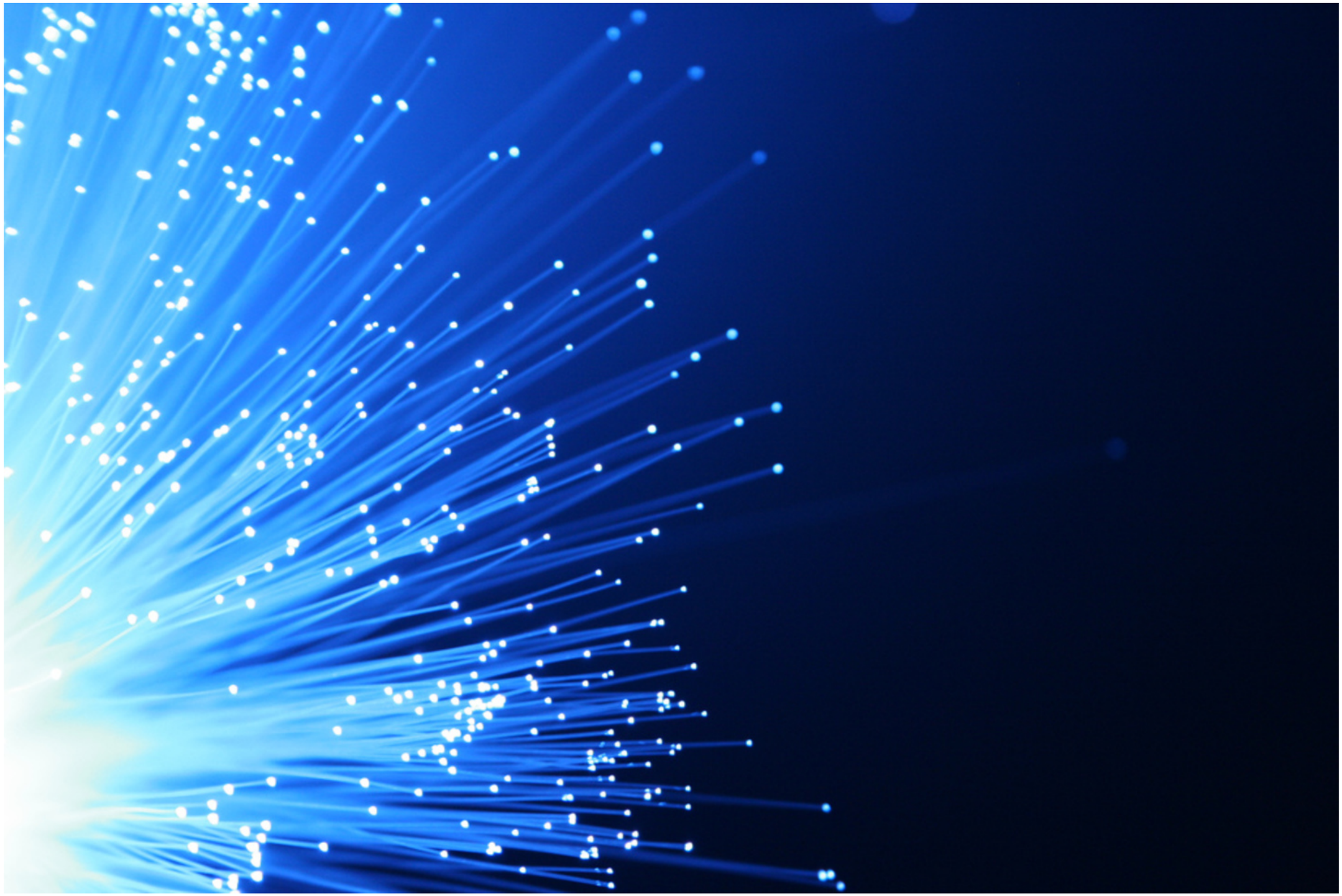


Designing Computing System Architecture and Models for the HL-LHC era

Stephen Gowdy - FNAL

Motivation

- The original computing LHC computing models (from 2005!) were designed with certain assumptions and realities in mind. For example in CMS the design focused on scalability and minimal dependencies between sites.

- The limitations were in particular appropriate for the period in which the WLCG was brought into production (say 2005-2010), along with the analysis models of the experiments.

Motivation

- We would like to explore other options for the geographical distribution of storage and processor capacity, including increased use of the network after removing or reducing restrictions on data locality.
- To get a better sense of the phase space of choices, and their impact, create a simple simulation of the computing system.

  See also next talk "Improvements in the CMS Computing System" by Maria Girone for information on evolution already being implemented for Run2.

Computing system simulation

- Event driven discrete simulation (time based events)

- Take into account job slots at sites, storage, network limitations and initial data placement

- Allows for transfers between sites, with checks on bandwidth

- Code is at https://github.com/gowdy/sitesim

Simulation technical specifics

- Flat files read to load in site, network, job and file/data information

- Plus key simulation parameters: CPU efficiency, remote read penalties and file transfer rates

- First it sets up sites and network links, then a catalog of the data.

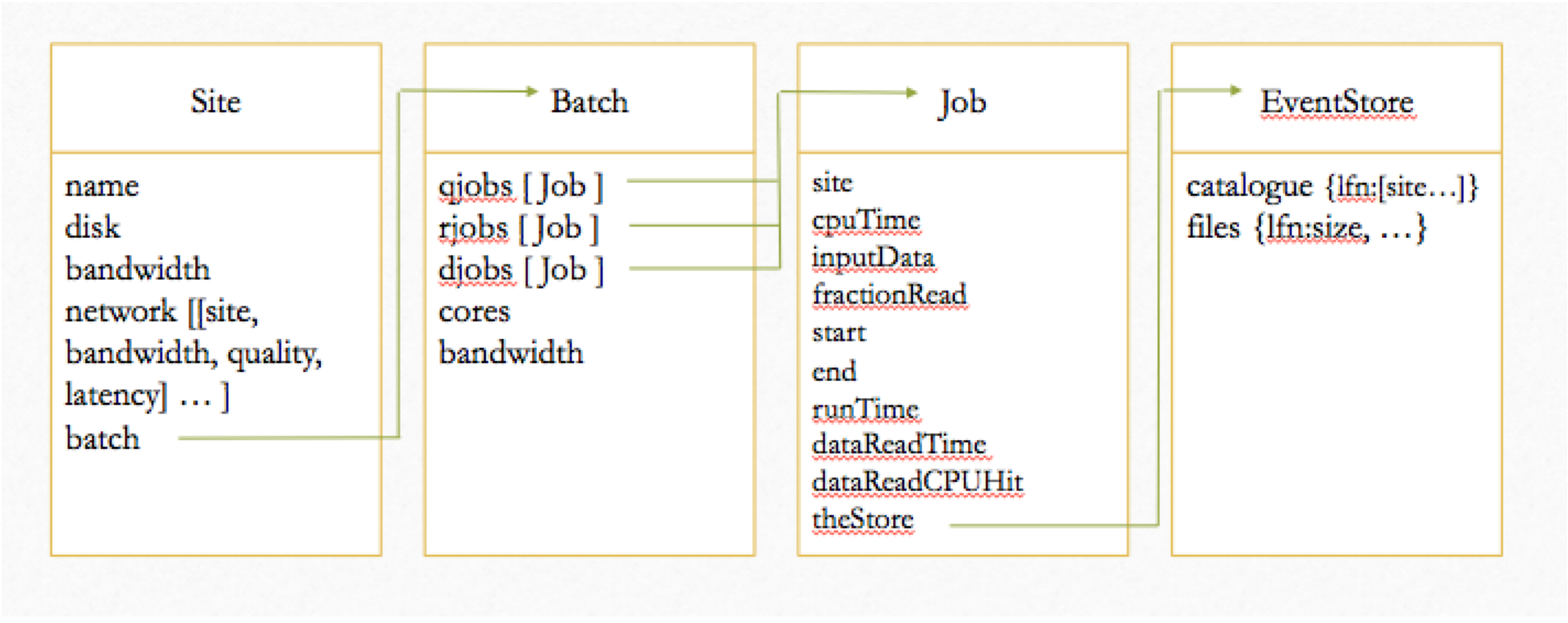
- A queue of jobs is processed in sequence.

- Initial studies use a list of analysis jobs whose runtime, input files, etc. is taken from the CMS Dashboard.

Simulation Parameters

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

Simulation Data Diagram



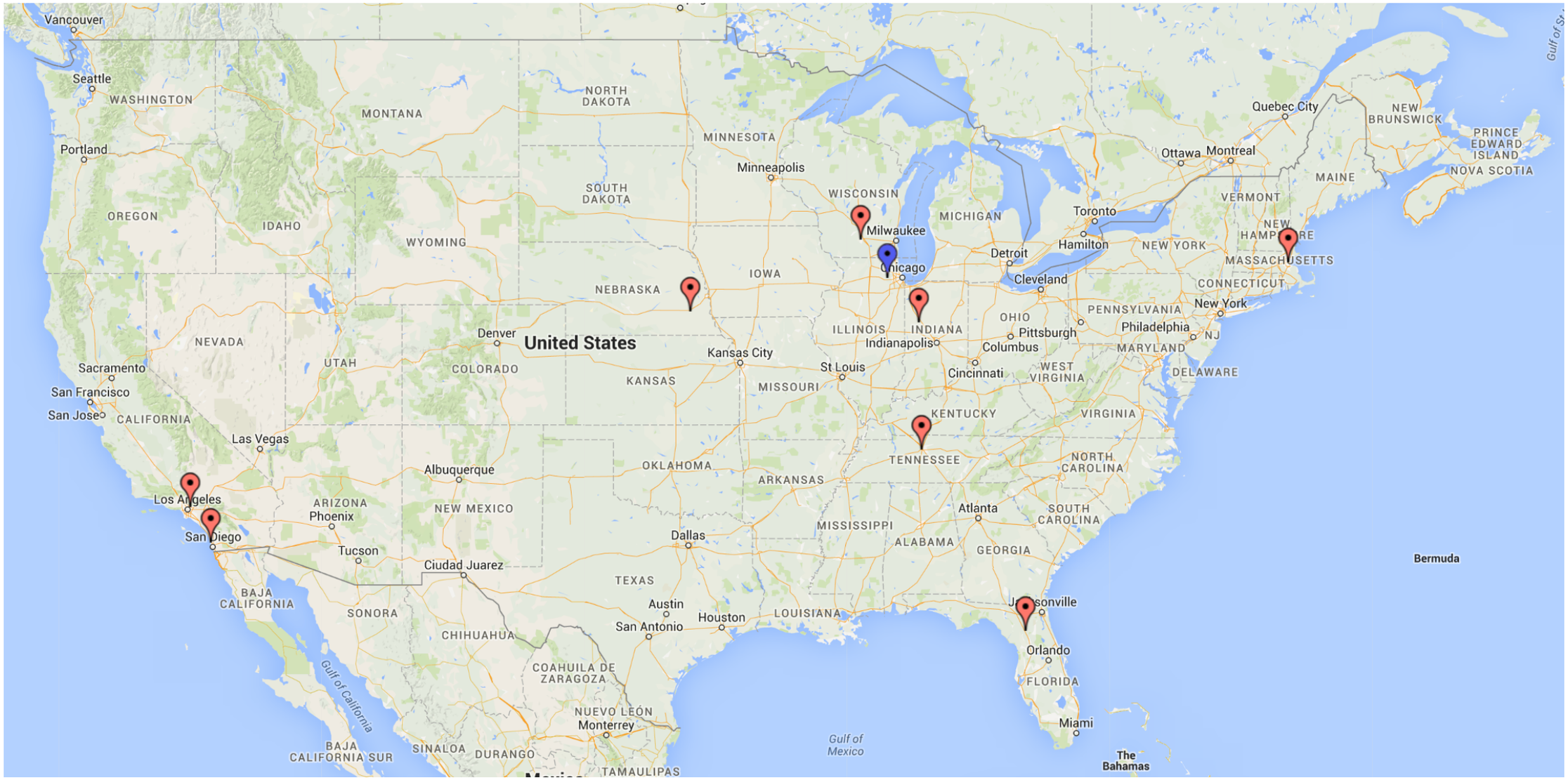
Site Information

- Originally we tried to simulate the entire set of CMS sites with information extracted from the CMS SiteDB database, which contains info about site pledges (2014 information, most recent). This is a bit too complex.

- For simplicity we however dropped back to a simpler system: focus on the set of US Tier1/Tier2 sites (9 total), sizes taken from REBUS

- FNAL, Caltech, Florida, MIT, Nebraska, Purdue, UC-San Diego, Wisconsin, Vanderbilt

Sites used in simulation



FNAL: 10400 cores and 10.4PB disk

All others: 1224 cores and 1.2PB disk each

Job Information

- Site, Start Time, Wall Clock, CPU time, files read
- Extracted job information from dashboard
  - A week from 15th to 22nd February, 2015
  - 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

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 examined

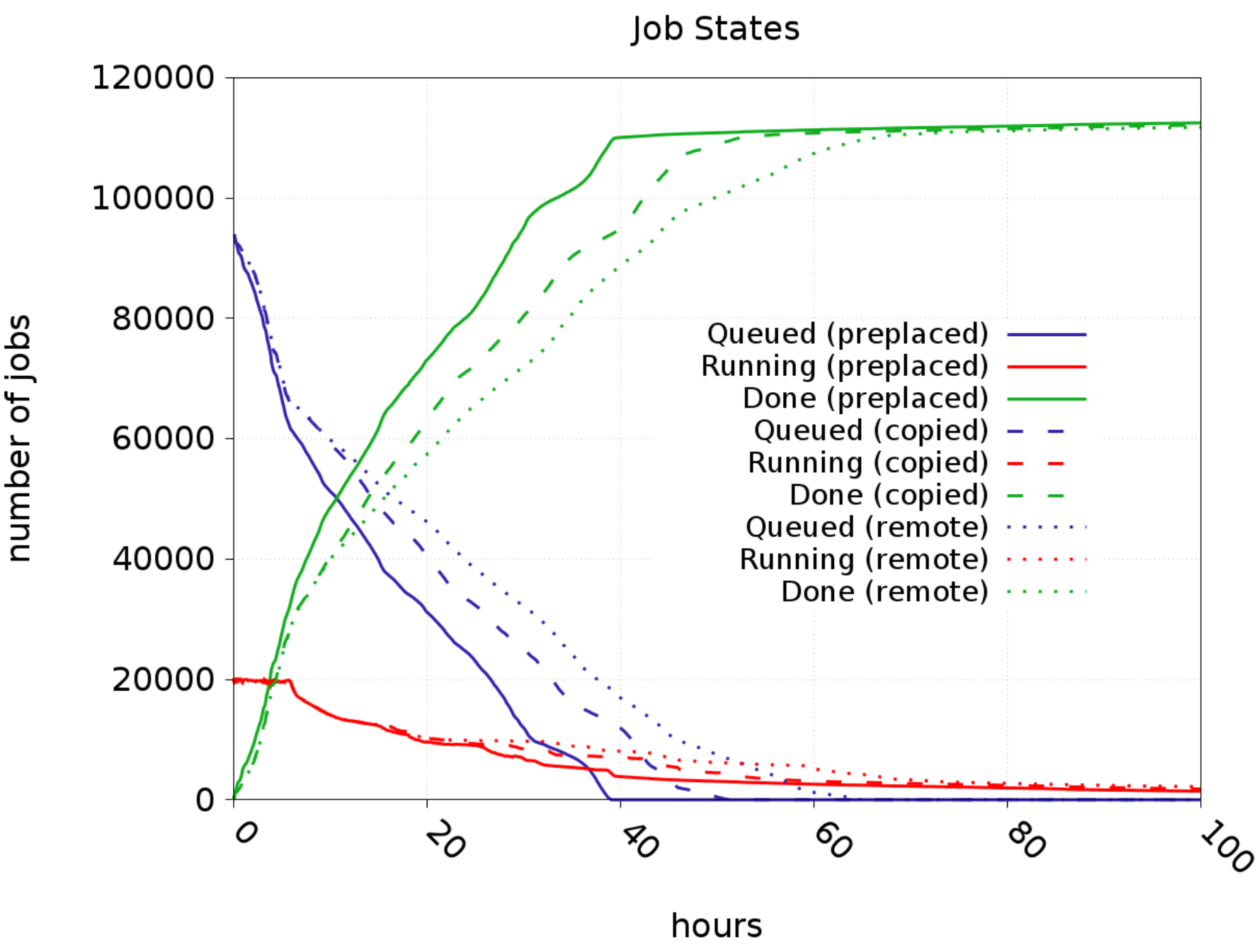
1. All data at FNAL and read via xrootd from FNAL

2. All data at FNAL and file transferred from FNAL at beginning of job

3. Data preplaced in advance at sites as during test period from Dashboard (altough there was some use of the data federation already here)

- First tests are all replay of jobs from Dashboard: jobs run again in the same sites, but data location has been changed. Worst case scenarios with all analysis jobs, no production jobs, were simulated.

Results - Job states



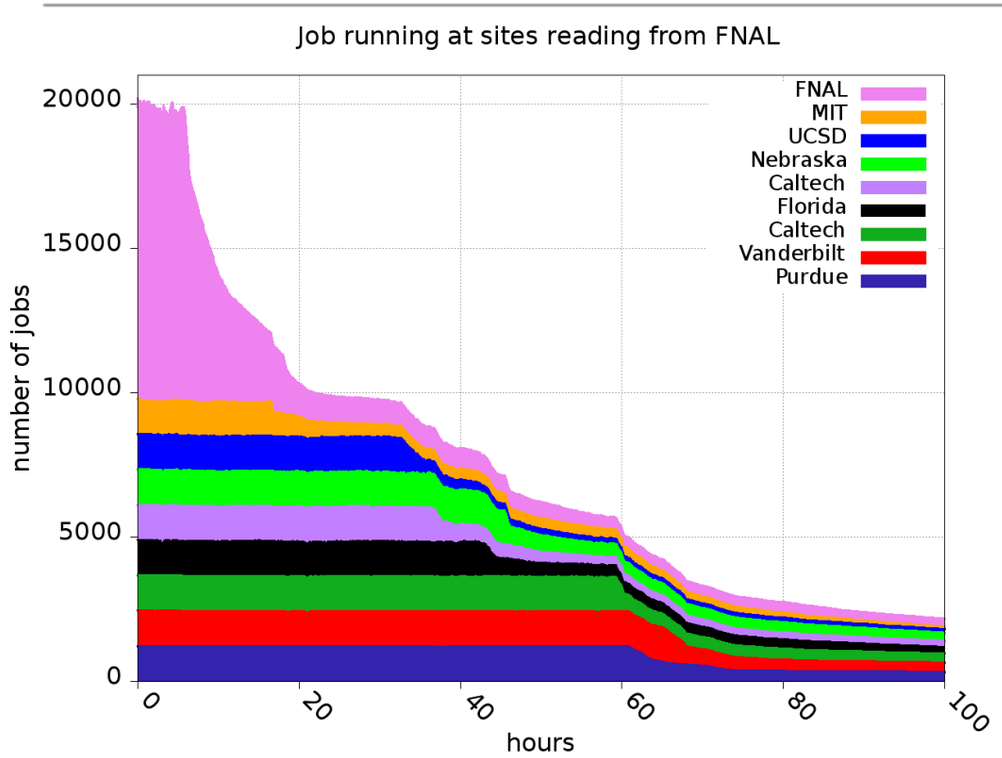
Total Wall Clock time for all jobs

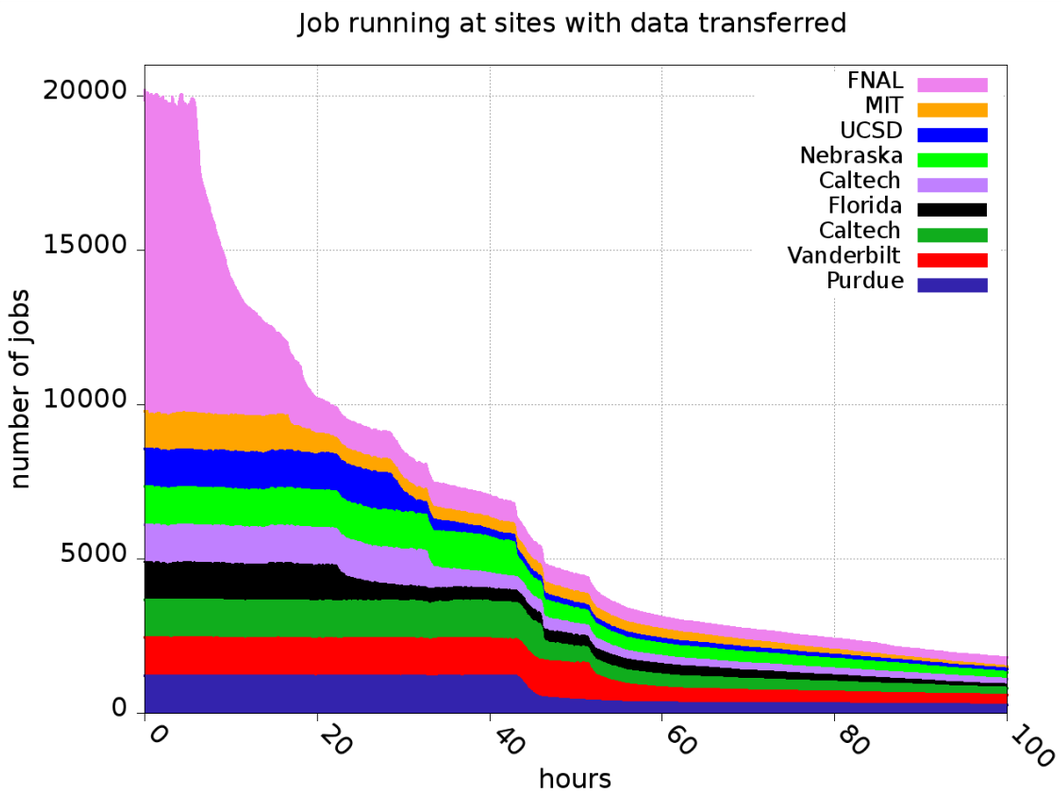
Remote:
1.1 Mhour

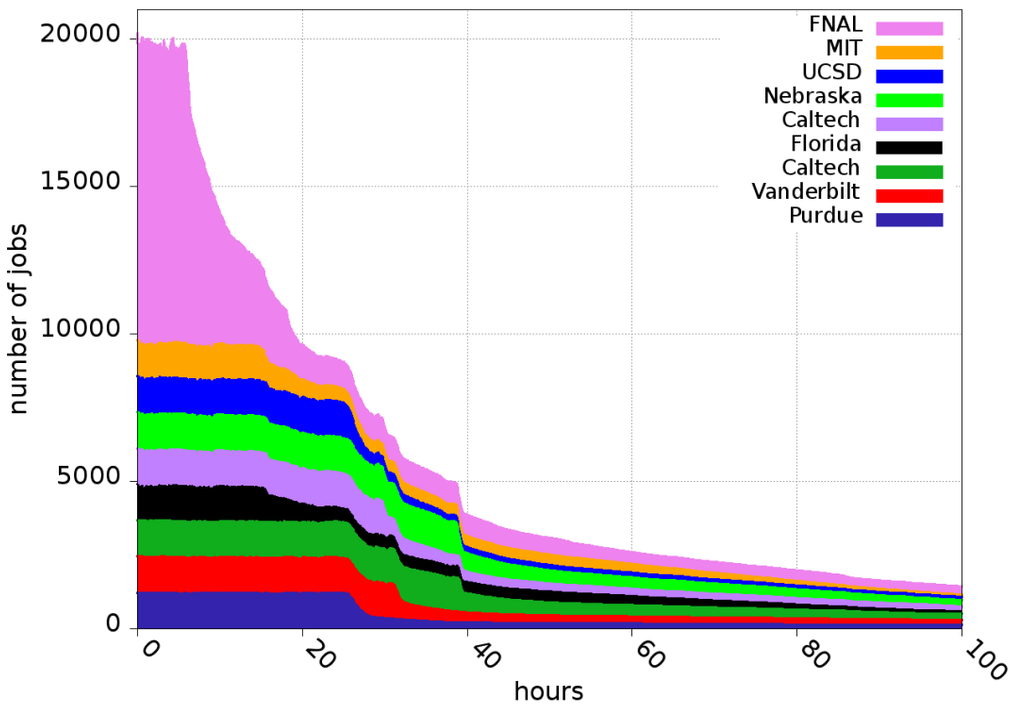
Copied:
944 khour

Preplaced:
777 khour

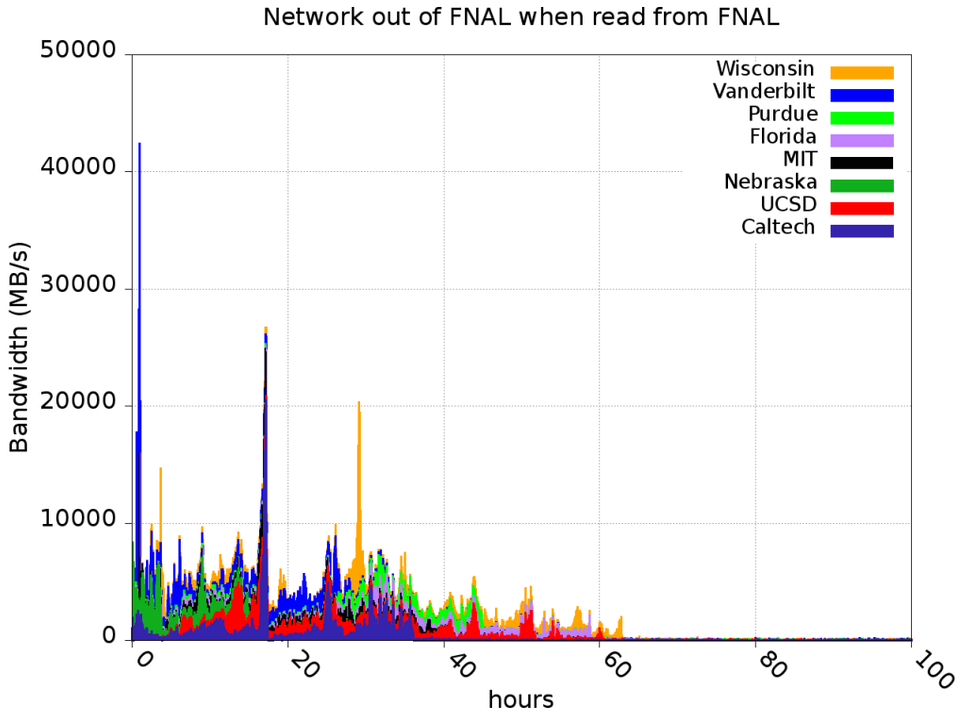
Results - Jobs running in each site

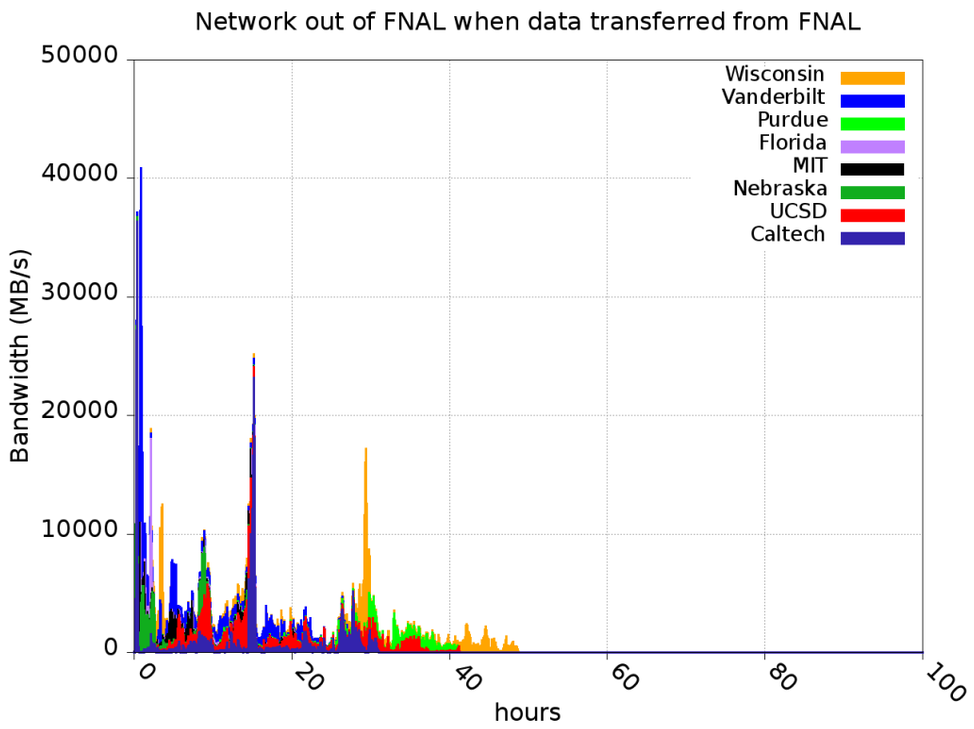


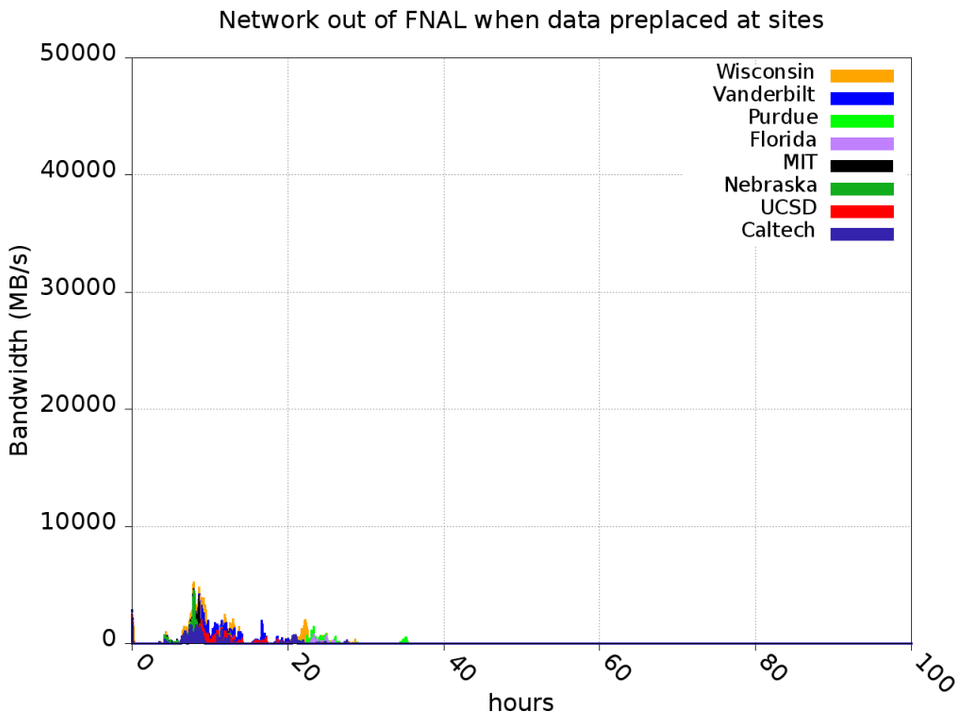




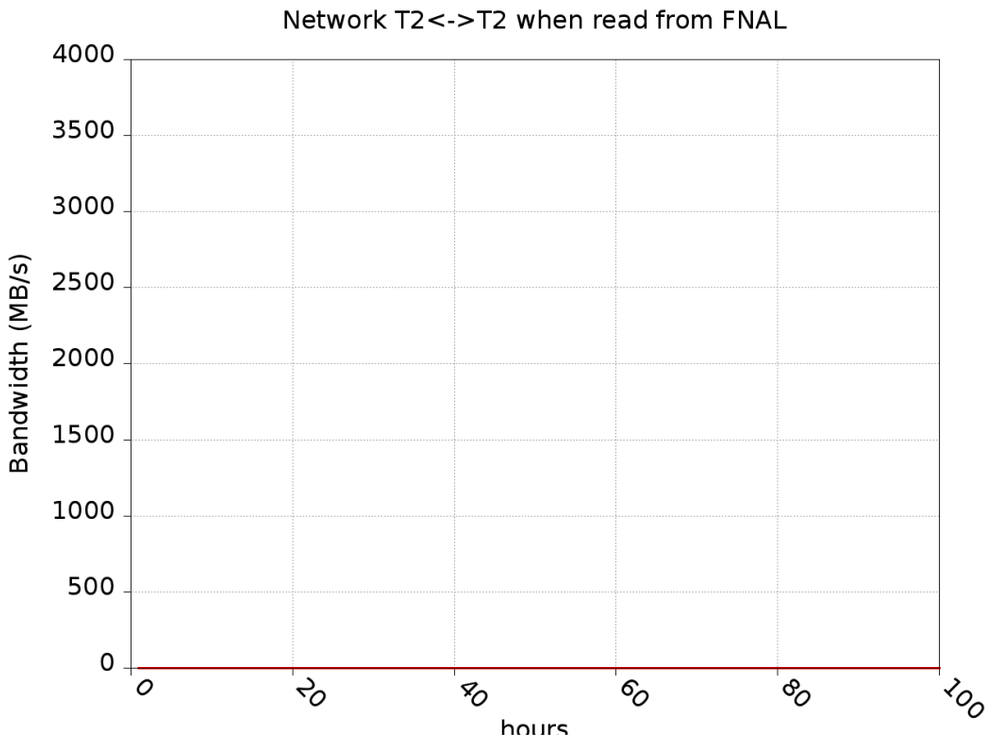
Results - Network traffic out from FNAL (aggregate)

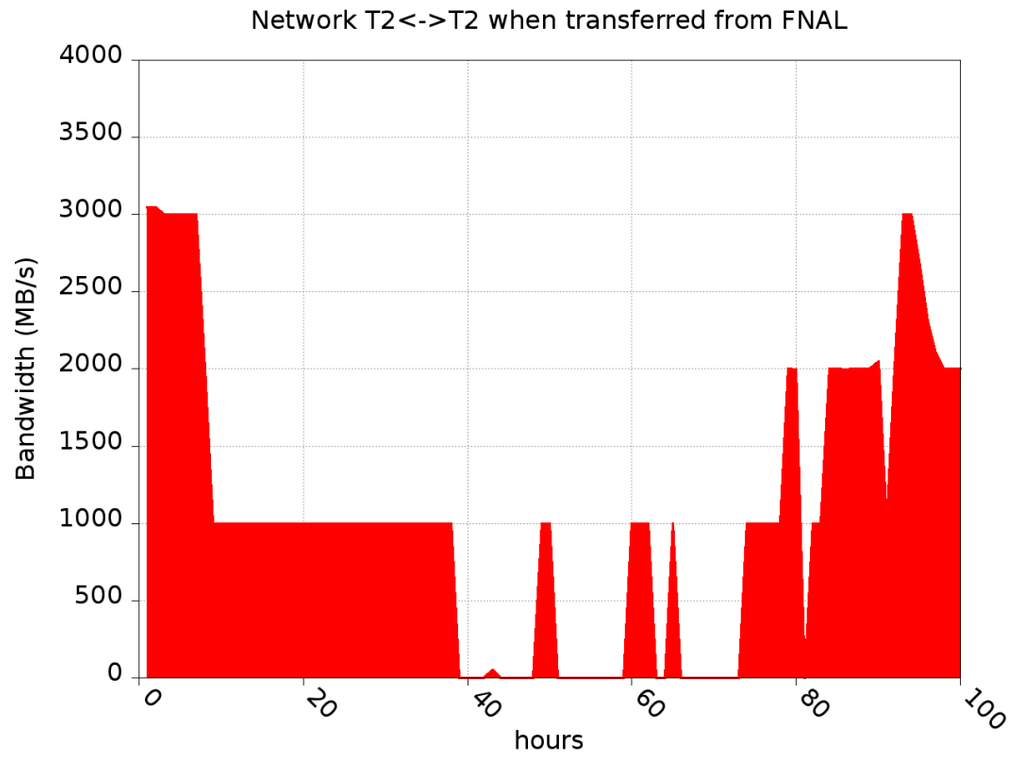


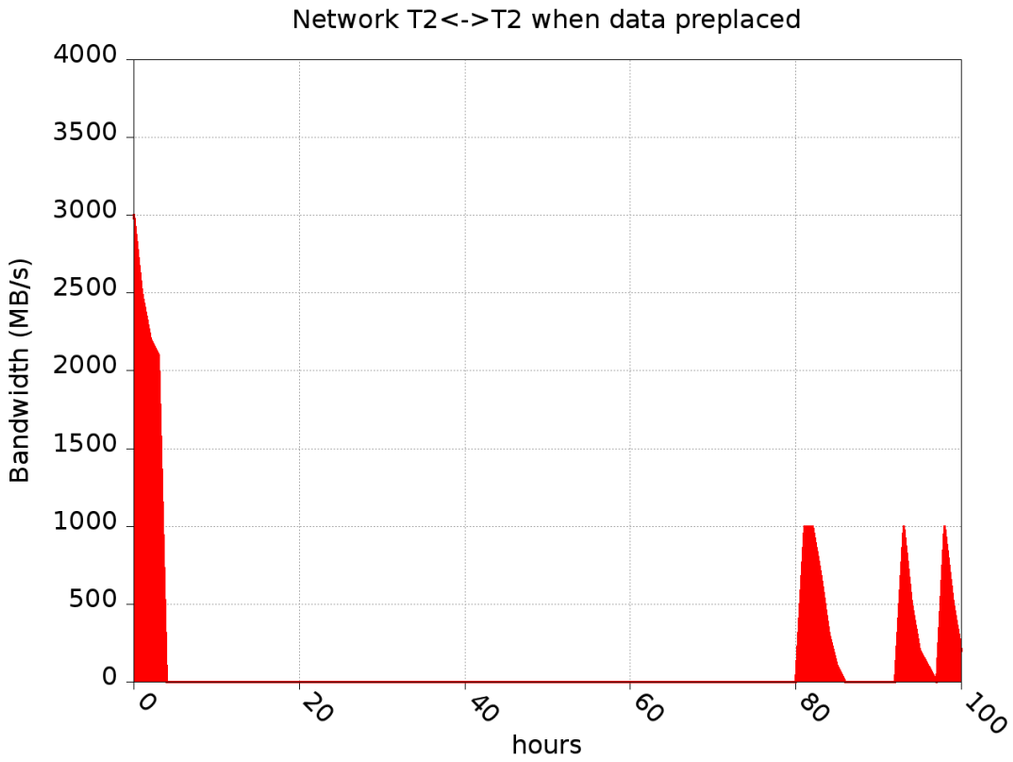




Results - network traffic between Tier2 sites

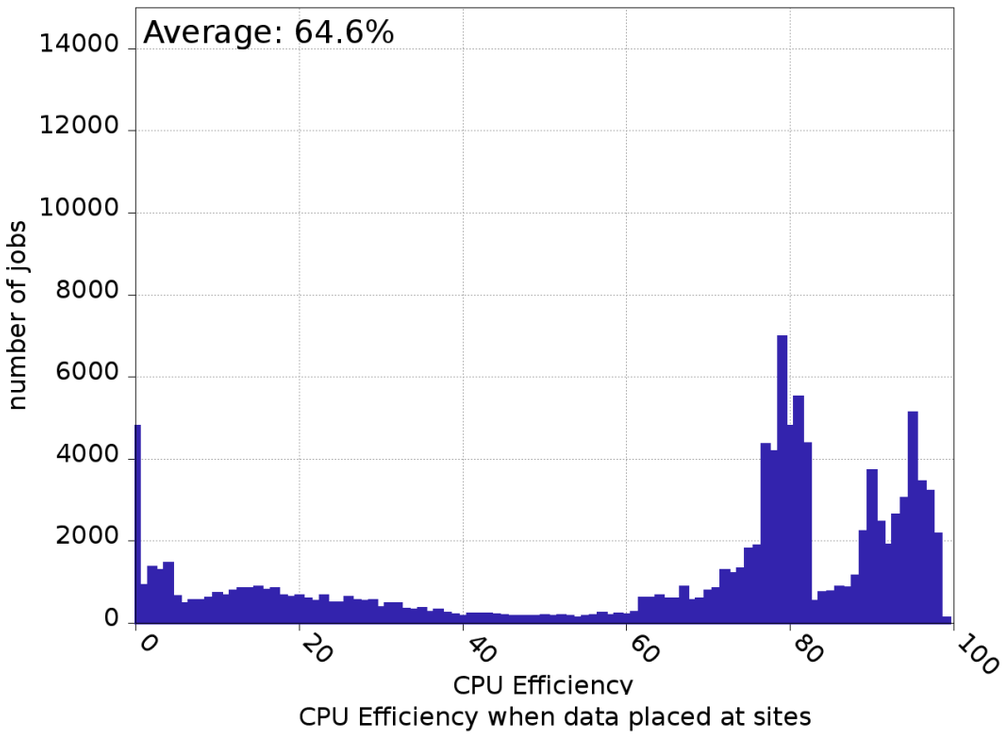




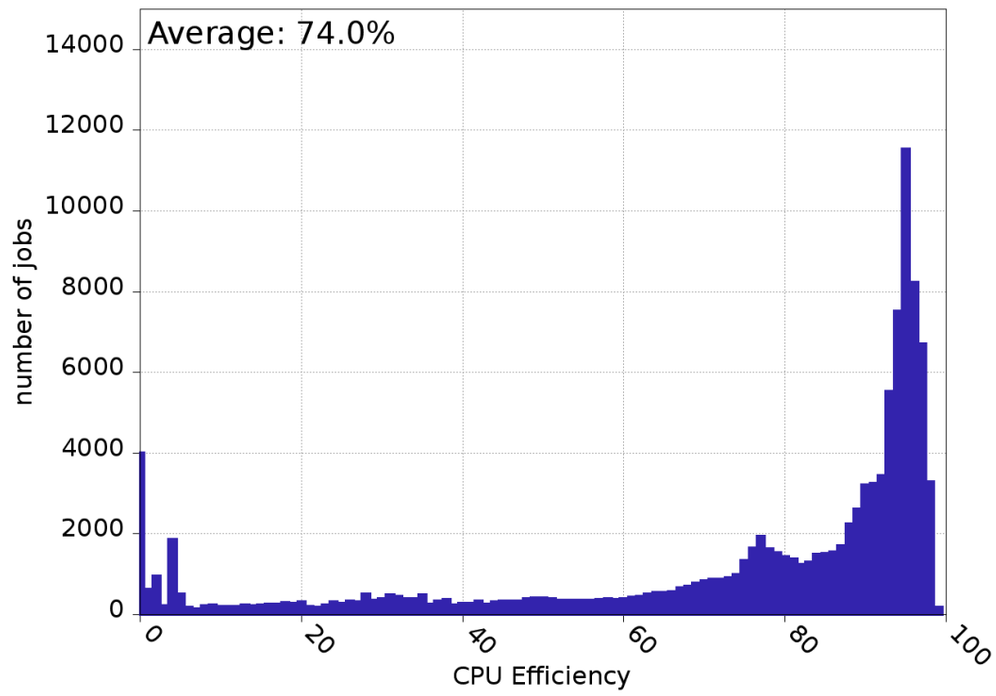


Results - CPU efficiency

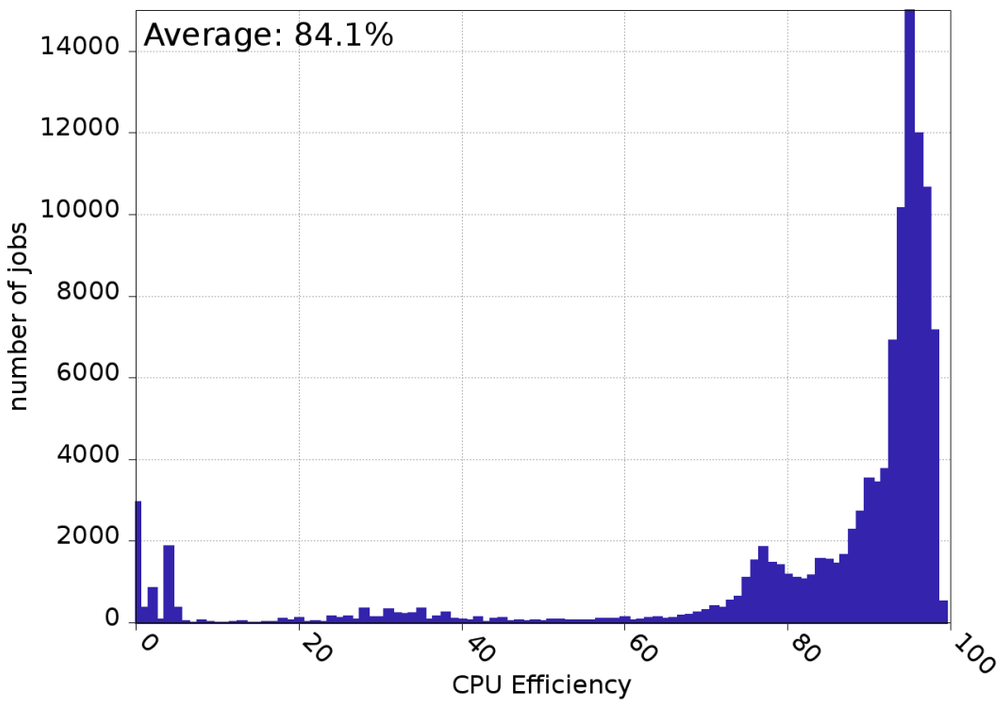
CPU Efficiency when data read from FNAL



CPU Efficiency when data transferred from FNAL



CPU Efficiency when data placed at sites



Summary

- We are interested in modeling alternate scenarios for the computing system architecture for the HL-LHC era, in particular strategies for data storage, placement and access.

- A simple simulation tool has been created to allow such modeling, taking into account job slot, storage and bandwidth limitations as well as other parameterized features of the system.