Lustre Metrics

New techniques for monitoring Lustre

Scott Nolin
Andrew Wagner
14 April 2015

About SSEC

The University of Wisconsin Space Science and Engineering Center (SSEC) is a research and development center focusing on geophysical research and technology to enhance our understanding of the atmosphere of Earth, the other planets in our Solar System, and the cosmos.

Major SSEC Initiatives

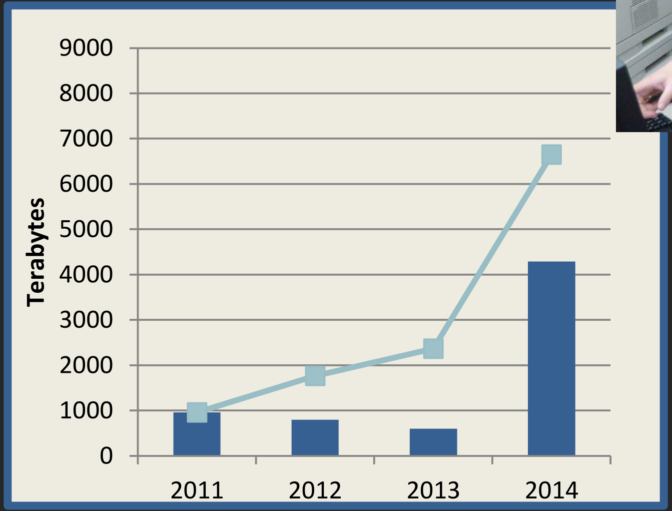
- atmospheric studies of Earth and other planets
- interactive computing, data access, and image processing
- spaceflight hardware development and fabrication

Noted Scientific Work

- satellite-based and other weather observing instruments
- remote sensing applications in earth and atmospheric science
- spaceflight instrumentation
- planetary meteorology
- data analysis and visualization
- diagnostic and numerical studies of the atmosphere

credit (NASA-GRC)

Why we care about this at SSEC



New techniques and open source tools enable a powerful new approach for monitoring Lustre.

the M word

- We are concerned with Lustre time series data and visualization.

Metrics is a popular term for recent projects working with time series data, often used as shorthand for the whole process. So we will use Lustre Metrics to describe our work.

So what is it?

- Time series database of whatever you want to measure.
- Very easy to add or change metrics.
- Visualization dashboards an analyst customizes as part of a normal workflow.

Logs not included

- For a good overview of modern log analysis see Kalpak Shah, LUG 2014
  - http://cdn.opensfs.org/wp-content/uploads/2014/04/D3_S32_LustreLogAnalyzer.pdf

Current Lustre Monitoring Tools

- LMT
- lltop and xltop
- Ganglia with collectl
- Vendor tools

These work!

But customizing can be difficult and they're not extremely flexible. The Ganglia system is the most easily customized, but the visualization component is not.

New Methods: Advantages

- General tools and techniques, not Lustre-specific.
- Flexible system that can evolve
- Measure “everything”
- Analyst has powerful visualization tools.

New Methods: Limitations

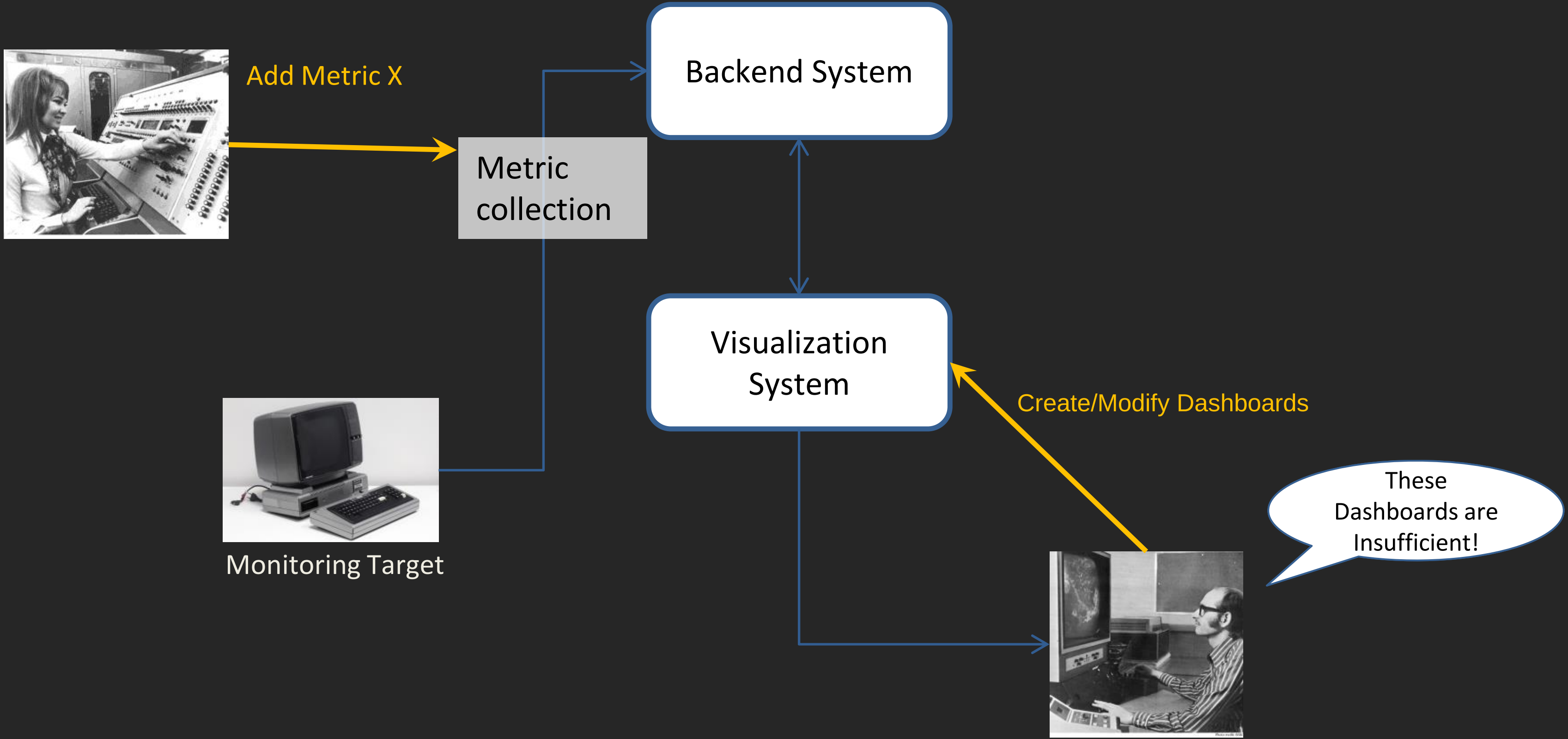
Time series databases make these things harder.

- Reports
- Archive of Historical Data
- Temporary metrics (jobstats)

Key: Iterative Prototyping

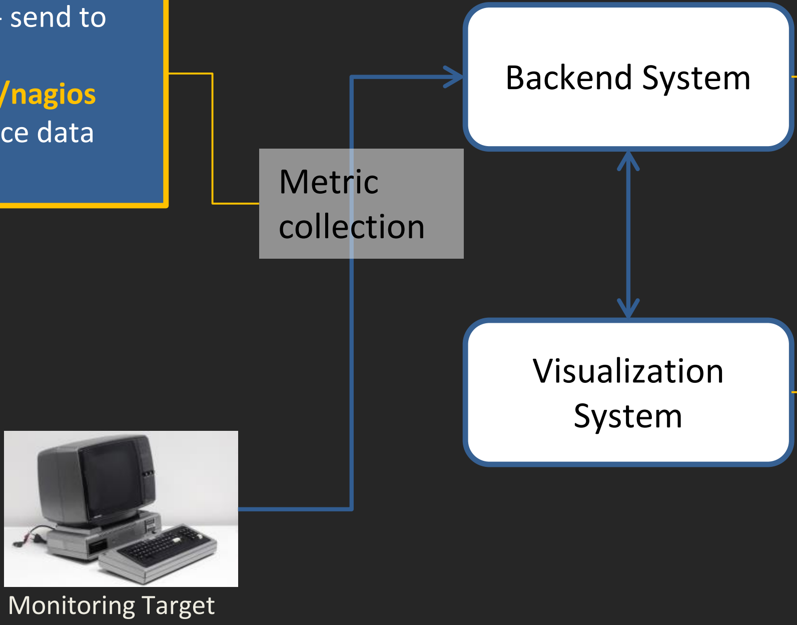
- System is immediately useful
- Start with low hardware investment and scale as needed
- Learn what Lustre statistics matter for you, and grow the monitoring system.

New Metrics System Workflow



New Tools: Many Choices

- **Collectd**
- **DIY** script - send to socket
- **Check_mk/nagios** performance data
- **Ganglia**



- **Graphite** - the most popular back end.
- **OpenTSDB and InfluxDB** are newer alternatives

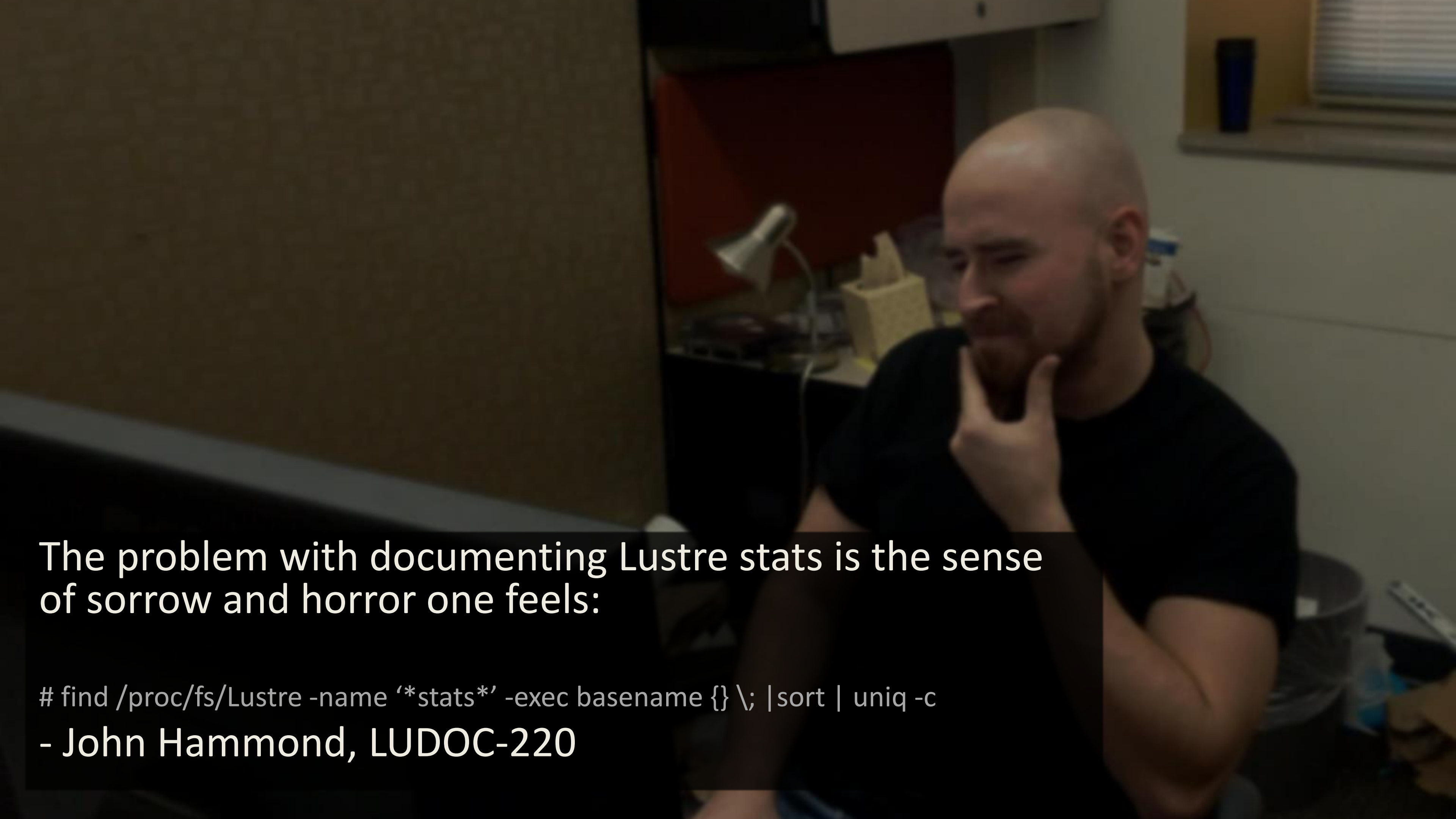
- **Grafana** is our favorite dashboard.
- **Graphite built in dashboard** tool.
- **Graphite URL api** to get data directly (csv, png, etc)
- **graph explorer** https://github.com/vimeo/graph-explorer: adds metadata to metrics

Case Study: Lustre Metrics at SSEC

Andrew Wagner

Prerequisites for SSEC Lustre Monitoring

- Configuration Management
- Hardware: Graphite back end / Grafana for dashboards. We used one server.
- Pick a Graphite name space.
  - This will be wrong (iterative prototype!)
- Decide what metrics to measure from Lustre stats files. This is challenging.



The problem with documenting Lustre stats is the sense of sorrow and horror one feels:

```
# find /proc/fs/Lustre -name '*stats*' -exec basename {} \; |sort | uniq -c
```

- John Hammond, LUDOC-220

Where to Look for Lustre Metrics?

- Lustre Manual
- Information in various presentations.
  - Daniel Kobras, LAD2012 http://www.eofs.eu/fileadmin/lad2012/06_Daniel_Kobras_S_C_Lustre_FS_Bottleneck.pdf
  - Florent Thery, LAD2013. http://www.eofs.eu/fileadmin/lad2013/slides/11_Florent_Thery_LAD2013-Lustre-bull-monitoring.pdf
  - Daniel Rodwell and Patrick Fitzhenry, LUG2014. http://www.opensfs.org/wp-content/uploads/2014/04/D3_S31_FineGrainedFileSystemMonitoringwithLustreJobstat.pdf
    - This is where we learned about graphite and new techniques, thanks!
  - Gabriele Paciucci and Andrew Uselton, LAD2013.http://www.eofs.eu/fileadmin/lad2013/slides/15_Gabriele_Paciucci_LAD13_Monitoring_05.pdf
  - Eric Focht, LAD2014 http://www.eofs.eu/fileadmin/lad2014/slides/14_Erich_Focht_LAD2014_Monitoring.pdf
- Lustre Mailing List
- Searching source of tools like LMT and lltop.

Version 1: Metadata Stats

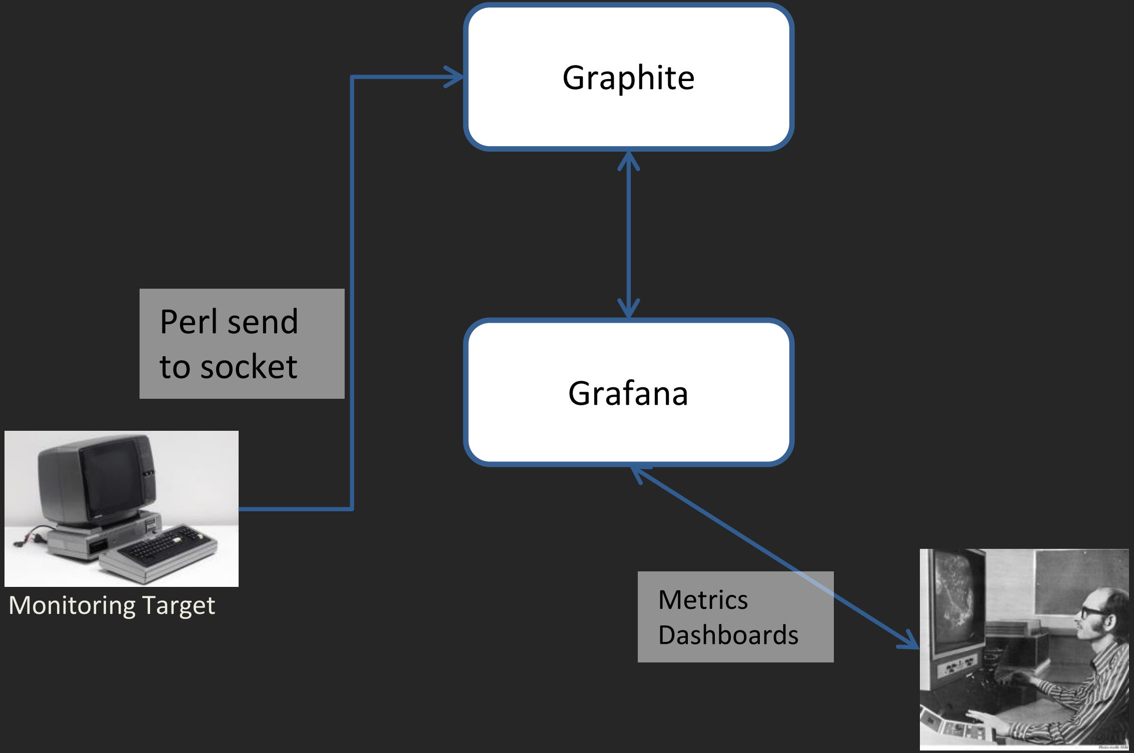
You have multiple filesystems.

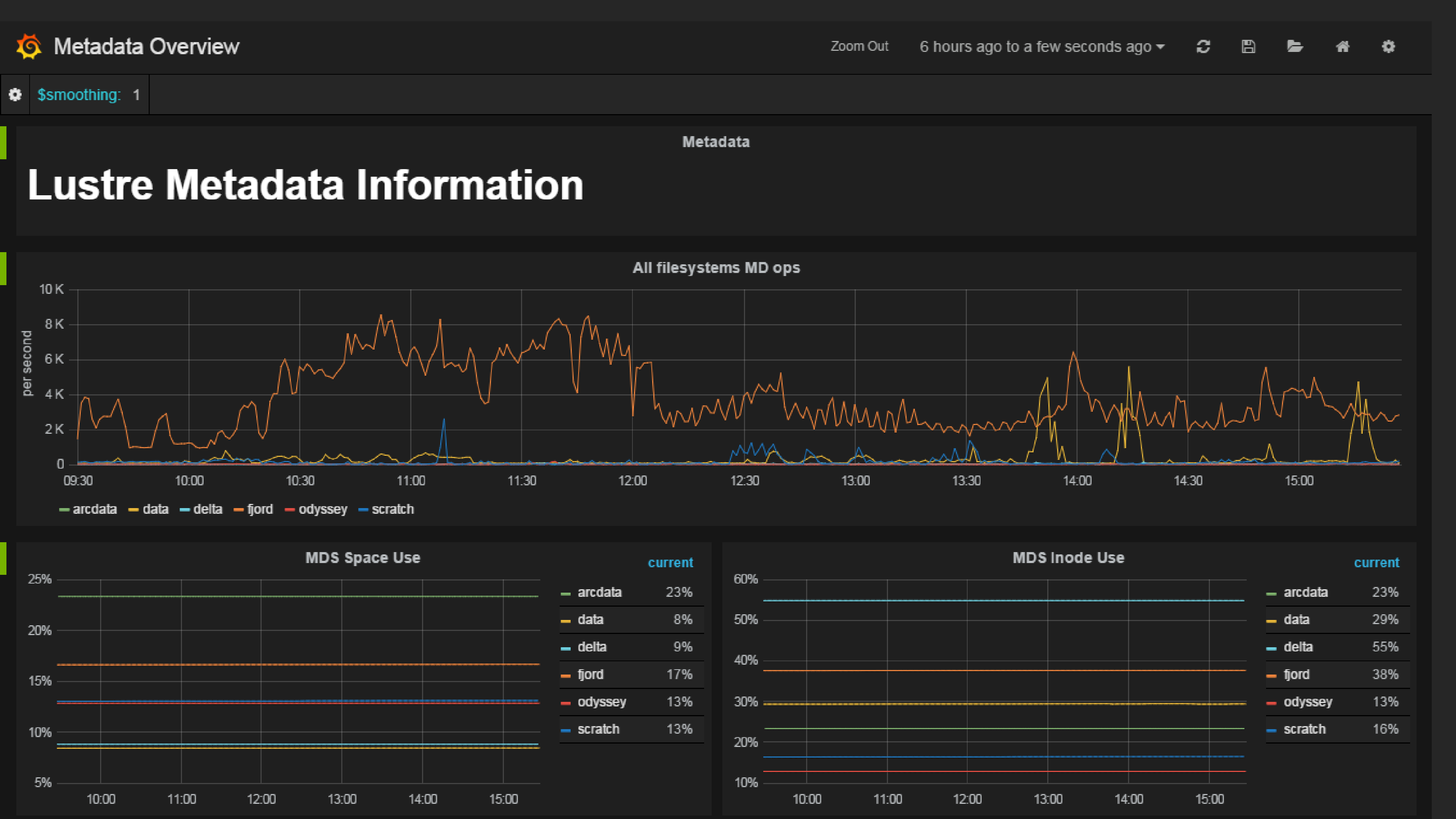
Which one is getting crushed with metadata operations?

Version 1: Metadata Stats

- Perl script run on MDS via cron every minute, and send data to Graphite socket. Simple, and easy to debug.
- On the MDS:
  - lctl get_param mdt.*.md_stats
    - read, open, close, stat, etc
  - lctl osd-*.*MDT*.filesfree and filestotal
    - available and total inodes
  - lctl osd-*.*MDT*.kbytesfree and kbytestotal
    - available and total disk space

Version 1


Graphite
Grafana
Perl send to socket
Monitoring Target
Metrics Dashboards



Lustre Metadata Information

All filesystems MD ops

MDS Space Use

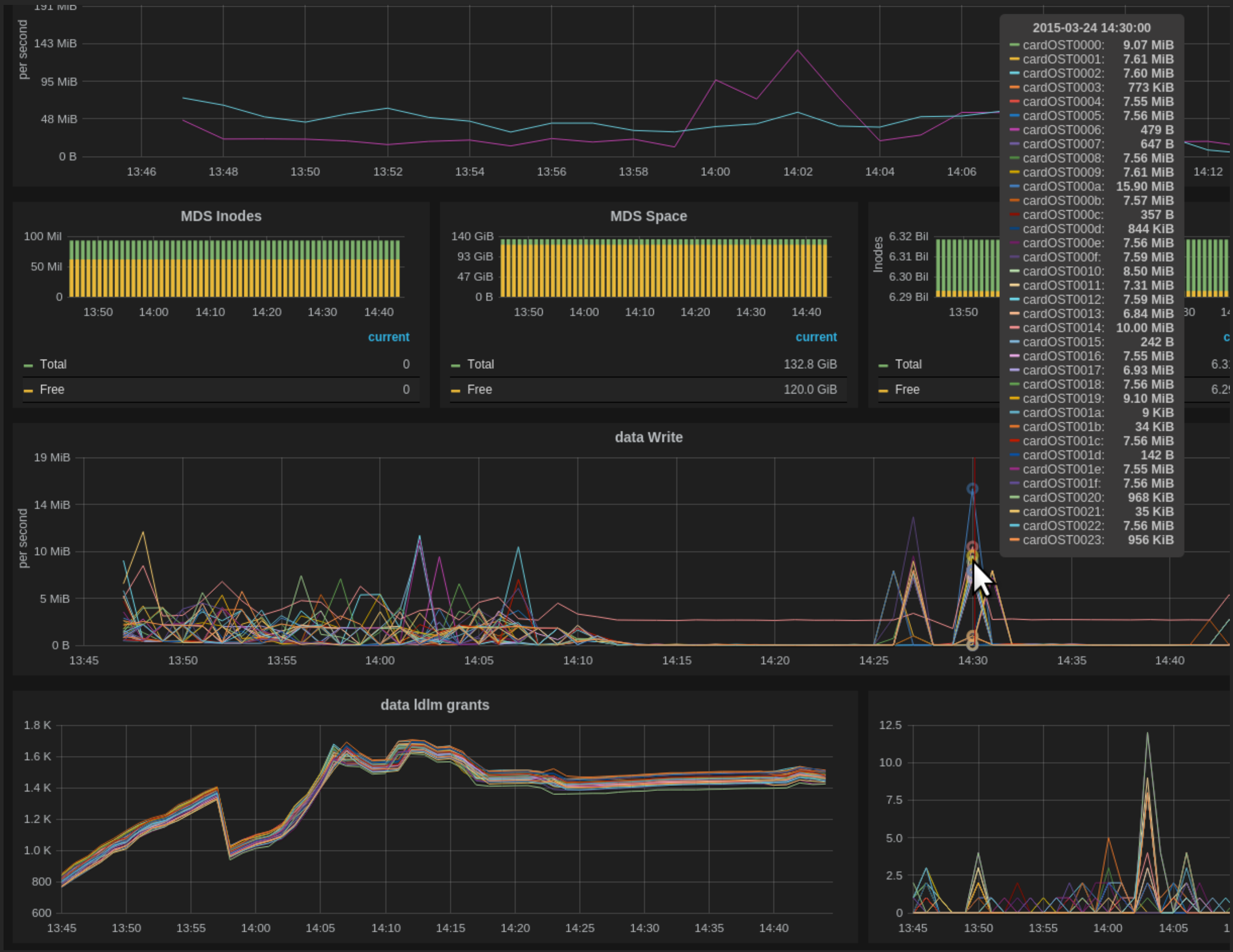
	current
arcdata	23%
data	8%
delta	9%
fjord	17%
odyssey	13%
scratch	13%

MDS Inode Use

	current
arcdata	23%
data	29%
delta	55%
fjord	38%
odyssey	13%
scratch	16%

Version 2: Add OST Stats

- lctl get_param obdfilter.*.stats
  - Stats per OST. Read and write data is particularly interesting
- lctl get_param obdfilter.*OST*.kbytesfree
  - kbytestotal, filesfree, filestotal
- lctl get_param ldlm.namespaces.filter-*.pool.granted
  - grant_rate, cancel_rate



```
2015-03-24 14:30:00
cardOST0000:   9.07 MiB
cardOST0001:   7.61 MiB
cardOST0002:   7.60 MiB
cardOST0003:    773 KiB
cardOST0004:   7.55 MiB
cardOST0005:   7.56 MiB
cardOST0006:      479 B
cardOST0007:      647 B
cardOST0008:   7.56 MiB
cardOST0009:   7.61 MiB
cardOST000a:  15.90 MiB
cardOST000b:   7.57 MiB
cardOST000c:      357 B
cardOST000d:    844 KiB
cardOST000e:   7.56 MiB
cardOST000f:   7.59 MiB
cardOST0010:   8.50 MiB
cardOST0011:   7.31 MiB
cardOST0012:   7.59 MiB
cardOST0013:   6.84 MiB
cardOST0014:  10.00 MiB
cardOST0015:      242 B
cardOST0016:   7.55 MiB
cardOST0017:   6.93 MiB
cardOST0018:   7.56 MiB
cardOST0019:   9.10 MiB
cardOST001a:      9 KiB
cardOST001b:     34 KiB
cardOST001c:   7.56 MiB
cardOST001d:      142 B
cardOST001e:   7.55 MiB
cardOST001f:   7.56 MiB
cardOST0020:    968 KiB
cardOST0021:     35 KiB
cardOST0022:   7.56 MiB
cardOST0023:    956 KiB
```

MDS Inodes

	current
Total	0
Free	0

MDS Space

	current
Total	132.8 GiB
Free	120.0 GiB

Total	6.3
Free	6.2

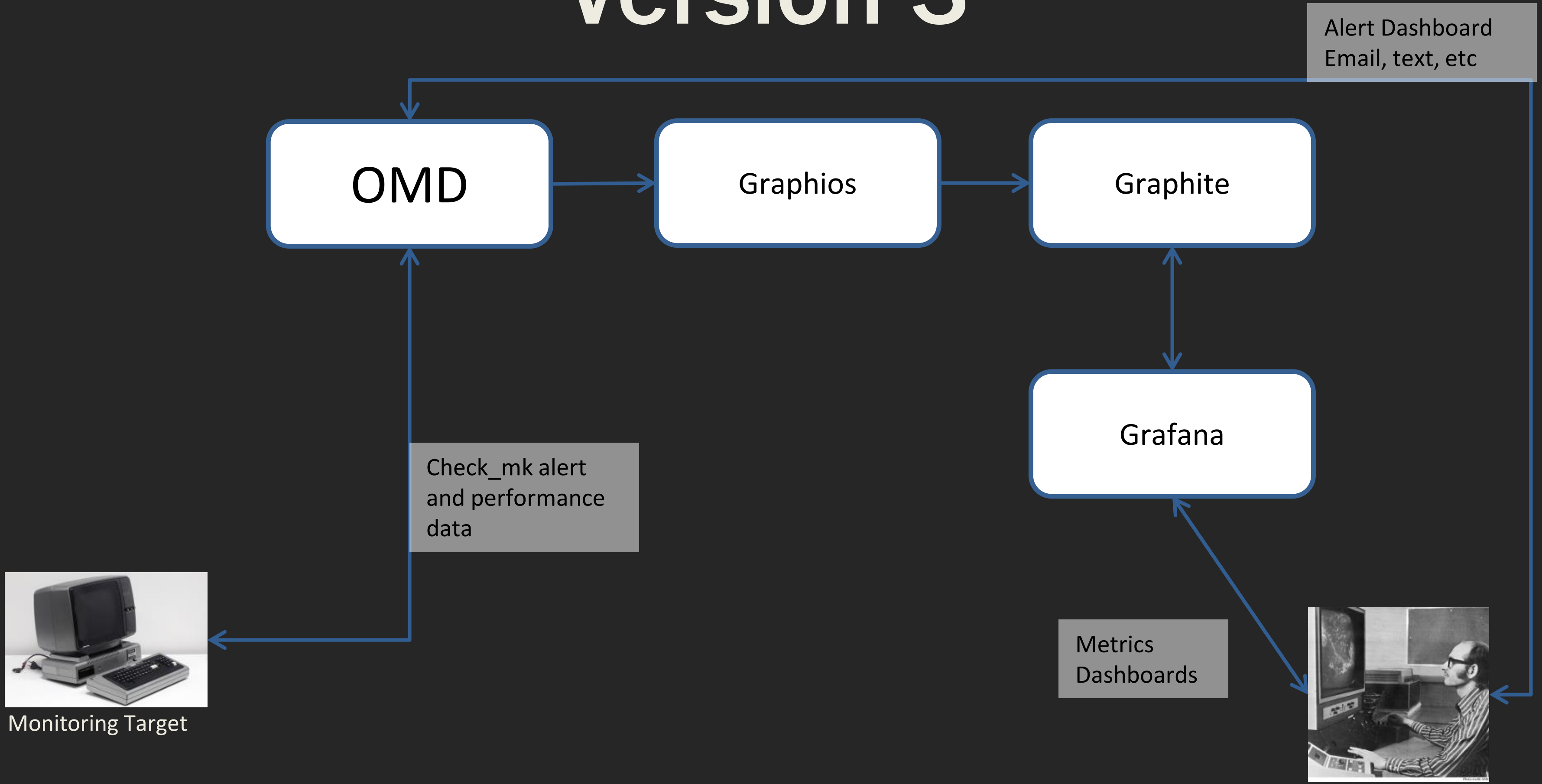
Non-Lustre System Metrics?

- We also care about MDS and OSS system metrics such as cpu load and memory.
- What can affect or help diagnose Lustre performance?

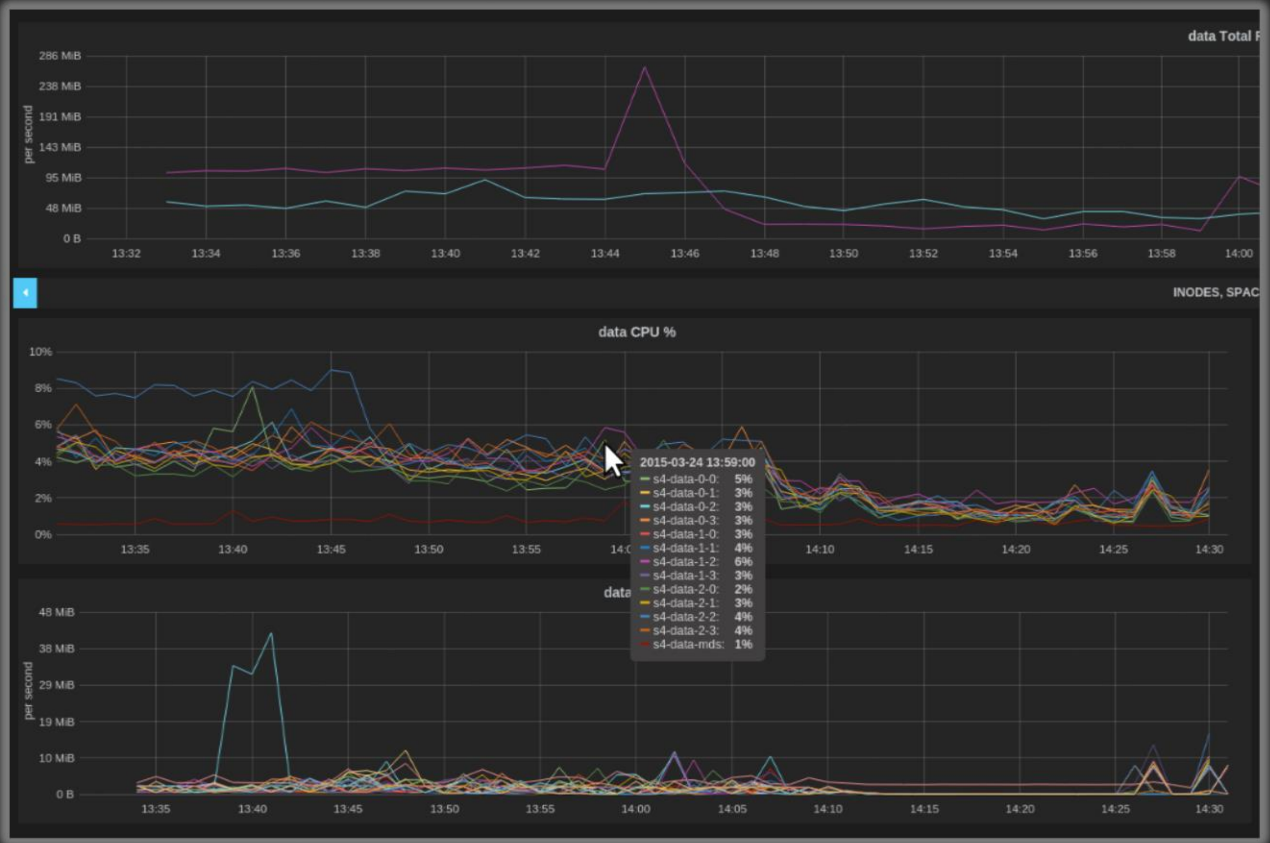
Version 3: Refactor Data Collection

- Change from standalone Perl scripts to Check_MK local checks
  - We already had Check_MK in place on hosts
  - Sample any metrics with a full featured system using Check_MK performance data
  - Alerts/notifications provided via Check_MK

Version 3


Alert Dashboard
Email, text, etc
OMD
Graphios
Graphite
Grafana
Check_mk alert and performance data
Metrics Dashboards
Monitoring Target

Version 3: Server CPU Metrics



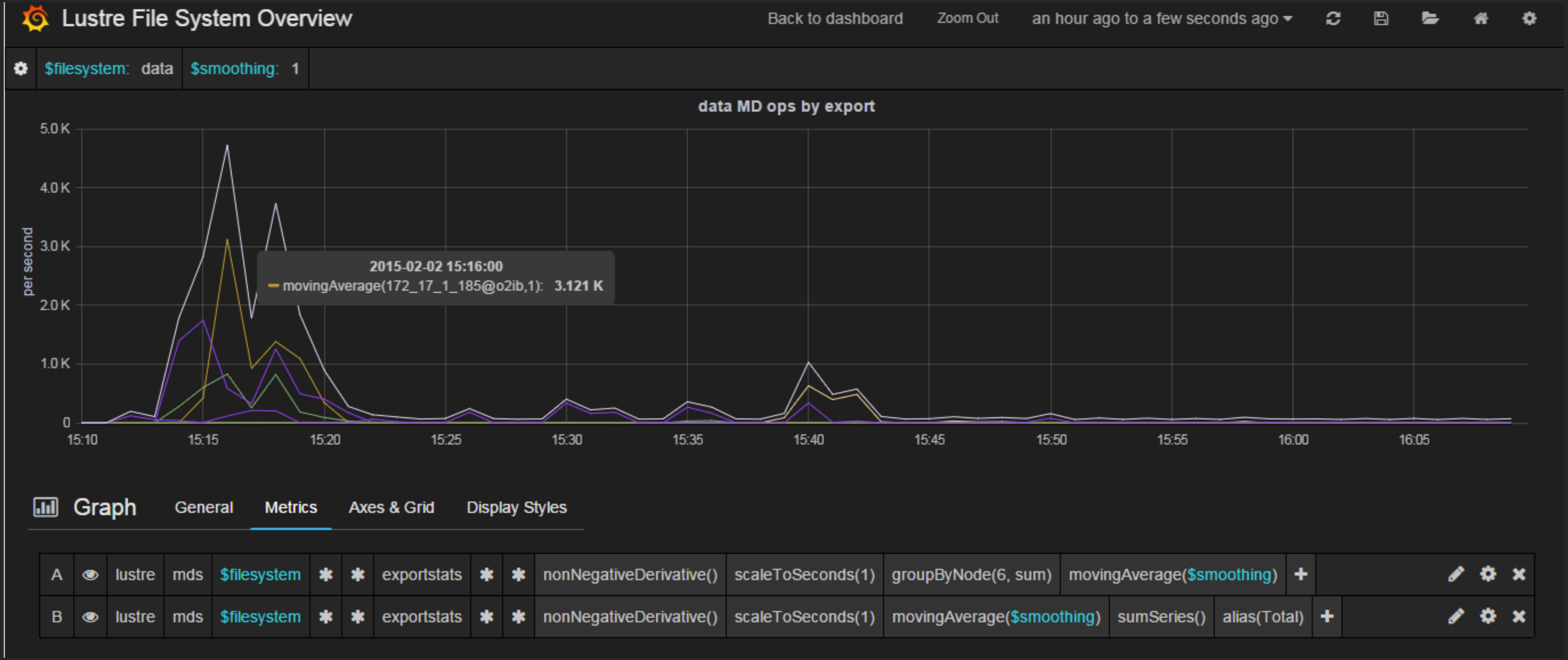
2015-03-24 13:59:00

s4-data-0-0:	5%
s4-data-0-1:	3%
s4-data-0-2:	3%
s4-data-0-3:	3%
s4-data-1-0:	3%
s4-data-1-1:	4%
s4-data-1-2:	6%
s4-data-1-3:	3%
s4-data-2-0:	2%
s4-data-2-1:	3%
s4-data-2-2:	4%
s4-data-2-3:	4%
s4-data-mds:	1%

Version 4: Add Export Stats

- Per-export stats
  - The exports tree lists client connections by NID. The exports are named by interfaces.
- MDS
  - lctl get_param mdt.*MDT*.exports.*@*.stats
- OSS
  - lctl get_param obdfilter.*OST*.exports.*@*.stats

Version 4: Add Export Stats



2015-02-02 15:16:00
movingAverage(172_17_1_185@o2ib,1): 3.121 K

A	lustre	mds	$filesystem	*	*	exportstats	*	*	nonNegativeDerivative()	scaleToSeconds(1)	groupByNode(6, sum)	movingAverage($smoothing)	
B	lustre	mds	$filesystem	*	*	exportstats	*	*	nonNegativeDerivative()	scaleToSeconds(1)	movingAverage($smoothing)	sumSeries()	alias(Total)

Sum of metadata operations per export. Each colored line = 1 export.

Version 5: Jobstats

- Just more of the same right?
  - lctl get_param mdt.*.job_stats

Jobstats are Different

- Every job will make a pile of metrics.
- Jobs are temporary and numerous.
- Storage problem and visualization performance issue.
    - System assumes all metrics are valid for all time ranges. So most operations mean: Open the whisper database file for every single job that has ever run.

Version 5: Coping with Jobstats

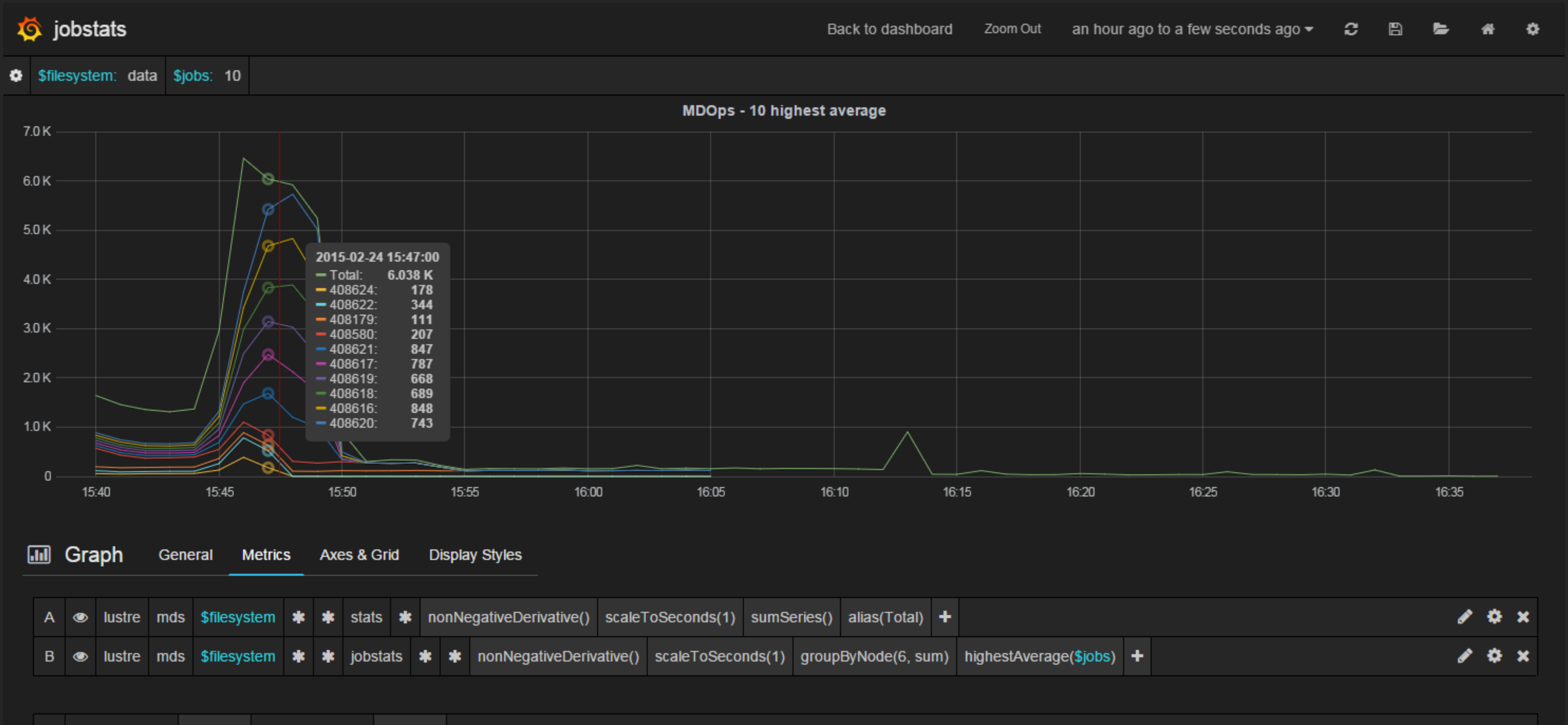
These ideas require actions outside of the normal system.

Stress Reduction Kit
Bang Head Here

Directions:
1. Place kit on FIRM surface.
2. Follow directions in circle of kit.
3. Repeat step 2 as necessary, or until unconscious.
4. If unconscious, cease stress reduction activity.

- Keep a smaller window of data
  - 'find and rm' cron job
- View only the subset of data you want
  - Use a script to find the start and stop time of a job, and build a custom url to a scripted Grafana dashboard.

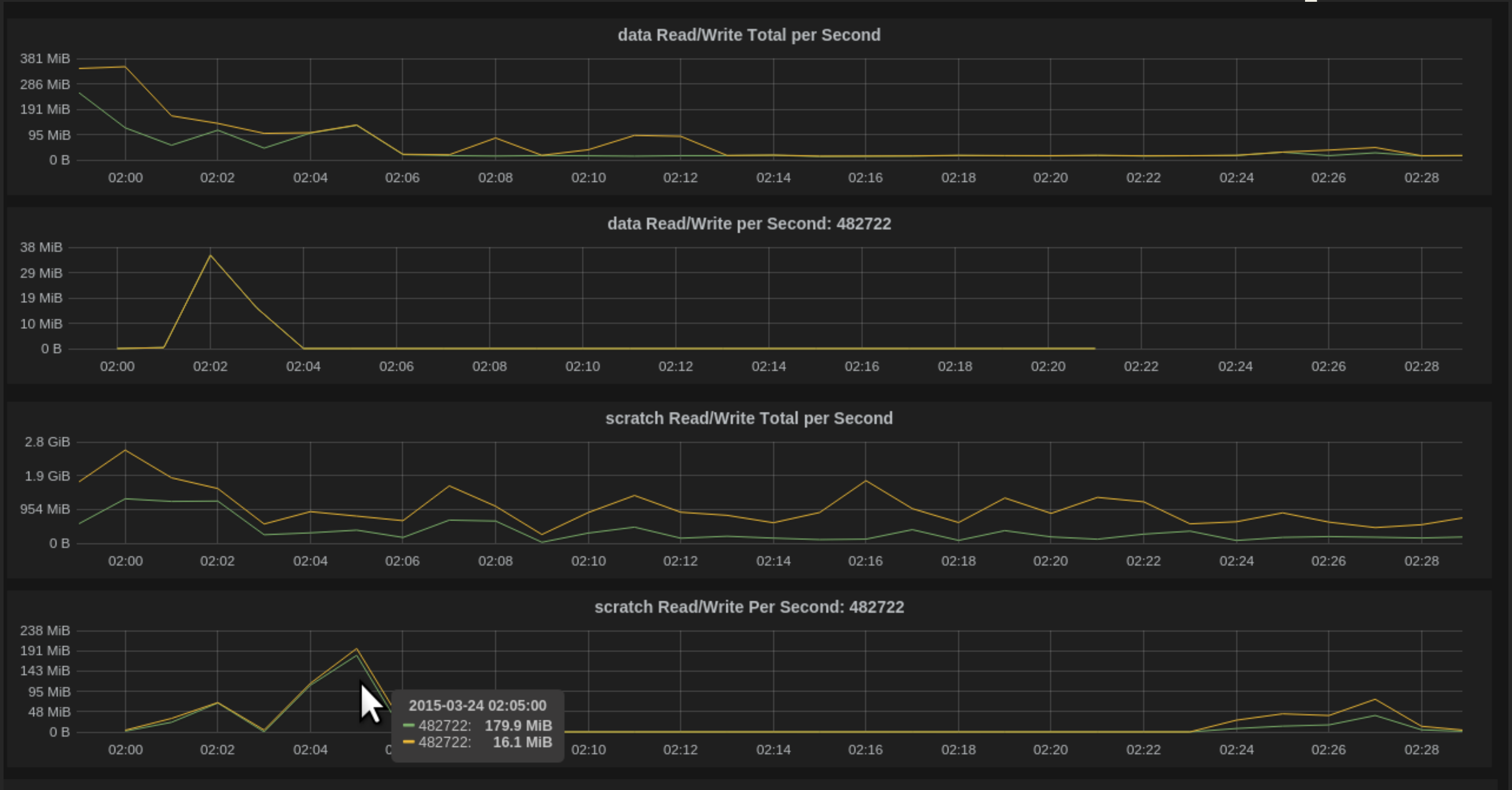
Version 5: Jobstats



```
2015-02-24 15:47:00
Total:    6.038 K
408624:       178
408622:       344
408179:       111
408580:       207
408621:       847
408617:       787
408619:       668
408618:       689
408616:       848
408620:       743
```

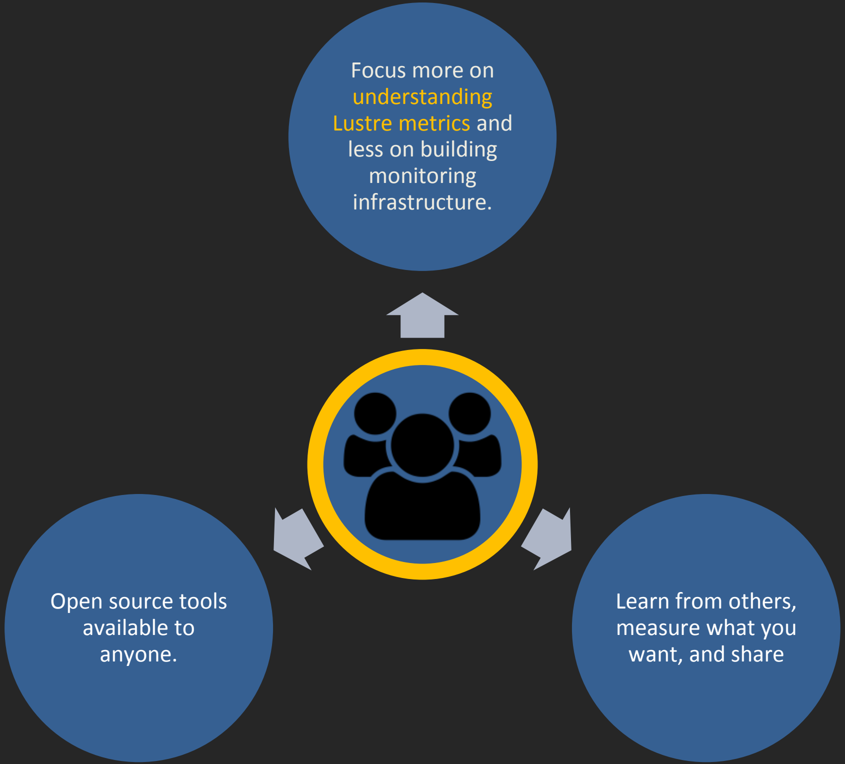
```
A  lustre mds $filesystem * * stats * nonNegativeDerivative() scaleToSeconds(1) sumSeries() alias(Total)
B  lustre mds $filesystem * * jobstats * * nonNegativeDerivative() scaleToSeconds(1) groupByNode(6, sum) highestAverage($jobs)
```

Job Data Selected via Script



2015-03-24 02:05:00
🟩 482722: **179.9 MiB**
🟨 482722: **16.1 MiB**

A Community Approach



Focus more on understanding Lustre metrics and less on building monitoring infrastructure.

Open source tools available to anyone.

Learn from others, measure what you want, and share

Community Documentation

- How about one community focused resource for Lustre stats information?
  - http://wiki.opensfs.org/Lustre_Monitoring_and_Statistics_Guide
  - http://wiki.opensfs.org/UW_SSEC_Lustre_Statistics_How-To

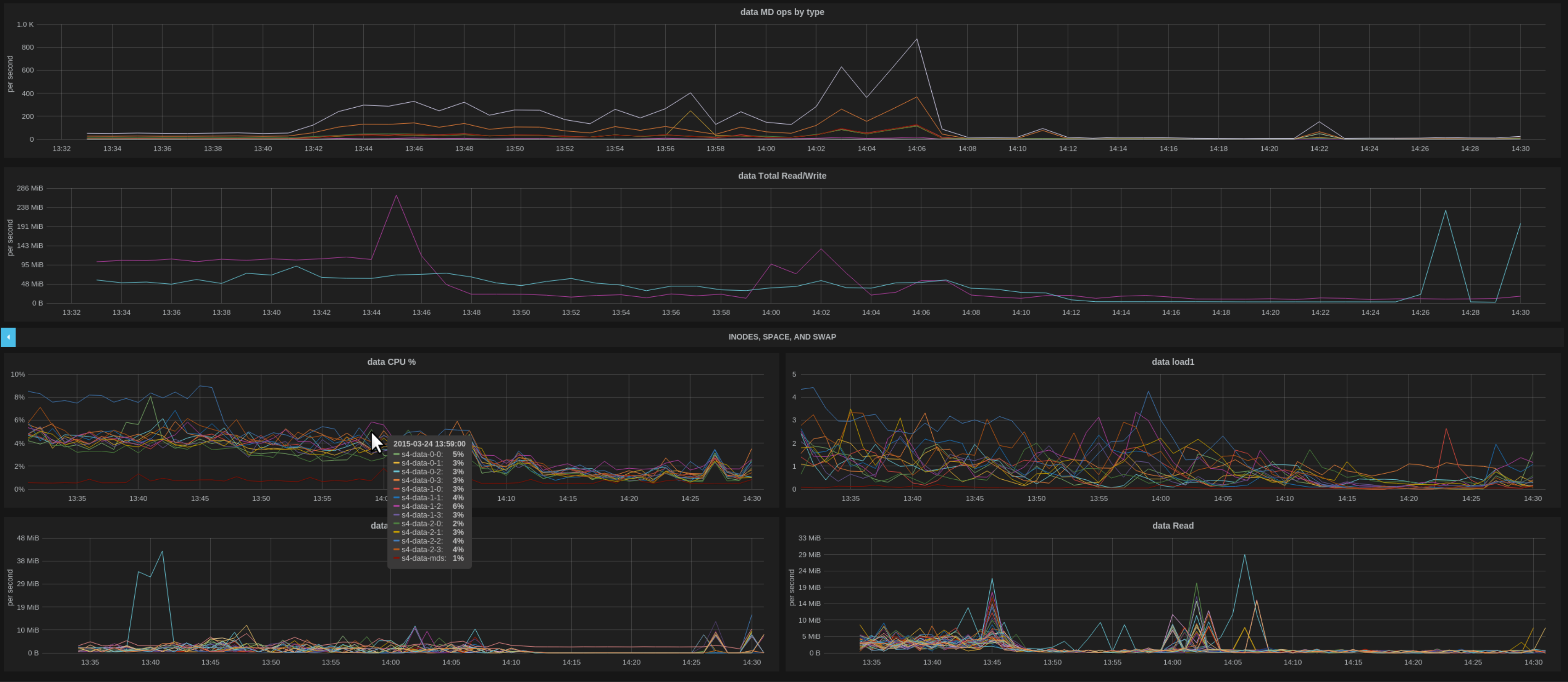
Conclusion

- New techniques and tools for monitoring enable a flexible and robust system for monitoring Lustre metrics.

Appendix

- The following slides are for reference or discussion

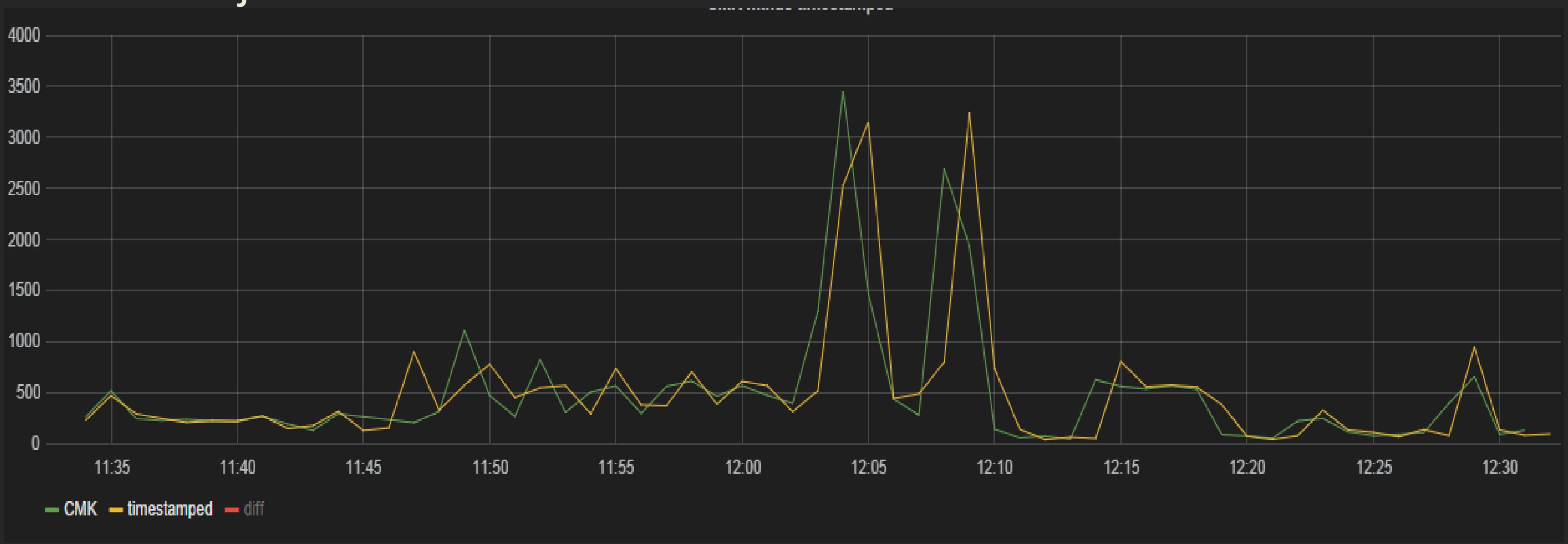
Case Study Version 2 Dashboard



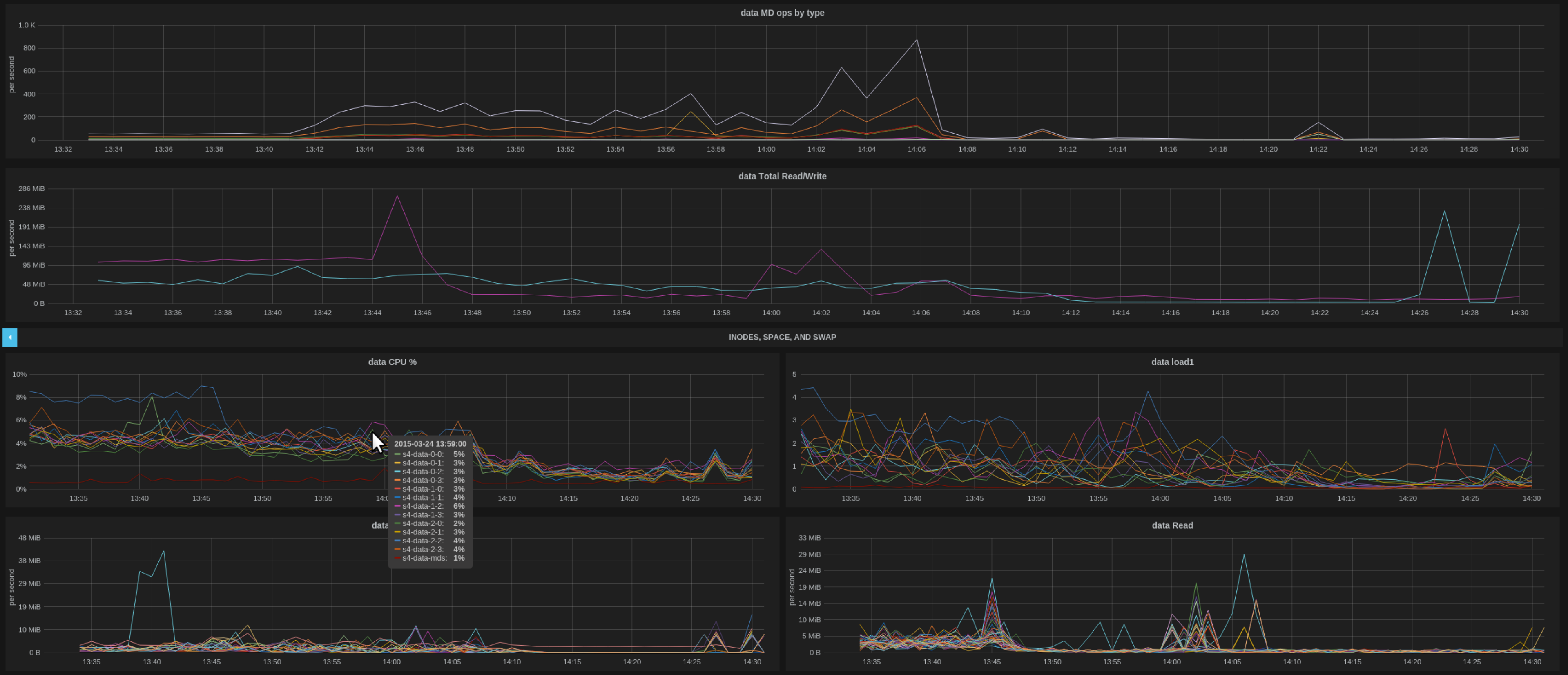
```
2015-03-24 13:59:00
s4-data-0-0:   5%
s4-data-0-1:   3%
s4-data-0-2:   3%
s4-data-0-3:   3%
s4-data-1-0:   3%
s4-data-1-1:   4%
s4-data-1-2:   6%
s4-data-1-3:   3%
s4-data-2-0:   2%
s4-data-2-1:   3%
s4-data-2-2:   4%
s4-data-2-3:   4%
s4-data-mds:   1%
```

Version 3: Refactor Data Collection

- Instead of sending exact timestamps from the stats file via perl, we now use timestamps based on when the check ran. This introduces some jitter.

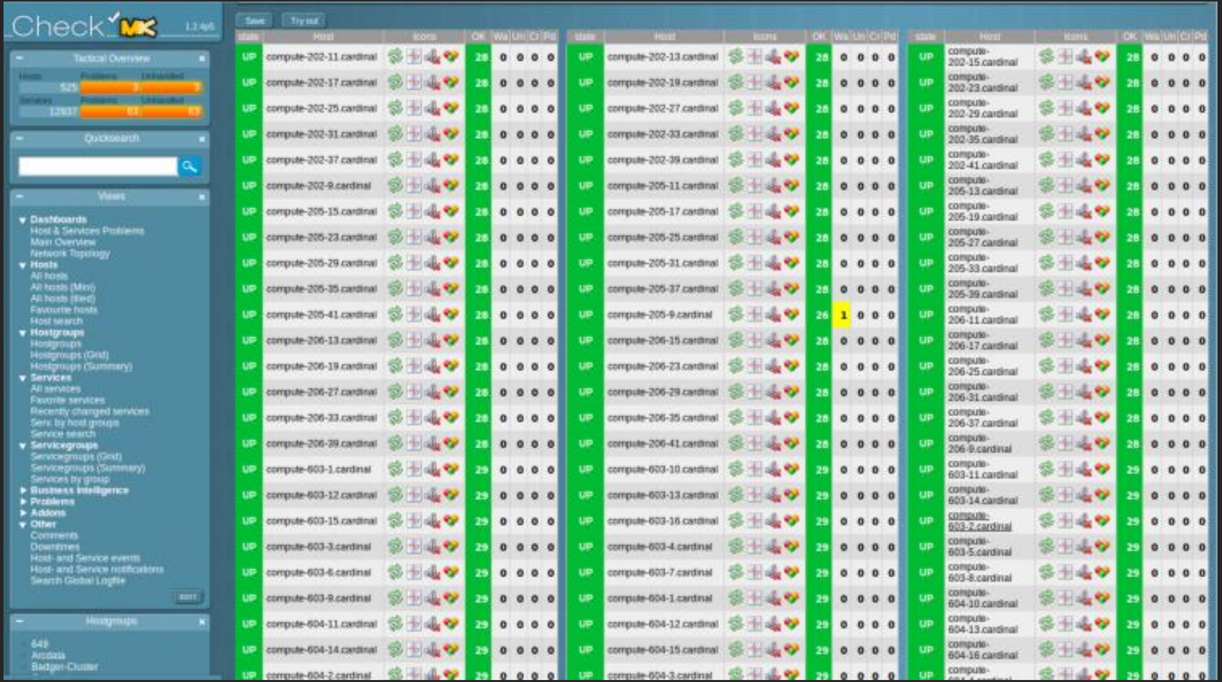


Case Study Version 3 Dashboard



```
2015-03-24 13:59:00
s4-data-0-0:   5%
s4-data-0-1:   3%
s4-data-0-2:   3%
s4-data-0-3:   3%
s4-data-1-0:   3%
s4-data-1-1:   4%
s4-data-1-2:   6%
s4-data-1-3:   3%
s4-data-2-0:   2%
s4-data-2-1:   3%
s4-data-2-2:   4%
s4-data-2-3:   4%
s4-data-mds:   1%
```

Version 3 Alert Dashboard



Tactical Overview

Hosts	Problems	Unhandled
525	3	3

Services	Problems	Unhandled
12837	63	63

Quicksearch

Views

- Dashboards
  - Host & Services Problems
  - Main Overview
  - Network Topology
- Hosts
  - All hosts
  - All hosts (Mini)
  - All hosts (tiled)
  - Favourite hosts
  - Host search
- Hostgroups
  - Hostgroups
  - Hostgroups (Grid)
  - Hostgroups (Summary)
- Services
  - All services
  - Favorite services
  - Recently changed services
  - Serv. by host groups
  - Service search
- Servicegroups
  - Servicegroups (Grid)
  - Servicegroups (Summary)
  - Services by group
- Business Intelligence
- Problems
- Addons
- Other
  - Comments
  - Downtimes
  - Host- and Service events
  - Host- and Service notifications
  - Search Global Logfile

EDIT

Hostgroups

- 649
- Arcdata
- Badger-Cluster

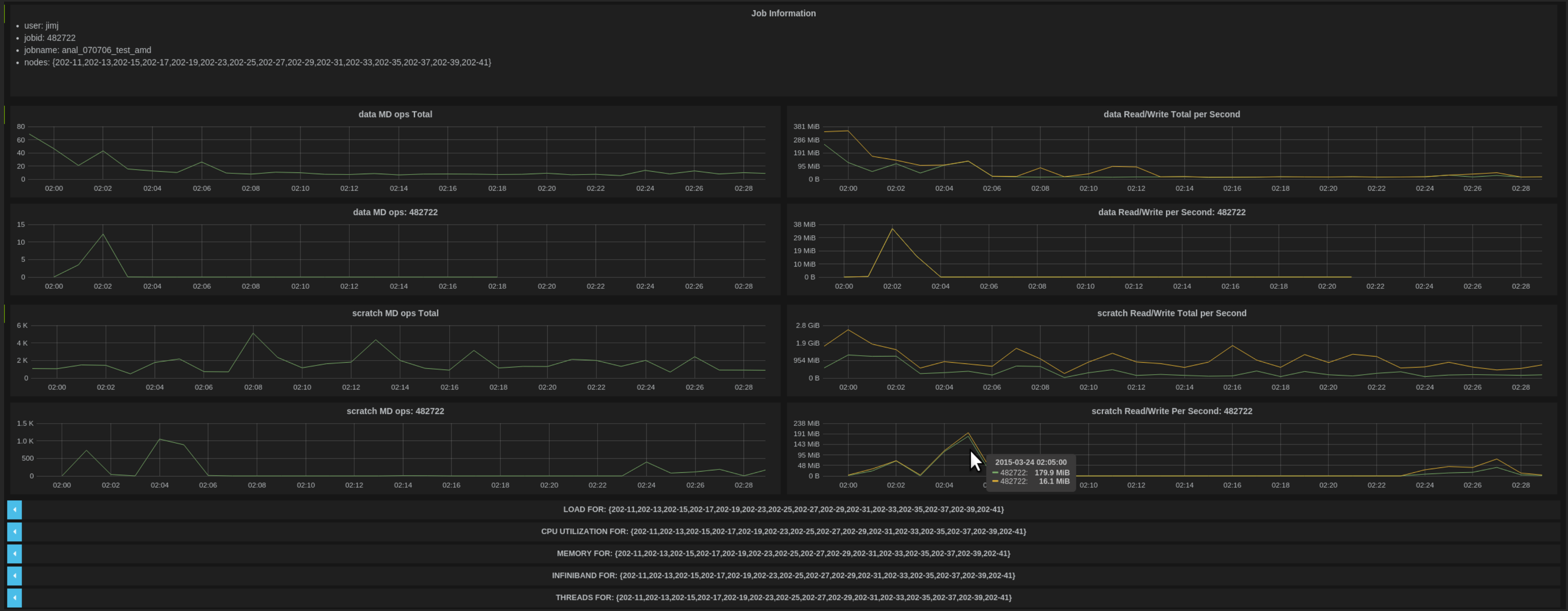
state	Host	Icons	OK	Wa	Un	Cr	Pd
UP	compute-202-11.cardinal		28	0	0	0	0
UP	compute-202-17.cardinal		28	0	0	0	0
UP	compute-202-25.cardinal		28	0	0	0	0
UP	compute-202-31.cardinal		28	0	0	0	0
UP	compute-202-37.cardinal		28	0	0	0	0
UP	compute-202-9.cardinal		28	0	0	0	0
UP	compute-205-15.cardinal		28	0	0	0	0
UP	compute-205-23.cardinal		28	0	0	0	0
UP	compute-205-29.cardinal		28	0	0	0	0
UP	compute-205-35.cardinal		28	0	0	0	0
UP	compute-205-41.cardinal		28	0	0	0	0
UP	compute-206-13.cardinal		28	0	0	0	0
UP	compute-206-19.cardinal		28	0	0	0	0
UP	compute-206-27.cardinal		28	0	0	0	0
UP	compute-206-33.cardinal		28	0	0	0	0
UP	compute-206-39.cardinal		28	0	0	0	0
UP	compute-603-1.cardinal		29	0	0	0	0
UP	compute-603-12.cardinal		29	0	0	0	0
UP	compute-603-15.cardinal		29	0	0	0	0
UP	compute-603-3.cardinal		29	0	0	0	0
UP	compute-603-6.cardinal		29	0	0	0	0
UP	compute-603-9.cardinal		29	0	0	0	0
UP	compute-604-11.cardinal		29	0	0	0	0
UP	compute-604-14.cardinal		29	0	0	0	0
UP	compute-604-2.cardinal		29	0	0	0	0

state	Host	Icons	OK	Wa	Un	Cr	Pd
UP	compute-202-13.cardinal		28	0	0	0	0
UP	compute-202-19.cardinal		28	0	0	0	0
UP	compute-202-27.cardinal		28	0	0	0	0
UP	compute-202-33.cardinal		28	0	0	0	0
UP	compute-202-39.cardinal		28	0	0	0	0
UP	compute-205-11.cardinal		28	0	0	0	0
UP	compute-205-17.cardinal		28	0	0	0	0
UP	compute-205-25.cardinal		28	0	0	0	0
UP	compute-205-31.cardinal		28	0	0	0	0
UP	compute-205-37.cardinal		28	0	0	0	0
UP	compute-205-9.cardinal		26	1	0	0	0
UP	compute-206-15.cardinal		28	0	0	0	0
UP	compute-206-23.cardinal		28	0	0	0	0
UP	compute-206-29.cardinal		28	0	0	0	0
UP	compute-206-35.cardinal		28	0	0	0	0
UP	compute-206-41.cardinal		28	0	0	0	0
UP	compute-603-10.cardinal		29	0	0	0	0
UP	compute-603-13.cardinal		29	0	0	0	0
UP	compute-603-16.cardinal		29	0	0	0	0
UP	compute-603-4.cardinal		29	0	0	0	0
UP	compute-603-7.cardinal		29	0	0	0	0
UP	compute-604-1.cardinal		29	0	0	0	0
UP	compute-604-12.cardinal		29	0	0	0	0
UP	compute-604-15.cardinal		29	0	0	0	0
UP	compute-604-3.cardinal		29	0	0	0	0

state	Host	Icons	OK	Wa	Un	Cr	Pd
UP	compute-202-15.cardinal		28	0	0	0	0
UP	compute-202-23.cardinal		28	0	0	0	0
UP	compute-202-29.cardinal		28	0	0	0	0
UP	compute-202-35.cardinal		28	0	0	0	0
UP	compute-202-41.cardinal		28	0	0	0	0
UP	compute-205-13.cardinal		28	0	0	0	0
UP	compute-205-19.cardinal		28	0	0	0	0
UP	compute-205-27.cardinal		28	0	0	0	0
UP	compute-205-33.cardinal		28	0	0	0	0
UP	compute-205-39.cardinal		28	0	0	0	0
UP	compute-206-11.cardinal		28	0	0	0	0
UP	compute-206-17.cardinal		28	0	0	0	0
UP	compute-206-25.cardinal		28	0	0	0	0
UP	compute-206-31.cardinal		28	0	0	0	0
UP	compute-206-37.cardinal		28	0	0	0	0
UP	compute-206-9.cardinal		28	0	0	0	0
UP	compute-603-11.cardinal		29	0	0	0	0
UP	compute-603-14.cardinal		29	0	0	0	0
UP	compute-603-2.cardinal		29	0	0	0	0
UP	compute-603-5.cardinal		29	0	0	0	0
UP	compute-603-8.cardinal		29	0	0	0	0
UP	compute-604-10.cardinal		29	0	0	0	0
UP	compute-604-13.cardinal		29	0	0	0	0
UP	compute-604-16.cardinal		29	0	0	0	0
UP	compute-		29	0	0	0	0

We were already using Check_MK for routine host alerts

Job Data Selected via Script



Job Information

- user: jimj
- jobid: 482722
- jobname: anal_070706_test_amd
- nodes: {202-11,202-13,202-15,202-17,202-19,202-23,202-25,202-27,202-29,202-31,202-33,202-35,202-37,202-39,202-41}

data MD ops Total

data Read/Write Total per Second

data MD ops: 482722

data Read/Write per Second: 482722

scratch MD ops Total

scratch Read/Write Total per Second

scratch MD ops: 482722

scratch Read/Write Per Second: 482722

LOAD FOR: {202-11,202-13,202-15,202-17,202-19,202-23,202-25,202-27,202-29,202-31,202-33,202-35,202-37,202-39,202-41}

CPU UTILIZATION FOR: {202-11,202-13,202-15,202-17,202-19,202-23,202-25,202-27,202-29,202-31,202-33,202-35,202-37,202-39,202-41}

MEMORY FOR: {202-11,202-13,202-15,202-17,202-19,202-23,202-25,202-27,202-29,202-31,202-33,202-35,202-37,202-39,202-41}

INFINIBAND FOR: {202-11,202-13,202-15,202-17,202-19,202-23,202-25,202-27,202-29,202-31,202-33,202-35,202-37,202-39,202-41}

THREADS FOR: {202-11,202-13,202-15,202-17,202-19,202-23,202-25,202-27,202-29,202-31,202-33,202-35,202-37,202-39,202-41}