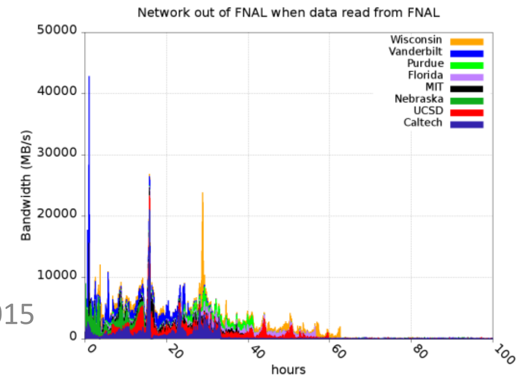


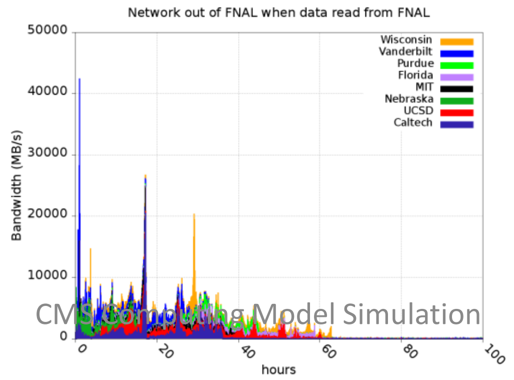


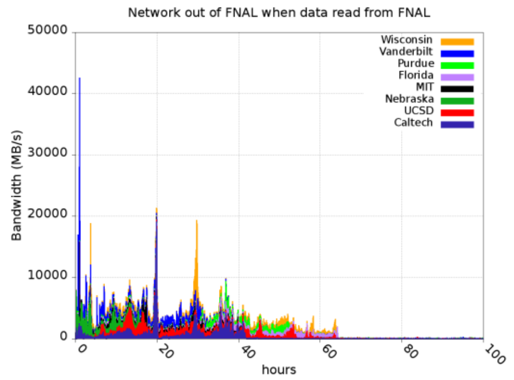




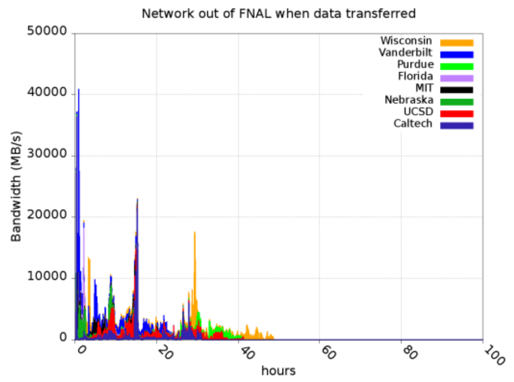


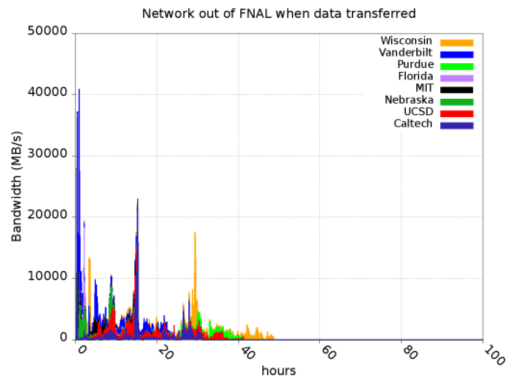


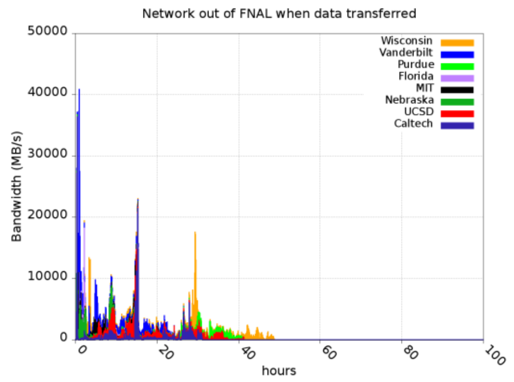


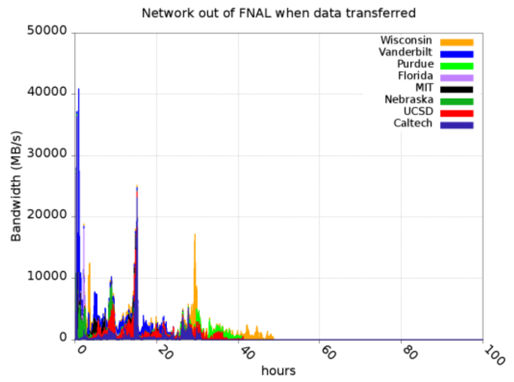


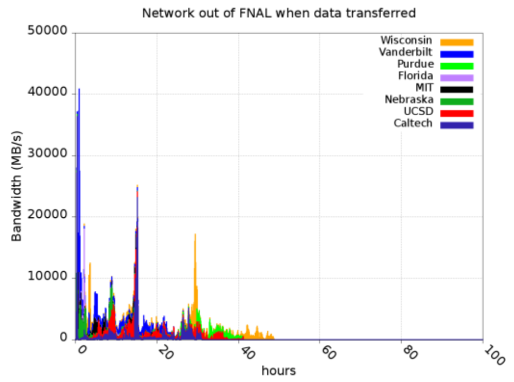
Rate from FNAL when data copied from FNAL

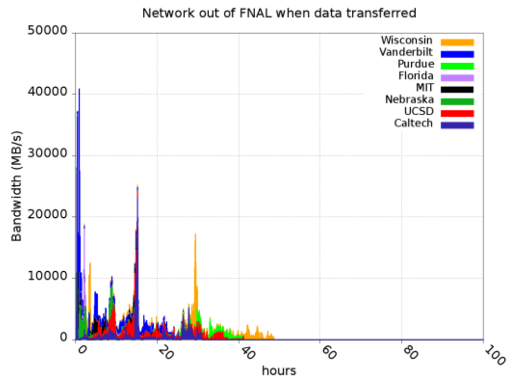


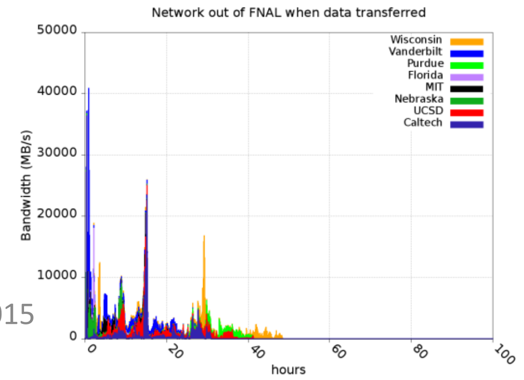


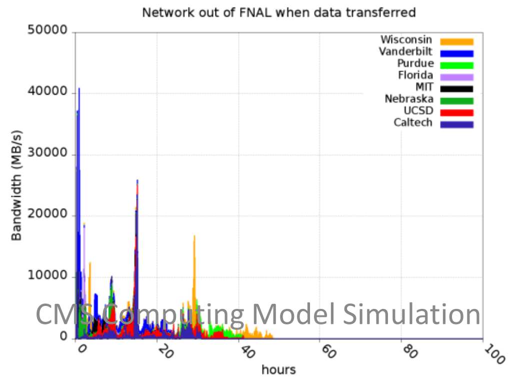


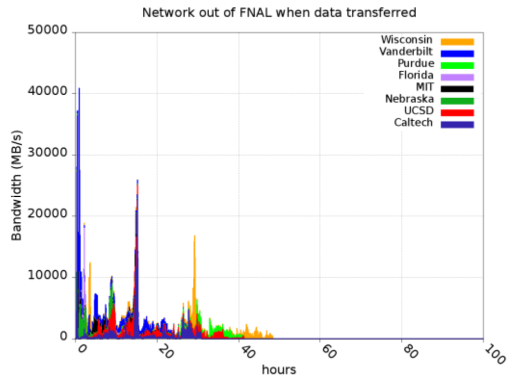




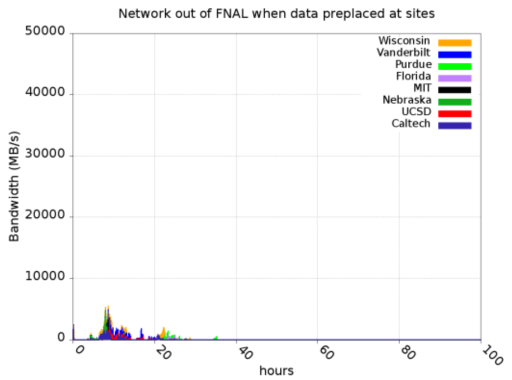


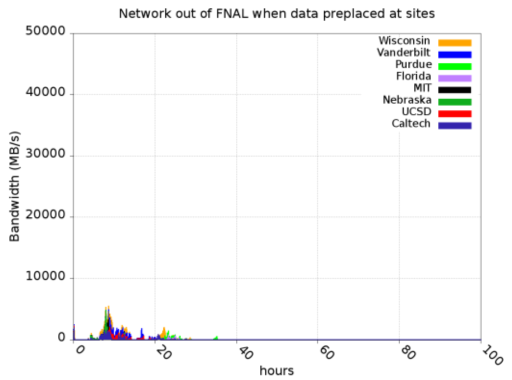


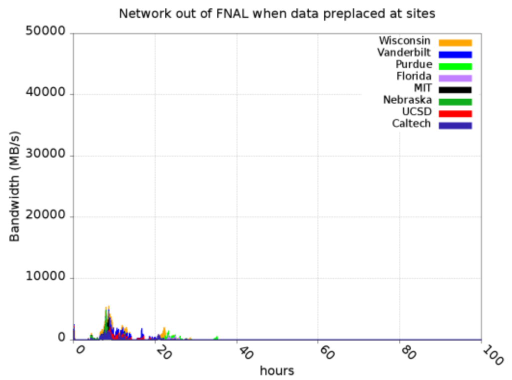


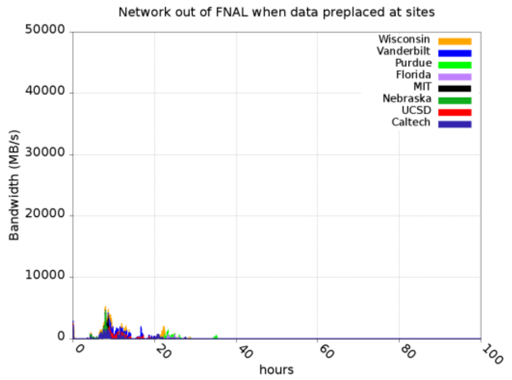


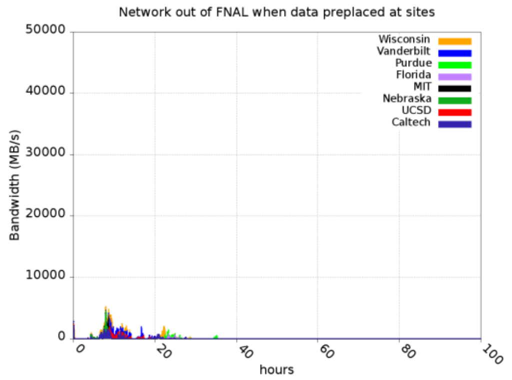
Rate from FNAL when data mostly preplaced

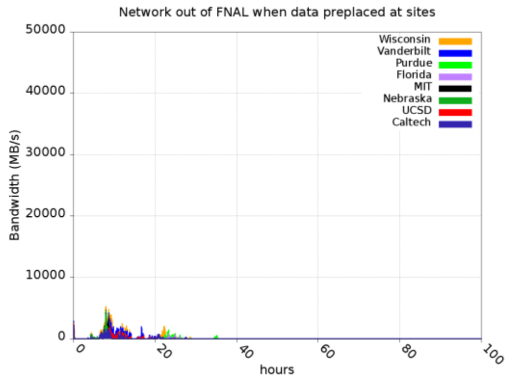


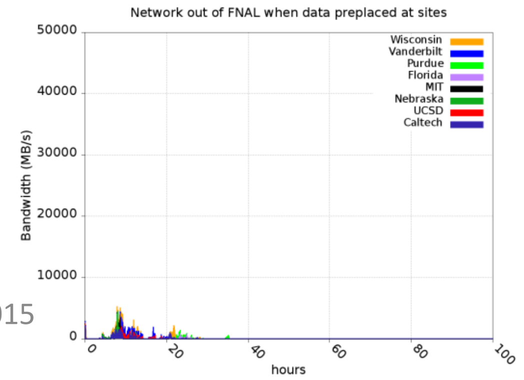


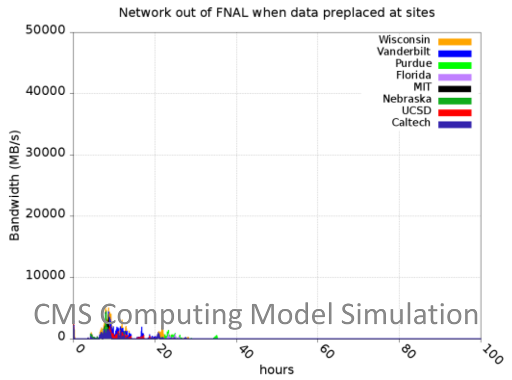


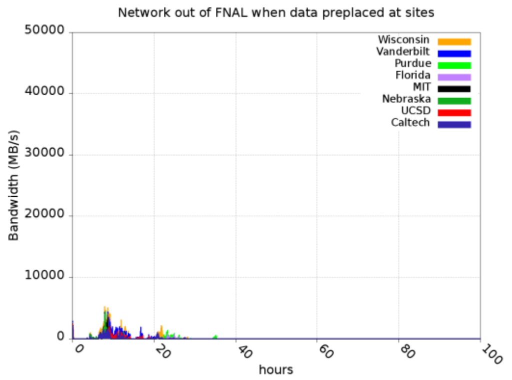




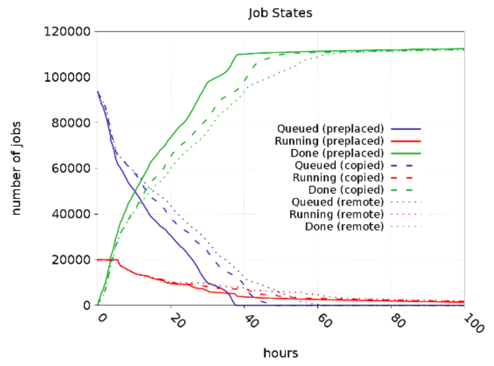


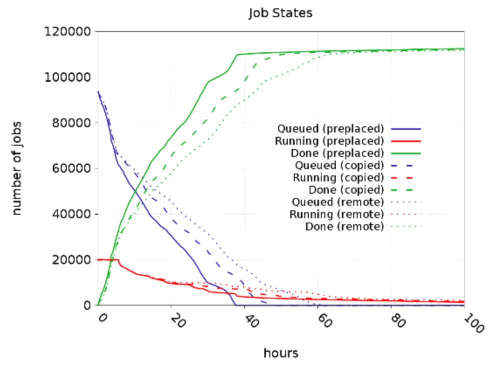


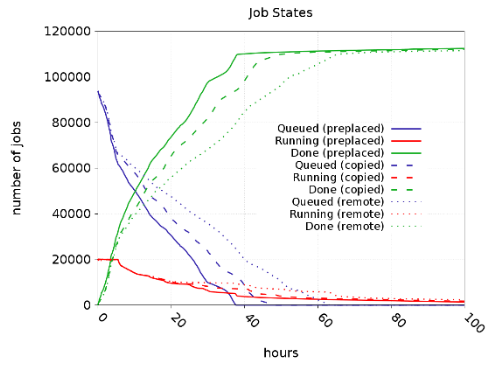


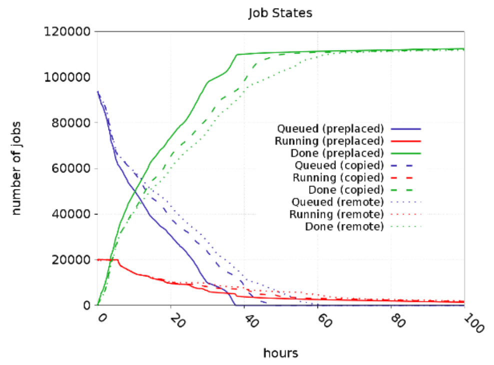


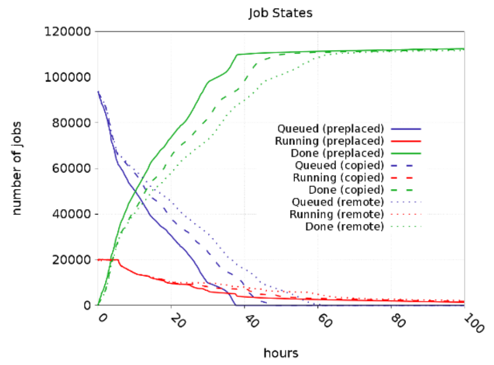
Job States

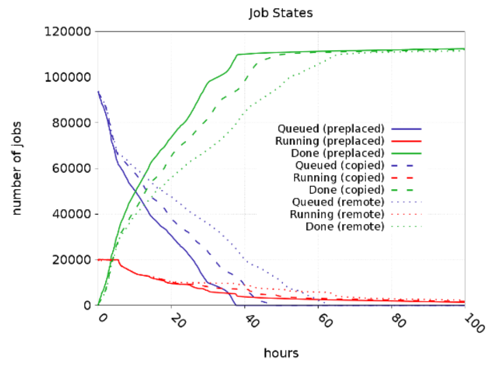

Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)

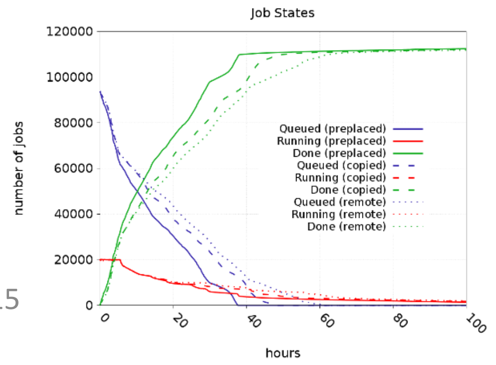

Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)

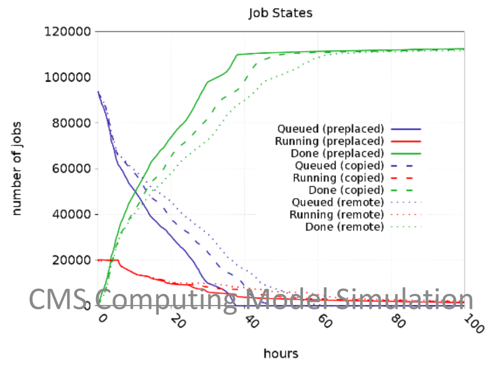

Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)

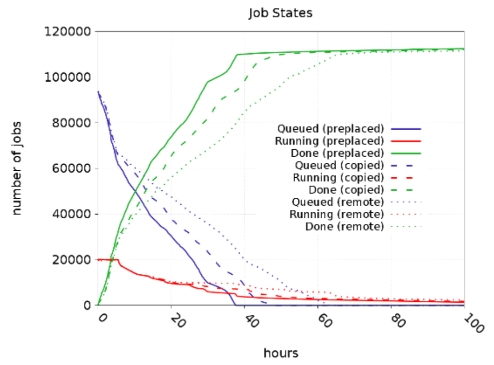

Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)


Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)


Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)


Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)


Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)

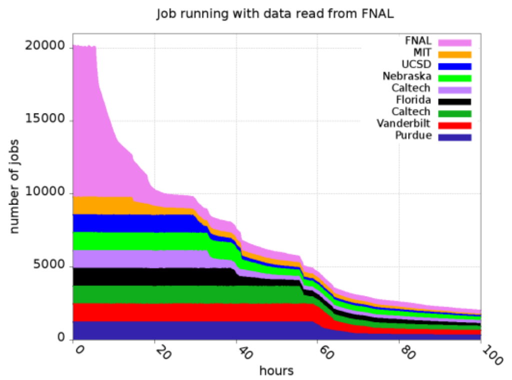

Job States
number of jobs
hours
0
20000
40000
60000
80000
100000
120000
0
20
40
60
80
100
Queued (preplaced)
Running (preplaced)
Done (preplaced)
Queued (copied)
Running (copied)
Done (copied)
Queued (remote)
Running (remote)
Done (remote)

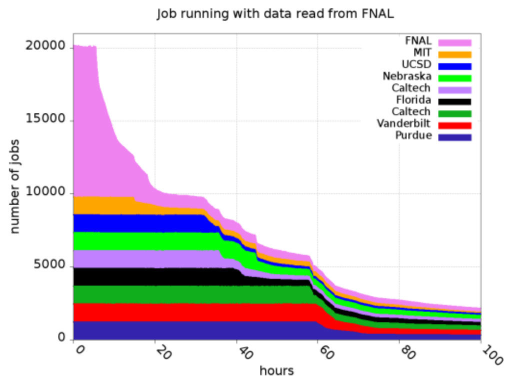
Summary

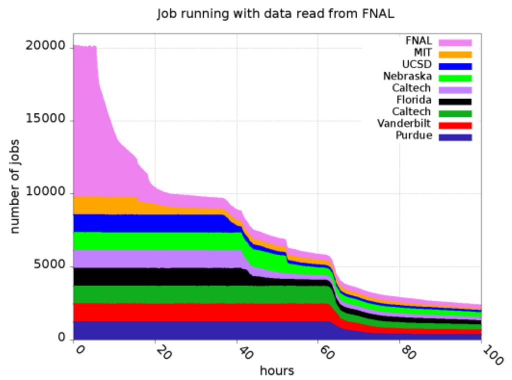
- Can simulate CMS computing system
- Concentrating on US infrastructure, simpler system to understand, and perhaps experiment on
  - Eg Turn off local disk access for a short amount of time
- Can use current infrastructure to determine input parameters better
- Scale up job throughput to capacity of system

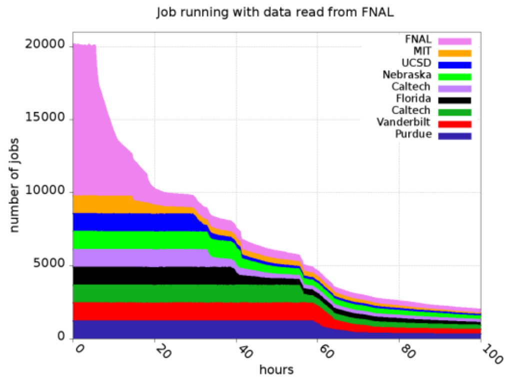
BACKUP SLIDES

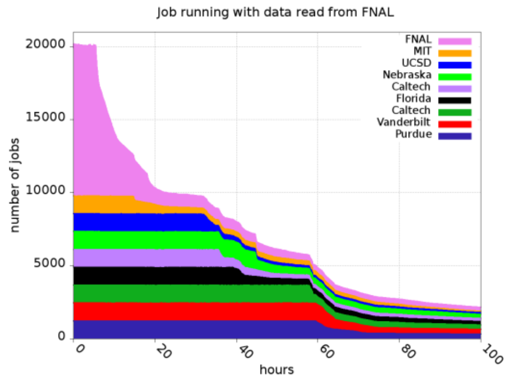
Jobs running when data read from FNAL

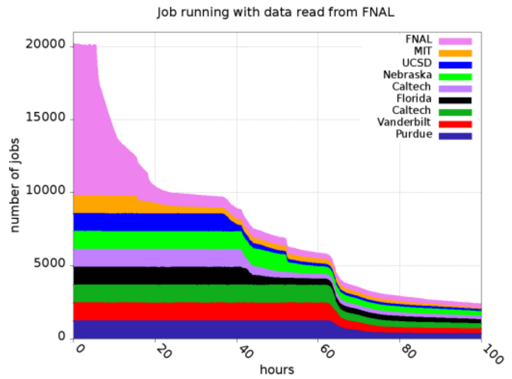


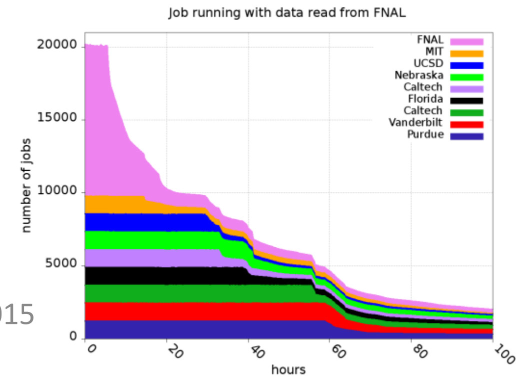


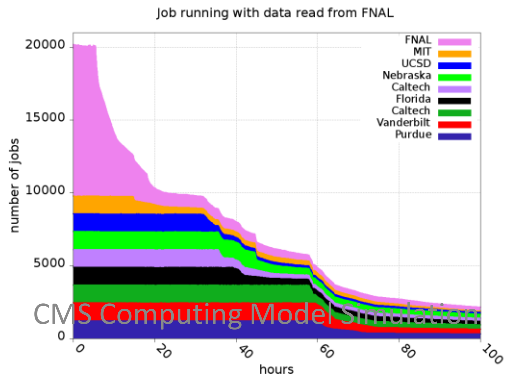


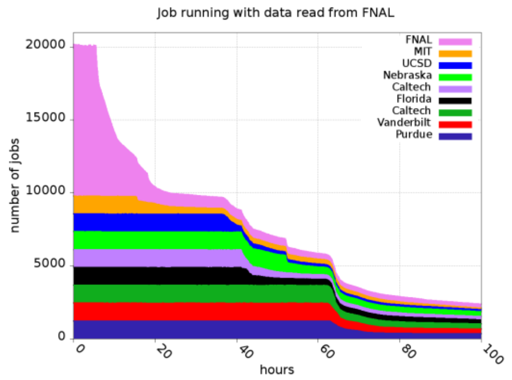




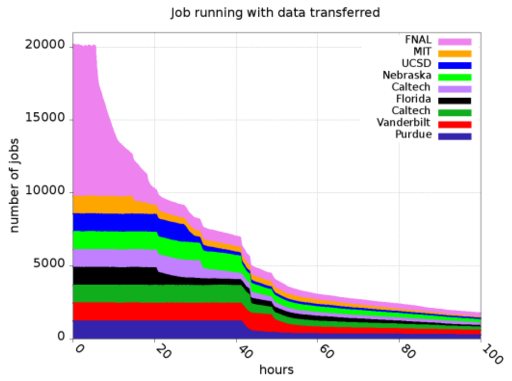


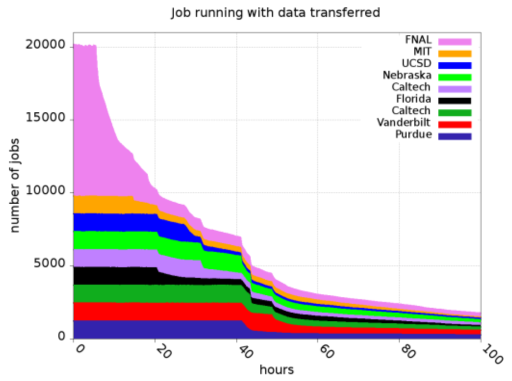


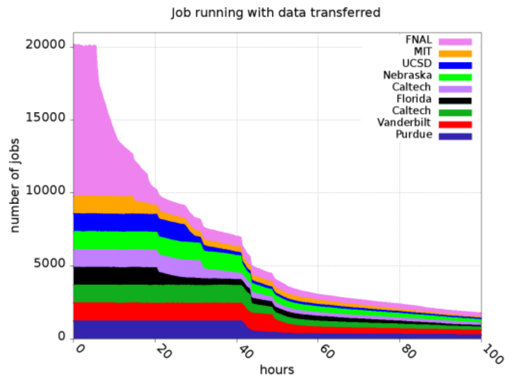


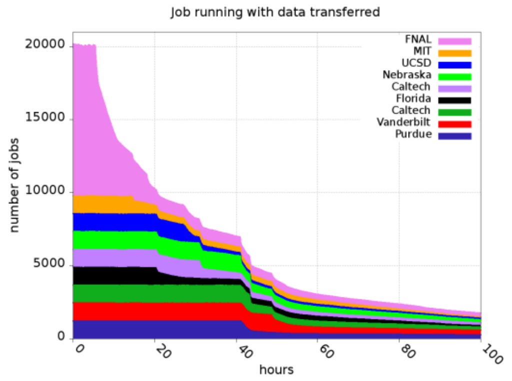


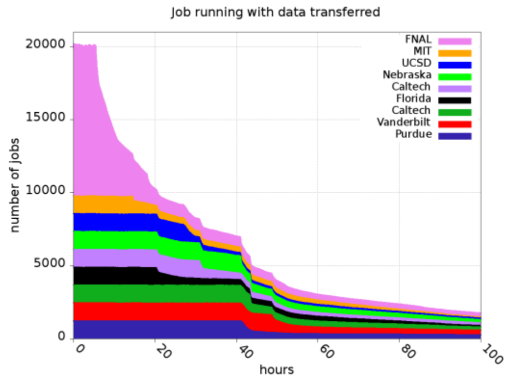
Jobs running when copied from FNAL

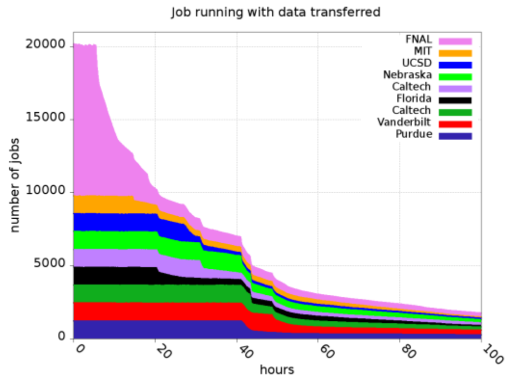


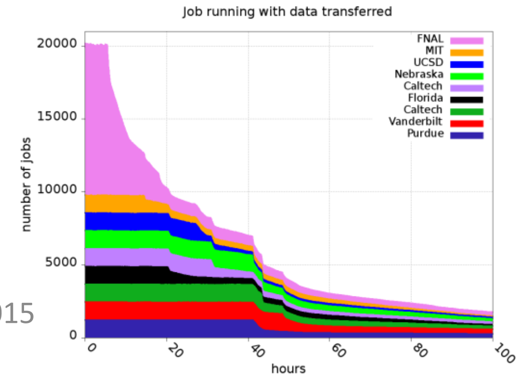


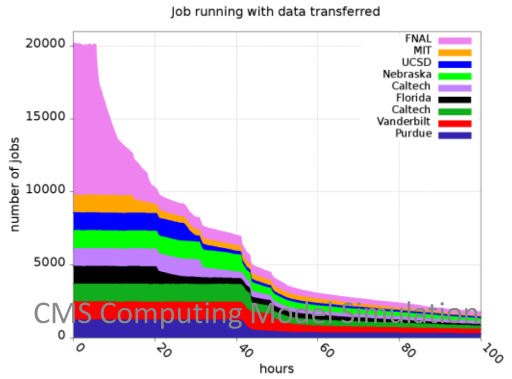


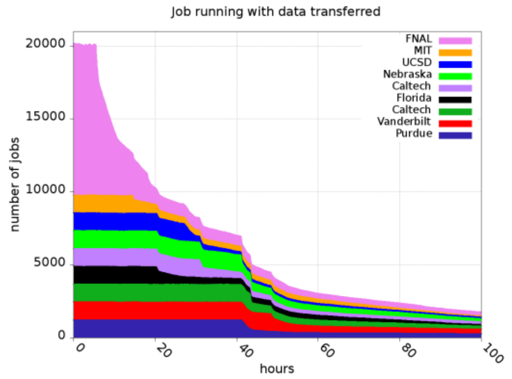




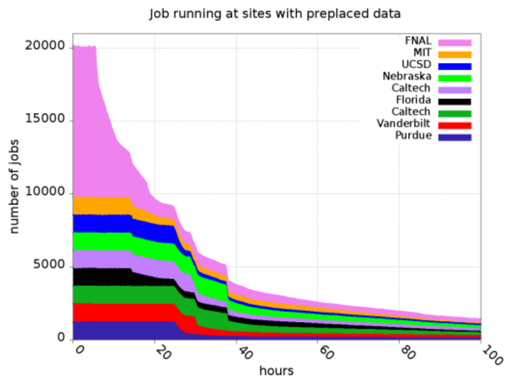


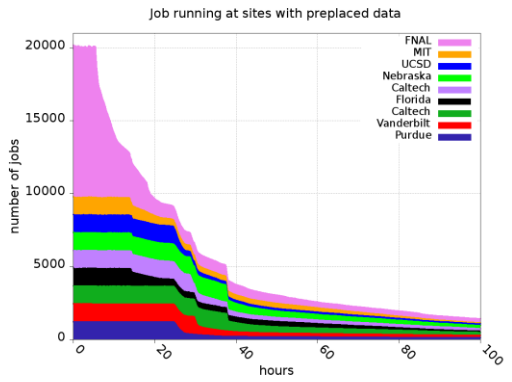


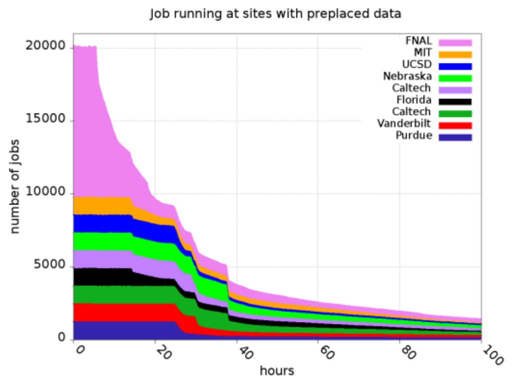


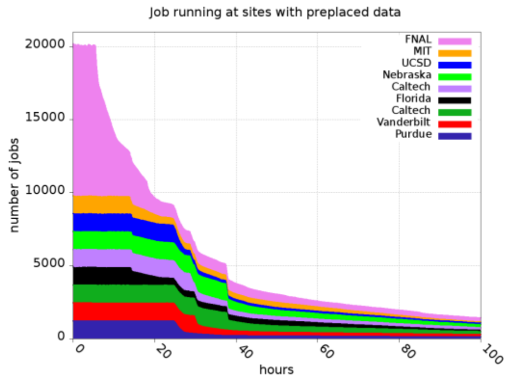


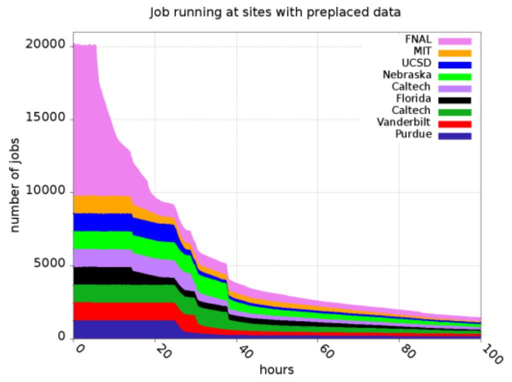
Jobs running when data mostly preplaced

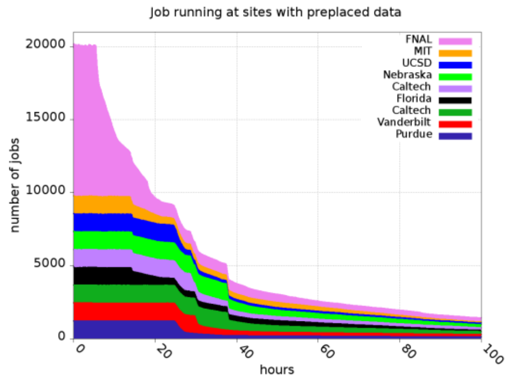


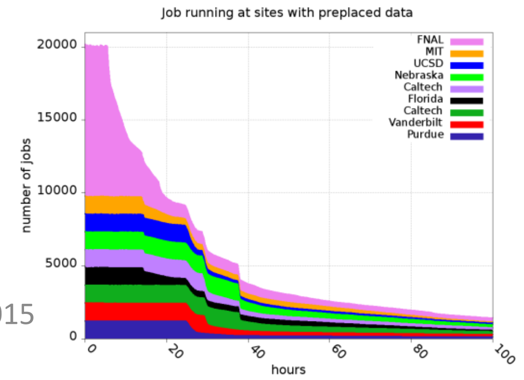


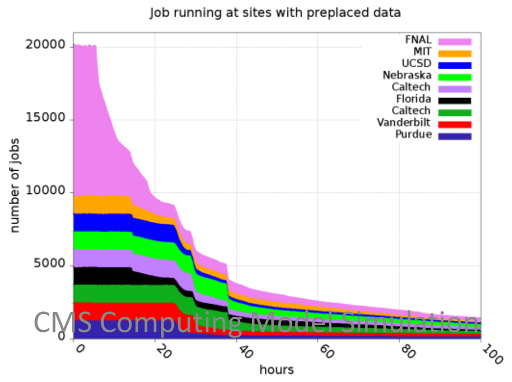


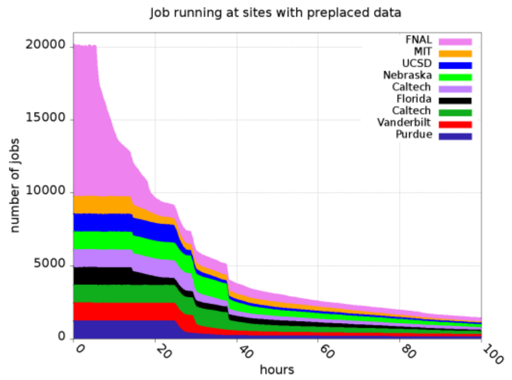




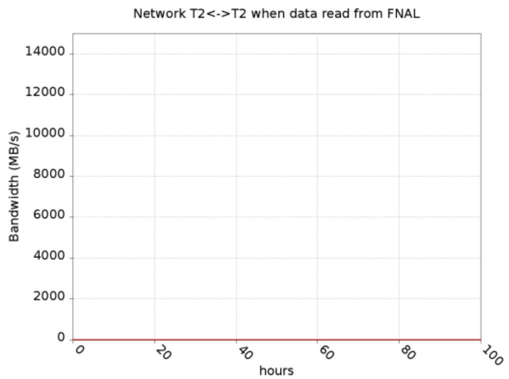


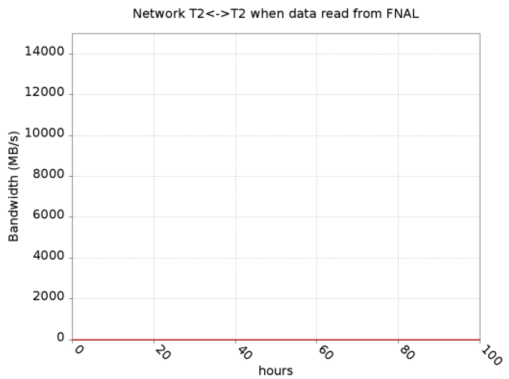


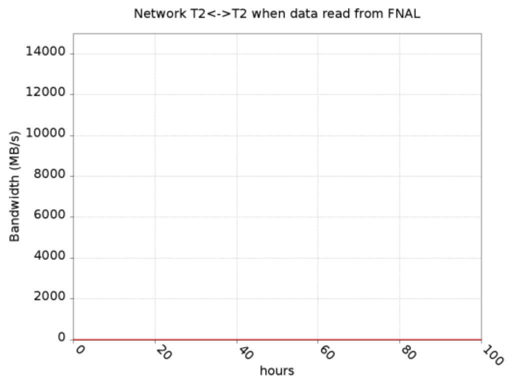


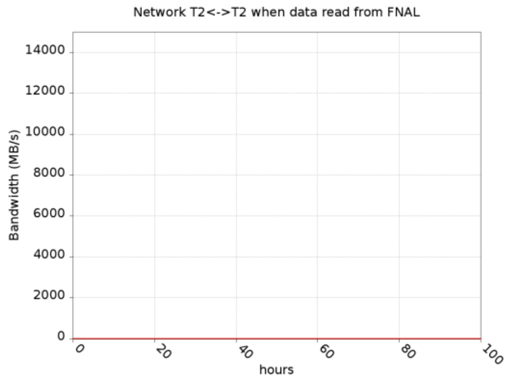


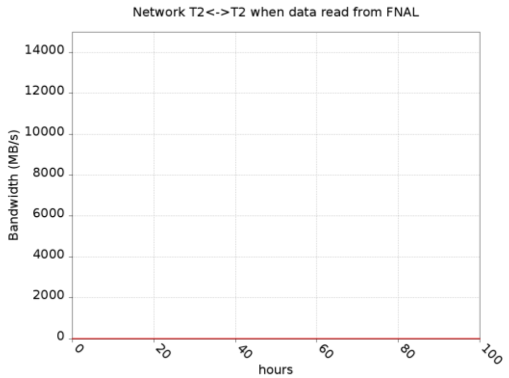
Inter-Tier2 rate when data read from FNAL

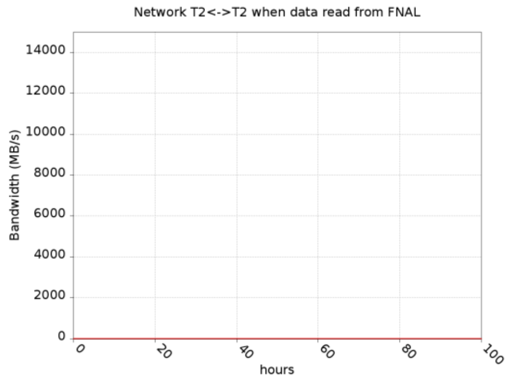


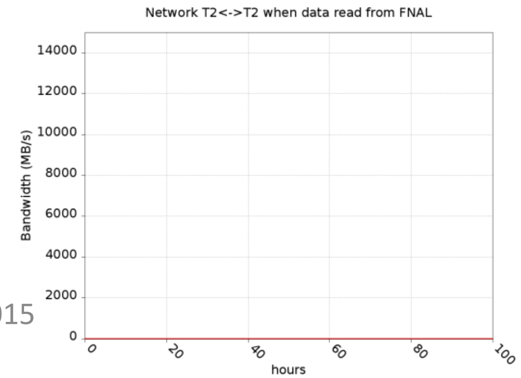


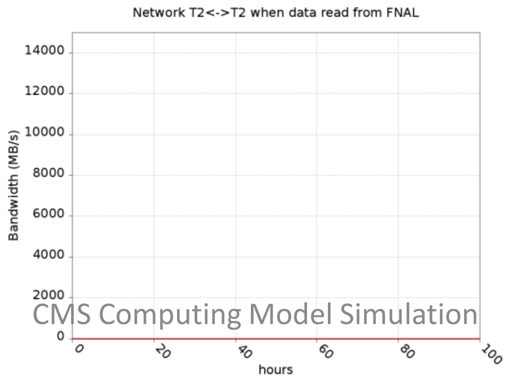


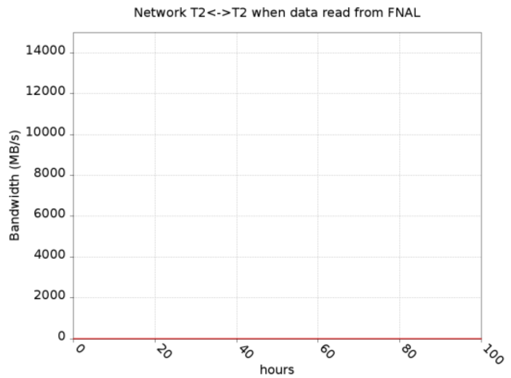




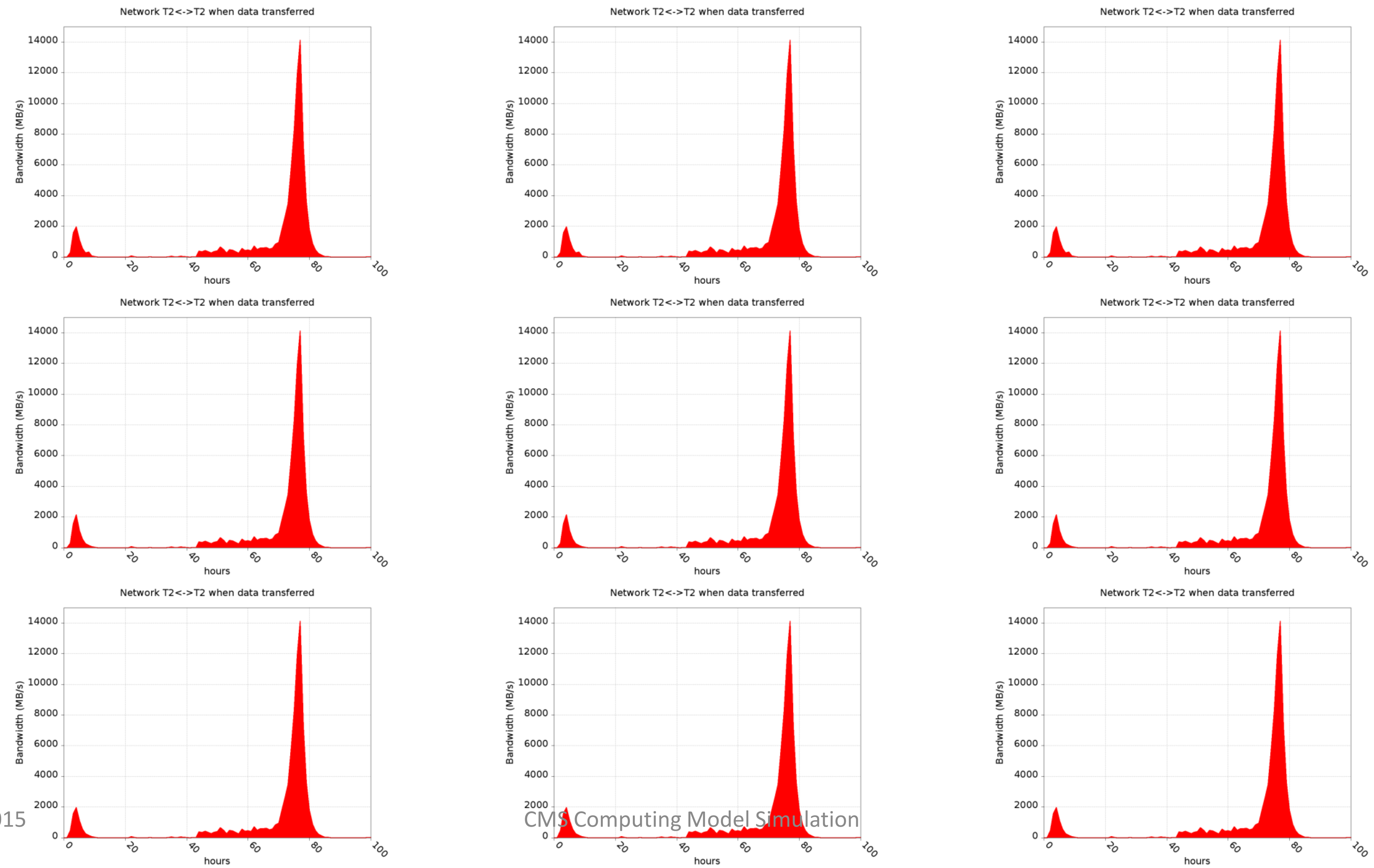




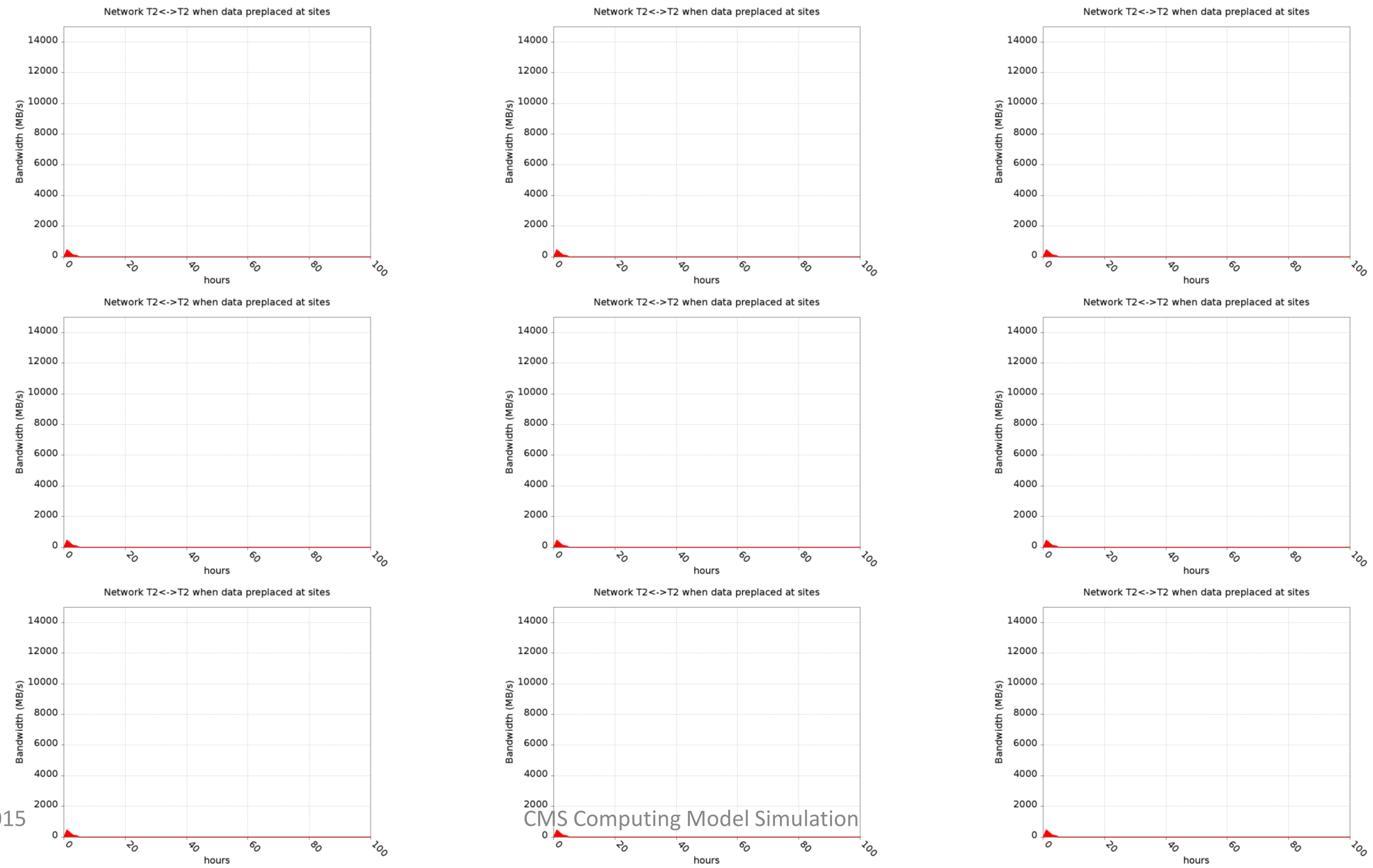




Inter-Tier2 rate when data copied from FNAL



Inter-Tier2 rate when mostly preplaced


Network T2<->T2 when data preplaced at sites
Bandwidth (MB/s)
hours