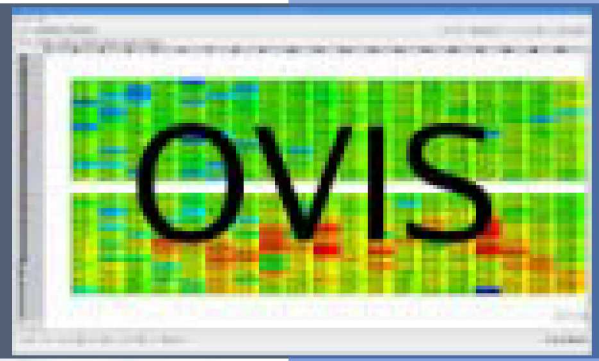
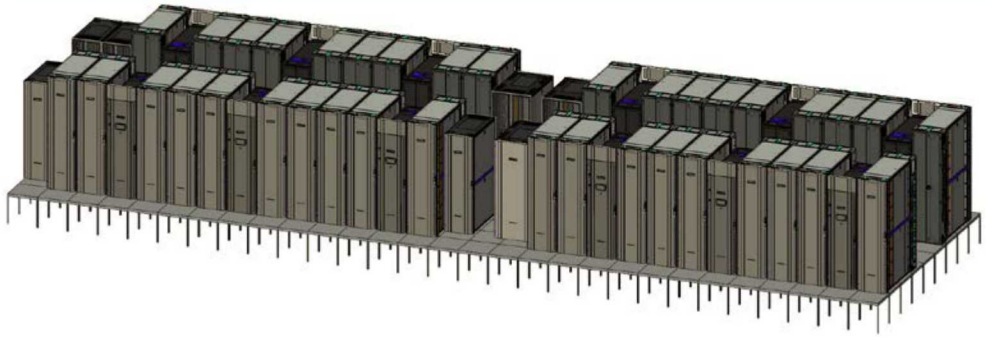
SAND2019-3484PE

Sandia National Laboratories

High Performance Computing Monitoring





PRESENTED BY

Benjamin Allan

U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
Unclassified, Unlimited Release.

Outline

- What is HPC monitoring?
- Why?
- Who dunnit?
- Data deluge
- Examples
  - Watch yourself
  - Application behavior
  - Abnormal behavior
  - Seeing the network

What is HPC monitoring?

- Understanding how modern high performance computing systems and applications behave and what to do about it.
  - A problem in big data collection and analysis.
  - 1-2 Petabyte/year for our unclassified HPC
    - 7 year history desired
    - Mostly time-series data
    - Also system log files
    - Possibly application log files
  - *Collecting must not interfere with applications (CPU, net, RAM)*
  - *Coherent snapshots at moderate frequencies*
    - *Synchronous across entire computing center at 0.01-1 Hz (not cloud!)*
    - *Track shared resources (parallel file systems, NFS, etc)*

Why and what are we monitoring?

- Losing one node will kill the entire application (this is not the cloud)
- HPC monitoring systems have a lot of data sources:
  - Kernel, CPUs, GPUs, RAM event counters
  - Storage devices, network devices
  - Power supplies, fans
  - Pipes, pumps, valves, tanks, cooling towers
  - Electric utility lines
  - People (users and admins)
- A problem in any of these can affect the applications (and therefore the process of getting the science & engineering results).

  The relationship is two-way:

  suddenly stopping a 2 MW calculation

  really bothers the electric company.

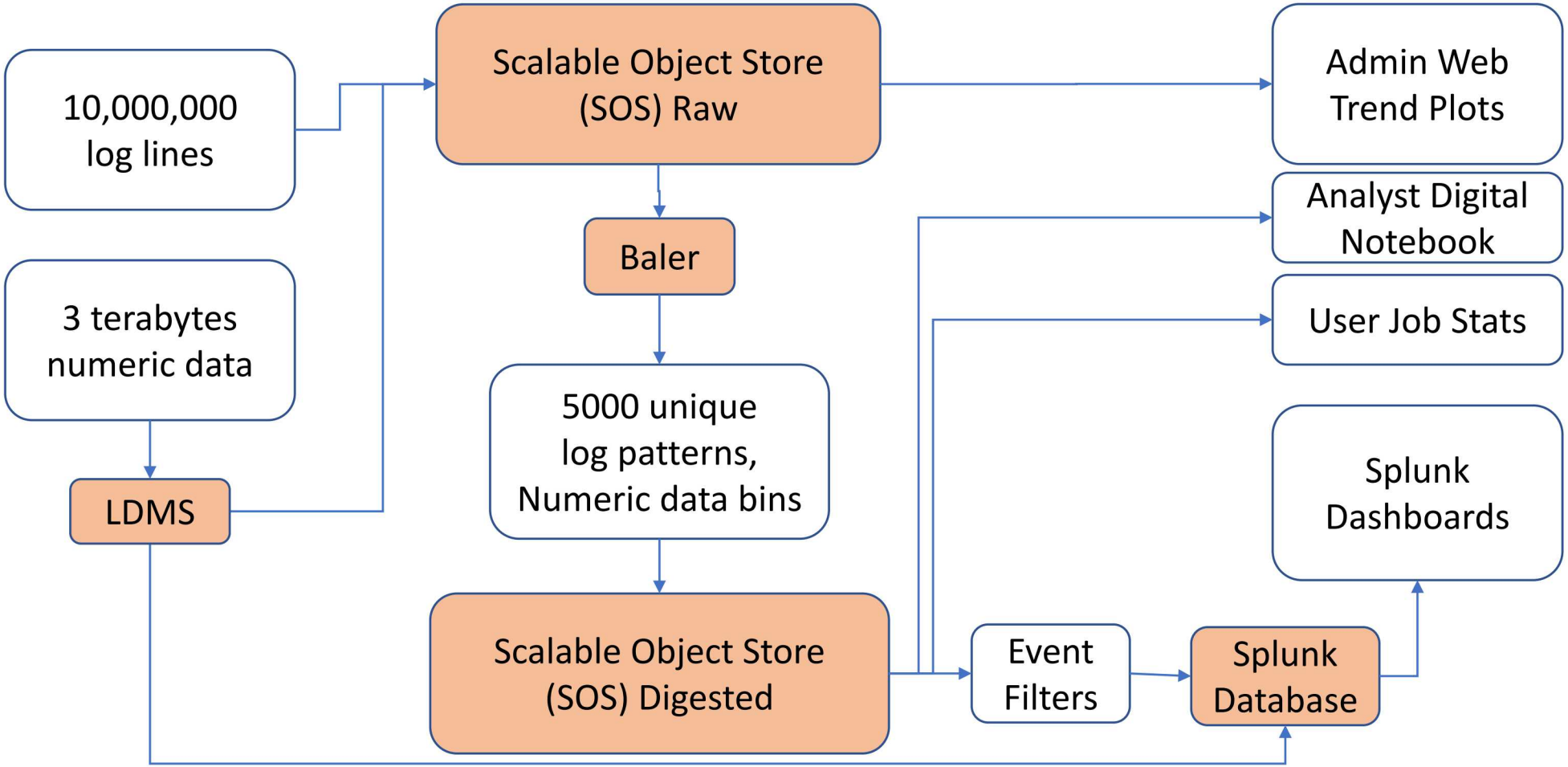
Whodunnit?

Monitoring development is a team sport

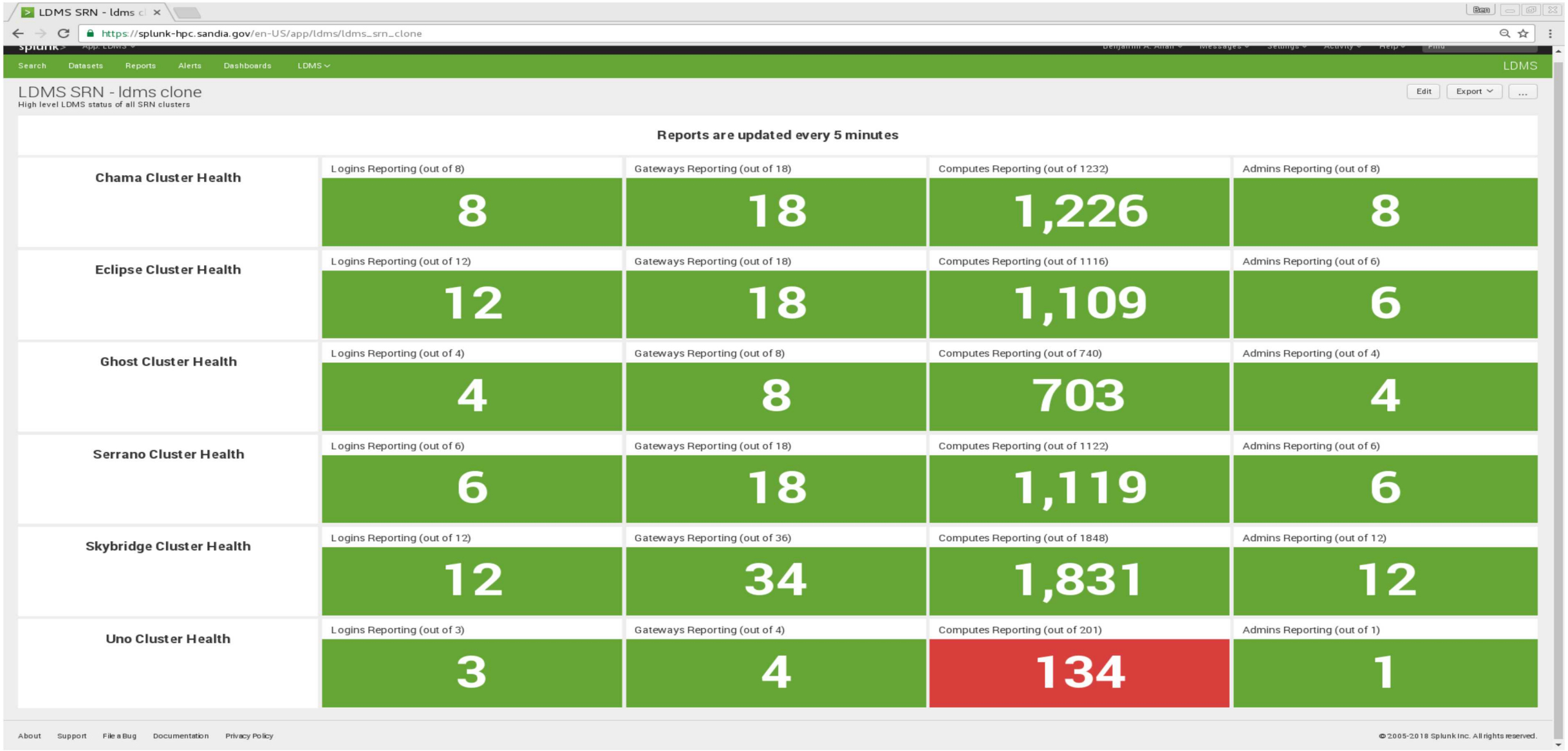
- System administrators
- Managers, system designers, system vendors
- Software developers (monitoring tools and application users)
- OVIS team includes many organizations:
  - Sandia, Los Alamos, Lawrence Livermore, NCSA
  - Open Grid Computing, Cray
  - Boston U., U. Central Florida, U. New Mexico, U. Illinois
  - ***Add your name here!***

Data flows and OVIS tools daily (the plan)

Logs + sensor data + counters from all the clusters



Splunk dashboard (watch yourself)



Production application (what's normal?)

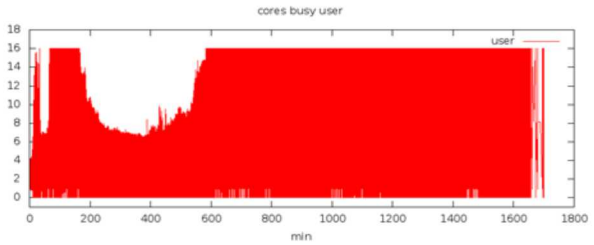
Black box – no source or library tinkering allowed

- Real application runtime is days/weeks
- OpenMP threaded, single node

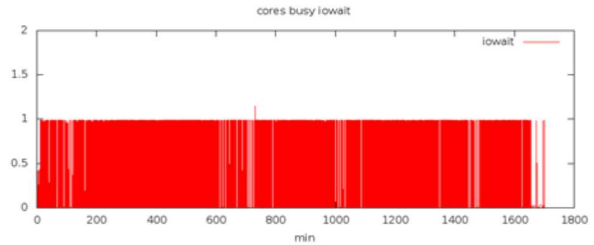
Results suggest trying 2 jobs/node allocation

1 day

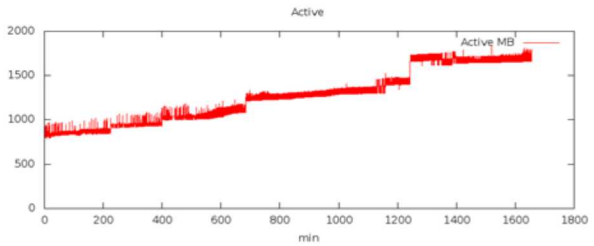
Cores busy user



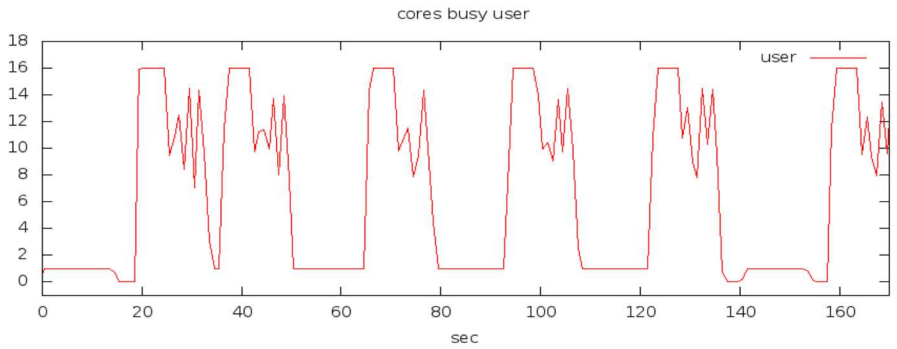
Cores in iowait

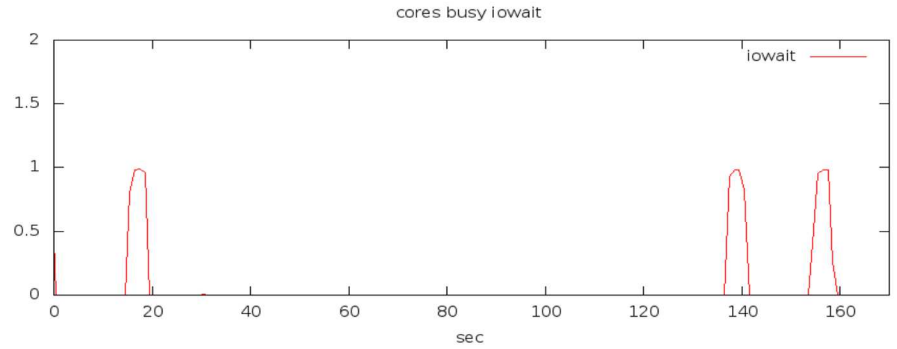


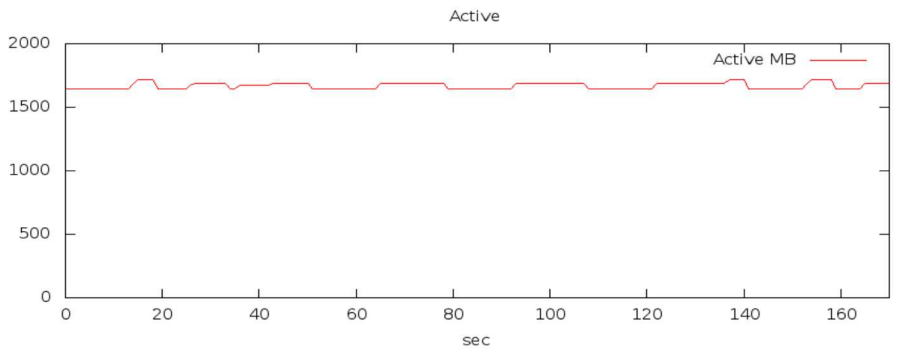
RAM used



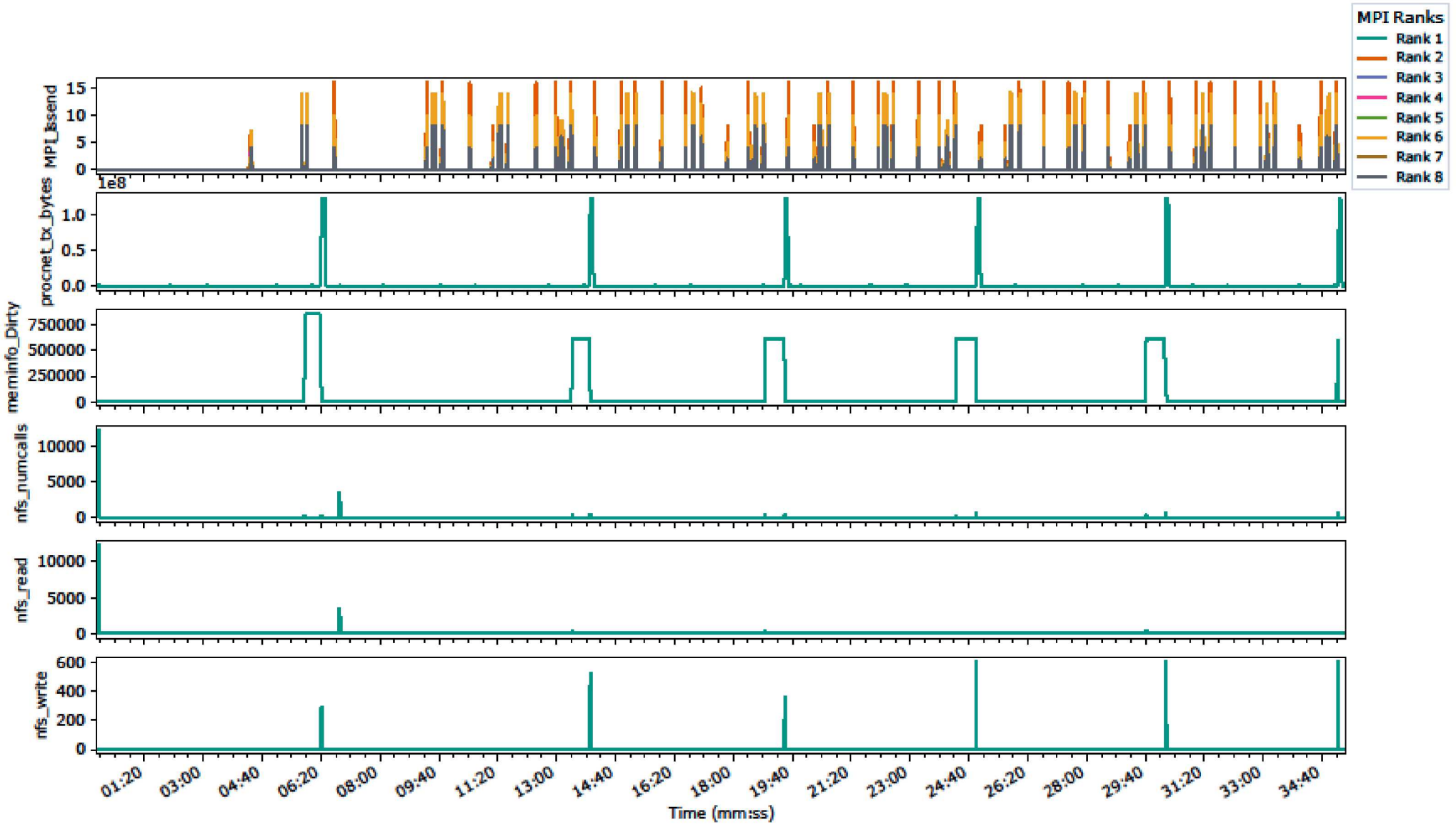
2 minutes (zoomed)



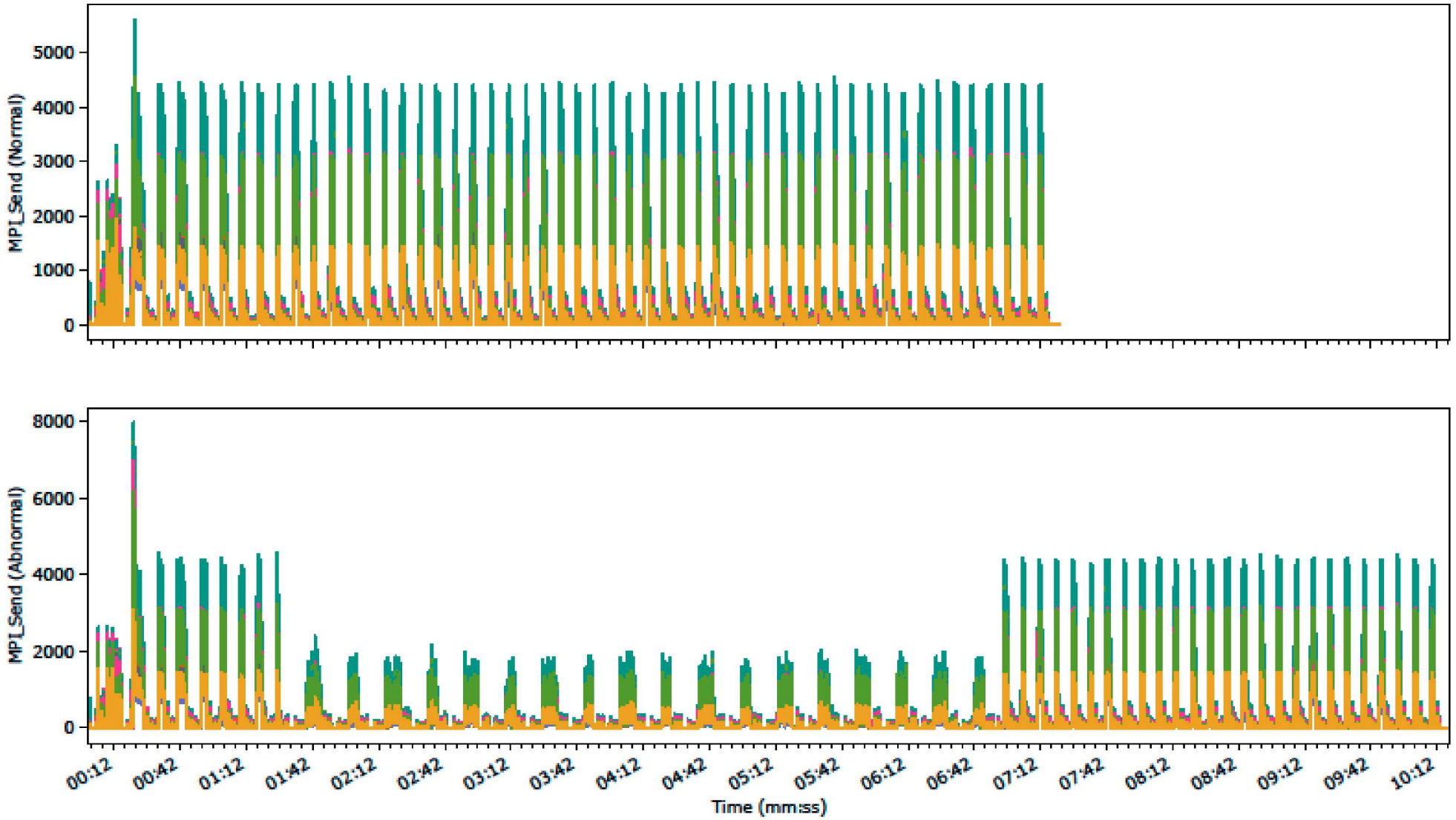




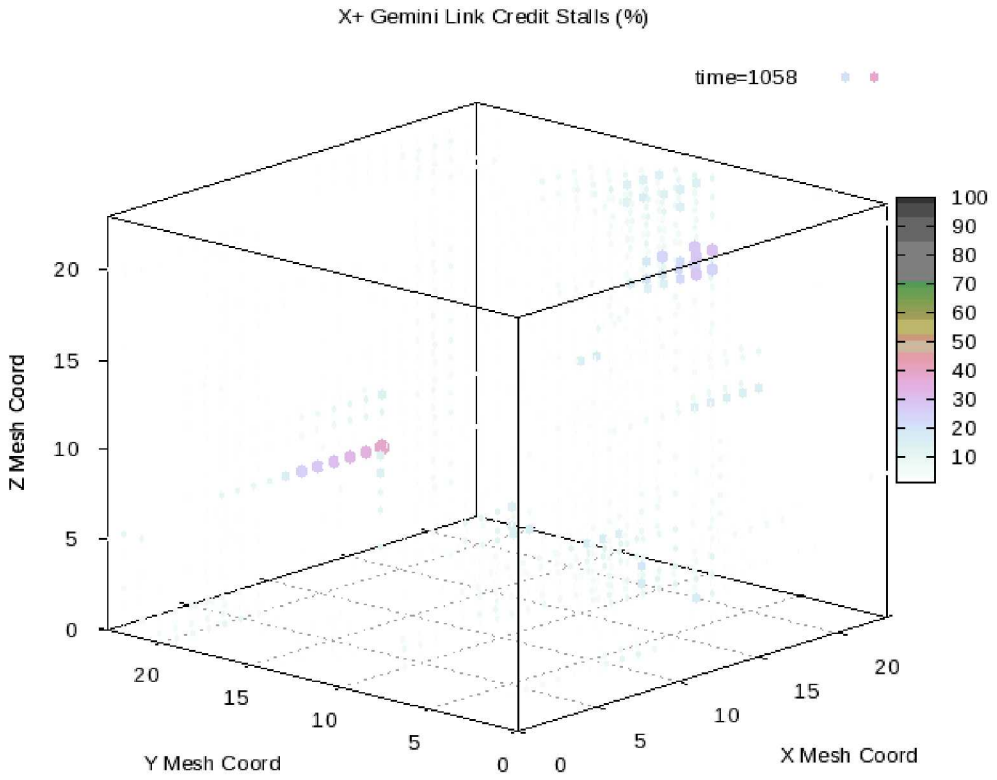
MPI + system data: quicker understanding



Comm data: 30% longer (what's wrong?)



Where is the Cray network problem?

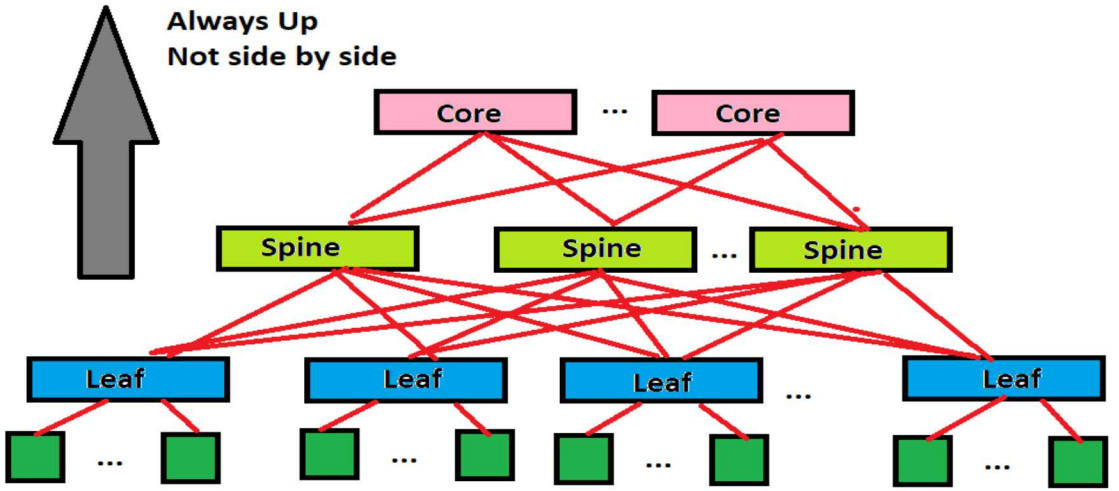


InfiniBand switch port monitoring

Skybridge (1900 nodes)

268 Switches, 9648 Switch Ports

- Single collector rate possible:
  - once per 20 sec
- 10 collector rate possible:
  - 60Hz



Comprehensive, Synchronous, High Frequency Measurement of InfiniBand Networks in Production HPC Systems, (SNL) Aguilar et al, Open Fabrics Alliance Workshop 2018.

Measuring Minimum Switch Port Metric Retrieval Time and Impact for Multi-layer InfiniBand Fabrics, Aguilar et al, IEEE Cluster, 2017.

Questions we can answer (often) using OVIS

How much of each resource (net, disk, RAM, processor, power) does each application use?

Where should we spend extra money to update current systems?

Why is this application running slower than yesterday?

What demands should we design/build/test the next system around?

What is degrading and likely to fail in the near future?

The End

See also:

https://ovis.ca.sandia.gov

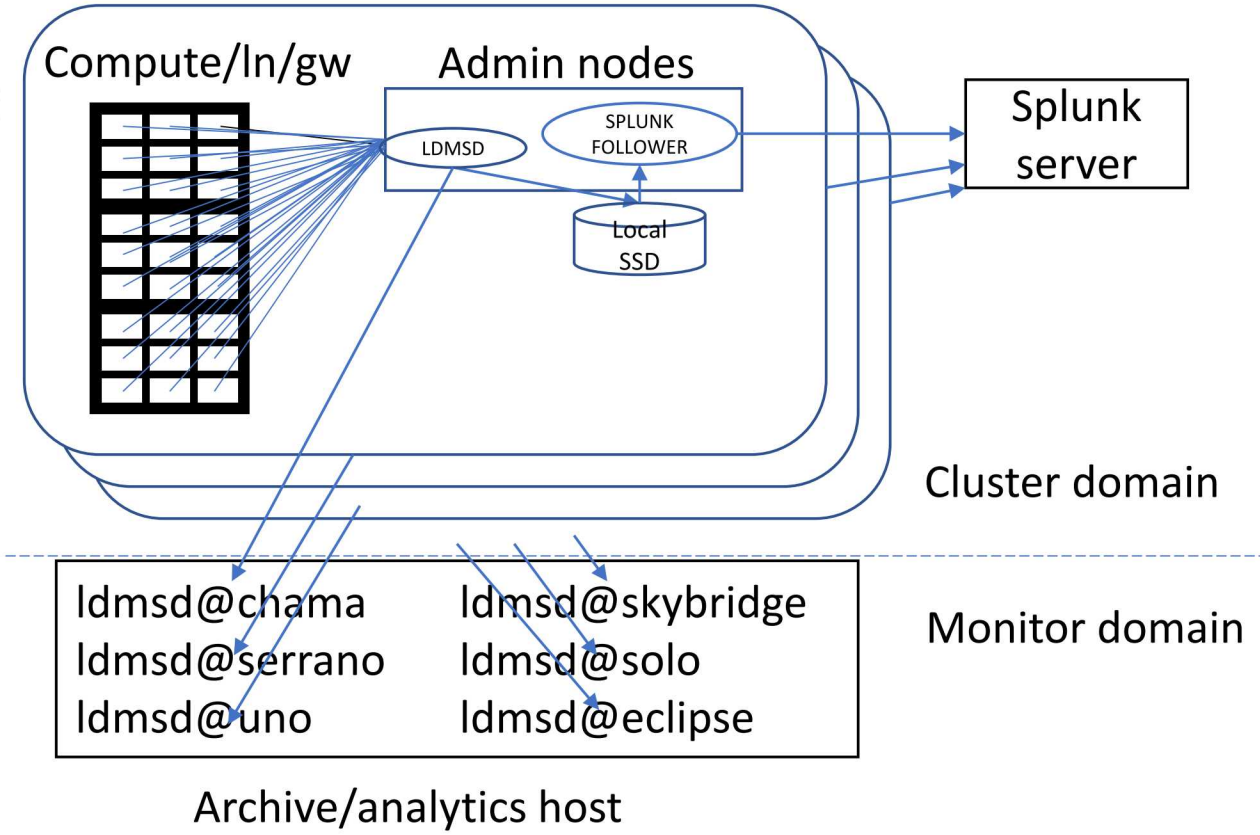
Lightweight distributed metric service

Low overhead, high rate system performance data collection

- Plug-ins for collecting, transporting, storing metrics

How:

- LDMSD collection on all nodes (L0)
- L1 aggregation @ admin nodes
  - In cluster
  - SSD buffer, splunk store (optional)
- L2 aggregation
  - Outside clusters admin domain
  - Archive store
  - Web interfaces
- Robust to L2, store outages



Development process

Internal gitlab (Open Grid Computing, university and lab partners)

External github

Tests/examples suite included in source, browsable on github

SNL Jenkins build/package testing (CLE6, TOSS 3) of public releases

Public releases for TOSS

Pre-release test versions for collaborators

Changes based on feedback from production admins

Data sets monitored on SNL TOSS clusters

- **Kernel**: /proc/interrupts, proc/meminfo, /proc/stat, /proc/vmstat,
- **Filesystems**: Lnet stats <pending>, Lustre client stats, NFSv3
- **Network**: InfiniBand HCA, OmniPath HFI, /proc/net/dev
- **Queue**: Job info (fed by SLURM)
- **Hardware**: EDAC (RAM errors) <pending>
- **Monitoring**: LDMSD (daemon self metrics)