SAND2014-3299P

Large Scale HPC Monitoring

NMSU Las Cruses New Mexico

April 23, 2014

Overview

- Technology Description
- Project Goals
- Deployment configuration and architecture of data collection component
- Use cases including data visualizations
  - Sandia
  - National Center for Supercomputing Applications
- Overview of current analysis and visualization status
- Conclusions
- Future Work

Description of Technology

- What is it?
  - Highly scalable platform independent system for HPC resource monitoring and understanding
- What does it do?
  - Collects, transports, and stores both numeric and log run-time information
  - Provides plugin capabilities for analysis, visualization and decision support including direct application/system feedback
  - Admin collection for troubleshooting, root cause analysis, and understanding of application/system interaction
  - User driven collection for understanding system/application interaction
- Addresses Large scale HPC system understanding and troubleshooting
  - System snapshots
  - Correlation of events and behaviors to degraded performance/failure

OVIS Framework

Sample

Bundle

Sample

Bundle

Aggregate

- Notification
- Feedback

Original Set

Bloom Filter

Filtered Set

Filter

Raw

- Analyze
- Visualize
- Model

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

Store

Sandia National Laboratories

Significance of Technology

- Technological advantage
  - Enable whole system state snapshots that can:
    - Pinpoint issues driving run-to-run application performance variability
    - Help system administrators troubleshoot problems
    - Relatively fine grained information can help users discover performance issues and solutions without having to re-compile or re-link their programs
  - Scalable auto-discovery of problems
  - Decision support for problem identification and resolution

Competitiveness

- How does this technology compare with its competition?
  - Complete solution (others tackle a piece of the problem)
  - Low overhead (small CPU and memory footprints)
  - Easy to build, configure, and use
  - Modular for plug-n-play functionality
- Crucial factors
  - Support for RDMA transport across a variety of interconnects
  - Sampler plug-ins (both kernel and user) that support run-time modification of sampling frequency
  - Large fan-in ratios (> 10,000:1)
- Intellectual property position
  - The software is open sourced and publically available
  - SNL and OGC hold joint copyright

Wow! Factor

- What is compelling about this technology?
  - First comprehensive **platform independent** HPC monitoring, transport, storage, analysis, visualization, and feedback **system**
  - Low overhead (both CPU and memory)
  - Simple to build, configure, and use
  - Extensible
  - Modular (Plug-n-play)
  - Highly scalable (no known limitations due to parallel everything)
  - Large fan-in ratio (> 10,000:1)

Project Goals

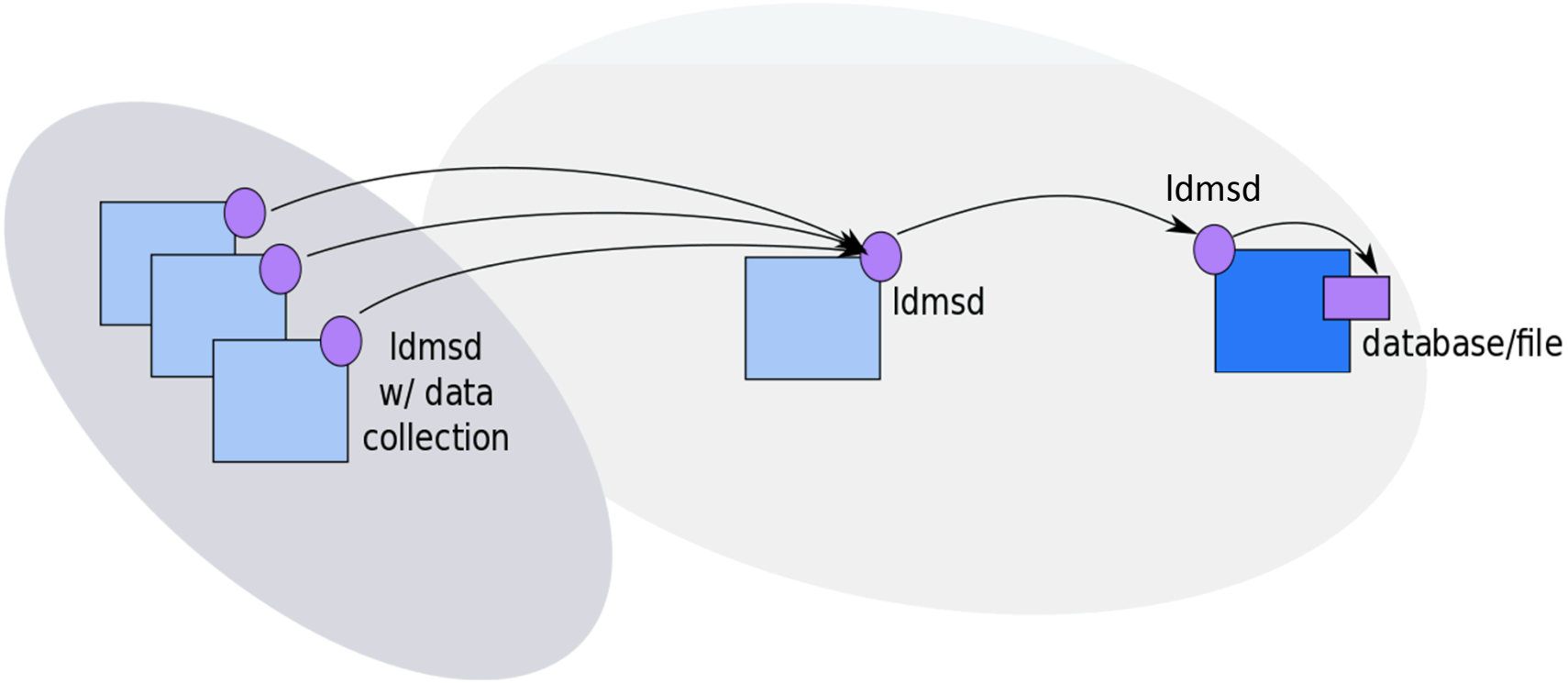
Monitor system resource state and utilization as a system service, running asynchronous to applications, for both administrators and users of High Performance Computing (HPC) systems

- Provide:
  - Run-time collection of metrics of interest on time scales of interest
  - Per-node resource utilization understanding
  - Application profiles
- Lightweight (negligible application impact)
  - ~1.5MB memory footprint
  - CPU overhead is dependent on number of metrics, frequency of collection, and communication technology
- High fidelity (~< 100Hz)
  - Typically run at intervals of seconds
- Platform independent (i.e. portable across Linux OSs)
  - Fedora, RHEL, SUSE, CentOS, Mint, Ubuntu
- Support for Socket and RDMA on major network technologies
  - Ethernet, Infiniband, Cray Gemini/Aries
- Simple configuration

Data Collection

Lightweight Distributed Metric Service (LDMS)

Deployment Configuration: Data Collection, Transport, and Storage



LDMS Metric Set Examples

cn1/meminfo

- U64 160032 MemFree
- U64 181728 Buffers
- U64 3443332 Cached
- U64 33076 SwapCached
- U64 2987544 Active

cn1/procstatutil

- U64 1826564 cpu0_user_raw
- U64 699631 cpu0_sys_raw
- U64 663843760 cpu0_idle_raw
- U64 201018 cpu0_iowait_raw

cn1/vmstat

- U64 40008 nr_free_pages
- U64 122286 nr_interactive_anon
- U64 321902 nr_active_anon
- U64 465532 nr_inactive_file
- U64 424986 nr_active_file

Metric sets:

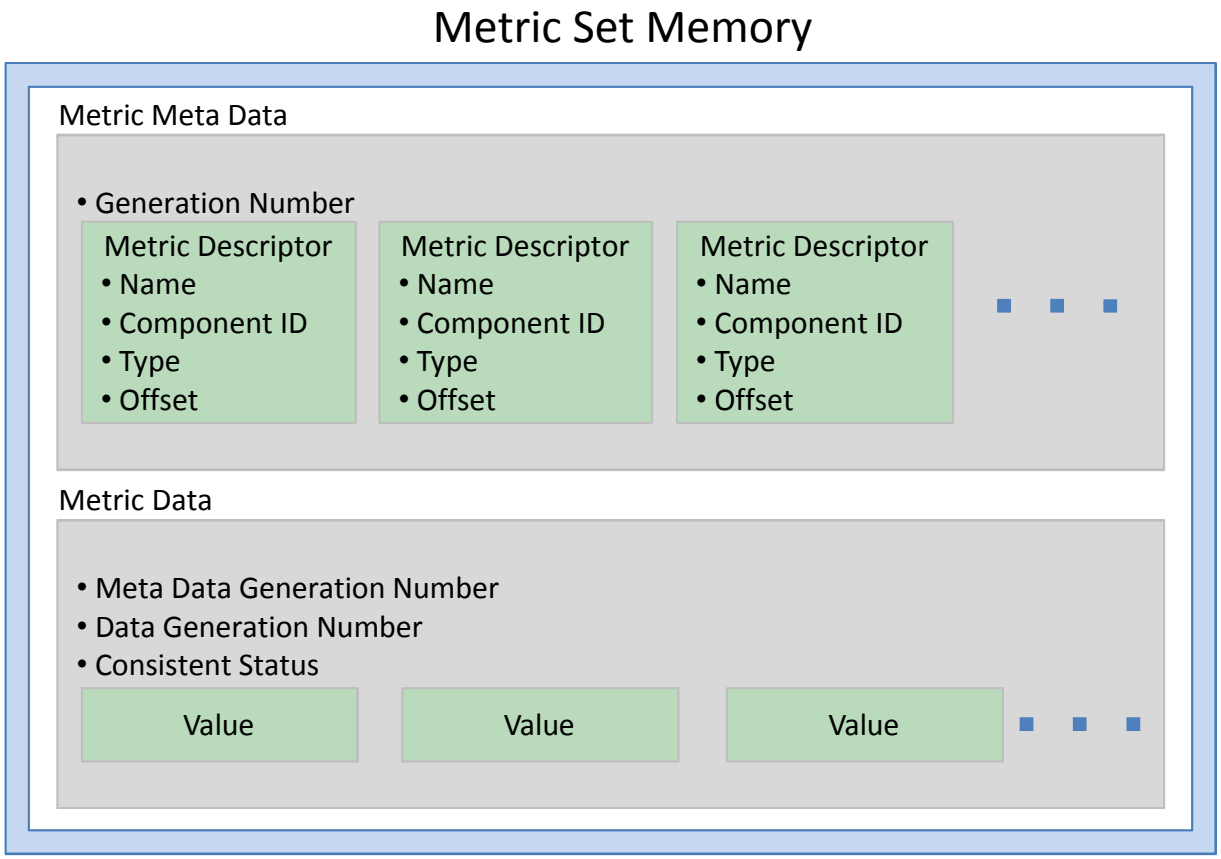
- (datatype, value, metricname) tuples
- optional per metric user metadata e.g., component id

API:

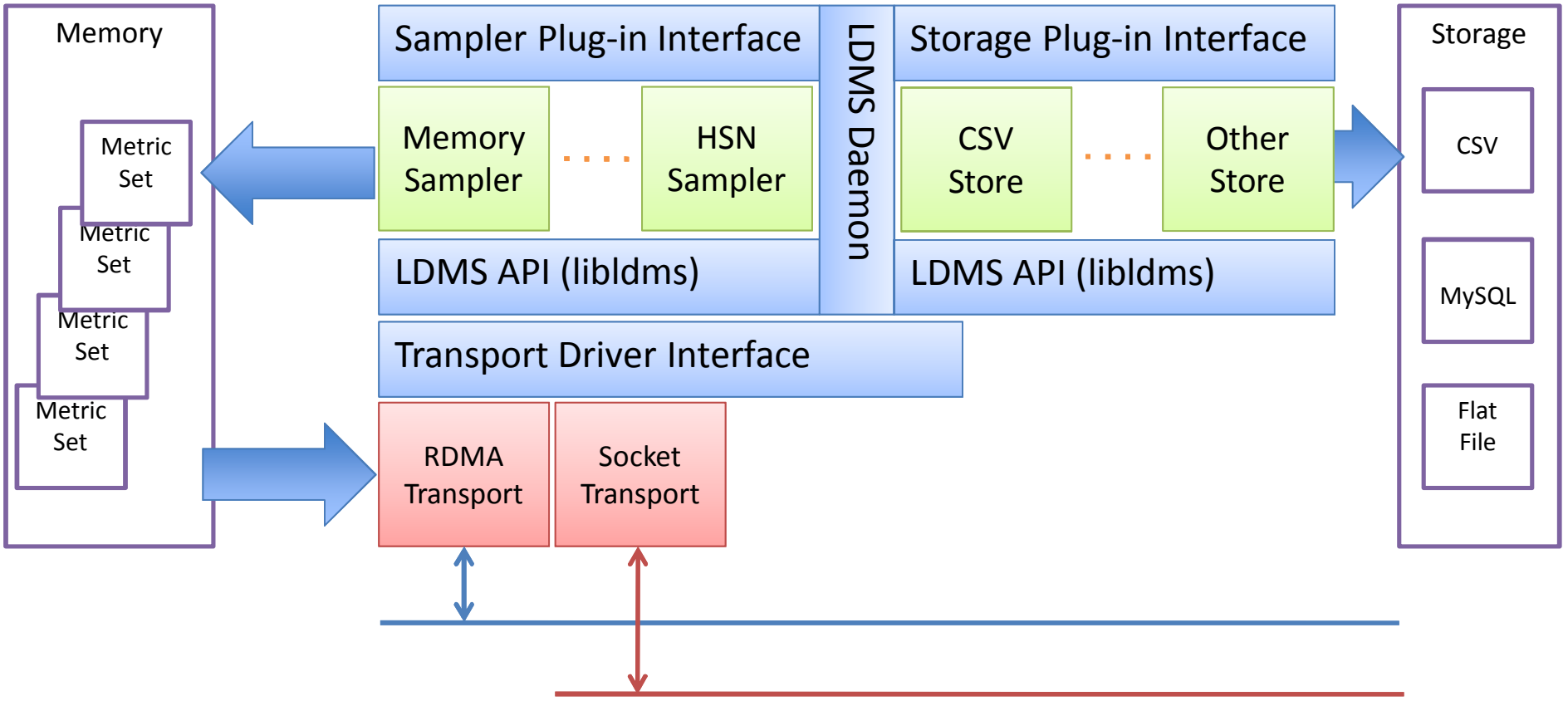
- *ldms_get_set*
- *ldms_get_metric*
- *ldms_get_u64*

- Same API for on-node and off-node transport

Metric Set Format



LDMS Architecture



Major Functional Components and Features

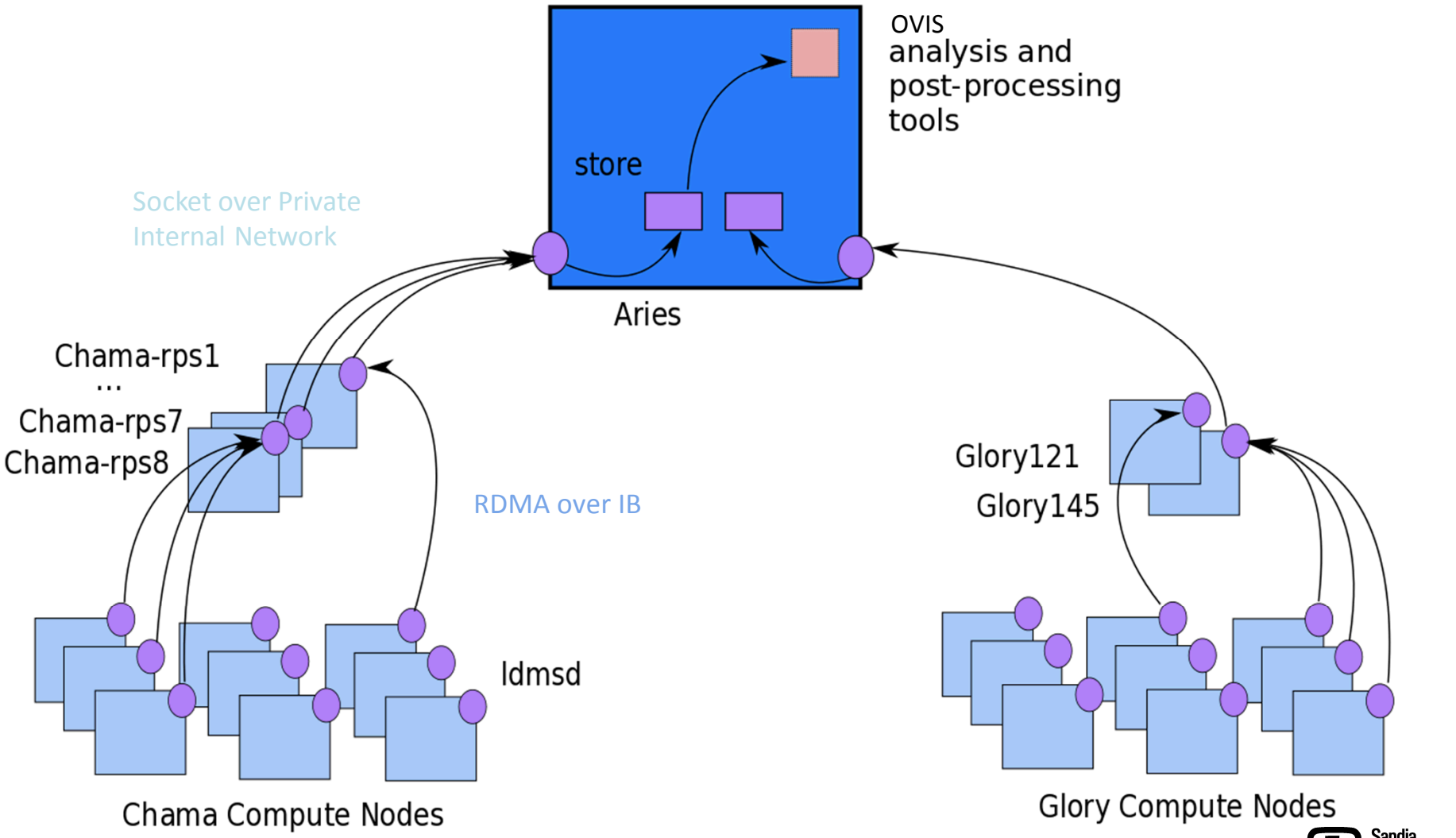
- Collection
  - Run-time loadable monitor plugins (collect data into a “metric set”)
  - Run-time configurable sampling period from ~100Hz to days
  - Variety of collectors draw from /proc, /sys, lm-sensors, perf-event
  - Control is at the granularity of a “metric set”
  - Synchronous option enables “system view” to within clock skew
- Aggregation
  - Fan-in of thousands to 1
  - Support for failover configuration
  - Supports daisy-chaining
    - Aggregate from collectors and/or aggregators
- Storage
  - Support for: CSV, flatfile, MySQL, custom

Current Monitor Plugins

- /proc
  - meminfo, vmstat, net/dev/stat, interrupts, nfs
  - kgnilnd (Cray specific)
  - Lustre
- cray_system_sampler (Cray specific)
  - Gemini Tile and NIC counters w/ link aggregation
  - Lustre llite counters
  - A variety of metrics from other sources
- perf_event
  - Generic interface for acquisition of hardware counters e.g., data cache misses, instruction cache misses, hyper-transport bandwidth
- rsyslog (Cray specific)
  - SEDC (RAS) and ALPSdata
- lmsensors (`/sys`)
  - Temperatures, fan speeds, voltages
- IB traffic counters (`/sys`)

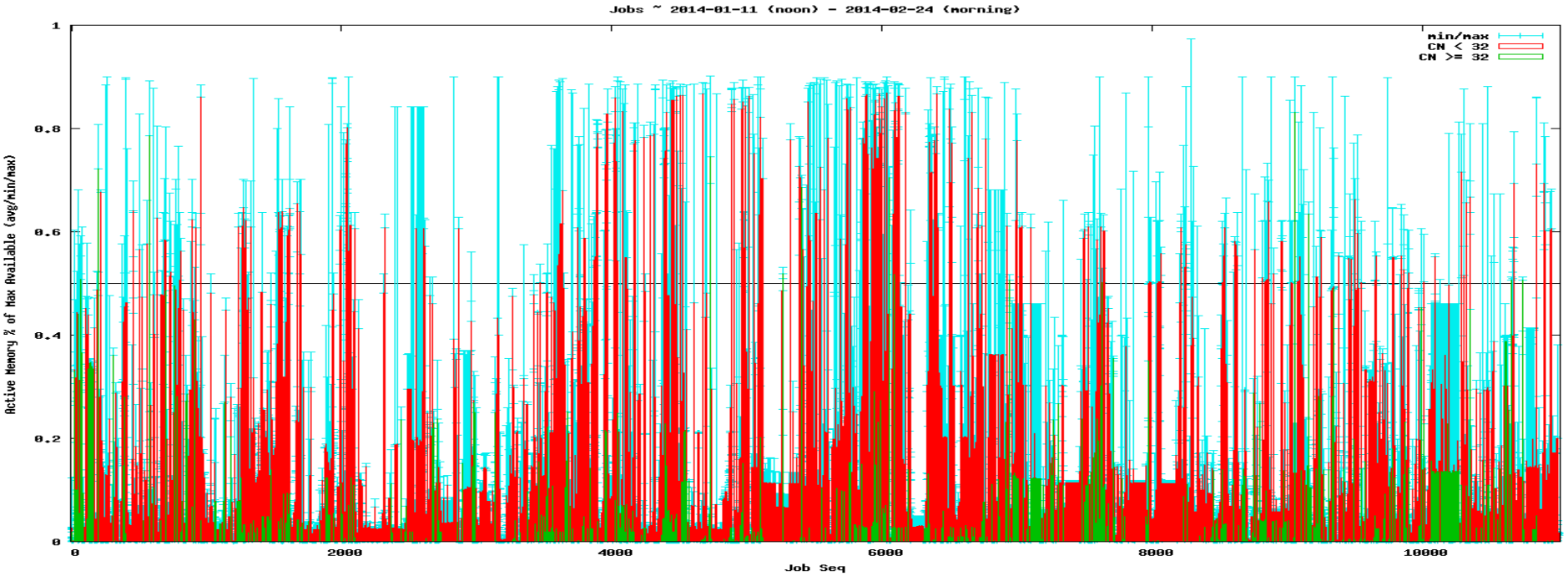
USE CASES: PROFILING SYSTEM AND APPLICATION RESOURCE UTILIZATION

Chama: 1232 node TLCC2 cluster (SNL)
Glory: 288 node TLCC1 cluster (SNL)



Sandia National Laboratories

Memory Use Across All Jobs
(Chama)

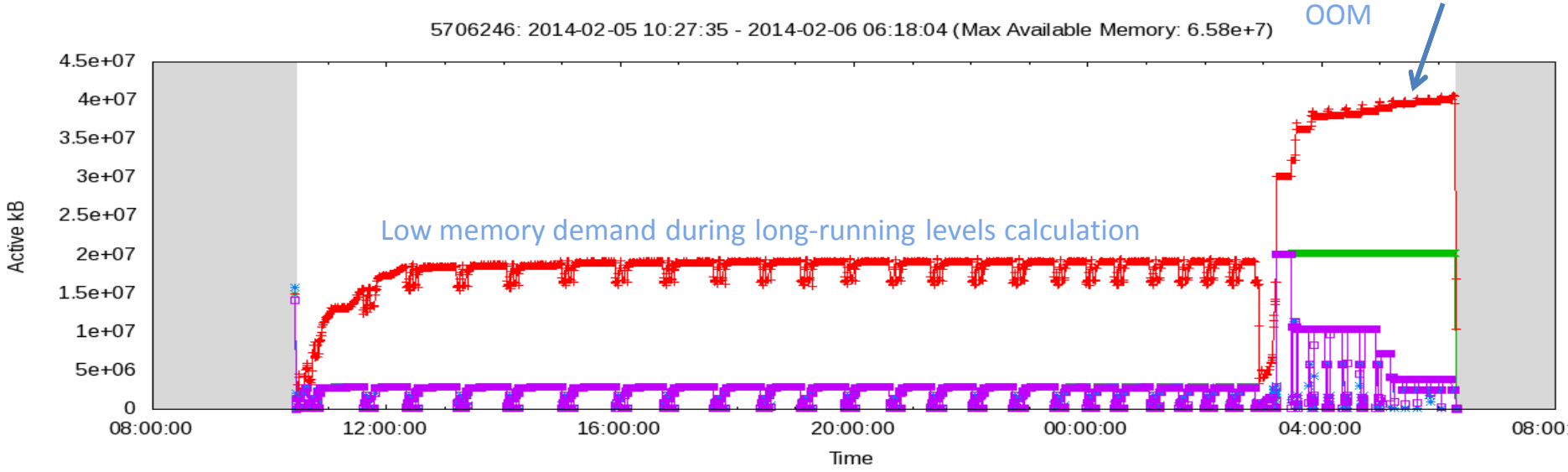


- Green represents jobs with greater than or equal to 32 nodes
- Blue (error bars) shows high water mark while green and red are average over job

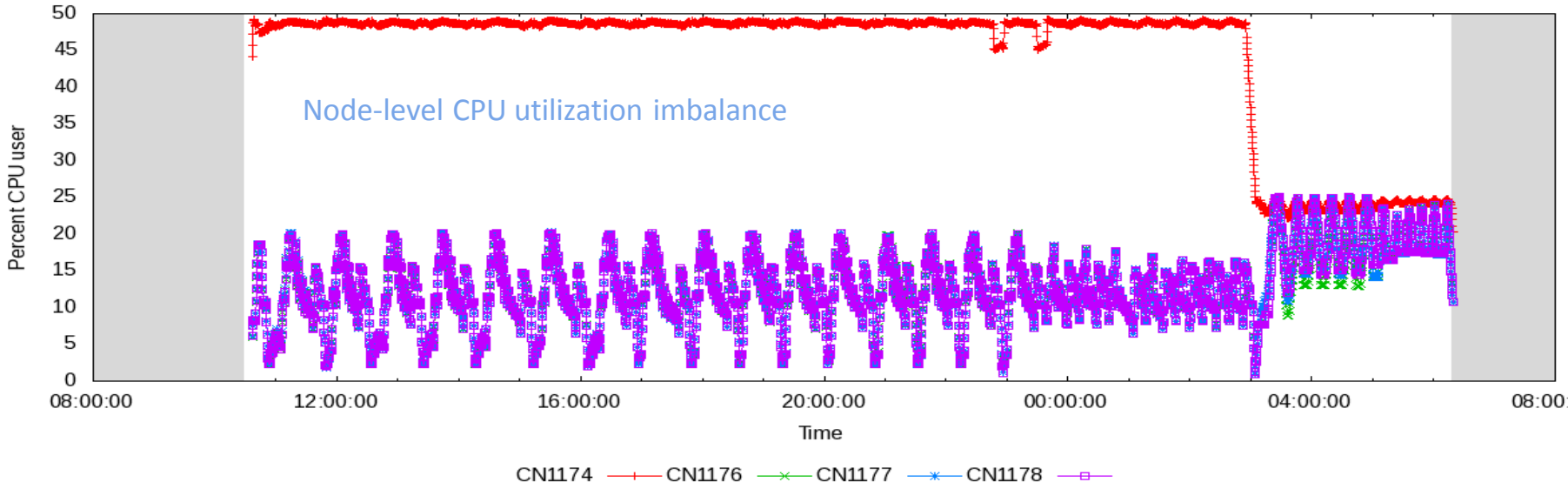
Sandia National Laboratories

Application Profile: Gaussian

High memory demand during DFT may result in OOM

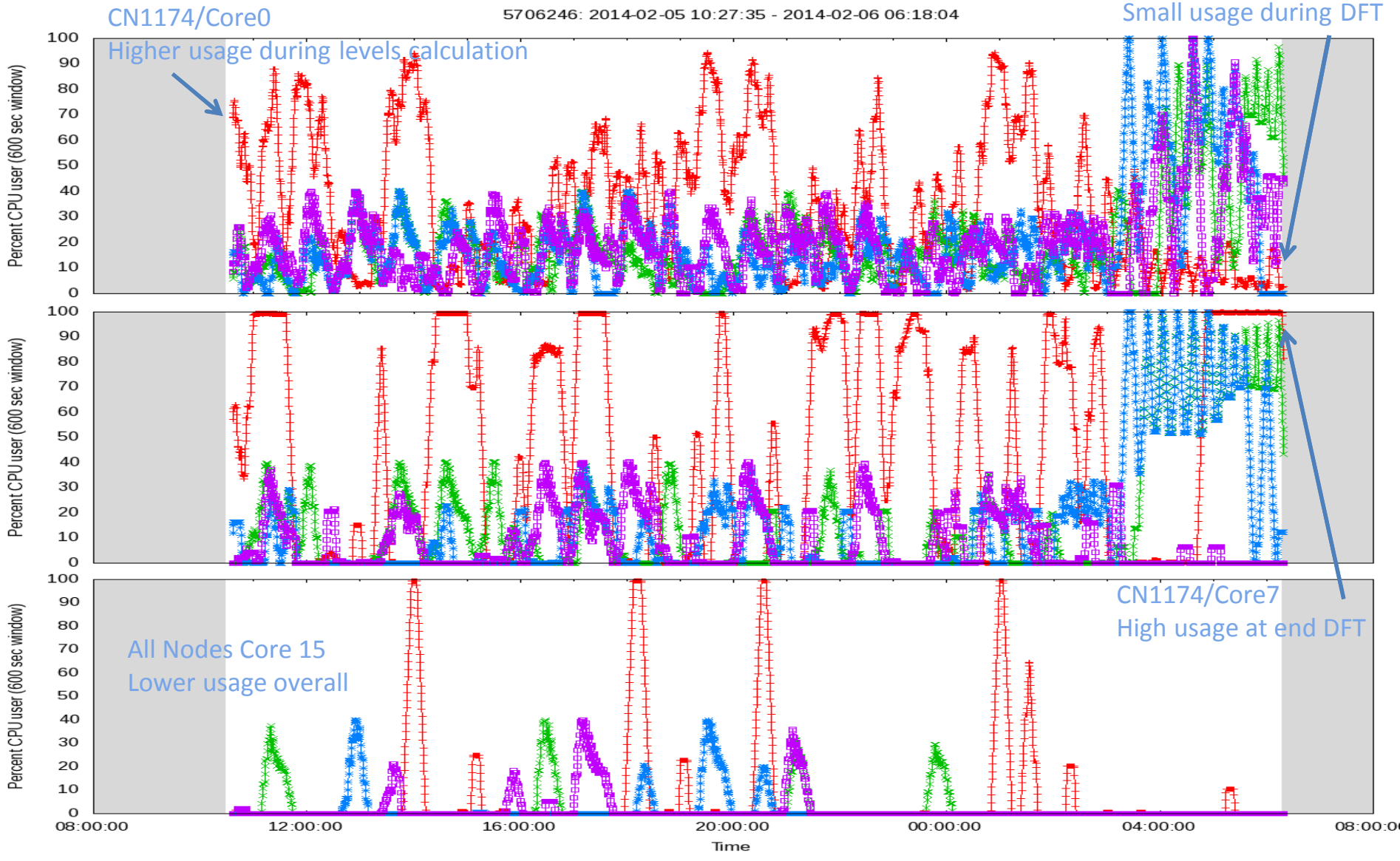


Two-execution phases are potentially separable enabling better application-to-resource mapping

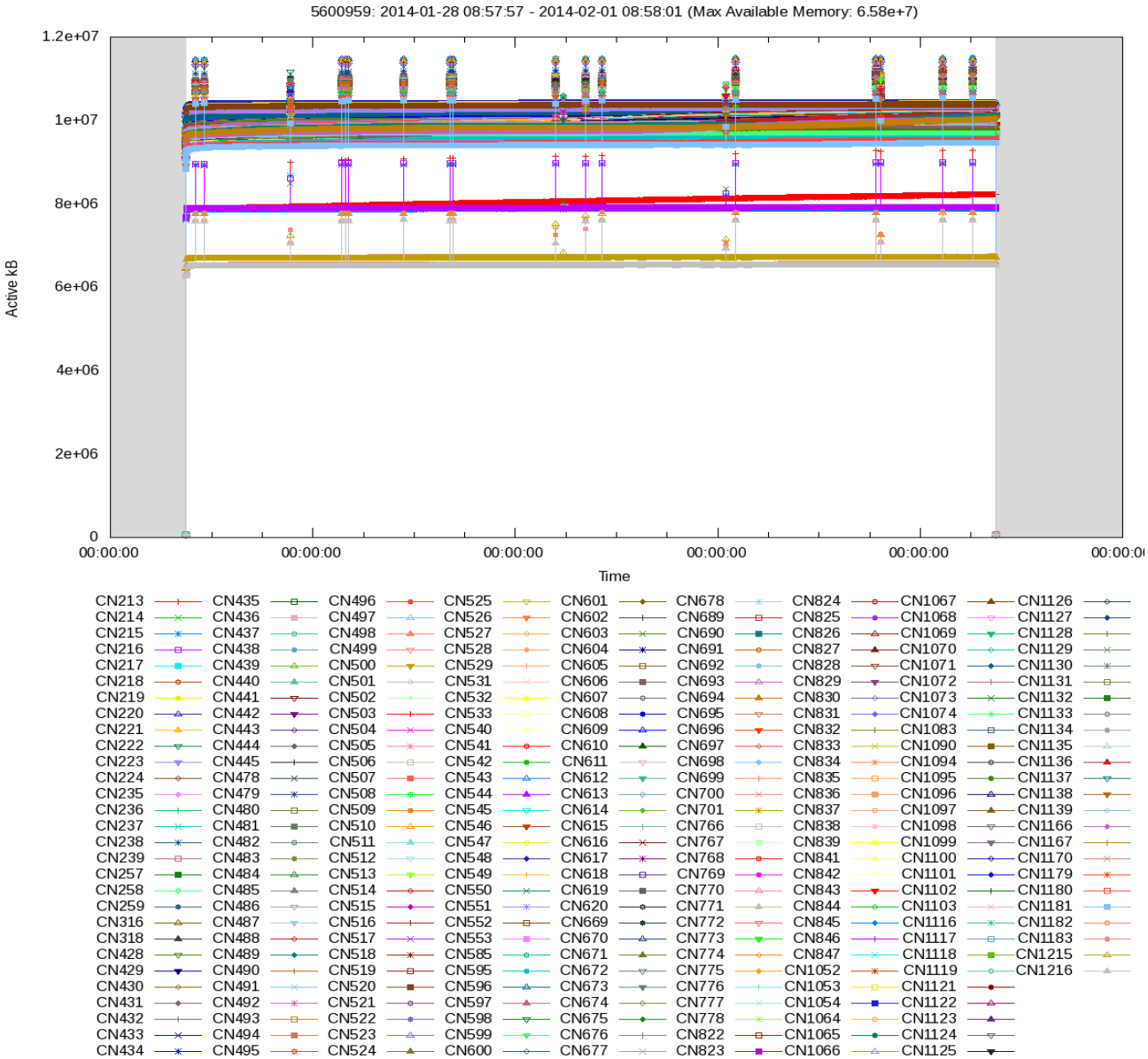


Gaussian: Per core CPU Load

(Cores 0/7/15)

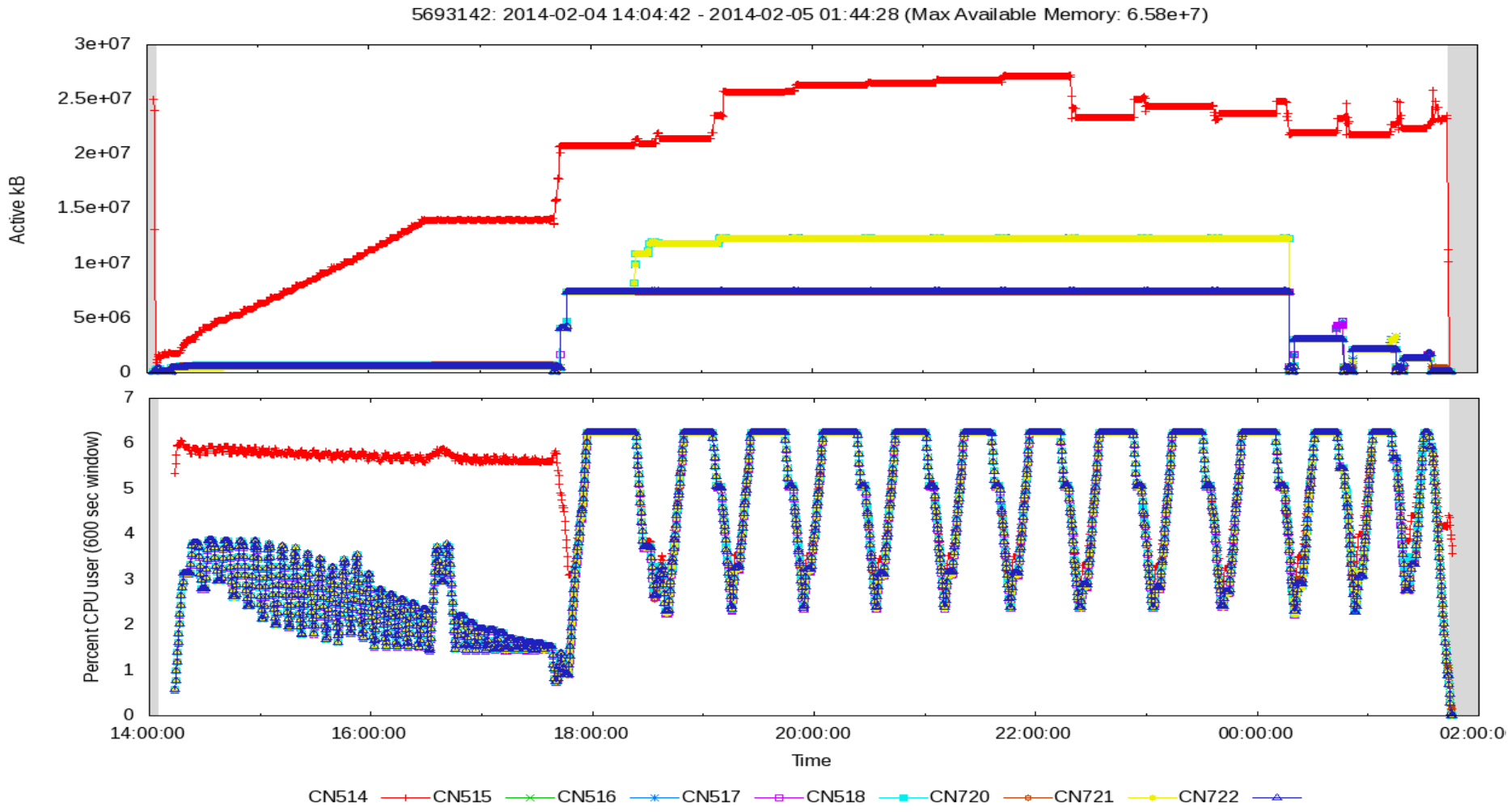


Application Profile: LAMMPS



- Generally well balanced in memory usage
- Running on fewer nodes can increase memory usage (17%->25%)
- However application is CPU bound, running nearly 100% user time on all cores.

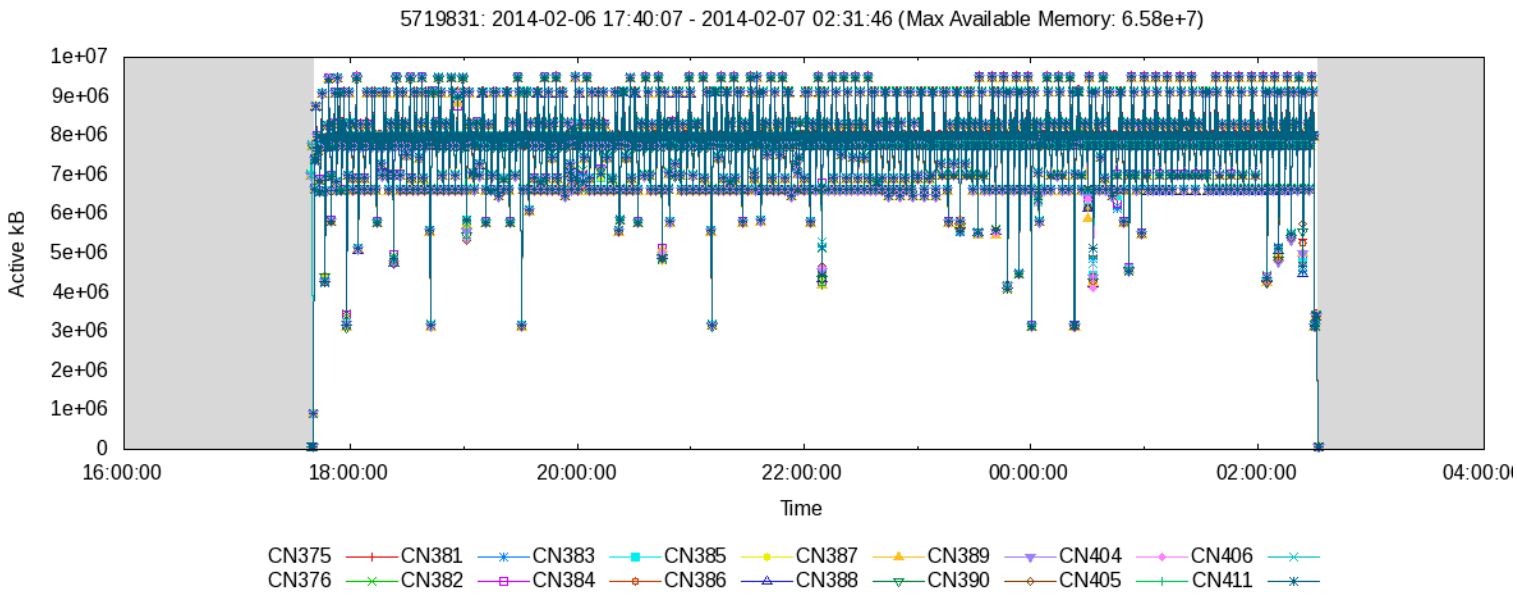
Application Profile: Unknown



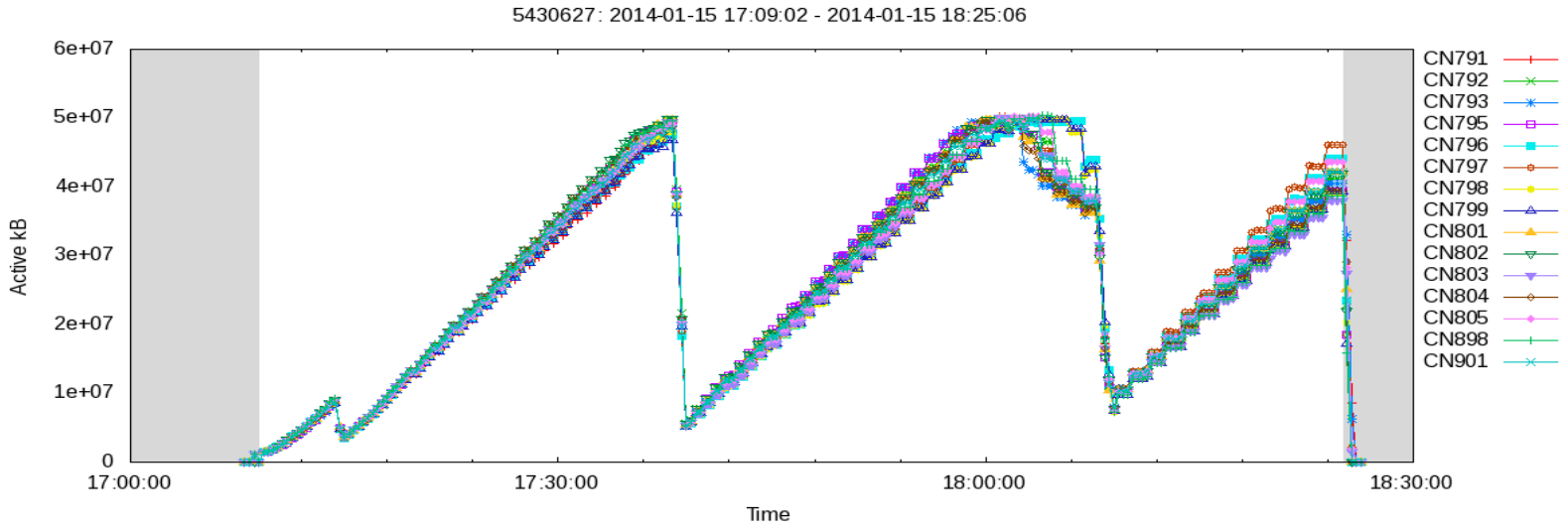
- Neither CPU nor memory bound. Investigating other quantities

Application Memory Usage Patterns

(Chama)



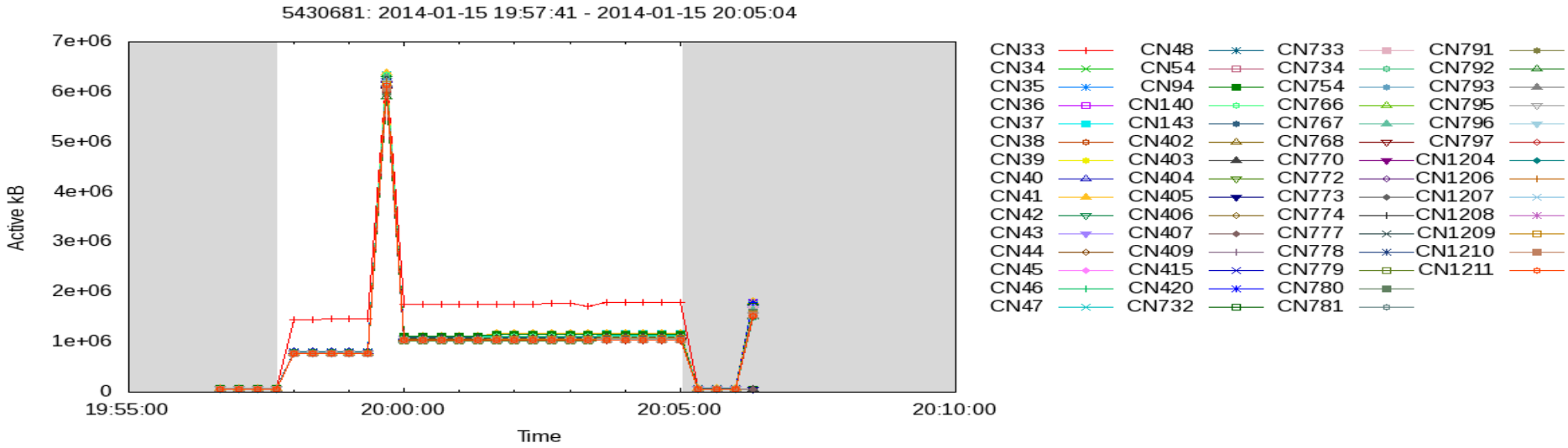
Soccoro

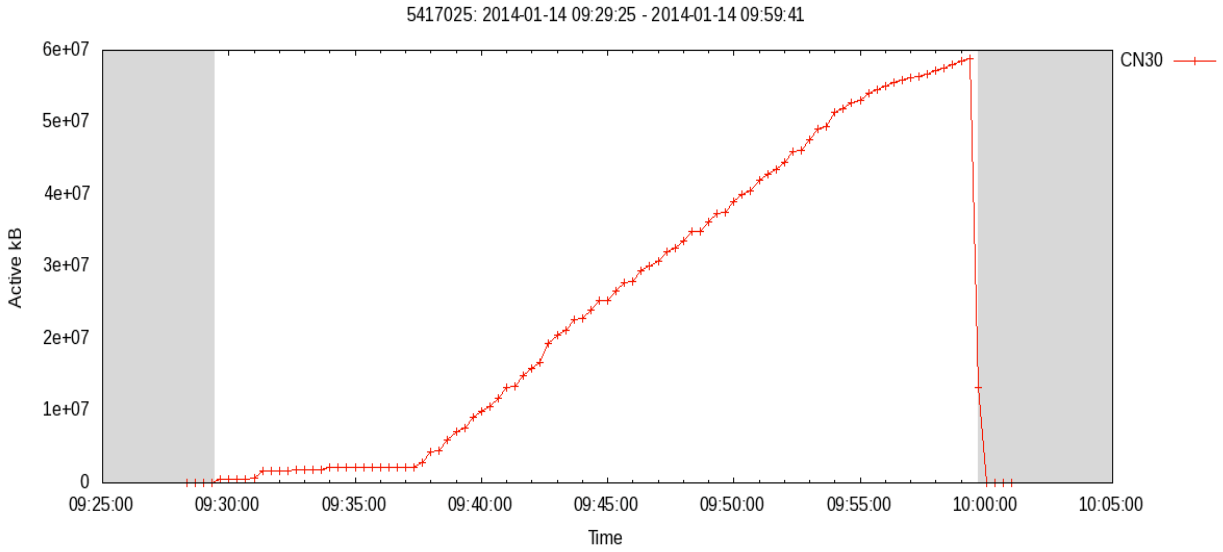


Sandia National Laboratories

Application Memory Usage Patterns

(Chama)



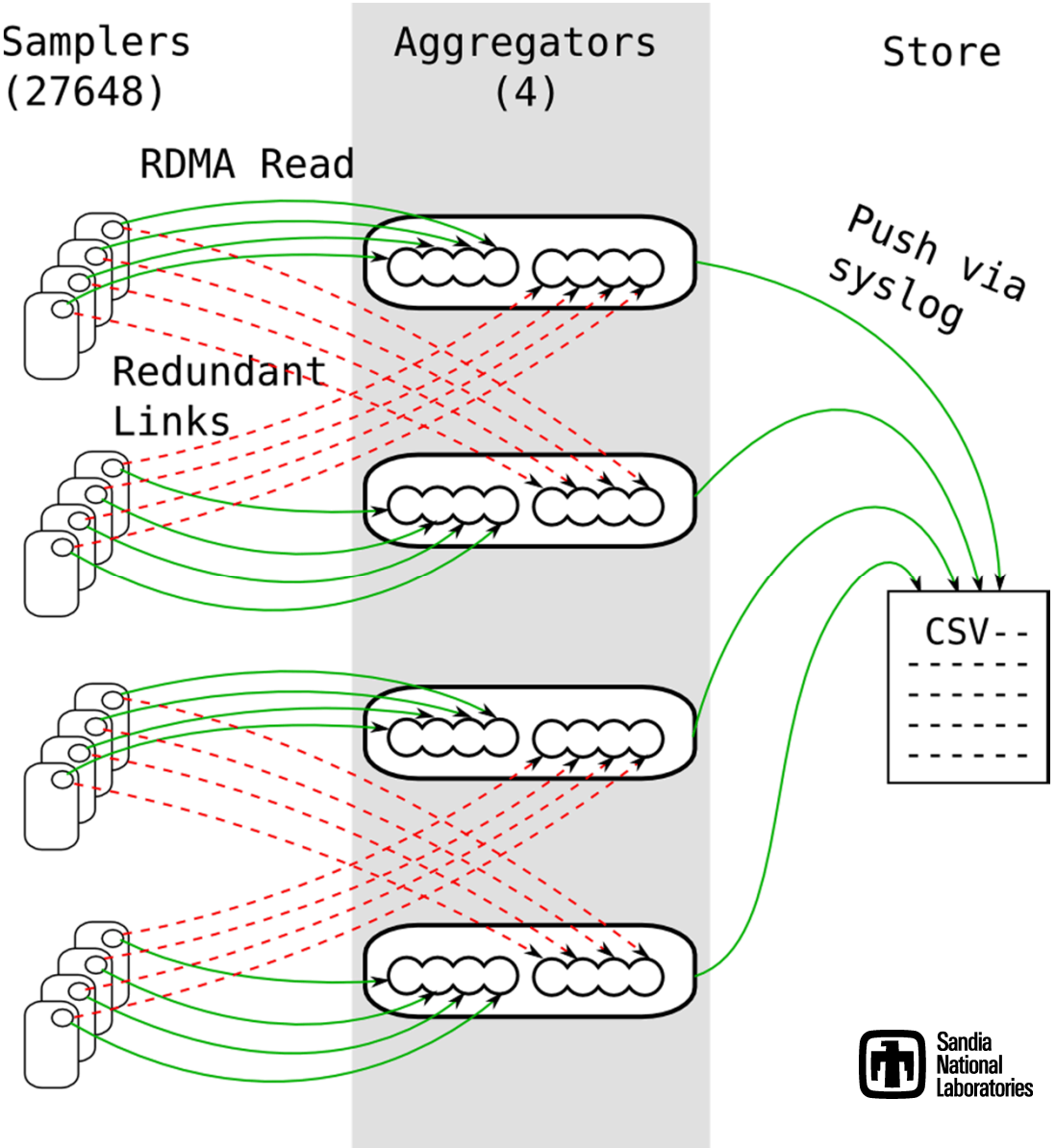


- Memory usage results in OOM

Sandia National Laboratories

NCSA's Blue Waters: 27648 node Cray XE6/XK7

- All metric sets identical independent of node
  - 194 metrics
- Sample period
  - 60 seconds (normal)
  - 1 second (high)
- Each aggregator primary for 6912 nodes
  - Pull model using RDMA read
- Each aggregator secondary for 6912 nodes
  - RDMA connection established
- In event of failover aggregator collects from 13824 nodes
- Data is pushed to store (MySQL database) using syslog-ng
- One day data set for 60 second collection period contains ~35 million data points per metric and 6.8 billion data points overall



Sandia National Laboratories

Cray System Sampler Metric Set

- Gemini Mesh coordinates
- HSN network utilization and status:
  - Directional Gemini network counters:
    - +/- XYZ traffic, packets, inq stall, credit stall, send link status, recv link status
  - Derived directional Gemini network status:
    - link BW, Used Link BW,aved packet size, % stalls
  - NIC counters
- OS traffic
  - Enables attribution of user vs OS traffic
- Lustre stats
- Scheduling pages to disk
  - nr_dirty nr_writeback
- Memory util
- Load averages

LDMS metric set Example (data)

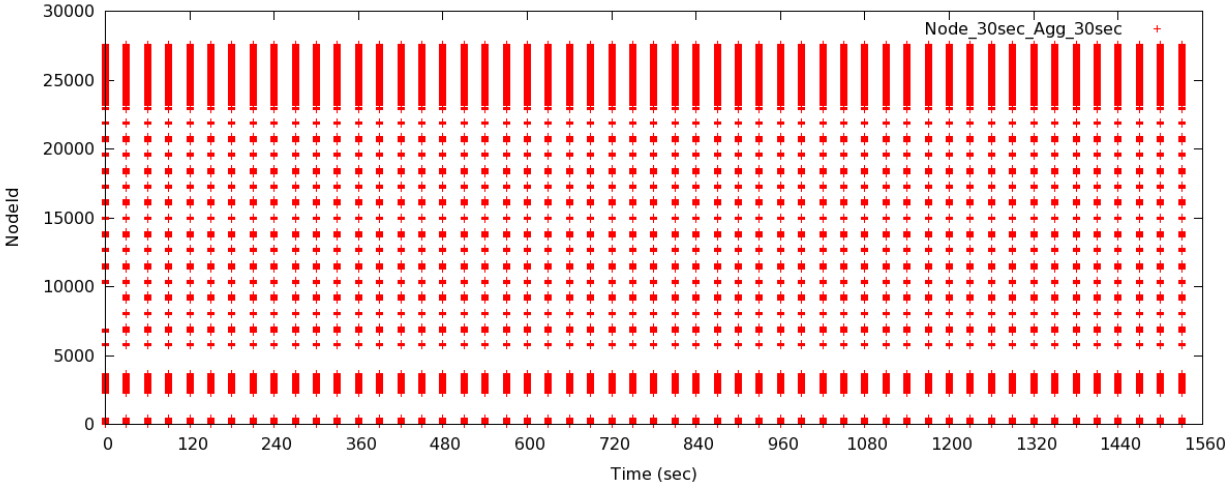
```
# ldms_ls -h nid00044 -x ugni -p 412 -l
nid00044/cray_system_sampler_r: consistent, last update: Wed Apr 09 08:52:40 2014 [726us]
 U64 1              nettopo_mesh_coord_X
 U64 1              nettopo_mesh_coord_Y
 U64 6              nettopo_mesh_coord_Z
 U64 3265901109447   X-_traffic (B)
 U64 21509840670687  Y-_traffic (B)
 U64 53884897461291  Z+_traffic (B)
 U64 89887627257     X-_packets (1)
 U64 475674895649    Y-_packets (1)
 U64 1333216704813   Z+_packets (1)
 U64 40775903446     X-_inq_stall (ns)
 U64 711117651410    Y-_inq_stall (ns)
 U64 544039347642    Z+_inq_stall (ns)
 U64 48             X-_sendlinkstatus (1)
 U64 24             Y-_sendlinkstatus (1)
 U64 24             Z+_sendlinkstatus (1)
 U64 191            X-_SAMPLE_GEMINI_LINK_BW (B/s)
 U64 306            Y-_SAMPLE_GEMINI_LINK_BW (B/s)
 U64 344            Z+_SAMPLE_GEMINI_LINK_BW (B/s)
 U64 1              X-_SAMPLE_GEMINI_LINK_USED_BW (% x10e6)
 U64 2              Y-_SAMPLE_GEMINI_LINK_USED_BW (% x10e6)
 U64 2              Z+_SAMPLE_GEMINI_LINK_USED_BW (% x10e6)
 U64 19             X-_SAMPLE_GEMINI_LINK_PACKETSIZE_AVE (B)
 U64 19             Y-_SAMPLE_GEMINI_LINK_PACKETSIZE_AVE (B)
 U64 19             Z+_SAMPLE_GEMINI_LINK_PACKETSIZE_AVE (B)
 U64 0              X-_SAMPLE_GEMINI_LINK_INQ_STALL (% x10e6)
 U64 0              Y-_SAMPLE_GEMINI_LINK_INQ_STALL (% x10e6)
 U64 0              Z+_SAMPLE_GEMINI_LINK_INQ_STALL (% x10e6)
 U64 13071017859520  totaloutput_optA
 U64 1551040415605   read_bytes#stats.snx11024
 U64 111681033094    write_bytes#stats.snx11024
 U64 33185713        open#stats.snx11024
 U64 33459578        close#stats.snx11024
 U64 200             loadavg_latest(x100)
 U64 203             loadavg_5min(x100)
 U64 2               loadavg_running_processes
 U64 217             loadavg_total_processes
 U64 32069868        current_freemem
 U64 180128670       SMSG_ntx
 U64 84138092941     SMSG_tx_bytes
 U64 179201767       SMSG_nrx
 U64 62591572089     SMSG_rx_bytes
 U64 2463841         RDMA_ntx
 U64 166910425701    RDMA_tx_bytes
 U64 5995457         RDMA_nrx
 U64 265128956892    RDMA_rx_bytes
 U64 207633071910    ipogif0_rx_bytes
 U64 116299863623    ipogif0_tx_bytes
```

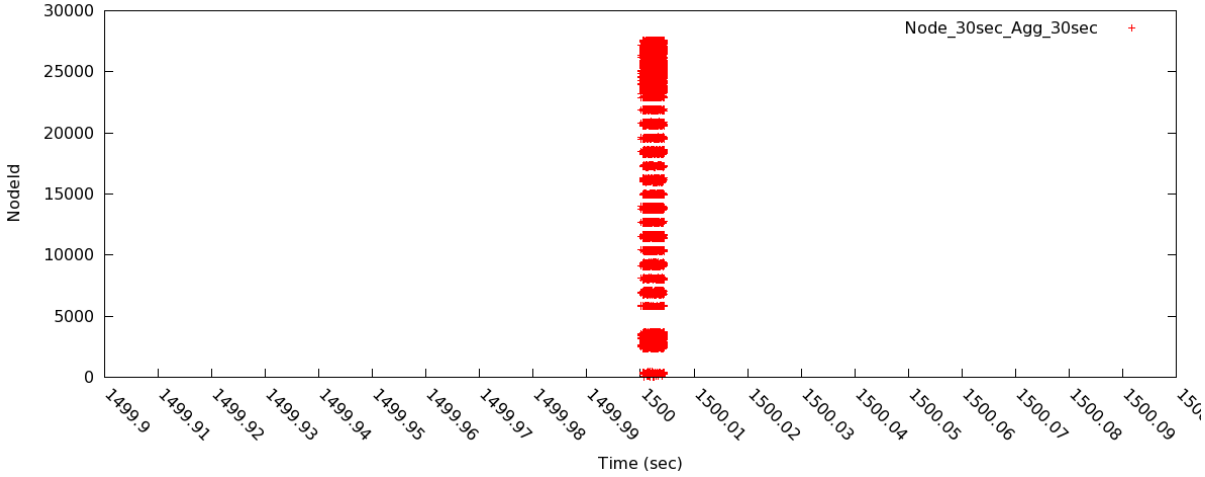
Sandia National Laboratories

Blue Waters Related Enhancements

- Synchronization
- Minimize Image Footprint
- Node type independent metric set
- Single Metric Set
  - Single Time Attribution
- Storage
  - CSV
  - Split sec and fraction with comma

Synchronous Collection





Synchronized collection across all nodes:

- Enables a coherent system snapshot

Asynchronous option spreads network load

Synchronous:

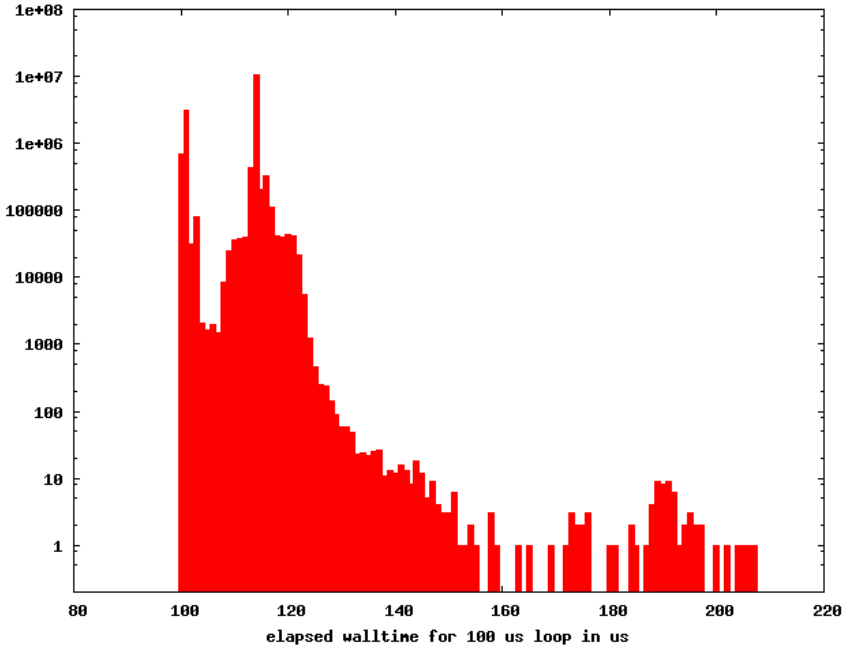
- Variance in collection timestamps ~ 4ms

Note: Clock skew not accounted for

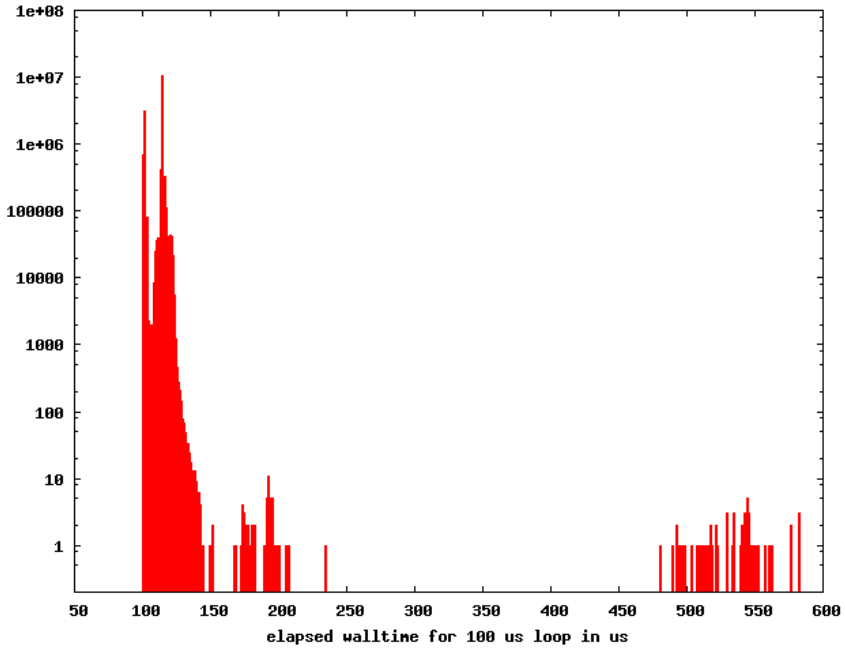
Collection occurrences over 10000 nodes on Blue Waters

Sandia National Laboratories

Low-Impact: System Noise



PSNAP No LDMSD 32 cores 5.5 sec total run. Target loop time is 100 usec. Offset from 100 usec is due to system noise.

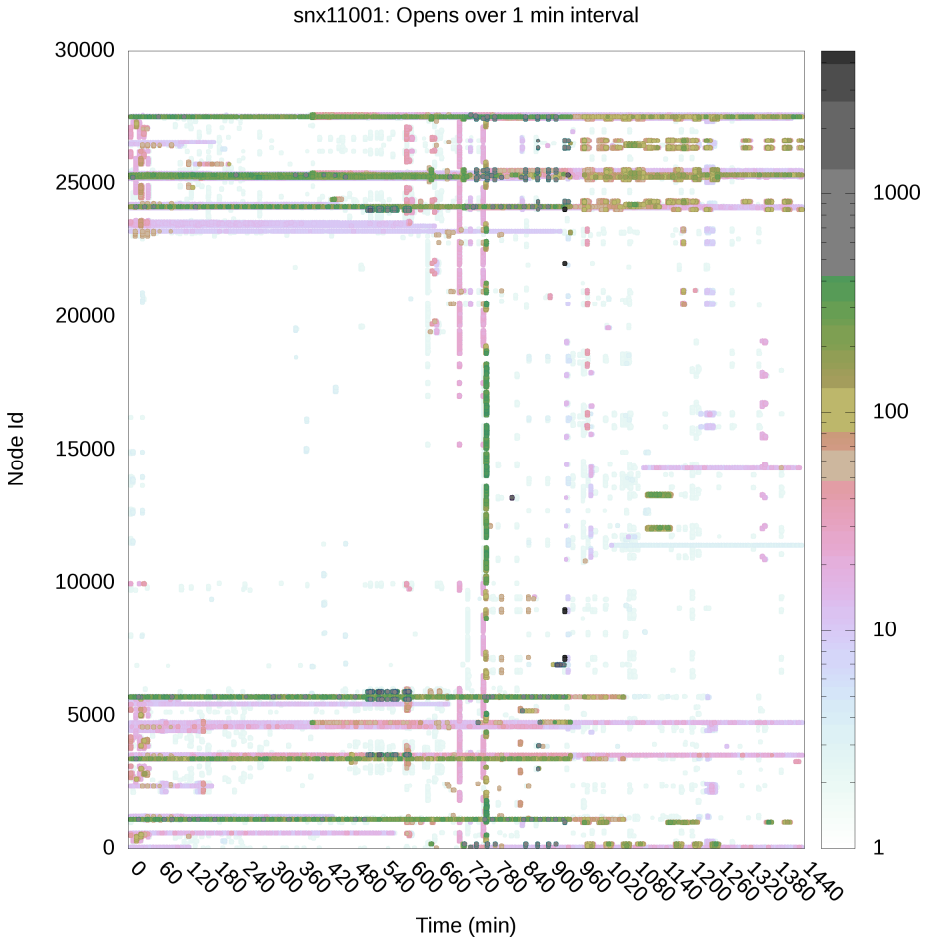


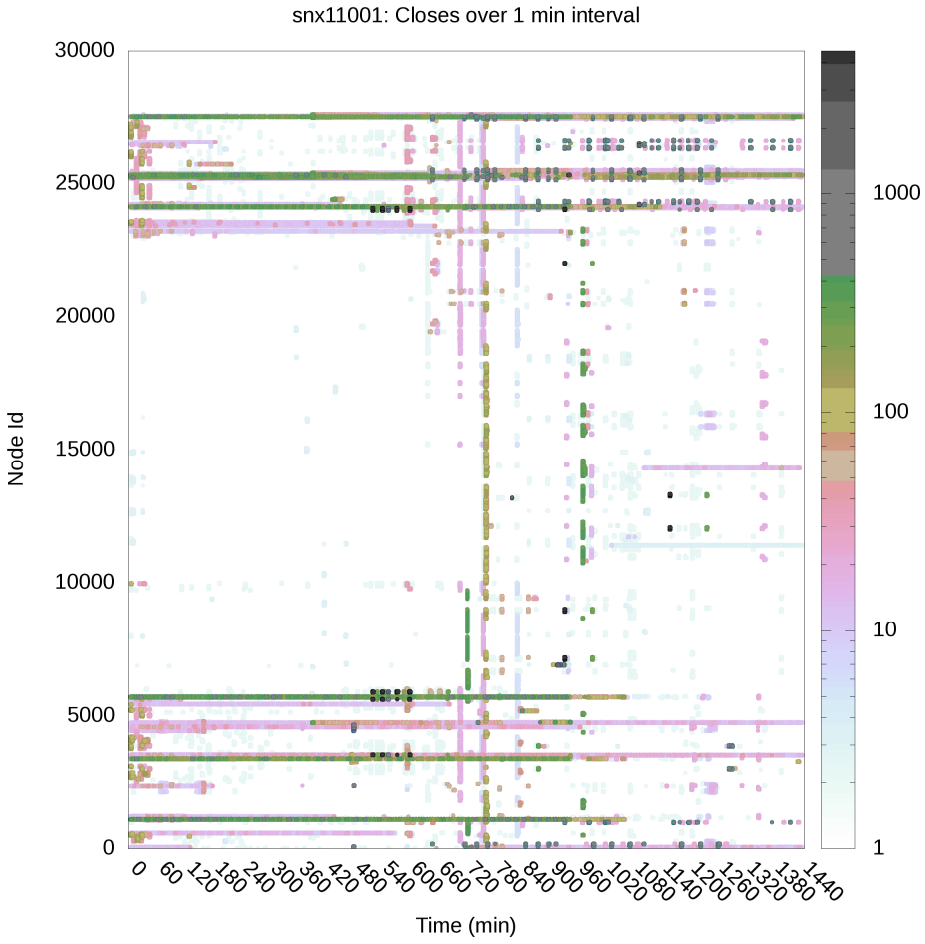
PSNAP w/ LDMSD (1s) 32 cores 5.5 sec. Same profile except for the occasions when LDMS runs -- once per sec. Adds ~ 400us to those times. 60 (6 instances per node) points of the entire run.

No statistically significant difference in system noise w/LDMS (PSNAP)
No statistically significant impact on MILC, IMB AllReduce, minighost

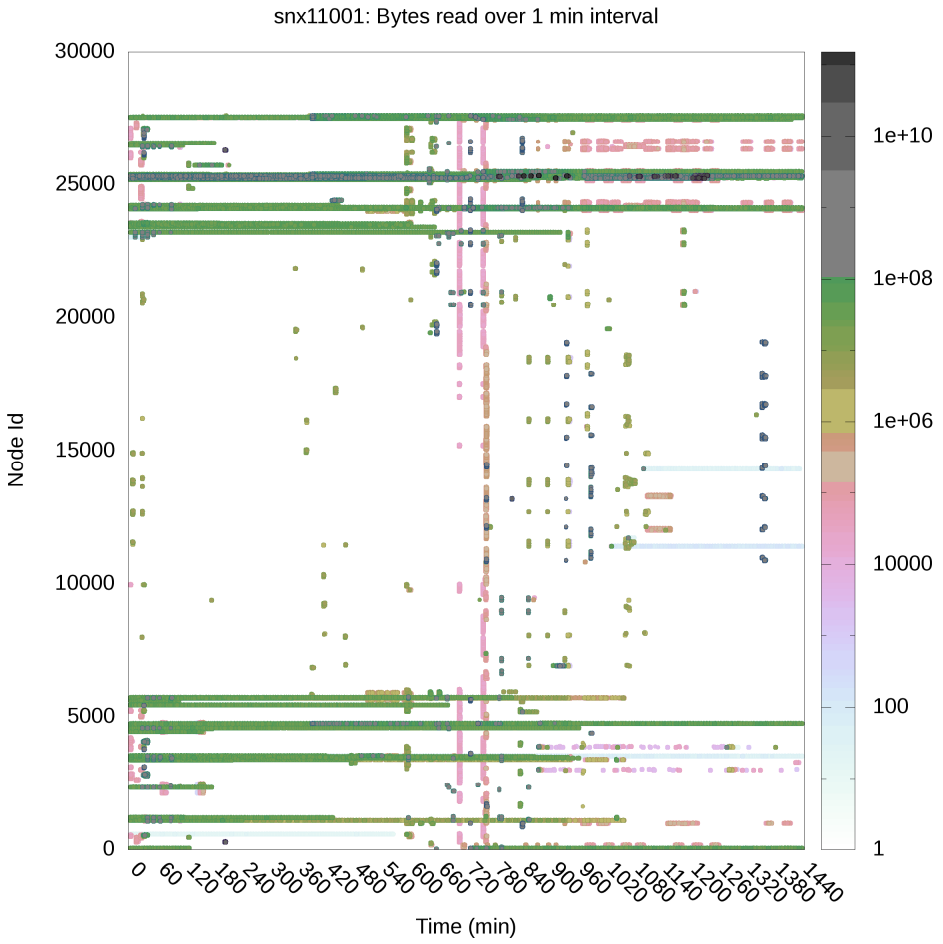
Sandia National Laboratories

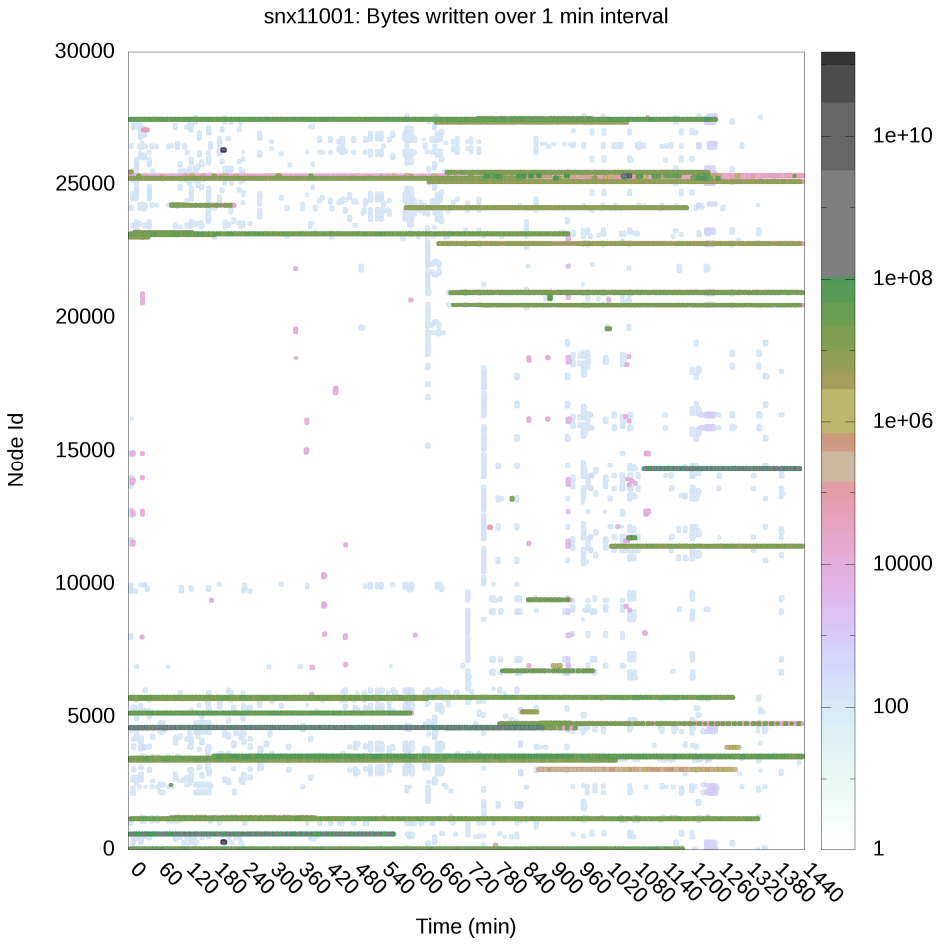
Lustre Opens/Closes





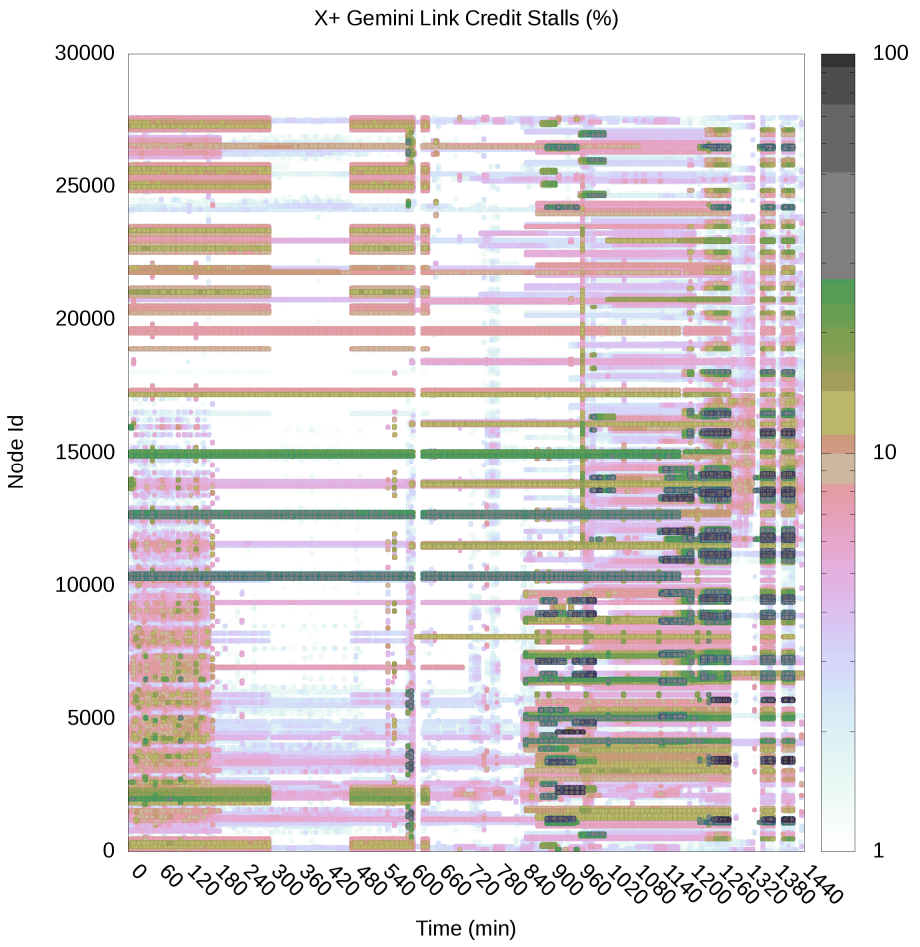
Lustre Reads/Writes

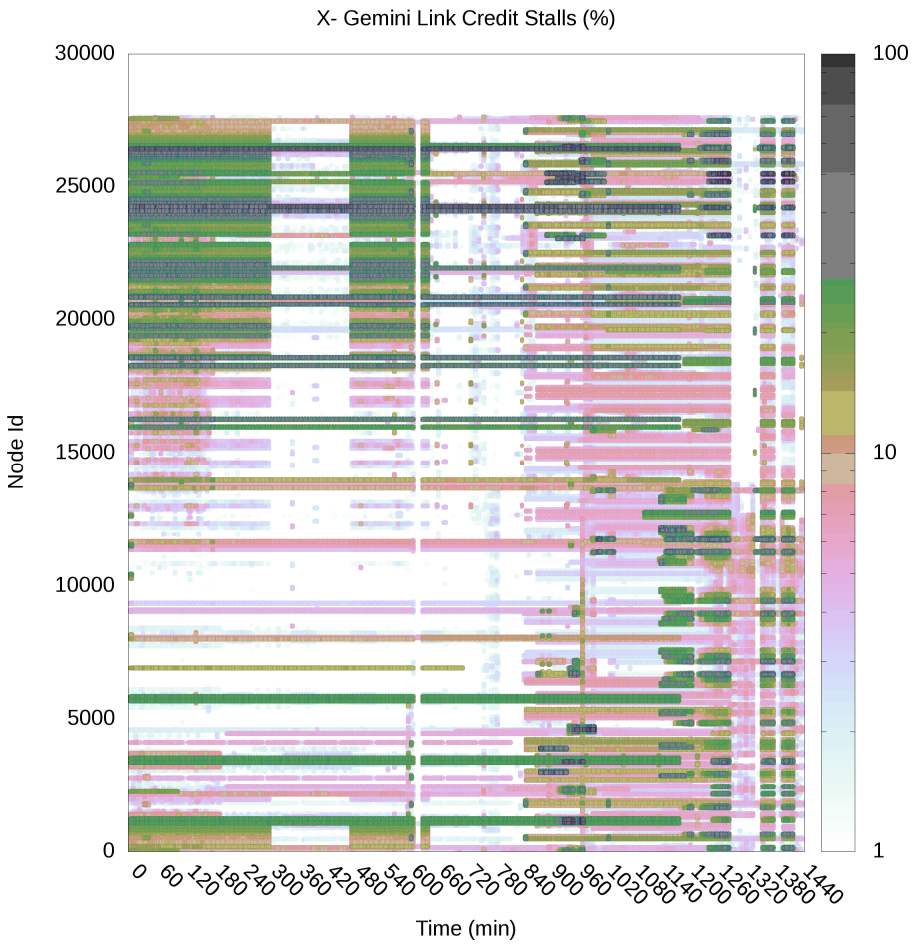




Sandia National Laboratories

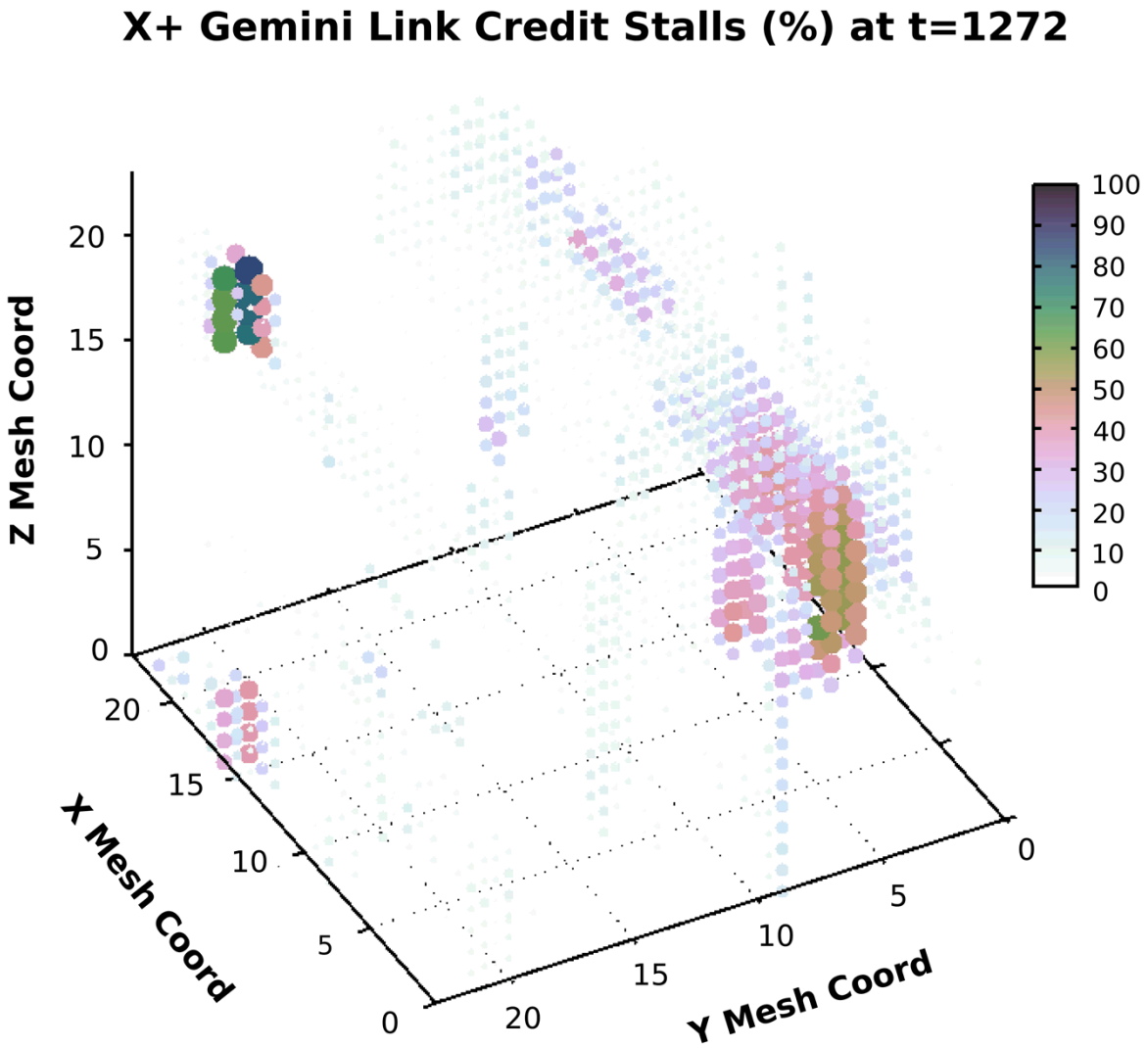
HSN Output Stalls (X)





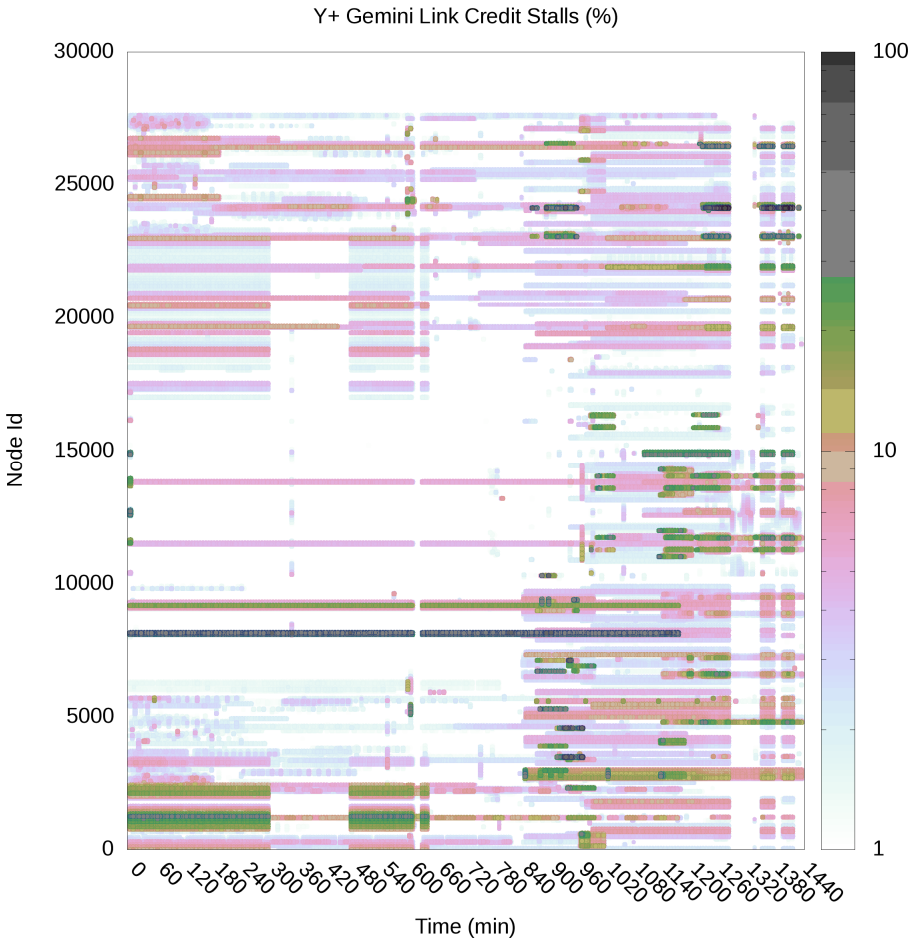
Sandia National Laboratories

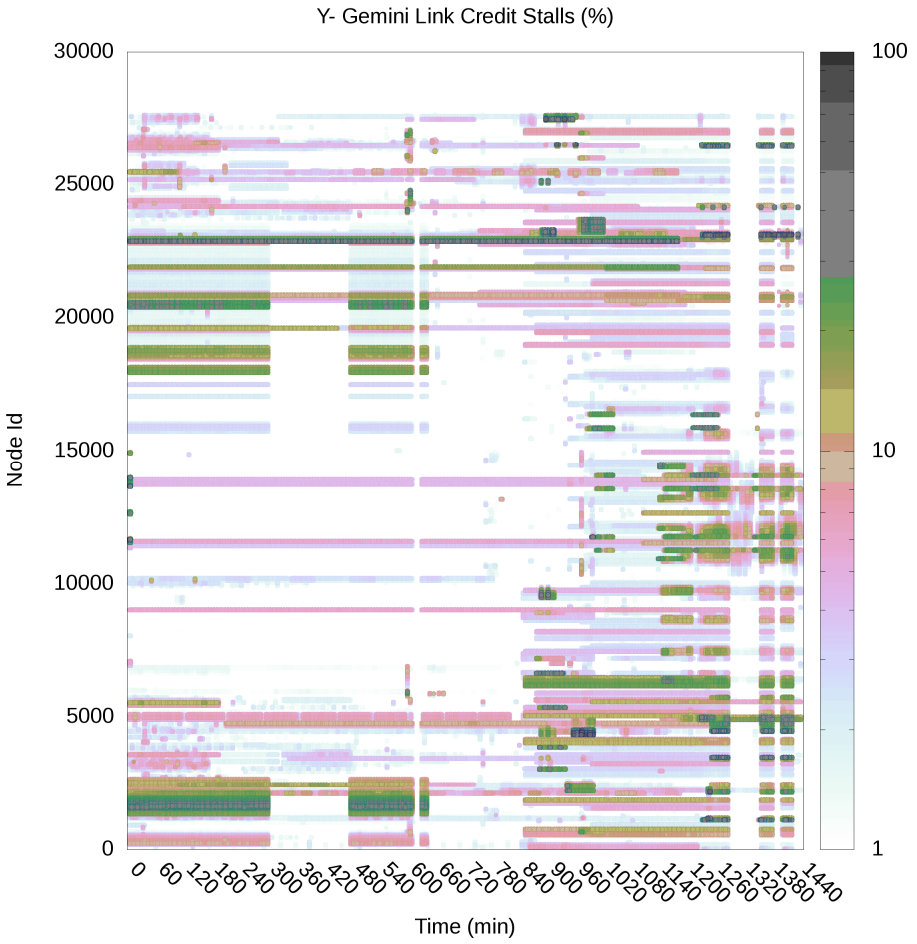
HSN Output Stalls (X) Time Slice



Sandia National Laboratories

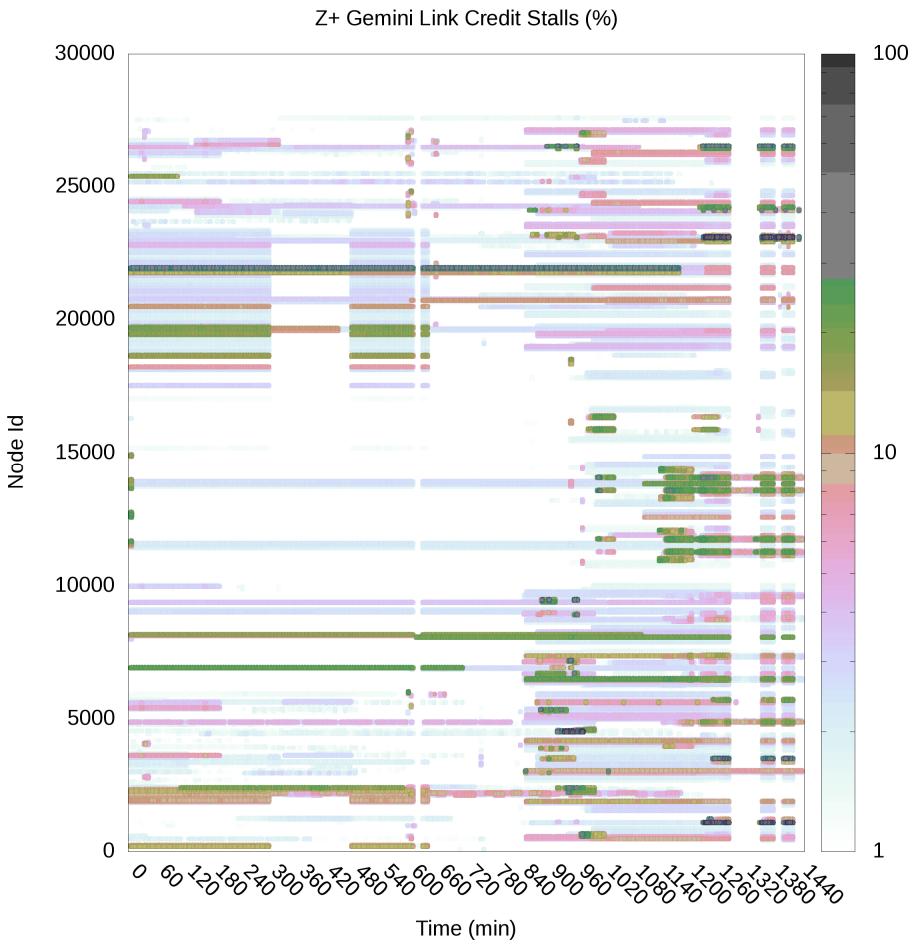
HSN Output Stalls (Y)

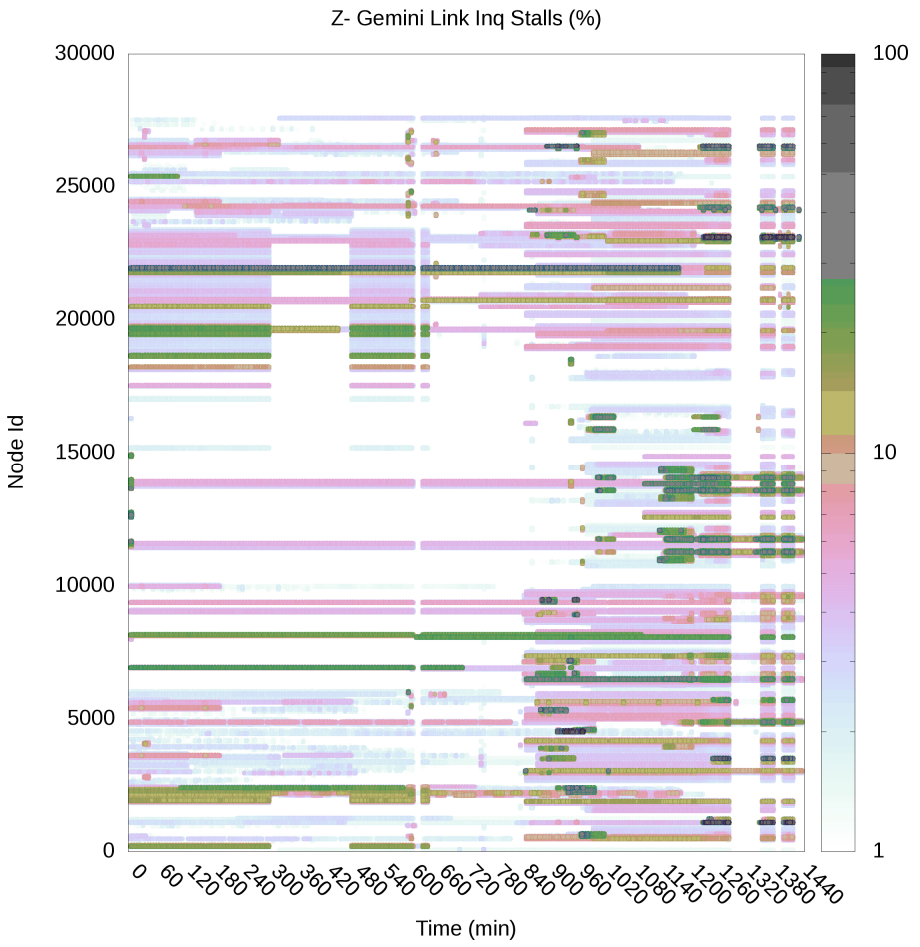




Sandia National Laboratories

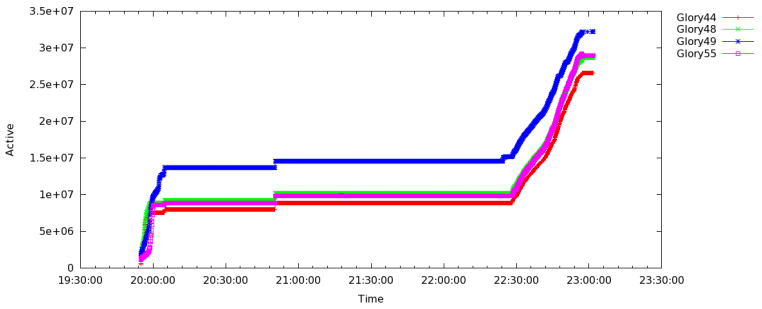
HSN Output Stalls (Z)





Sandia National Laboratories

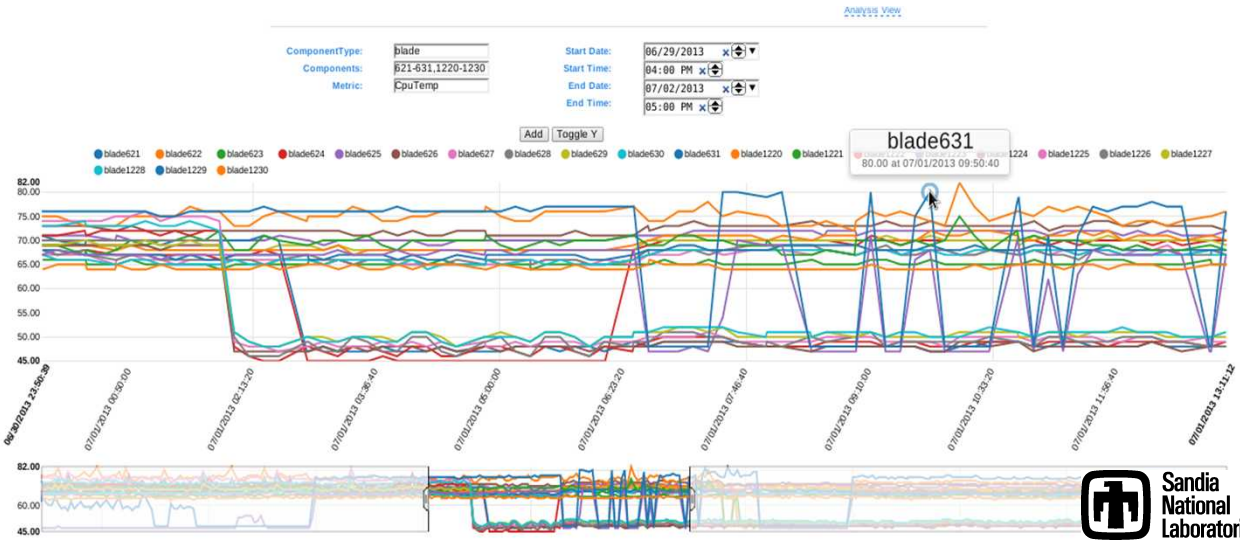
Current Analysis and Post Processing




```
================================================================================
COMP_TYPE: node METRIC_NAME: Active
================================================================================
 CompId   NPts     Ave      Std          Min              MinTime              Max              MaxTime
------- ------- -------- --------- ---------- ------------------- ---------- -------------------
     45    2242 1.023e+07 4.537e+06     525824 12/14/12 19:54:52 2.65823e+07 12/14/12 23:01:15
     49    2242 1.161e+07 4.805e+06 1.61549e+06 12/14/12 19:54:52 2.87072e+07 12/14/12 23:01:20
     50    2195 1.545e+07 4.362e+06  1.7361e+06 12/14/12 19:54:52 3.22476e+07 12/14/12 23:01:35
     56    2242 1.124e+07 5.072e+06 1.19422e+06 12/14/12 19:54:52 2.91891e+07 12/14/12 22:56:55
------- ------- -------- --------- ---------- ------------------- ---------- -------------------
------- ------- -------- --------- ---------- ------------------- ---------- -------------------
 Group    NPts     Ave                  Min          MinCompId              Max          MaxCompId
------- ------- -------- --------- ---------- ------------------- ---------- -------------------
         8921 1.211e+07                525824                 45 3.22476e+07                 50
```

Post-Job Stats and Plots

Interactive Web Interface: System Layout and Time Series



Sandia National Laboratories

Conclusions

- The OVIS data collection, transport, and storage framework (LDMS) provides scalable whole system data access with no statistically significant adverse impact to applications
- Whole system snapshots of shared system resource utilization can provide valuable insights to system and application performance
- We need to develop new analysis and visualization tools to fully utilize the new wealth of data we are collecting

Future Work

- More Tools – both run-time and post processing
  - Analysis
  - Visualization
- Log collection without store for diagnostics
- More LDMS plugins to support derived data, high frequency metrics (e.g. power, memory access info.), runtime configuration, etc.

Miscellaneous

- Collaborative Interactions Welcome
- Monitoring and Analysis Workshop held in conjunction with IEEE Cluster 2014 in Madrid Spain
  - See call www.cluster2014.org
  - Submissions due May 23, 2013
    - Full papers
    - Mini-talks

Questions?