Site Report
OSG All Hands Meeting

03/23/2015

Azher Mughal
Dorian Kcira
Samir Cury

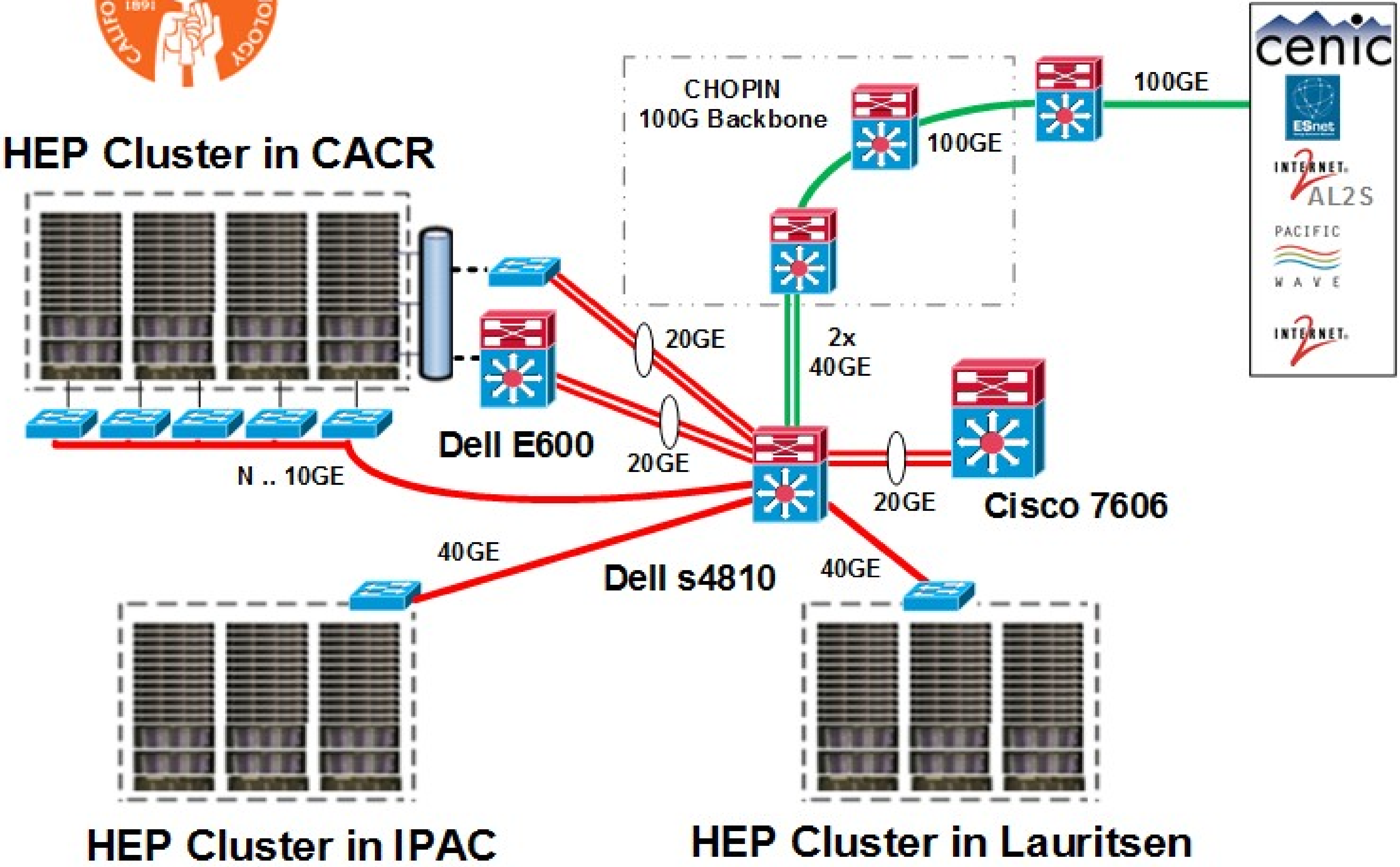
Caltech today - Resources

- 5824 Cores (98.2% online)
  - 363 servers / 16 Racks
- 2.057 PB of Usable storage
- 200 Cores of opportunistic access
  - +512 until the end of the year - 4 GB / core queue

Caltech today - Resources

- 5824 Cores (98.2% online)
  - 363 servers / 16 Racks
- 2.057 PB of Usable storage
- 200 Cores of opportunistic access
  - +512 until the end of the year - 4 GB / core queue

Caltech HEP Tier2



WAN = 100GE Link from CHOPIN Project (CC-NIE)

Software

- HDFS 2.0
- HTCondor 8.2
- All Grid Middleware on OSG 3.2
  - Xrootd : 4.0
  - CE1 : GRAM (Active) + HTCondor CE
  - CE2 : GRAM
  - CE Opportunistic : HTCondor CE

Challenges

- Main item : physical space
  - All the space provided by campus was used by our Tier-2 and associated projects.
  - All upgrades starting from 2015 will have to imply deprecation of the oldest generation of hardware.
    - Not necessarily bad.
  - Server recycling options are available. Unclear if policies will allow it.

Preparations for Run 2

- CMS will get slots when it asks for
  - OSG/Opportunistic job preemption. 48h Pilots.
- AAA will work when configured. In all resources (T3 included).
  - More flexible workflows are a fact. Networking activity needs more attention to prevent bottlenecks or failures.
- LAN is well-designed to support high throughput
  - T3 got uplink upgraded and started benefiting from T2 faster caches.
  - Some internal links were upgraded.
- Ensuring node-uniformity through Configuration Management and high level service monitoring
  - Special attention to potential black-hole nodes.
- CPU-only resources

Future goals

- Have optimal WAN usage through GridFTPs
  - Not spend too much resources to fully utilize WAN capacity.
  - Hope to have central middleware (PhEDEx, FTS) helping sites to achieve that.
- Continuous Integration for Configuration Management code

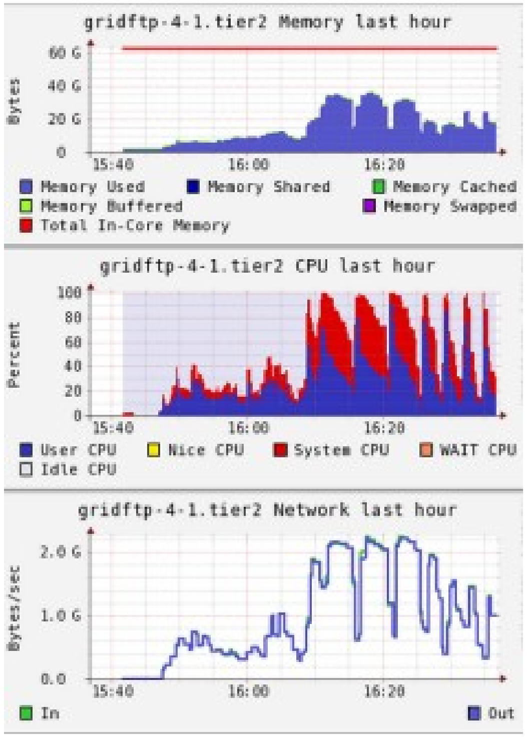
We're Hiring!

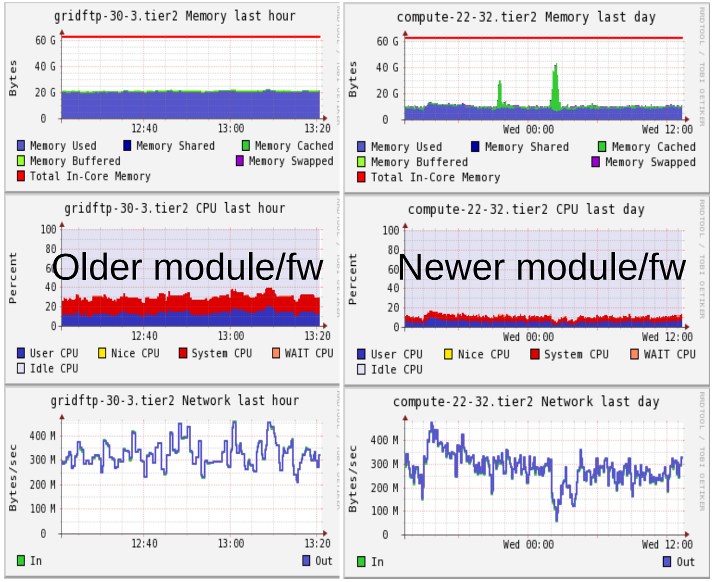
- Preferrably seasoned Site Admin / Sysadmin
  - CMS Experience desirable but not a requisite.
  - Replacing me at the Caltech T2.
  - Send resumes/references to dkcira@caltech.edu
- It was good to work with all of you and for CMS!

Strategy for transfer middleware

- GridFTP Strategy - 2 dedicated servers, pool of 6 “elastic GridFTPs”
  - More will be added if justified
  - Won’t lose 192 cores from batch system if transfers are calm.
  - Switchover could be automated
- Mellanox drivers improved significantly
  - 40 Gbps GridFTPs possible?

Older module/fw @ 40 Gbps





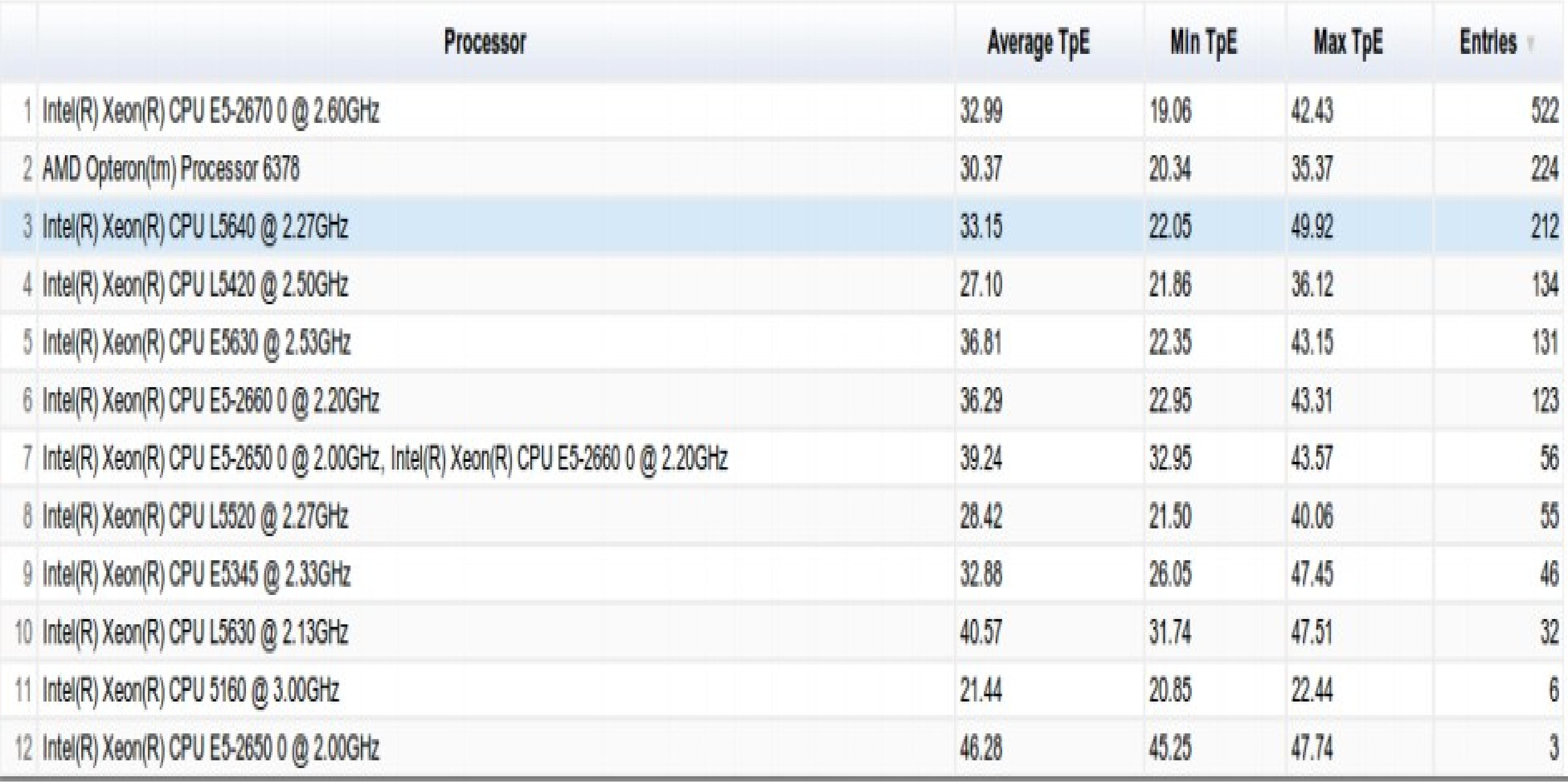
Systematic CMSSW Benchmarks

- HS06 is a good reference
  - Some suspect that it will eventually diverge from HEP software behavior
  - It’s not the actual software.
  - Requires license/deployment/execution effort.
    - Our Framework enables us to easily benchmark it.
- CMSSW is already deployed and working on worker-nodes
  - No deployment effort
  - Central reporting
  - See in details my HEPiX slides about this.
  - Code is available in GitHub

Status

- Currently have several running modes and PSets:
  - Running modes
    - Condor Benchmark - becomes a ClassAd
      - Thanks, Brian!
    - Whole node - isolated
    - Transparent - submit jobs to batch system
      - Optional CouchDB reporting
  - PSets :
    - Tier-0 reconstruction, 33 PileUp
    - Monte Carlo GENSIM

Monitoring CouchApp



	Processor	Average TpE	Min TpE	Max TpE	Entries
1	Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz	32.99	19.06	42.43	522
2	AMD Opteron(tm) Processor 6378	30.37	20.34	35.37	224
3	Intel(R) Xeon(R) CPU L5640 @ 2.27GHz	33.15	22.05	49.92	212
4	Intel(R) Xeon(R) CPU L5420 @ 2.50GHz	27.10	21.86	36.12	134
5	Intel(R) Xeon(R) CPU E5630 @ 2.53GHz	36.81	22.35	43.15	131
6	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	36.29	22.95	43.31	123
7	Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz, Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	39.24	32.95	43.57	56
8	Intel(R) Xeon(R) CPU L5520 @ 2.27GHz	28.42	21.50	40.06	55
9	Intel(R) Xeon(R) CPU E5345 @ 2.33GHz	32.88	26.05	47.45	46
10	Intel(R) Xeon(R) CPU L5630 @ 2.13GHz	40.57	31.74	47.51	32
11	Intel(R) Xeon(R) CPU 5160 @ 3.00GHz	21.44	20.85	22.44	6
12	Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz	46.28	45.25	47.74	3

Summary of transfer activities

- Goal
  - Scale up Grid Middleware to cope with new network speeds.
- Items that require effort/tuning :
  - Central - Transfer system (PhEDEx + FTS) be able to trigger a high number of parallel transfers.
  - Sites - Handle a high number of parallel transfers
    - Optimize individual transfer rate

So far

- Issues
  - PhEDEx configurations at sites was rather limiting
    - Thanks to all admins, that was fixed quickly
  - FTS Optimizer algorithm
    - Optimizer “bypass” got sites doing 20+ Gbps
    - High rate but not stable traffic.
  - Optimizer assumptions are rather optimistic :
    - Default : throttles if success rate < 99%
    - Most “aggressive” : throttles if success rate < 95%
    - Success rate : non-configurable for now.

Latest developments

- FTS3 deployed at Caltech
    - Improved control over configuration, better for tests.
- Found 2 other bottlenecks
    - PhEDEx will throttle transfers if too much recent failures between 2 sites. About 150.
    - PhEDEx queues - By setting a high LoadTest rate, you’re queueing several TB. High, Normal, Low priority queues have a limit of 15 TB.
        - In practice, one would manage to download from 3 other sites at most.

Conclusions

- It’s not “just about” raising LoadTest rates and having good, fast SEs on both sites.
  - It improved from what we had at the beginning.
- It might take more than 1 SA’s “free time” to brush out all the problems.
- It’s an interesting problem, and will benefit a large amount of sites when all works well.
- A number of sites have showed high rates with Xrootd, a good share of SRM transfers.
  - Are we ready to do the same with solely production SRM activity?

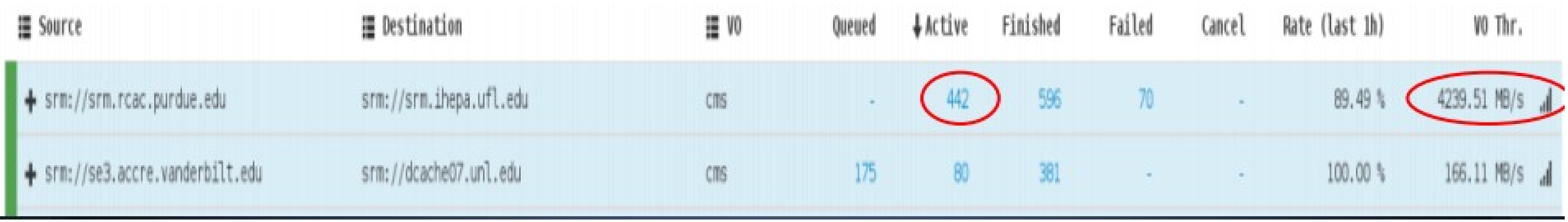
The End

Backup slides

Alternative for site rate testing

- Our Grid Middleware currently has a number of limits and algorithms that were fine for the previous scales.
- We’re finding/addressing as we go.
- For people that don’t want to be throttled at these several layers, there is an adptive SRM Client developed by LBL/UCLA :
    - Adaptive SRM client
- It will only depend on your client settings and the 2 sites.

TransferRate vs Success Rate



Source	Destination	VO	Queued	Active	Finished	Failed	Cancel	Rate (last 1h)	VO Thr.
srm://srm.rcac.purdue.edu	srm://srm.ihepa.ufl.edu	cms	-	442	596	70	-	89.49 %	4239.51 MB/s
srm://se3.accre.vanderbilt.edu	srm://dcache07.unl.edu	cms	175	80	381	-	-	100.00 %	166.11 MB/s

CPU-only resources

Now a reality

- Have a campus resource as a testbed.
  - Methods and tools used there could be easily applied to cloud resources.
- Main differences - site-local-config.xml; storage.xml
- Counts most on networks for I/O, but not all available clusters will have good networking.
  - Filter CMS jobs that are not too demanding for I/O.
  - Brian : receiving only production jobs is a good start.