SAND2013-6521C

Photos placed in horizontal position with even amount of white space between photos and header

Photos placed in horizontal position with even amount of white space between photos and header

Lightweight Distributed Metric Service (LDMS)

Run-time Resource Utilization Monitoring

Sandia National Laboratories,
Scientific Computing Systems, Albuquerque, NM
And
Open Grid Computing, Austin, TX

Sandia National Laboratories

Exceptional service in the national interest

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Overview

Sandia National Laboratories

- Motivation
- High-level Architecture
- Performance
- Usage
- Advanced Deployment Features
- Summary

Motivation

Sandia National Laboratories

Gain insight into resource utilization/bottlenecks (e.g., network bandwidth/hotspots, CPU utilization, Memory footprint/ bandwidth)

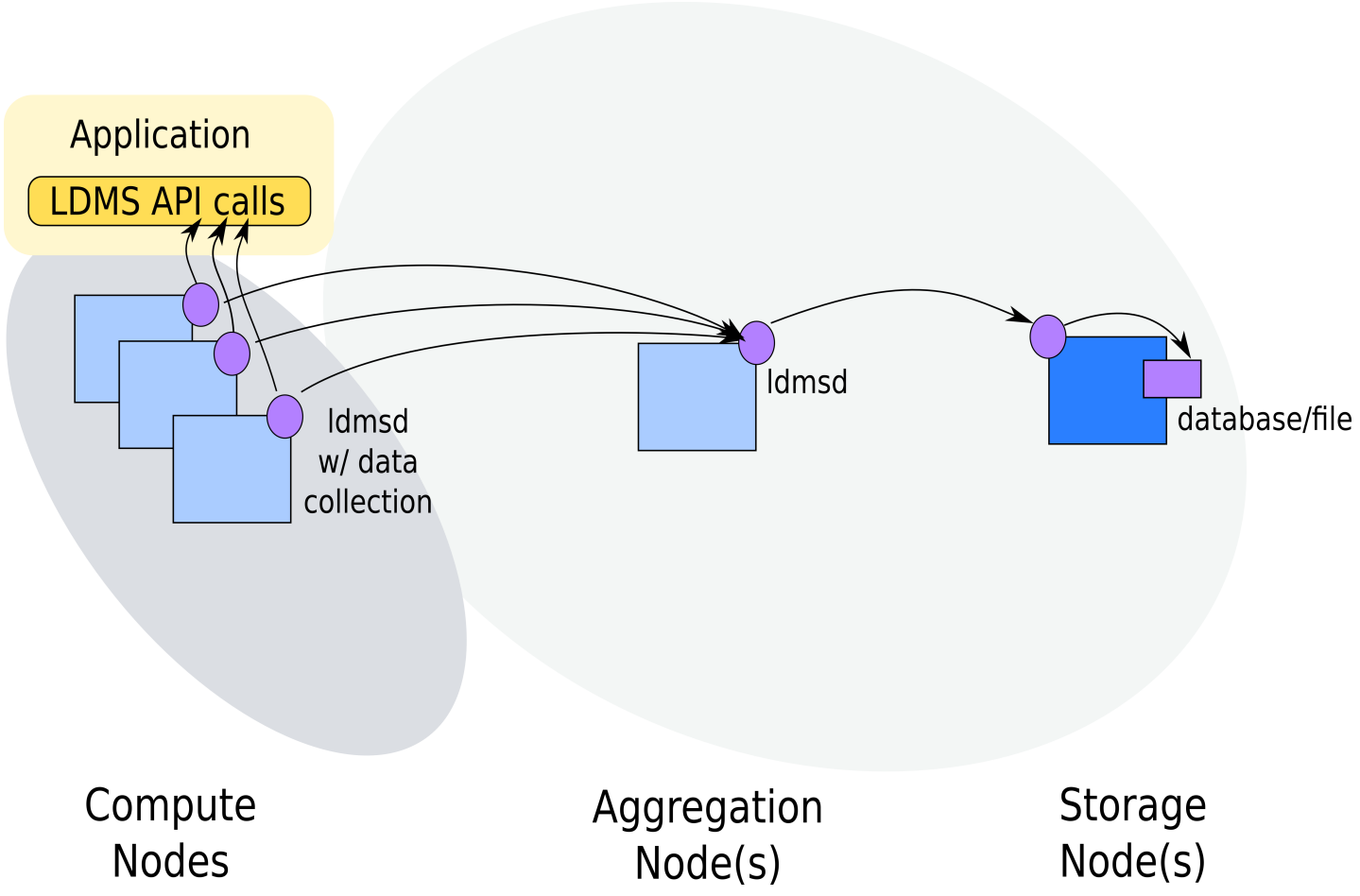
- Intelligent job placement
- Run-time workload partitioning/adaptation
- Historical comparison
- Anomaly detection

Monitoring System and Application Resource Utilization

Sandia National Laboratories

- Typical monitoring systems target failure detection, uptime, trend overview:
  - Information targeted to system administration
  - Collection intervals of minutes
  - Relatively high overhead (both compute node and aggregators)
- Application profiling/debugging/tracing tools:
  - Collection intervals of sub seconds (even sub-millisecond)
  - Typically requires linking, not run under real-world conditions (i.e., tools perturb the application profile)
  - Limits on scale
  - Don't account for external applications competing for the same resource
- Lightweight Distributed Metric Service (LDMS):
  - Continuous data collection, transport, storage as a system service
  - Targets system administrators, users, and applications
  - Enables collection of a reasonably large number of metrics with collection periods that enable job-centric resource utilization analysis and run-time anomaly detection
  - Variable collection period (~seconds)
  - On-node interface to run-time data
  - Multiple concurrent data clients and installations

LDMS High Level Overview



* Only the current data is retained on-node

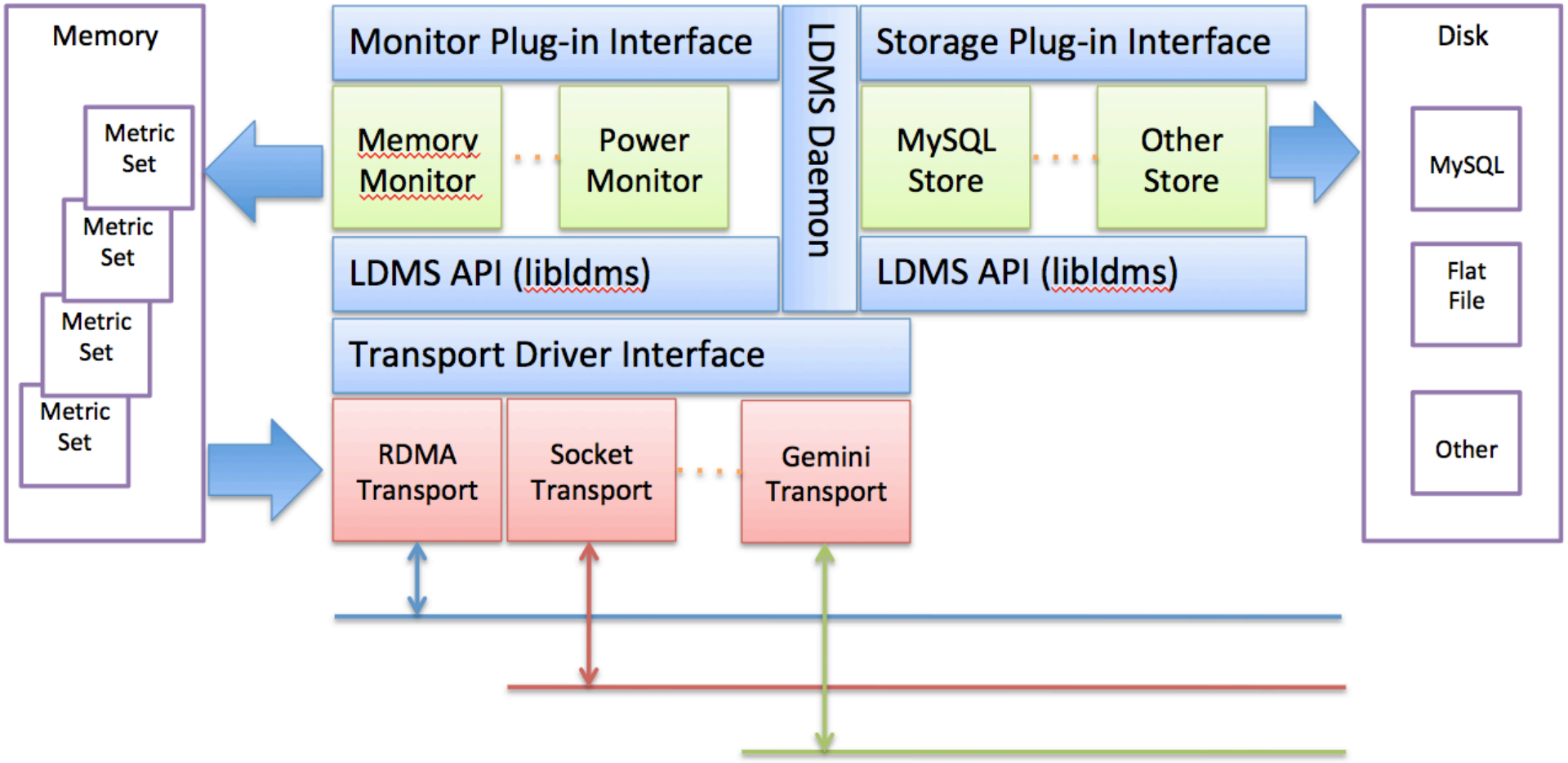
LDMS Functional Overview

Sandia National Laboratories

- Data is bundled into “Metric Sets” – “related” information
- Metric Sets have associated Data and Meta-data and include generation numbers for both
  - Meta-data is only transmitted during initial setup and when change occurs
- Run-time (Re)Configuration:
  - Run-time Collection plugin add, start, stop
    - Add new collection components
    - Start collection – begin scheduling data collection and make data visible to queries
    - Stop collection – stop scheduling data collection, last data set still visible to queries – no CPU overhead associated with this as no collection scheduled
    - Modify collection frequency on metric set basis
  - Run-time Storage plugin add, start, stop
    - Insert to new data storage containers
    - Add new metrics to the store
  - Run-time Add Hosts to Aggregator
- Data Queries can be either host local or remote
- Socket or RDMA transport options

LDMS Architecture: Modular Plugin Interface for Data Collectors, Transport, and Storage

Sandia National Laboratories



LDMS Metric Sets Example

Sandia National Laboratories

shuttle-cray.ran.sandia.gov_1/meminfo

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

shuttle-cray.ran.sandia.gov_1/procstatutil

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

shuttle-cray.ran.sandia.gov_1/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

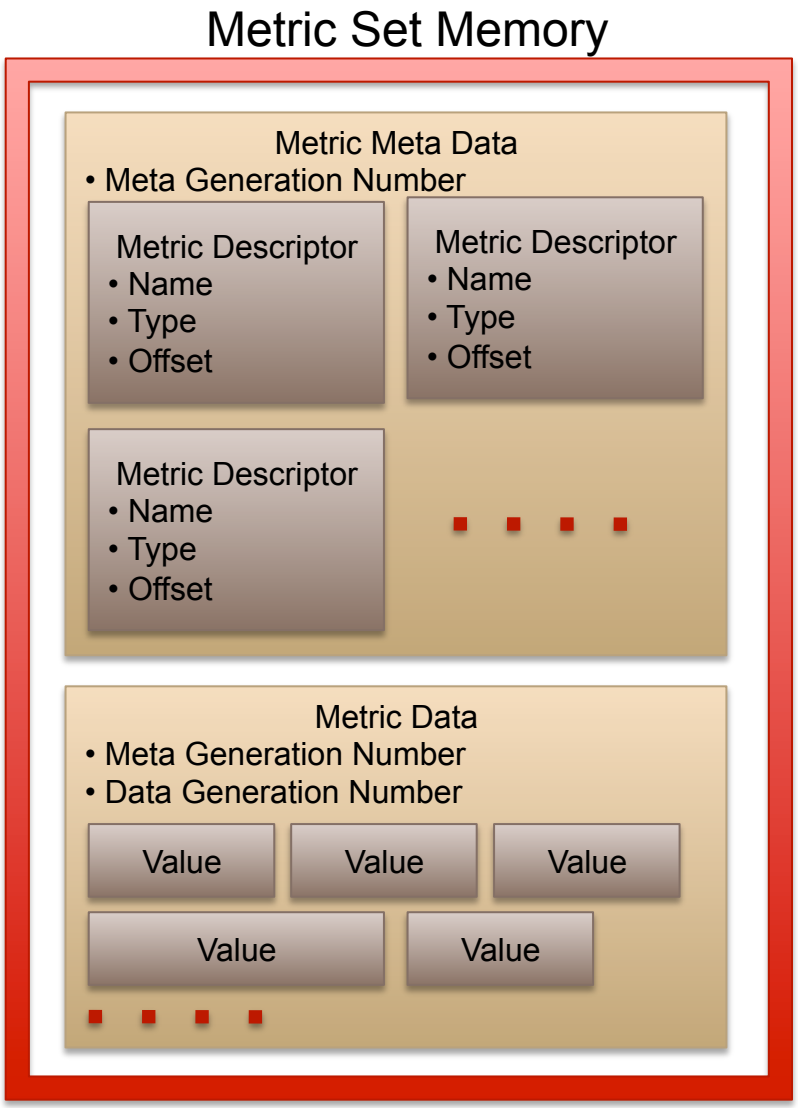
API:

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

- Same API for on-node and off-node (aggregator) transport

Metric Set Format

- Meta data generation number bumped whenever metrics are added or removed
- Data generation number changes whenever a value changes
- Meta data generation number is included with metric data to detect when cached local meta-data is stale



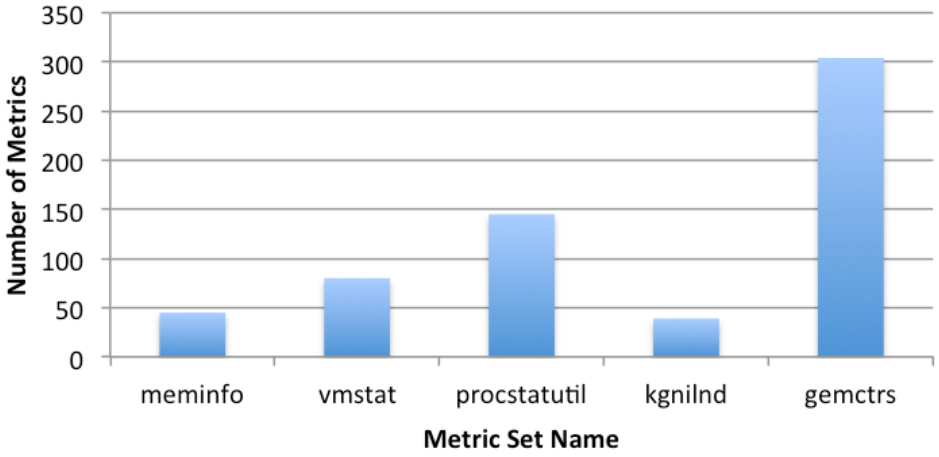
Current Data Collector/Storage Plugins

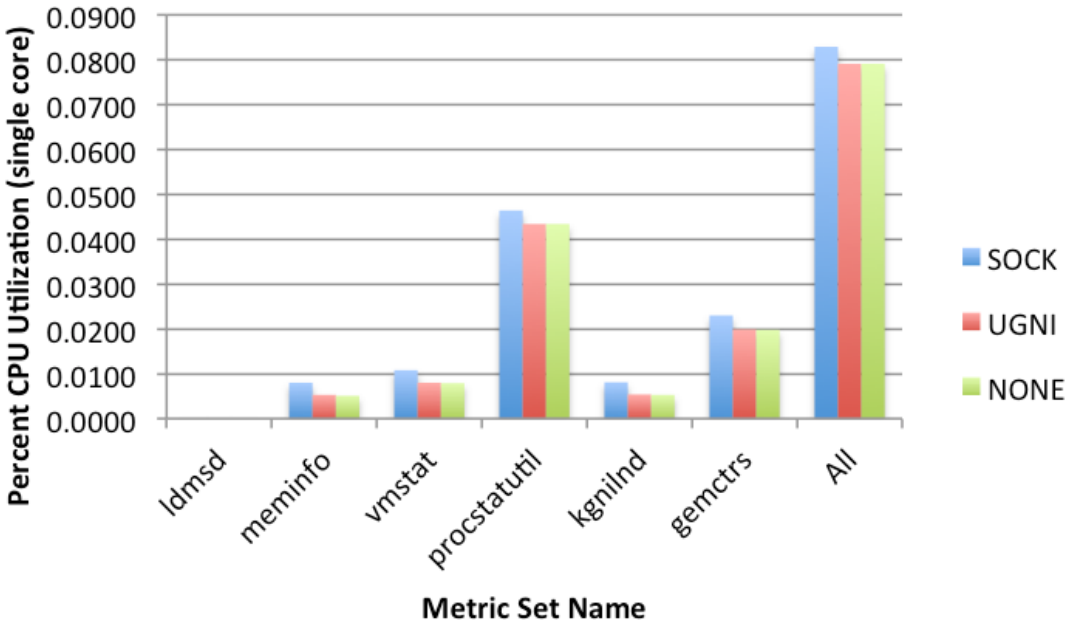
- /proc
  - meminfo, vmstat, stat, interrupts, pid/(stat, statm)
  - kgnilnd (Cray specific)
- Gem_link_perf (Cray specifc)
  - Gemini Tile and NIC counters w/ link aggregation
- perf_event
  - Generic interface for acquisition of hardware counters e.g., data cache misses, instruction cache misses, hyper-transport bandwidth (AMD)
- rsyslog (Cray specific)
  - SEDC (RAS) and ALPSdata
- lmsensors (/sys)
  - Temperatures, fan speeds, voltages
- Storage: Flatfile, CSV, SQL, NoSQL, Custom

Overhead

Collection Interval of 1 second on Cray XE6

- CPU overhead increases with number of metrics in a metric set for a particular gathering mechanism (e.g., /proc, /sys readers, ioctl calls)
- `gemctrs` – ioctl as opposed to reading from `/proc`. `gemctrs` with ~300 metrics has < 1/3rd the overhead of `procstatutil` but has ~twice the number of metrics



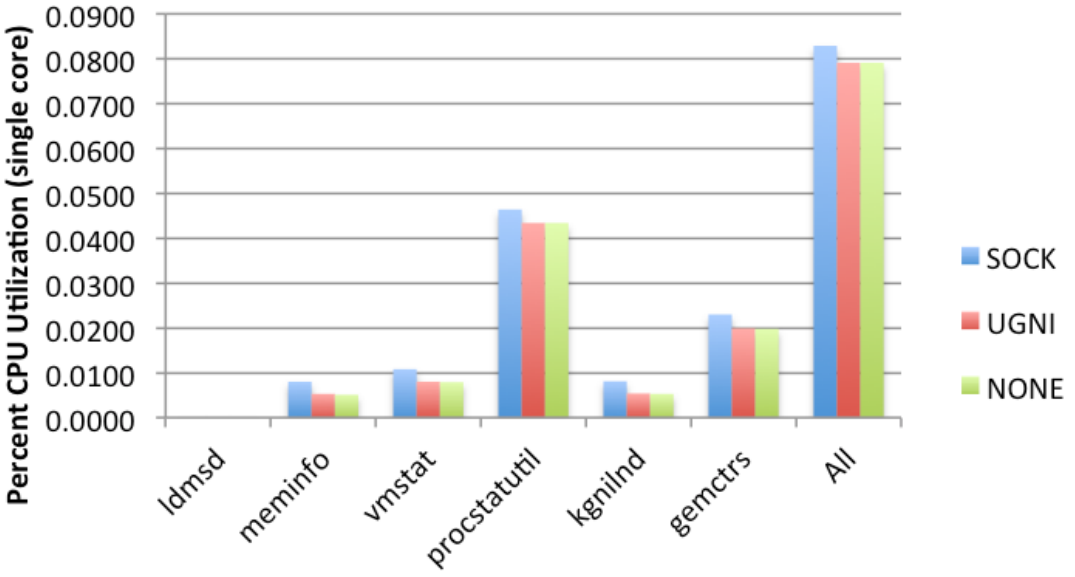


Overhead Summary

Sandia National Laboratories

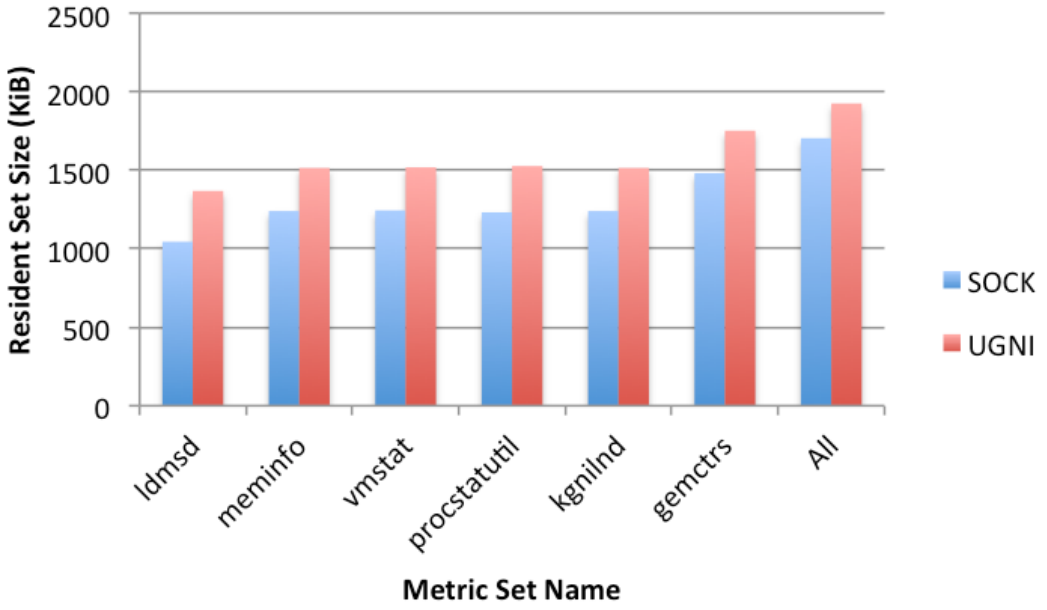
CPU Overhead

- Mostly due to data collection vs. transport
- RDMA (UGNI) has none past startup
- SOCK significant if collection overhead is small (e.g., small dataset)



Memory Footprint

- RDMA has a larger memory footprint than SOCK
- Except for `gemctrs`, sampler overhead, over *ldmsd* alone is about the same



Sandia National Laboratories

USAGE

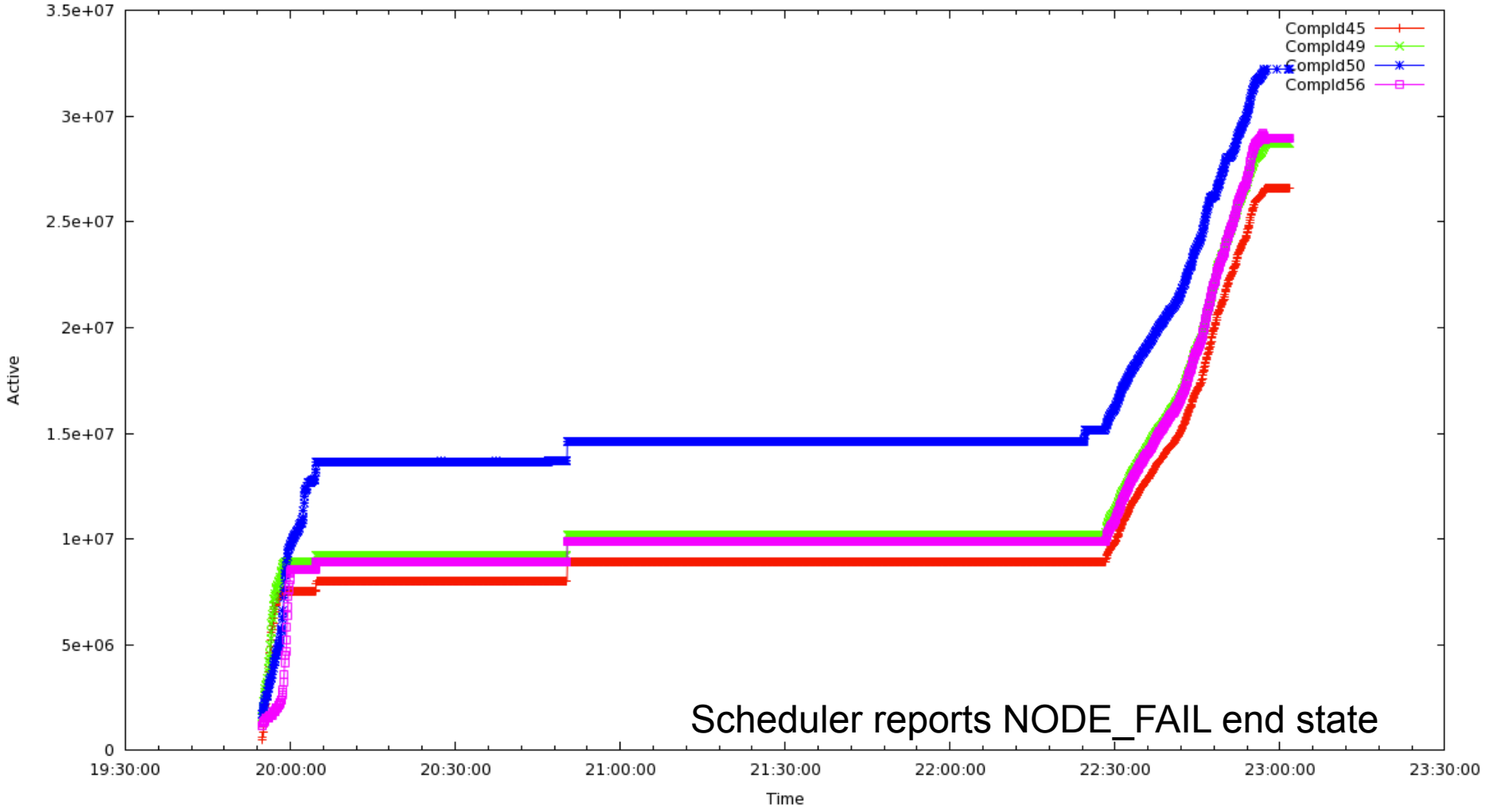
Data Collection and Analysis Tools as a System Service

Sandia National Laboratories

- LDMS data collection runs as a system service
  - Run-time adjustment of data collected, frequency
  - Requires NO changes or relinking to the application
  - Per-user LDMS instances*
- Run-time feed of resource utilization information
  - Early warning of problems
- Provide job profile data dumps on a per-user per-job basis
  - Analysis and plotting tools
  - Users can examine performance across runs
  - Post-mortem diagnosis
- On-node API for data access
  - Can be used for computational steering, dynamic allocation

Temporal Job Profile

(Per-job per-component graphical tools extract data from the database dump)



Statistical Profiling Tools

(Active Memory)

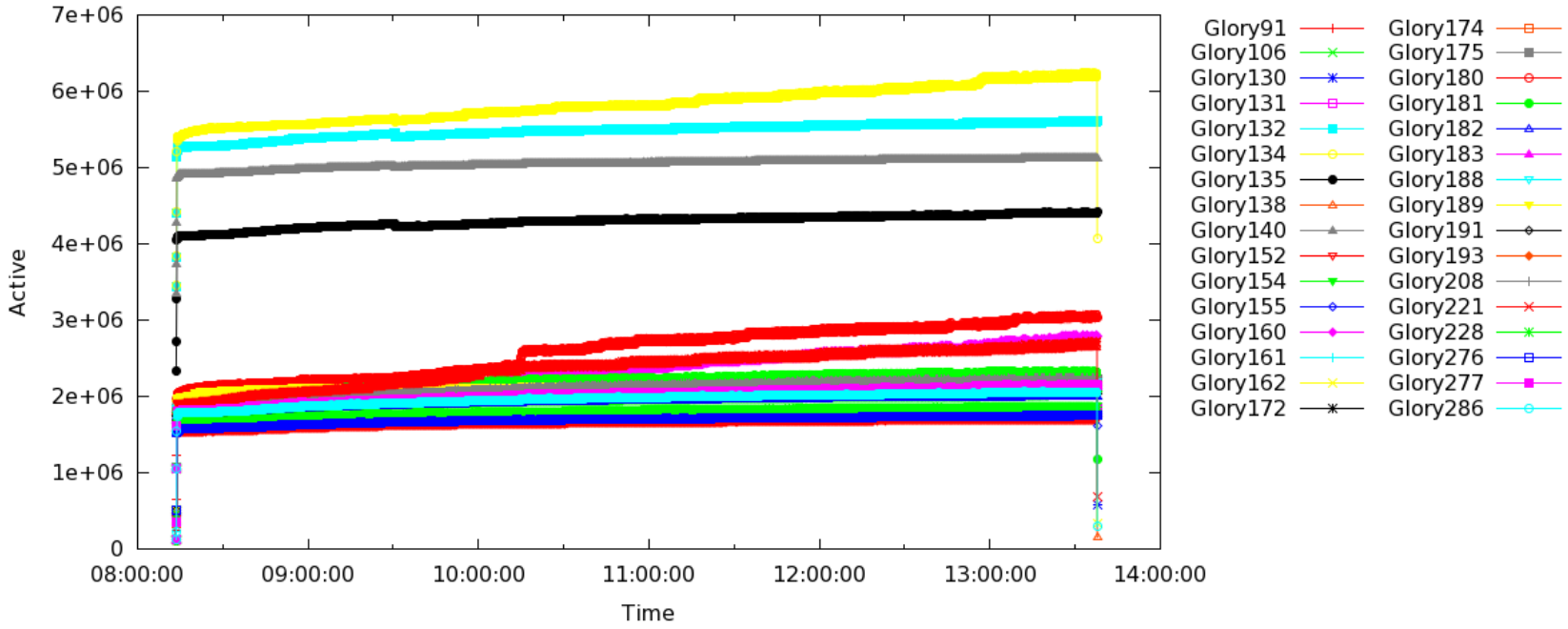
Sandia National Laboratories

```
./stats_multi -p /data1/store -j 6684780,"/var/log/glory_slurm/joblog",0 -o ./output_6684780/6684780 node:Active
```

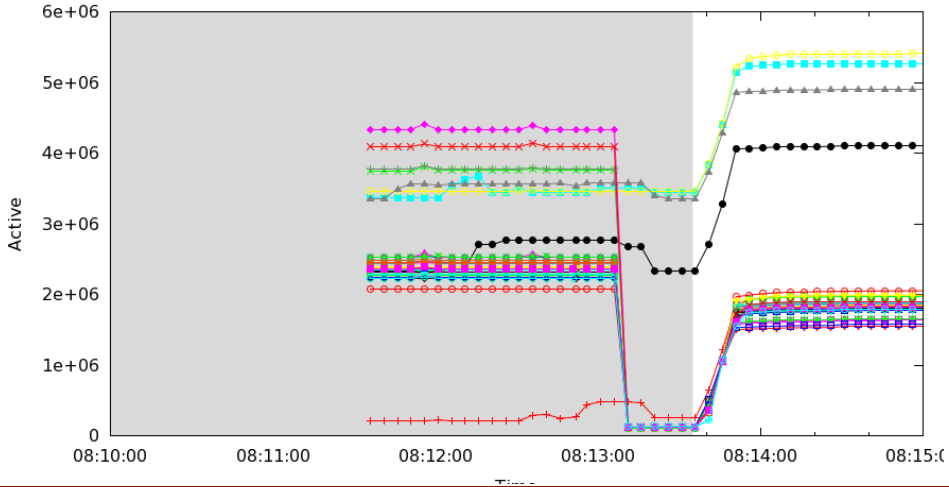
```
================================================================================
COMP_TYPE: node METRIC_NAME: Active
================================================================================
 CompId    NPts     Ave      Std            Min               MinTime                  Max                  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
```

- Descriptive statistics, high/low water mark, per node and per job basis
- One node averages %50 more active memory over job lifetime than other nodes
- High maximum active memory near job end in all nodes compared to job average, available memory

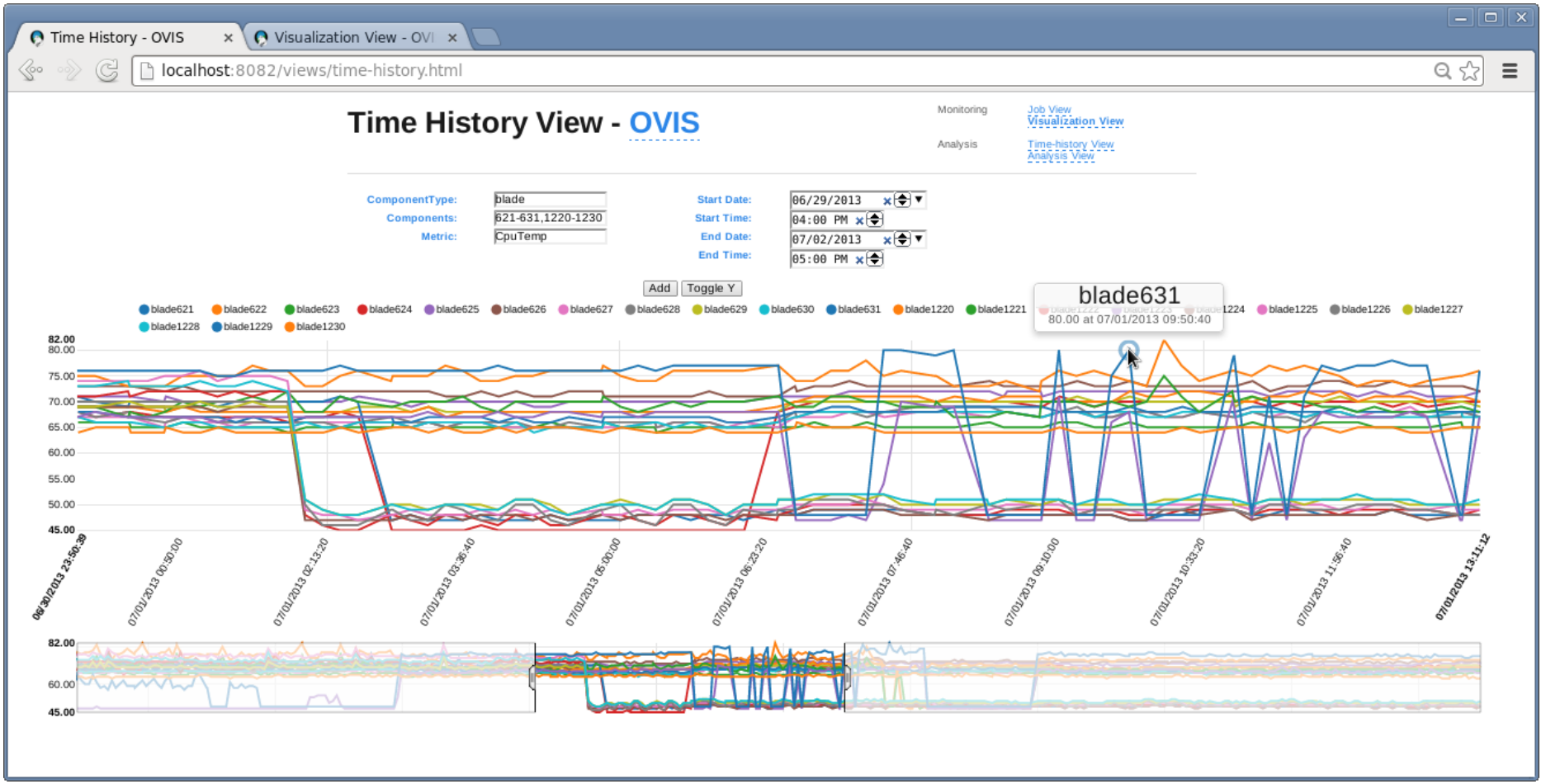
Multi-node: General Behavior



- All nodes similarly behaved but with offset
- Problem is initial node condition



Data Storage Options Enable User's Choice of Analysis Tools



Example: JavaScript Web Visualizations of MySQL data

Visualization View - OVI

localhost:8082/views/vis_view.html?type=blade&metric=CpuTemp

Visualization View - OVIS

Components: blade (1232) | rack (26) | chassis (140) | numanode (2464) | core (19712). Details

Monitoring: Job View, Visualization View

Analysis: Time-history View, Analysis View

Enter in a thing to draw (component type then components) then what to draw (metric). This view draws the historical value, at one point in time, of a metric for every component of the given type in a rectangular grid.

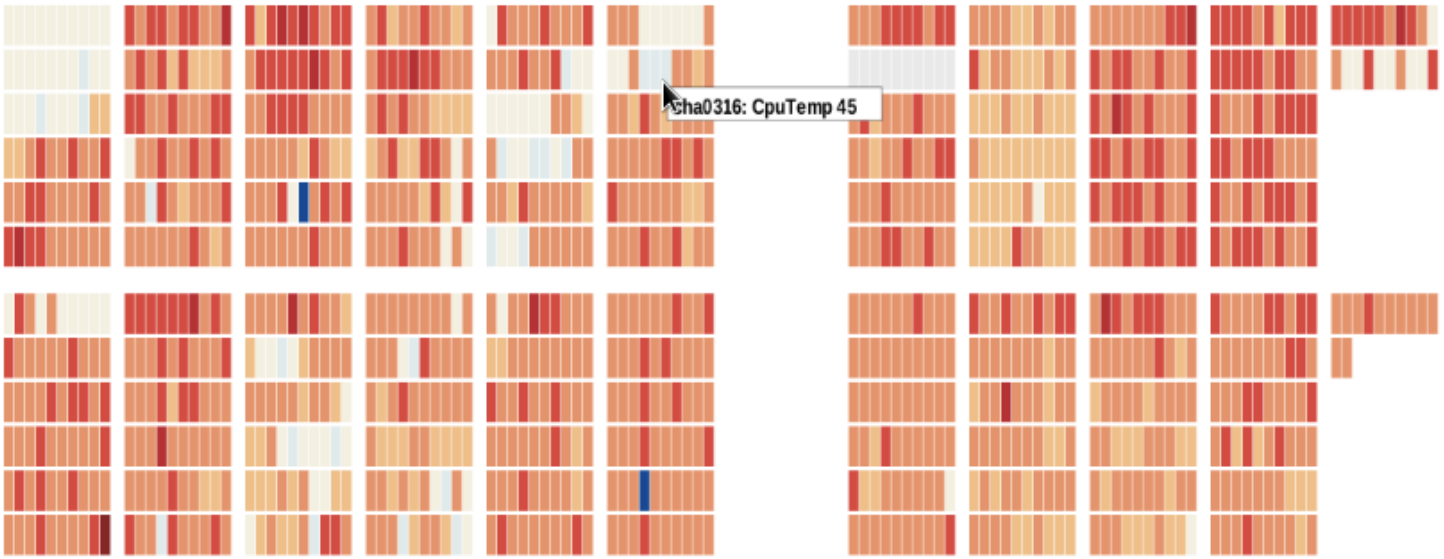
Component type: blade
Components: 1-1232
Metric: CpuTemp

Date: 07-03-2013
Time: 12:00:00
sec min hr day
(Current date)

Range: 91.00 / 0.00
Maximum:
Minimum:

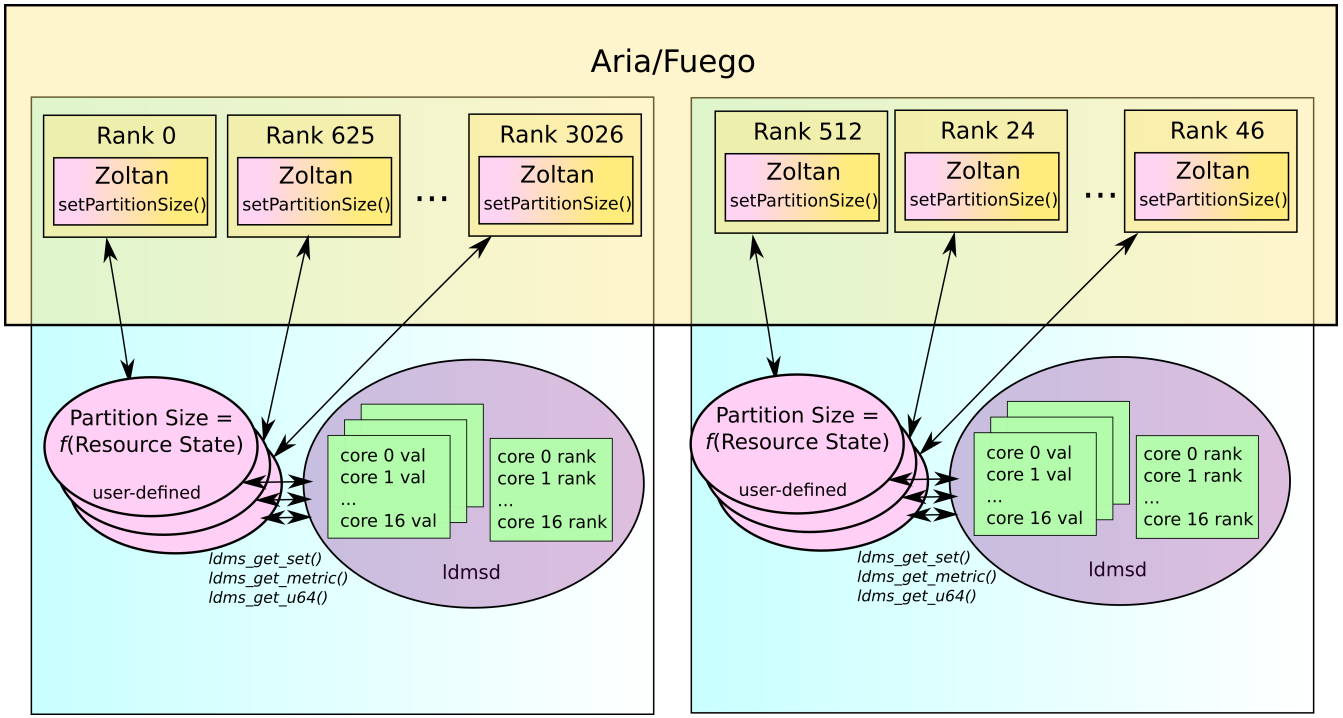
Scheme: Red-blue (rev) Spectra(rev) Gray(rev)
Actions: Play Stop Faster Slower

Visualize



Resource-Aware Applications

- Performance of an application depends on capabilities of the hardware and system software and on how the application utilizes resources.
- Utilize node level monitoring information to make run-time load balancing decisions.



- SIERRA Applications repartition using Zoltan
- Augmented Zoltan to call LDMS for resource utilization data via on-node interface and use that information in repartitioning calculation

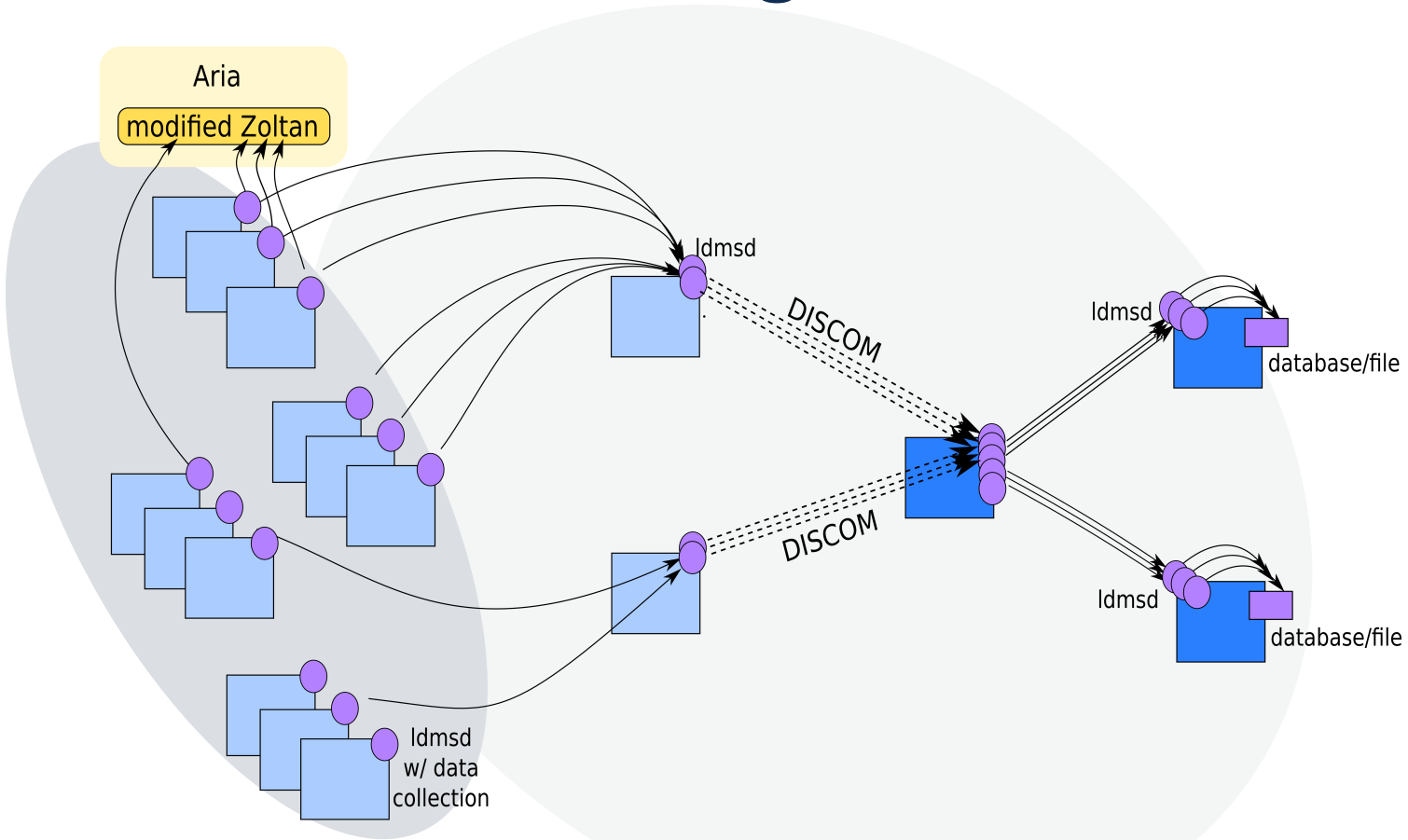
High Fidelity Data Collection and Transport Service Applied to the Cray XE6/XK6 (CUG 2013)

Sandia National Laboratories

ADVANCED DEPLOYMENT FEATURES

Cielo: Data streaming across DISCOM

Sandia National Laboratories

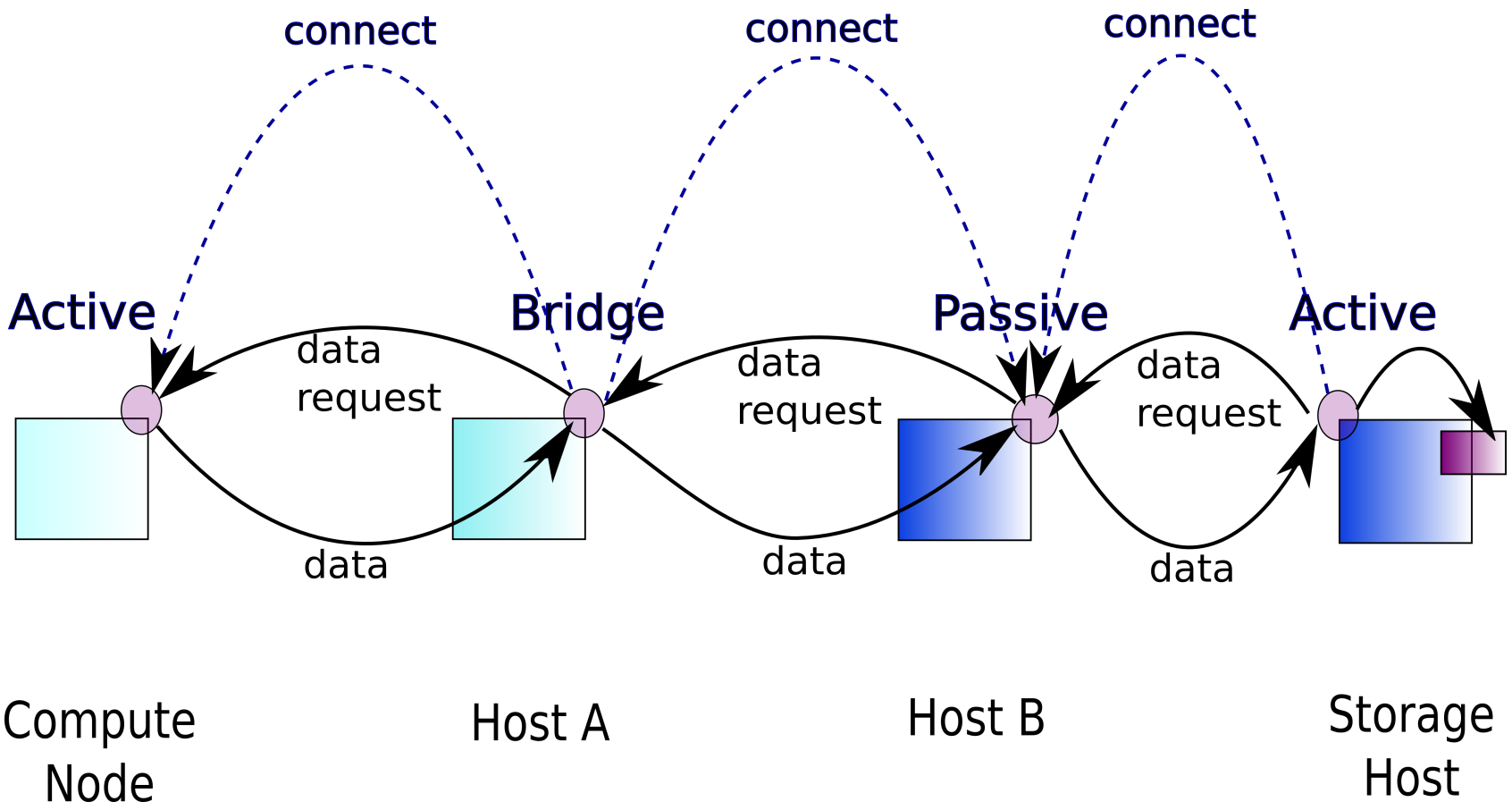


Cielo (Cray XE6)	**Cielo**	**Whitney**	**Whitney**
compute nodes: subset: 646 nodes, 10336 cores	viz nodes	132	admin nodes

FY11 ASC L2 Milestone:

Develop Feedback System for Intelligent Dynamic Resource Allocation to Improve Application Performance

Supporting Security Postures: Asymmetric Network Set-up



Concurrent Administrator and User LDMS Instances

Sandia National Laboratories

- Admin: Continuous system-wide LDMS deployment supporting administrator data of interest and data rates
- Users: transient job-wide LDMS's launched by the Resource Manager
  - User determines metrics and frequency based on interest and acceptable performance impact
  - User can adjust metrics and rate collected during run time (e.g., change rates, metrics based on what is happening in the computation).

Collaboration with LANL:
Dave Montoya, Mike Mason, Tim Randles, Cory Leuninghoener, Craig Idler

Security

- Reliably restrict access to data
  - Where are processes running and who is running?
    - Compute nodes – root/privileged user, user running on node
    - Storage servers – root/privileged user
    - Service nodes – root/privileged user
  - What ports would be open and on what servers
    - Compute nodes – privileged port, unprivileged port for user
    - Storage servers – privileged port
    - Service nodes – privileged port
- Privilege
  - Authentication – Unix AUTH
  - Daemons run as root or user/group ldms:ldms
- Communication
  - Trusted Networks (e.g. uGNI)
    - Privileged port or other access mechanism (e.g. PTags in uGNI)
  - Un-trusted Networks
    - Accessible through LDMS secure socket (SSL) transport
    - Authentication through private/public key + Unix AUTH
    - Data are encrypted via SSL

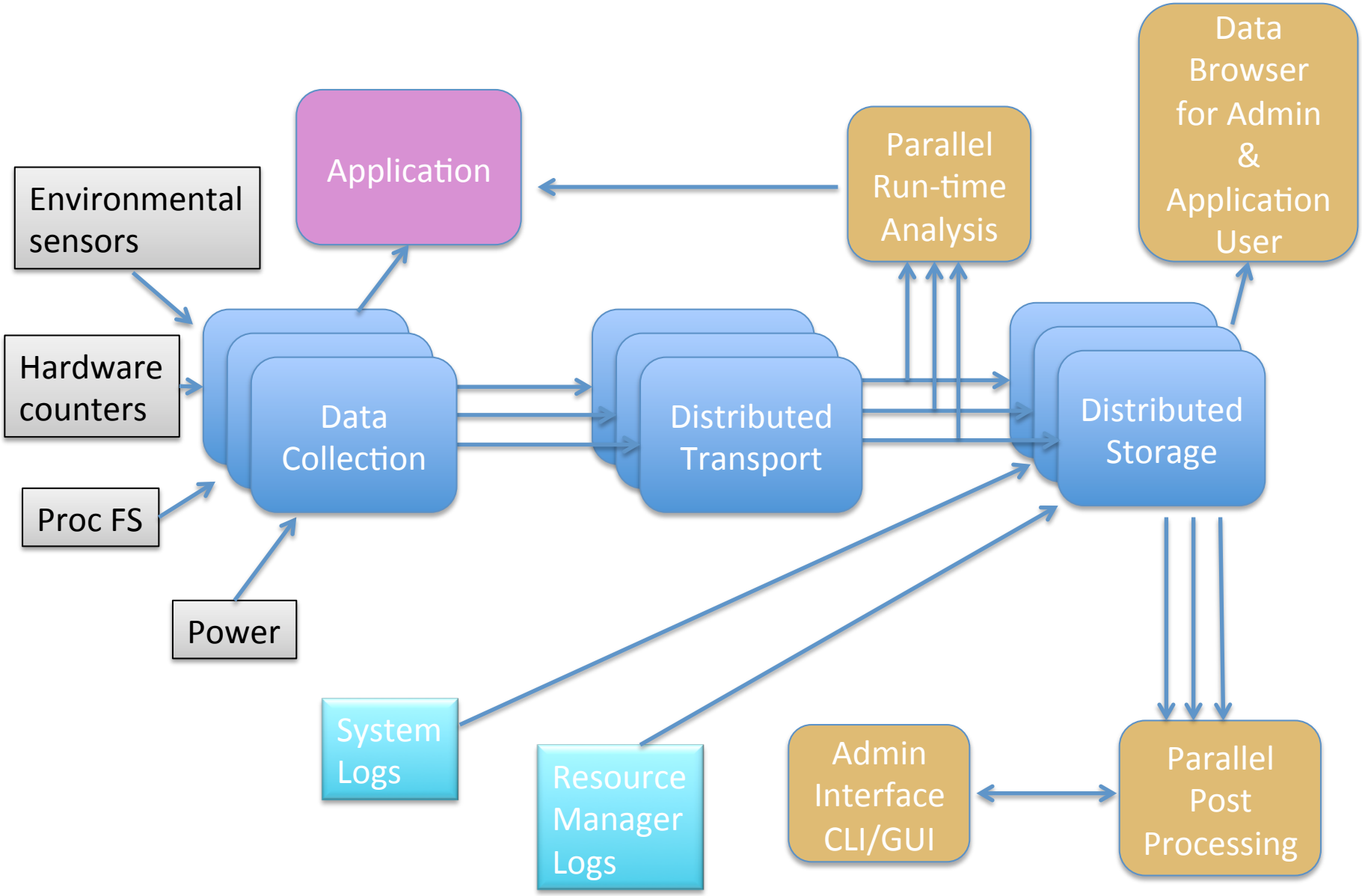
Summary

- Lightweight Distributed Metric Service (LDMS):
  - System service that provides low-overhead collection and storage of high-fidelity system related data
  - Continuous job and system resource utilization profiling
    - Supply to the user full data dumps, graphs, statistics etc
  - Resource utilization information in actual run conditions
  - On-node interface for run-time application interaction
- Developing production architecture to launch per-job LDMS instances
  - User dynamically selects metric sets of interest and frequency based on interest and acceptable performance impact
- Open Source
  - http://ovis.ca.sandia.gov
  - ovis@sandia.gov

Sandia National Laboratories

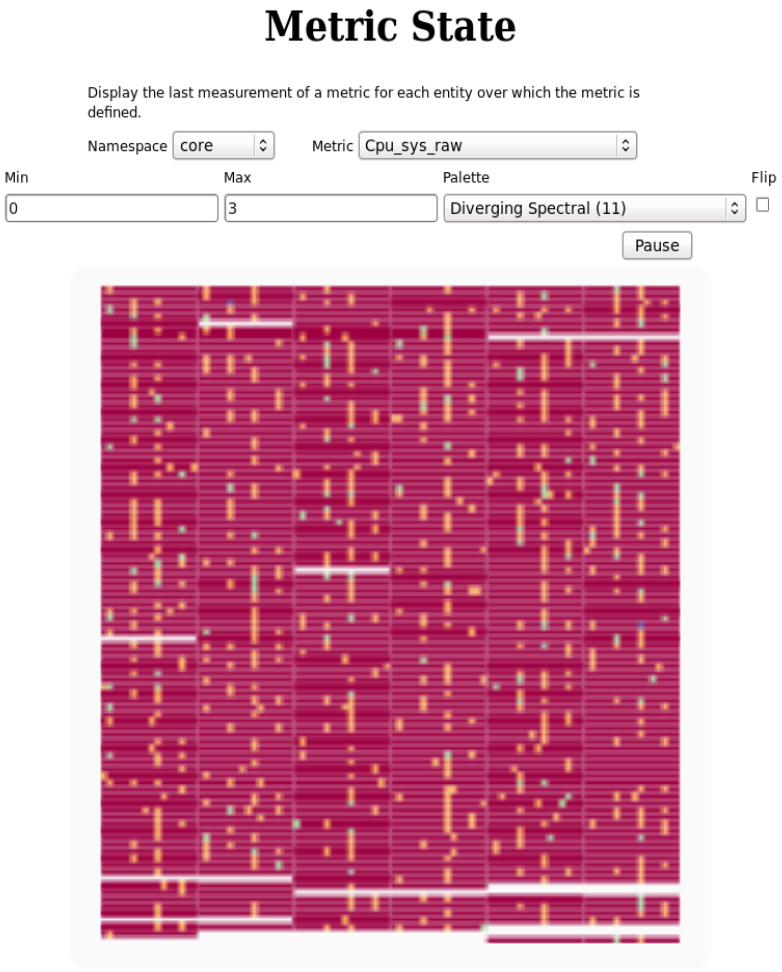
EXTRA SLIDES

Scalable HPC Information Processing



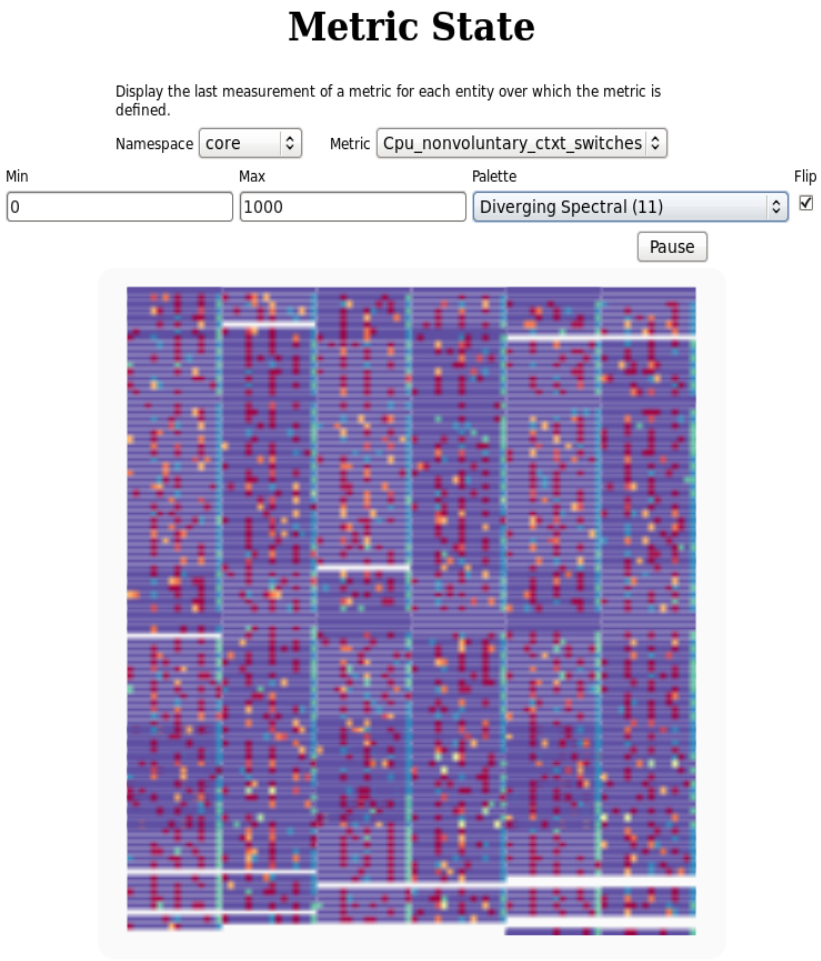
System CPU Utilization

Sandia National Laboratories



- 8310 Cores during idle on Cray XE6

Non-Voluntary Context Switches



- From 8310 processor application run on Cray XE6

Visualization View - OVI

localhost:8082/views/vis_view.html?type=chassis&metric=InletTemp0

Visualization View - OVIS

Components: blade (1232) | rack (26) | chassis (140) | numanode (2464) | core (19712). Details

Monitoring: Job View, Visualization View

Analysis: Time-history View, Analysis View

Enter in a thing to draw (component type then components) then what to draw (metric). This view draws the historical value, at one point in time, of a metric for every component of the given type in a rectangular grid.

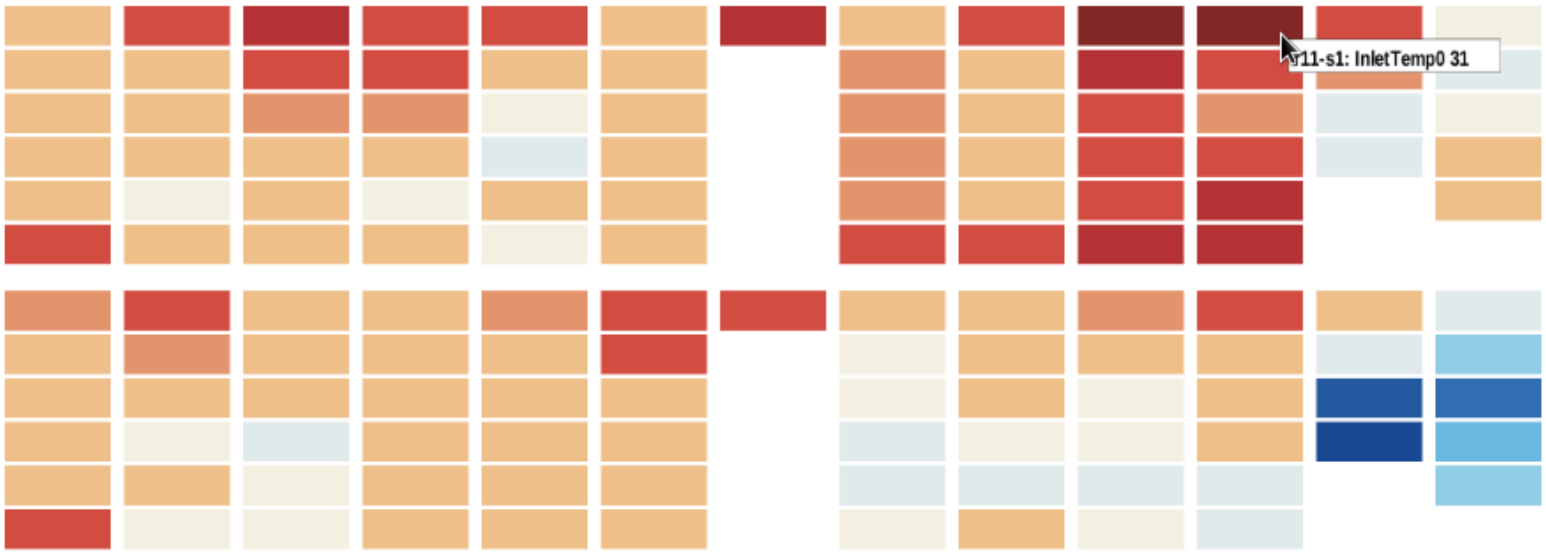
Component type: chassis
Components: 1-140
Metric: InletTemp0

Date: 07-03-2013
Time: 12:00:00
sec min hr day
(Current date)

Range: 31.00 16.00
Maximum:
Minimum:

Scheme: Red-blue (rev) Spectra(rev) Gray(rev)
Actions: Play Stop Faster Slower

Visualize



LDMS MetricSets Example

Sandia National Laboratories

ldms_ls -h localhost -x sock -p 60000: (Lists the metric sets)

- shuttle-cray.ran.sandia.gov_1/meminfo
- shuttle-cray.ran.sandia.gov_1/procstatutil
- shuttle-cray.ran.sandia.gov_1/vmstat

LDMS MetricSets Example

Sandia National Laboratories

ldms_ls -h localhost -x sock -p 60000 –v

- shuttle-cray.ran.sandia.gov_1/vmstat
- METADATA --------
- Size : 6954
- Inuse : 3792
- Metric Count : 97
- GN : 98
- DATA ------------
- Size : 808
- Inuse : 808
- GN : 105410
- -----------------
- shuttle-cray.ran.sandia.gov_1/procstatutil
- METADATA --------
- Size : 7805
- Inuse : 4192
- Metric Count : 118
- GN : 119
- DATA ------------
- Size : 976
- Inuse : 976
- GN : 128468
- -----------------

LDMS MetricSets Example

ldms_ls -h localhost -x sock -p 60000 –l:

- **shuttle-cray.ran.sandia.gov_1/meminfo**
- U64 1 component_id
- U64 160032 MemFree
- U64 181728 Buffers
- U64 3443332 Cached
- U64 33076 SwapCached
- U64 2987544 Active
- **shuttle-cray.ran.sandia.gov_1/procstatutil**
- U64 1 component_id
- U64 1826564 cpu0_user_raw
- U64 699631 cpu0_sys_raw
- U64 663843760 cpu0_idle_raw
- U64 201018 cpu0_iowait_raw
- **shuttle-cray.ran.sandia.gov_1/vmstat**
- U64 1 component_id
- U64 40008 nr_free_pages
- U64 122286 nr_inactive_anon
- U64 321902 nr_active_anon
- U64 465532 nr_inactive_file
- U64 424986 nr_active_file

RDMA Transport Overview

Sandia National Laboratories

- Metric Sets reside in pinned, registered memory.
  - Metric Sets may reside in either kernel or user memory
- Metric Set Remote Key/Virtual Address is shared with remote peer and returned with metric set directory information
- Metric Set contents are ‘mirrored’ on consumer
- Metric Set contents are made consistent with an RDMA READ that is issued through *ldms_update*

Security

Provide Compatibility with security postures of platforms

Sandia National Laboratories

- Reliably restrict access to data
  - Where are processes running and who is running?
    - Compute nodes – root/privileged user, user running on node
    - Storage servers – root/privileged user
    - Service nodes – root/privileged user
  - What ports would be open and on what servers
    - Compute nodes – privileged port, unprivileged port for user
    - Storage servers – privileged port
    - Service nodes – privileged port
  - Prefer not to run as root – have a trust relationship between source and consumer – currently investigating options

Security

- Privilege
  - Authentication – Unix AUTH
  - Daemons run as root or user/group ovis:ovis
- Communication
  - Trusted Networks (e.g. uGNI)
    - These networks must be physically secured from un-trusted clients.
    - Privileged port or other access mechanism (e.g. PTags in uGNI)
  - Un-trusted Networks
    - Accessible through LDMS secure socket (SSL) transport
    - Authentication through private/public key + Unix AUTH
    - Data are encrypted via SSL