UW Madison CMS T2 site report

D. Bradley, T. Sarangi, S. Dasu, A. Mohapatra
HEP Computing Group

Outline

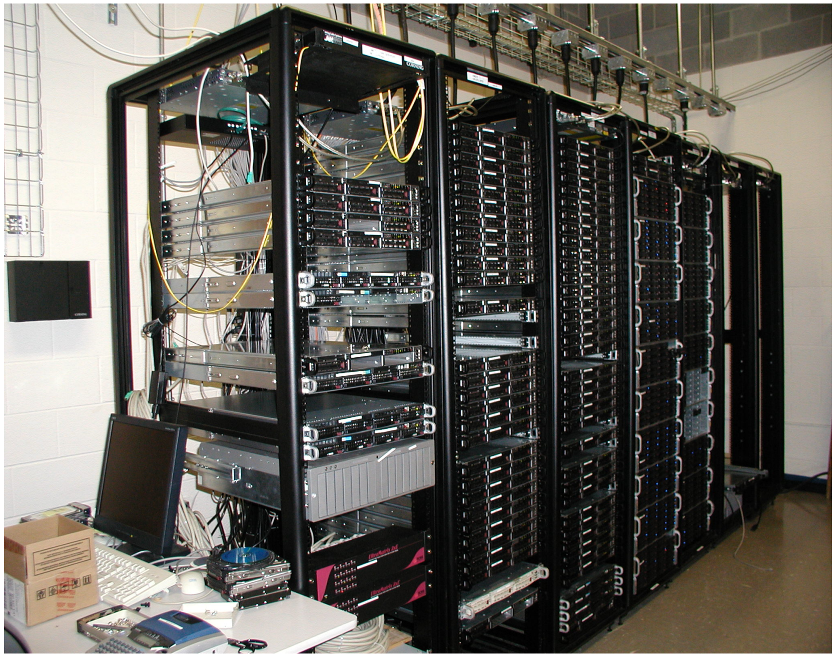
- Evolution
- Infrastructure
- Resources
- Management & Operation
- Contributions to CMS
- Summary

Evolution

- ✓ Started out as a grid3 site
- ✓ Played a key role in the formation of the Grid laboratory of Wisconsin (GLOW)
- ✓ HEP/CS (Condor team) collaboration
  - Designed standalone MC production system
  - Adapted CMS software, and ran it robustly in non-dedicated environments (UW grid & beyond)
- ✓ Selected as one of the 7 CMS Tier2 site in the US.
- ✓ Became a member of WLCG and subsequently OSG.
- ✓ Serving all OSG supported VOs besides CMS

Infrastructure

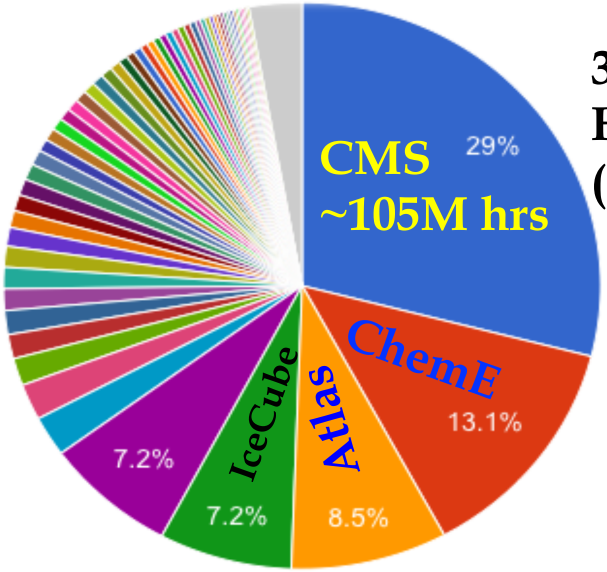
- ✓ 3 machine rooms, 16 racks
- ✓ Power supply – 650 KW
- ✓ Cooling
  - Chilled water based air coolers and POD based hot aisles



Compute / Storage Resources

✓ Compute (SL6)

- T2 HEP Pool – 4200 cores (38K HS06)
  - Adding 1000 cores soon
  - Dedicated to CMS
- GLOW Pool – 3200 cores
  - Opportunistic

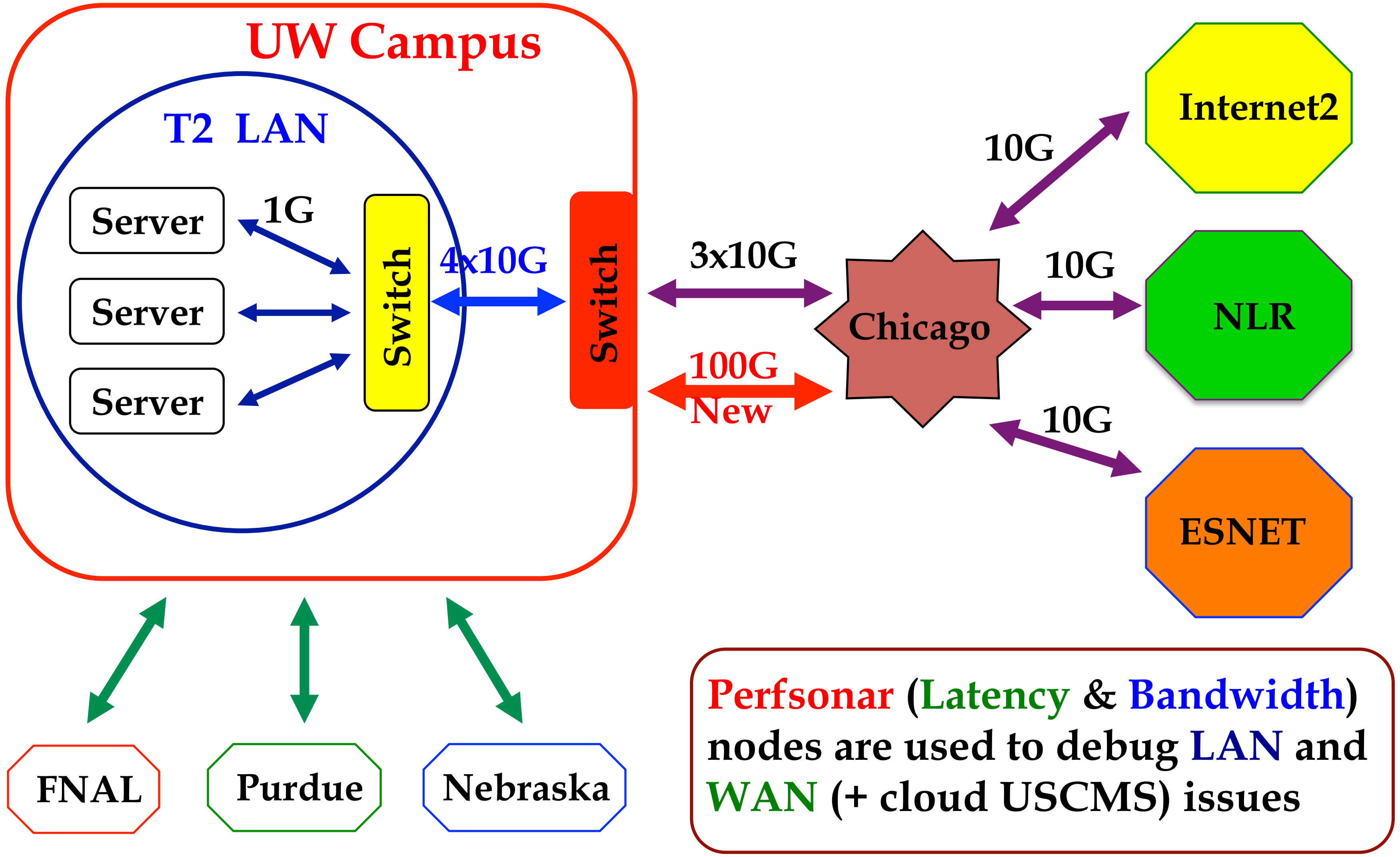


363M Hours (8 years)

✓ Storage (Hadoop)

- Migrated from dCache to hadoop 3 years ago.
- 3PBs distributed across 350 nodes
- Will add 1B soon.
- Being upgraded to hadoop-2.0

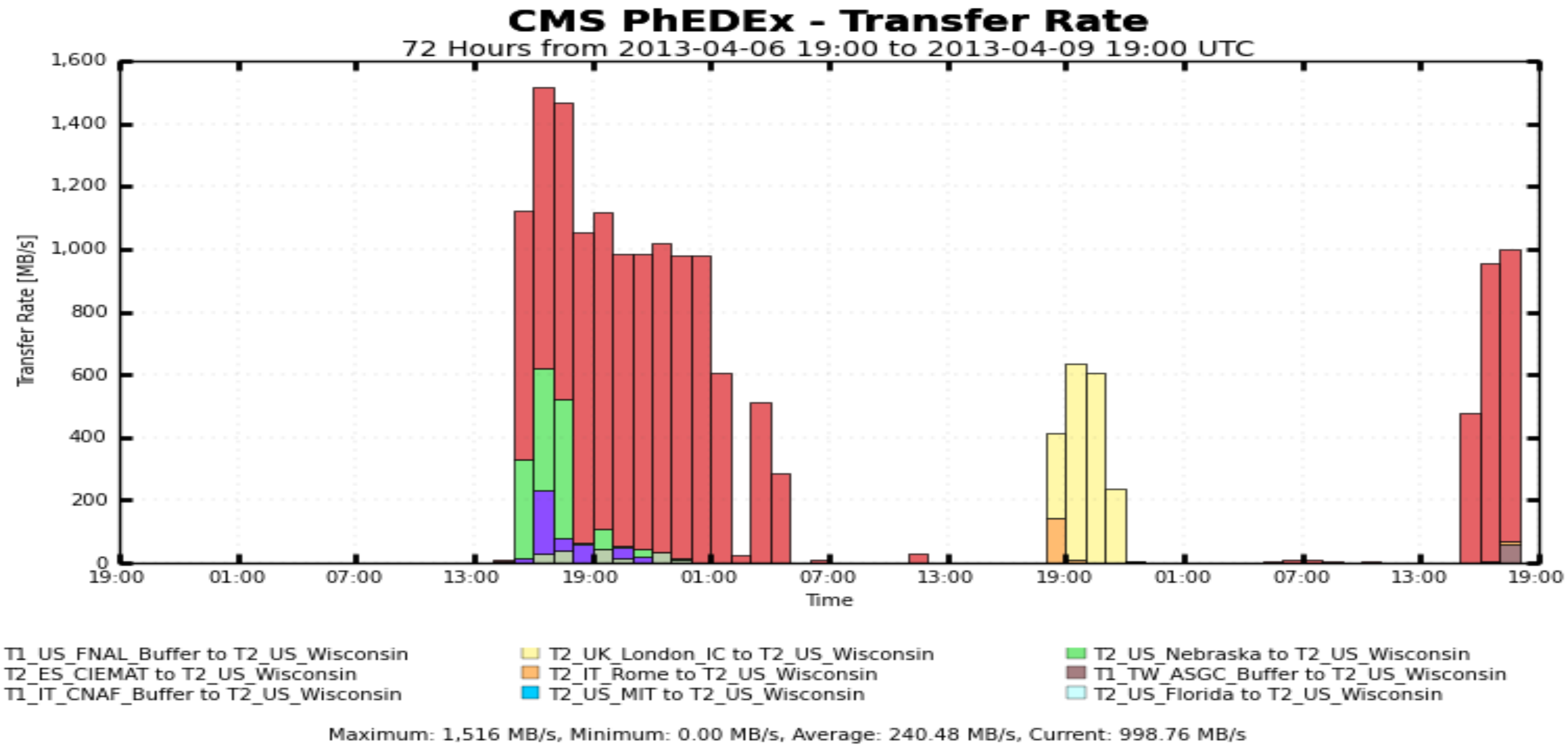
Network Configuration



Perfsonar (Latency & Bandwidth) nodes are used to debug LAN and WAN (+ cloud USCMS) issues

Network Configuration (2)

- ✓ Strong support from compus network team
- ✓ Good rate for CMS data transfer from T1/T2 sites enables data availability on demand and with low latency



Software and Services

- ✓ File systems & proxy service
  - AFS, NFS, CernVM-FS (cvmfs), Frontier/Squid
- ✓ Job batch system
  - HTCondor
- ✓ OSG software stack
  - Globus, GUMS, glexec, CEs, SEs, and a lot more
- ✓ Storage
  - Hadoop (hdfs), BestMan2(srm), gridFtp, Xrootd, etc.
- ✓ Cluster management & monitoring
  - Local yum repo, Puppet, Nagios, Ganglia, and a few dozen home grown scripts

Cluster Management & Monitoring

- ✓ Puppet
  - Migrated from Cfengine to Puppet this summer.
- ✓ Nagios
  - Hardware, disks etc.
- ✓ Ganglia
  - Services, memory, cpu/disk usage, I/O, network, storage
- ✓ OSG and CMS dedicated tools
  - RSV, SAM, Hammer Cloud, Dashboard
- ✓ Miscellaneous Scripts
  - Provide redundancy

Contributions to CMS

✓ HTCondor and Glidein technology

- condor_job_router
- condor_ssh_to_job
- condor_gangliad
- condor_defrag
- condor_shared_port & CCB
- file transfer scheduling
- scalability improvements

✓ MC production and analysis infrastructure

- ProdAgent, WMAgent

Any data, Anytime, Anywhere

✓ Goal : Make all CMS data transparently available to any CMS physicist, anywhere.

- Transparent and efficient local/remote data access : no need to know about data location
- Reliable access i.e. failures are hidden from the user's view
- Ability to run CMS software from non-CMS managed worker nodes
- Scheduling excess demand for CPUs to overflow to remote resources

- The technologies that make this possible:
  - xrootd (read data from anywhere)
  - cvmfs + parrot (read software from anywhere)
  - glideinWMS/HTCondor (send jobs anywhere)

Any data, Anytime, Anywhere

THE UNIVERSITY of WISCONSIN MADISON

✓ Underlying technology for data access : Xrootd

- Works with heterogenous storage systems
- Data access at registered sites in the data federation via local/global xrootd redirectors (fallback supported)
- Access to data via authentication

✓ Deploying the rest of the AAA technologies gave us added capabilities:

- Overflow CMS jobs to other HTCondor clusters on our campus
- Overflow CMS jobs to other sites in Open Science Grid
- Add Amazon EC2 nodes to our site to increase capacity at the drop of a credit card

Experience with Amazon EC2

- EC2 VMs were configured as WNs in our site
  - Authorized to join HTCondor pool via x509 credential
  - Did not make use of cloud storage
  - Read data via xrootd ← Wisconsin HDFS
  - Write data via SRM → gridftp → Wisconsin HDFS
  - Read software via CVMFS
- Used EC2 Spot market to reduce cost
  - Tradeoff : risk of termination of VM at unpredictable time, causing jobs to die and restart
- Only small-scale trials so far
  - 3 cores for a month
  - 100 cores for a week

Experience with Amazon EC2

- Results
  - Total cost: $0.035/equivalent-Wisconsin-core-hour
  - This depends a lot on scale and type of workflow:
    - 55% of cost was for CPU
    - 45% of cost was for transferring output (at $0.12/GB)
      - At larger volumes, price/GB decreases
  - Depending on workflow, inefficiency due to spot instance termination:
    - 5-15% loss in efficiency in our trials
    - spot bid was 2x the base spot price

Summary

- ✓ The site is in good health and performing well
- ✓ Making our best effort to maintain the high availability/reliability while productively serving CMS and the grid community.
- ✓ Looking forward to make use of the new 100G network to Chicago as soon as it's available.

Thank You !

Questions / Comments ?

Backup Slides

Cluster Management - Puppet

✓ Entire Tier-2 Cluster is managed by Puppet (open source project)

- Designed as a Master-Client framework
- Apache and rails based passenger-fusion supports the http backend for the puppet-master
- Configuration for each service such as AFS, HDFS, SRM, GUMS are designed as individual modules
- Configuration catalogs are propagated to each node via inbuilt SSL authentication
- New implementation and monitoring are done through regular cron jobs

Anydata, Any time, Any where (AAA)

✓ **Entire Tier-2 Cluster is managed by Puppet (open source project)**

- Designed as a Master-Client framework
- Apache and rails based passenger-fusion supports the http backend for the puppet-master
- Configuration for each service such as AFS, HDFS, SRM, GUMS are designed as individual modules
- Configuration catalogs are propagated to each node via inbuilt SSL authentication
- New implementation and monitoring are done through regular cron jobs