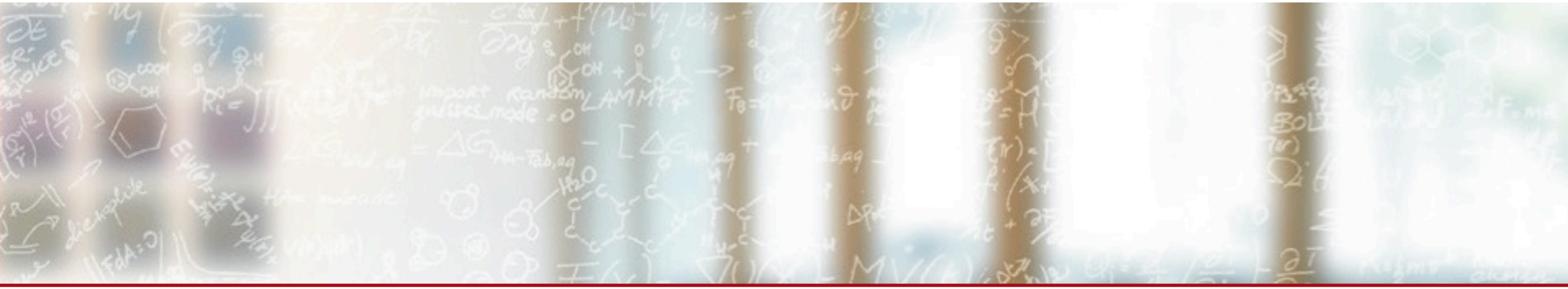
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Early experiences configuring a Cray CS Storm for Mission Critical Workloads

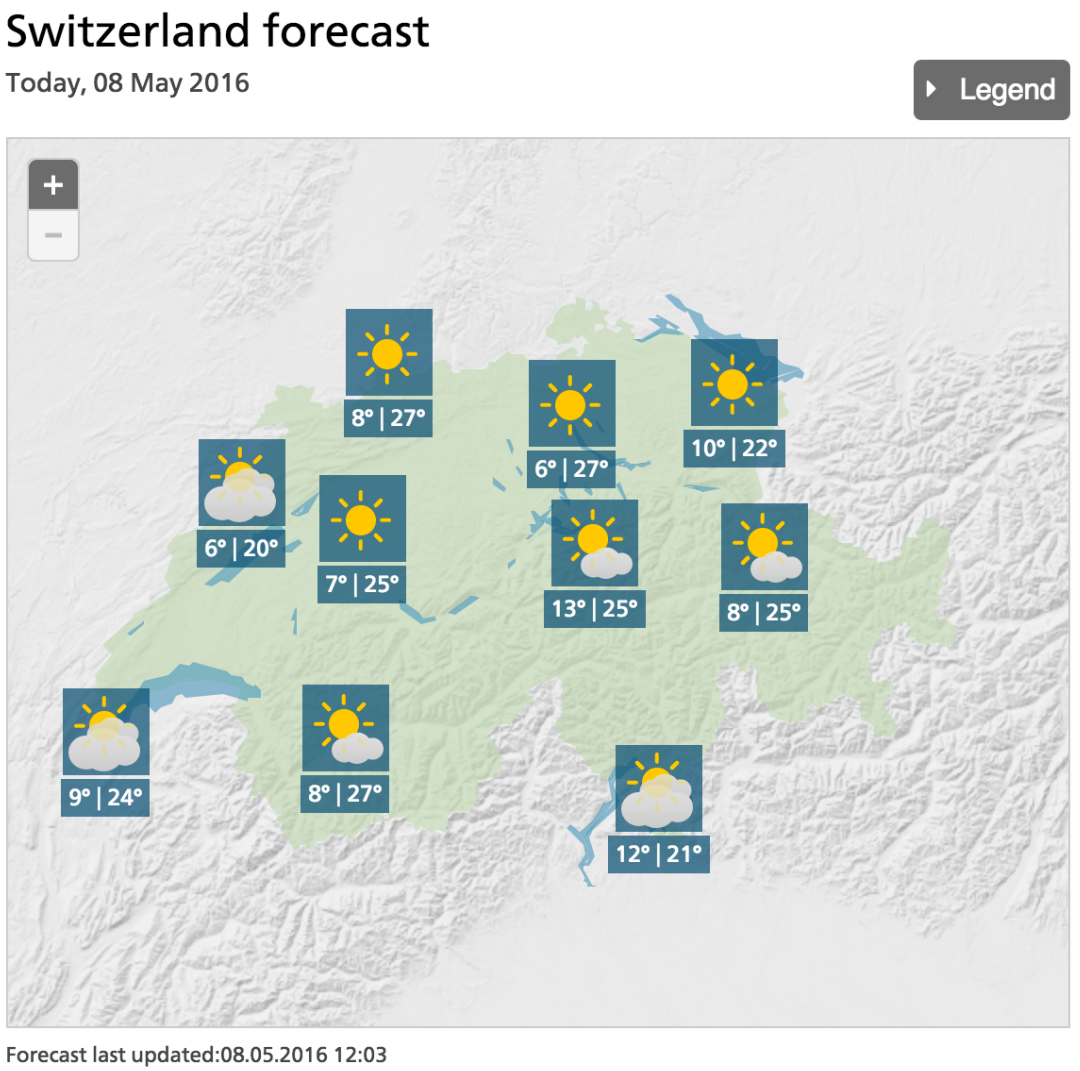
CUG2016

Mark Klein and Marco Induni, CSCS

May 11, 2016

Mission Critical

- MeteoSwiss
  - Weather Forecasting
    - 8 short-range forecasts per day
    - 2 long-range ensembles per day
  - Severe Weather Warnings
  - Air Traffic Safety
  - Disaster Modeling
    - Uses the short-range forecast model/data
    - Radiation/Chemical dispersion events



Mission Critical (cont)

- Short time between runs
  - Currently at most 1h 45m buffer
  - Not much time for scheduled maintenance
    - Can remove at most one per type of node for service actions
    - Theoretically, could lose a few GPUs per compute nodes
      - In practice, if GPUs are misbehaving, it's better to remove node and replace them ASAP
      - Most of the time a failed GPU takes out the node anyway.

Mission Critical (cont)

- Two Identical Systems
  - Data synchronization maintained by MeteoSwiss
  - In case of too many problems
    - Failover!
    - Can also be staged in case of maintenance
      - Upgrade one machine, shift production, upgrade the other
      - This limits the ability to failover, so still need to be quick
        - ACE makes it easy to roll back to old image if upgrade needs to be aborted for failover purposes

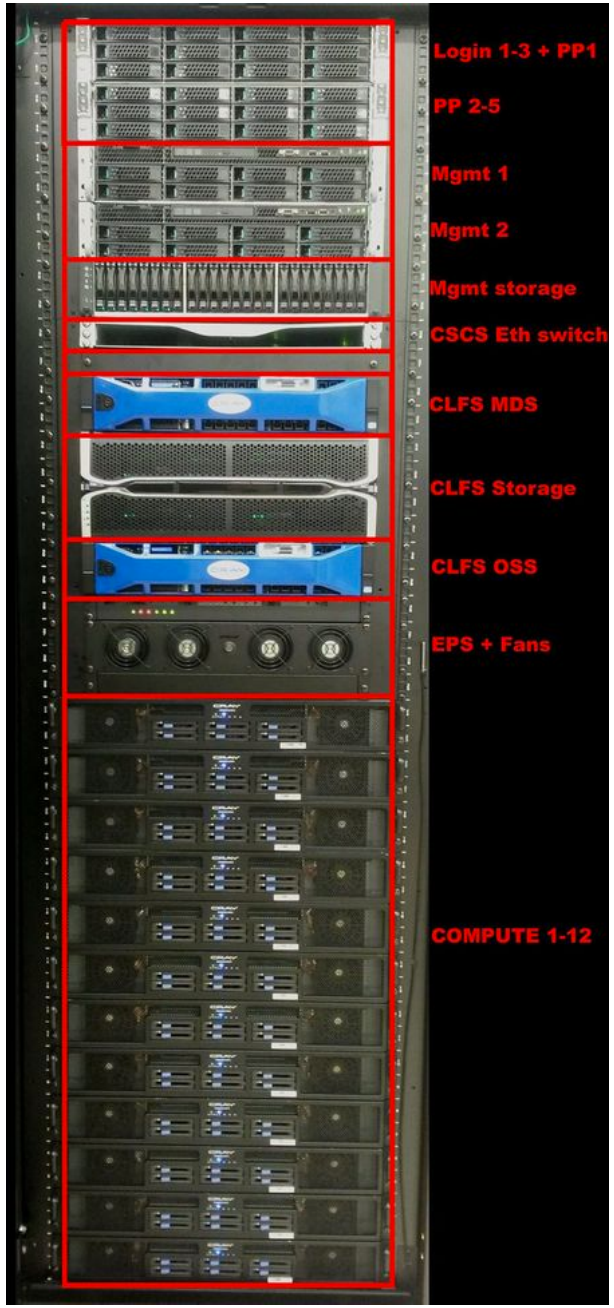
Moving from XE6 to CS

- Reduction in nodes
  - 72 nodes in XE6 System
  - 12 nodes in CS Storm
- Increased Compute Node density
  - Larger RAM
    - 32G -> 256G
  - Faster CPUs
    - Opteron 6172 -> Haswell
  - Addition of GPUs
    - Very dense GPU configuration: 16 cuda devices per node (192 total)
- Increased Computational Capability
  - Increased Forecast Resolution
  - Decreased maintenance windows

System Design

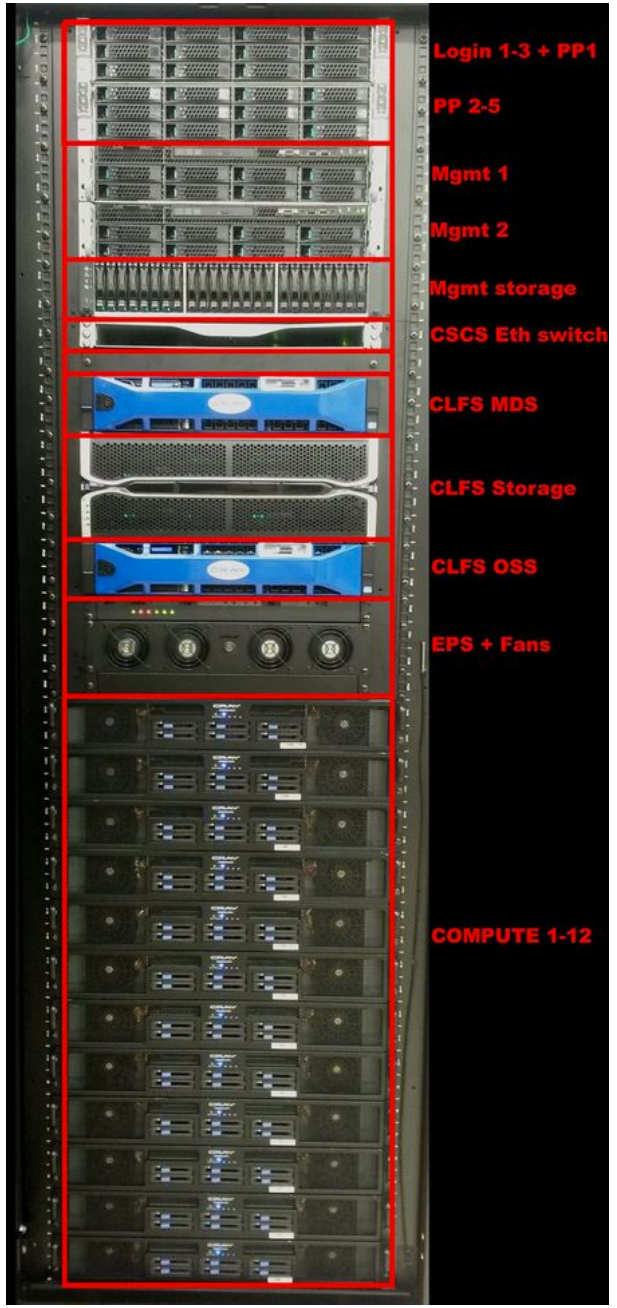
System Design

- Two Independent Identical Systems
  - Each Containing:
    - Two Management Nodes:
      - These are treated as appliances so hardware is really not that important, but:
        - Two Ivy Bridge E5-2680 v2
        - 64GB DDR3
        - Shared XFS HA Filesystem using NetApp 2724
        - Two ConnectX-3 HCAs



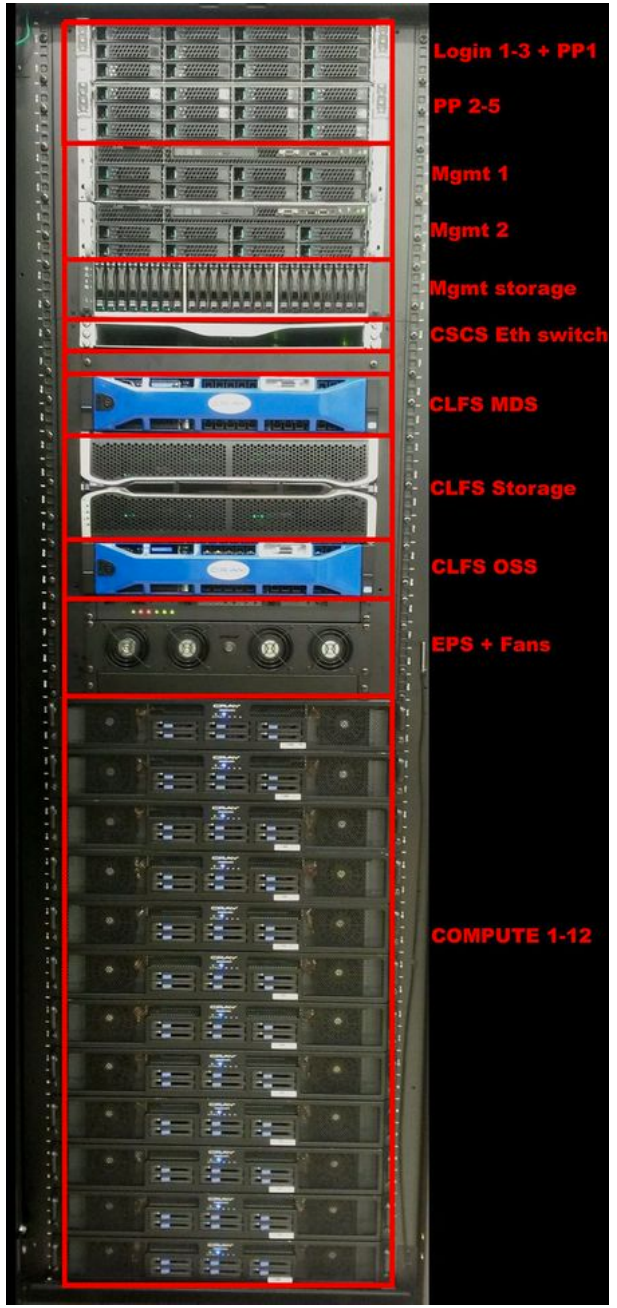
System Design (cont)

- Two Independent Identical Systems
  - Each Containing:
    - Three Login Nodes
      - Two Haswell E5-2690 v3
      - 128GB DDR4
      - Two FDR Infiniband Cards



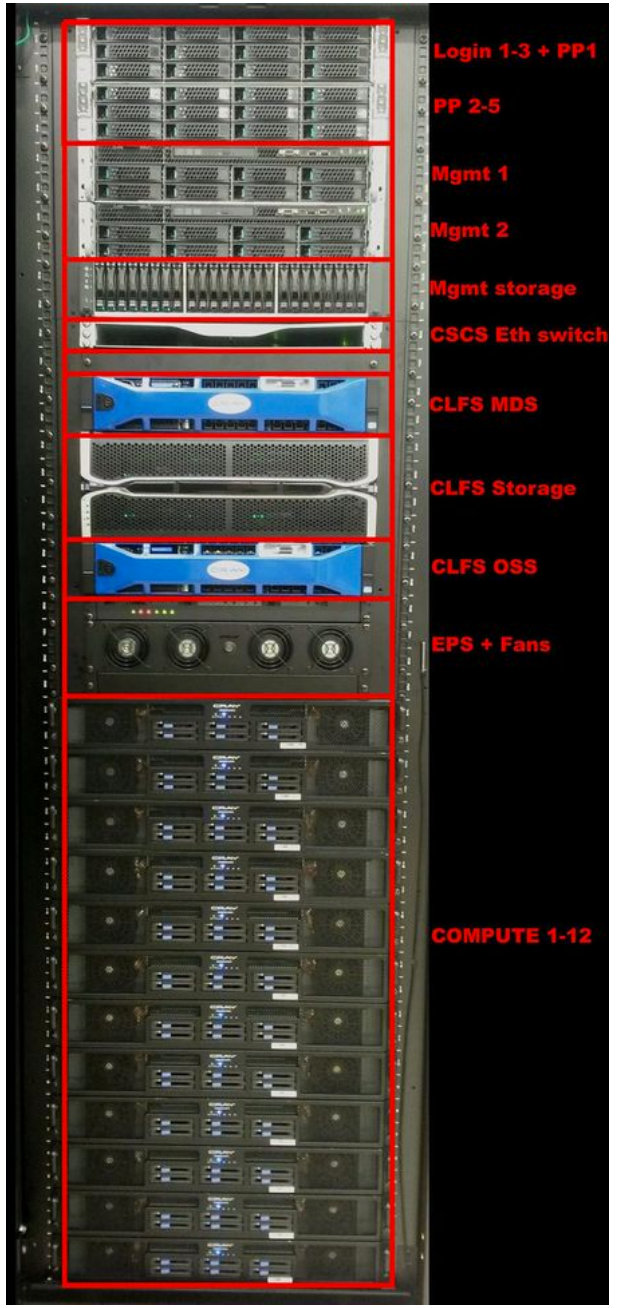
System Design (cont)

- Two Independent Identical Systems
  - Each Containing:
    - Five Post-Processing Nodes
      - Two Haswell E5-2690 v3
      - 128GB DDR4
      - Two FDR Infiniband Cards



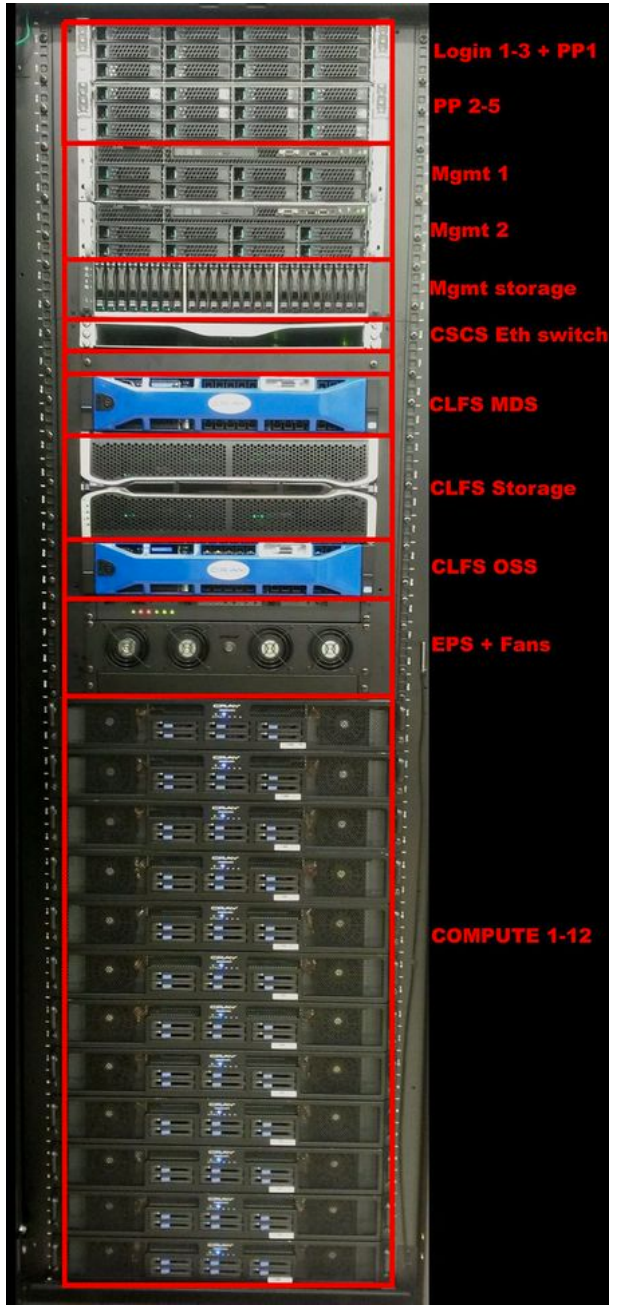
System Design (cont)

- Two Independent Identical Systems
  - Each Containing:
    - Twelve Compute Nodes
      - Two Haswell E5-2690 v3
      - 256GB DDR4
      - Eight K80 GPUs
        - 16 total CUDA devices
      - Two FDR ConnectIB Cards
        - Three total ports
        - One card attached to each PCI Root
      - Diskless



System Design (cont)

- The storage system is based on CLFS
  - Very small ~72TB operational scratch
    - 1 OSS
    - 1 MDS
  - Which means the storage has it's own management system
    - Bright Cluster Manager
      - In a separate rack, we needed room for an CSCS ethernet switch
  - Can see why it was done this way
    - CLFS is same release that shipped on the big Cray product lines
  - However, would be nice to maybe combine the Lustre deployment with ACE



Early Experiences

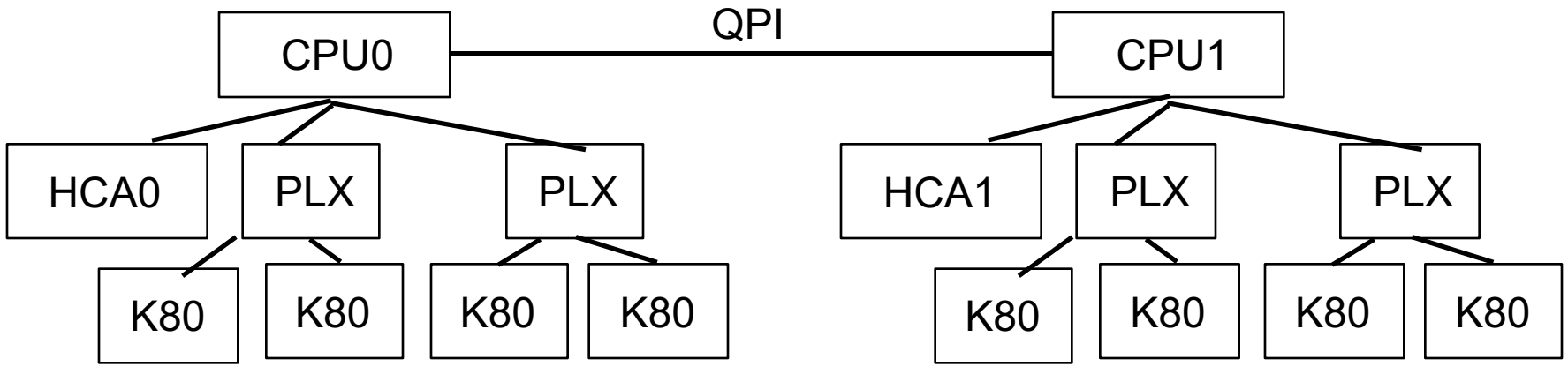
Early Experiences

- CS-Storm is a very new machine to us
  - Previously running on Cray XE6
    - xtopview vs ACE
      - Live changes to images are very limited
    - CPUs vs GPUs
      - GPU Affinity, GPUDirect RDMA
      - Performance fluctuations due to thermal on GPUs
    - Much different support model
      - Free to modify the OS of the cluster
        - No Compatibility Matrix, some trial and error required

Early Modifications to the System

- System shipped with
  - Compute nodes set up managed by Ace
  - Login/Post-Procs set up managed by Nothing
    - Why?
    - 8 additional installations to maintain (per system)
    - Not sustainable
  - Cray recreated these for us as Ace images
    - Treated as unique clusters
    - Required some custom boot scripts to bring up interfaces and daemons correctly

GPU Affinity for Slurm



- Benefit of Haswell Based CS-Storm
  - ConnectIB FDR cards now exist on both PCI Roots
    - Avoids the QPI link allowing all GPUs to be able to use GPUDirect RDMA
  - MVAPICH is good at making decisions in a bubble, but this is a shared system
    - Multiple Jobs are running on each node at a given time
    - Multiple GPUs are going to be assigned to a given job on a node
  - Slurm GPUs are handled as a GRES
  - Slurm CPU assignments are handled by taskselect/affinity
  - How to make this all play nicely?

GPU Affinity for Slurm

- Custom implementation
  - When machine shipped, 14.11 was latest available
    - Found Presentation from Bull at GTC
    - Early access to Bull's custom Slurm code
      - Hardcoded for specific test system
      - Did not use GRES: no GPU usage tracking/blocking
      - Found that all changes needed took place in slurmstepd
      - Took ideas and implemented as TaskProlog
  - 15.08 adds native affinity (--accel-bind), but we still use the Custom implementation
    - Some edge cases were failing on 15.08 method
      - When heavily loaded, GPU availability may not match CPU availability
      - Slurm's best-effort on GRES assignment doesn't always play nicely with the accel-bind.
      - Rather allow non-optimal GPU assignment than tasks failing for not enough GPUs
    - Currently slurm does not set MVAPICH ENV variables
    - TaskProlog remains flexible
      - Easy to fix edge cases that come up
        - CCE 8.4 OpenACC Regression:
    - Eventually would like to rewrite this into something a little more standard but it currently is working very well for the needs of MeteoSwiss

GPU Affinity for Slurm

- A little about our implementation
  - Two modes of operation:
    - G2G=1
      - Similar in design to the launch wrapper for xhpl
      - In this mode, each task sees a single GPU, and a single (optimal) network interface
      - Useful for codes that do not handle multiple GPU selections
    - G2G=2
      - In this mode, each tasks sees all the GPUs assigned to the job, and the task optimal network interface
      - Additional variables are exported to assist with the selection of GPU device
        - LOCAL_RANK, MV2_COMM_WORLD_LOCAL_RANK, etc.
        - The CUDA_VISIBLE_DEVICES are reordered in such a way for optimal selection by index
      - This mode is preferred if the application is able to handle GPU Selection
      - Additional Variables are currently set to work around a CCE 8.4 regression
        - In 8.4.0-8.4.5, OpenACC was ignoring device setting, placing all GPUs on Device 0
        - Can work around by exporting OMP_DEFAULT_DEVICE indexed to the GPU to run on
        - Fixed in 8.4.6

GPU Issues

- A very small number of thermal throttling has occurred
  - ~6 times over the life of the system so far
  - Have yet to trace it to a certain event
  - Ongoing investigation
- More frequently seen: GPU bandwidth degrades
  - Caught by regression testing
  - PCI links on nvidia-smi report full speed/width
  - The issue is usually one of the PLX chips
    - 2 GPUs slow is most likely the card
    - 4 GPUs slow is probably the cable between motherboard and riser
      - Rarely the riser needs replaced

Red Hat Kernel Bug

- System shipped with Red Hat 6.6
  - futex_wait() deadlock bug in kernels 2.6.32-504<=2.6.32-504.12.2
    - This was triggered by MeteoSwiss code
    - Very hard to diagnose
      - Attaching gdb or strace wakes process up and continues
    - Discovered mention of similar problem on a mailing list
      - Upgrading Kernel fixed problem confirming suspicions

ACE

- Things it does well
  - When it works, it is quite user friendly.
    - Able to figure out how to do most management tasks by looking in /opt/ace
    - Both the GUI and Command Line work well most of the time
    - Basic system monitoring works well
      - CPU Temperatures, Loads, Uptime
    - Ability to have multiple revisions of an image is useful
    - If all needed kernel modules are in the image, seems fairly intelligent in building initrd

ACE

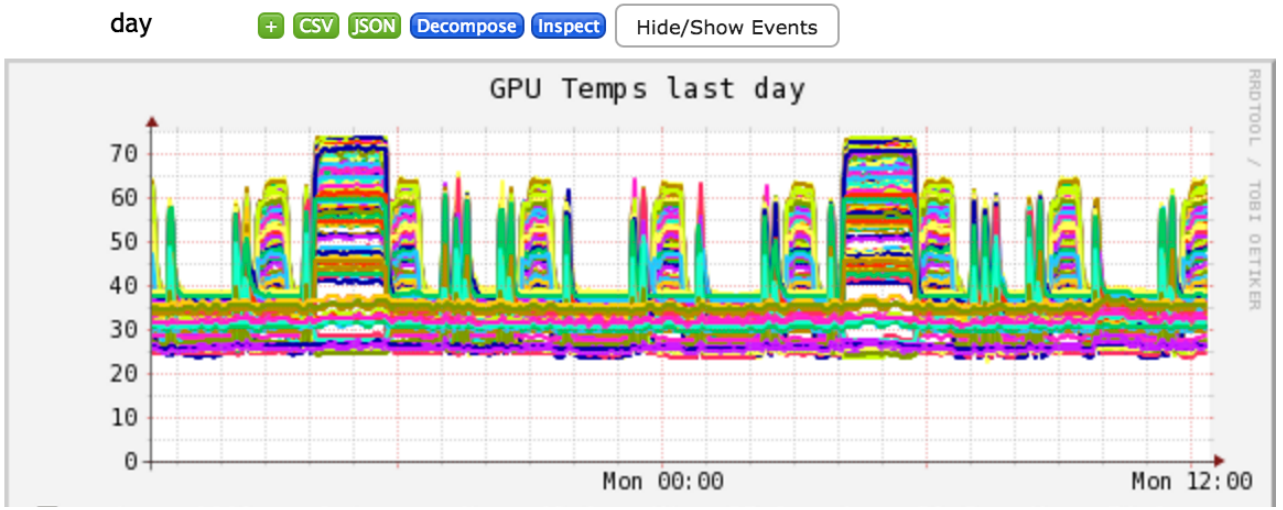
- Some lacking functionality
  - The Documentation was pretty much useless
    - Many tasks were left to us to figure out
      - Updating Kernel took a lot of trial and error
      - Needed to boot: gnbd, OFED, and Lustre
    - Documentation has gotten better
  - Weirdness with Image Management
    - By default, even a small change requires a reboot
      - Overcome by moving files to either ACEFS or the /global NFS mounts
    - Limit to 10 revisions
      - Not guaranteed to be consecutive revision numbers
        - Marco Induni wrote a nice acerev alias to sort by checkin date
      - Updating Kernel will take two revisions, one for Kernel update and one for nvidia/gpfs/etc updates
    - Export/Import images
      - Boot only one system the first boot after an import, or the image corrupts

ACE

- When it breaks, it breaks hard
  - Daily backups for the acedb are incredibly useful
    - We've seen AceDB corruption twice now
  - XFS Failover questionable
    - We've seen XFS corruption twice now that seems to be related to failover events
      - Still under investigation

System Monitoring

- ACE includes a custom Ganglia
  - In theory possible to add custom monitors
    - In practice likely caused one of the corrupted acedb events
  - Safer to just install real gmond on the nodes
    - Works very well for our small cluster, your mileage may vary
    - To minimize jitter, configured in unicast: Nodes->Management<->CSCS Central Services DB
    - Very useful monitors available for GPUs
      - Power, Clocks, Temperature, ECC Errors
      - Trace throttle events quickly at a glance
      - Yes, there really is a 35 degree gradient

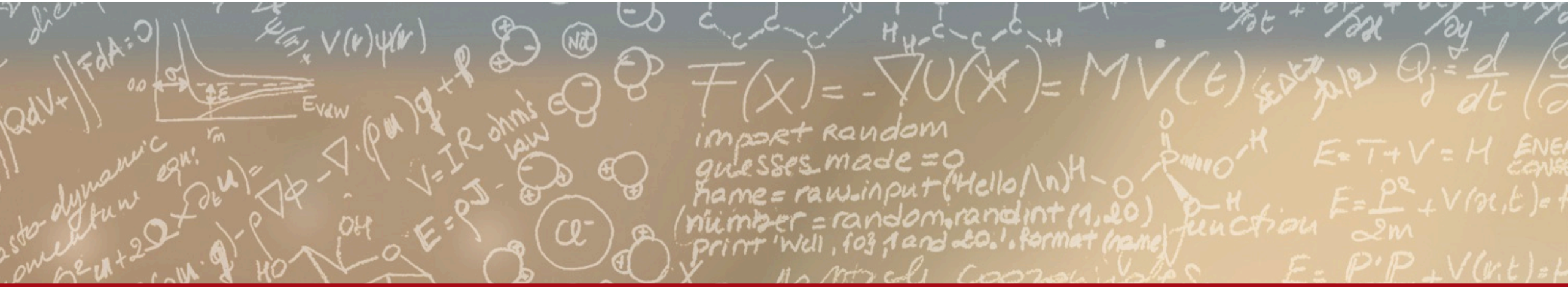


Support on the System

- Long term support is an open question
  - No place to go to check latest ACE versions
    - We had a bug that was crashing aced reproducibly
      - Given fix immediately, but the build date was over a year old
    - Recently told ACE is being completely replaced
      - Timeline unknown
      - Unsure what this means for the currently install
  - CLFS is end of life
    - Tried updating clients to a later release, caused MDS to die
      - Rolled back to older release, now running Cray C3.
        - Stable, but still getting some large performance fluctuations on certain codes
        - Investigation ongoing

Conclusion

- Had to make a number of modifications to the system to get to this point
  - Documentation is better but still needs a little work
    - I understand this is a difficult problem because each CS is unique
- Overall happy with the hardware
  - So far production has been going well with limited problems
- Overall ok with the software
  - Learning the new management system has been easy
  - A few open issues, but nothing show stopping
- Support has gotten better
  - Still a number of unanswered questions about the long term plans of the product line
    - CLFS EOLed
      - Currently stuck on Centos 6.4
      - Informed there is at least a 6.6 available
    - Recently learned ACE also going away
      - Not sure when replacement will be available or what this means for our system
      - Mission Critical means that we can't really take these systems down for extended lengths of time



Thank you for your attention.