



Monitoring GPFS with Grafana and InfluxDB

Aaron Knister
HPC System Administrator
NASA Center for Climate Simulation
Goddard Space Flight Center

Overview



Discover Supercomputer Highlights

- **~3600 Nodes**
 - ~3.5PF Peak Performance
 - 3400+ Compute Nodes
 - 60 GPFS I/O Nodes (NSD Servers)
 - 35 Interactive Login Nodes
 - 10 Gateway (Data Mover) Nodes
 - 100+ Miscellaneous Service/Support Nodes
 - 3x (Mostly) Non-Blocking Fat Tree InfiniBand fabrics
 - » 1x QDR, 2x FDR
- **~33PB Usable GPFS Storage (45PB raw)**
 - DDN SFA12K, DDN S2A9900, SGI IS5500 (NetApp E5400), Cisco Whiptail UCS

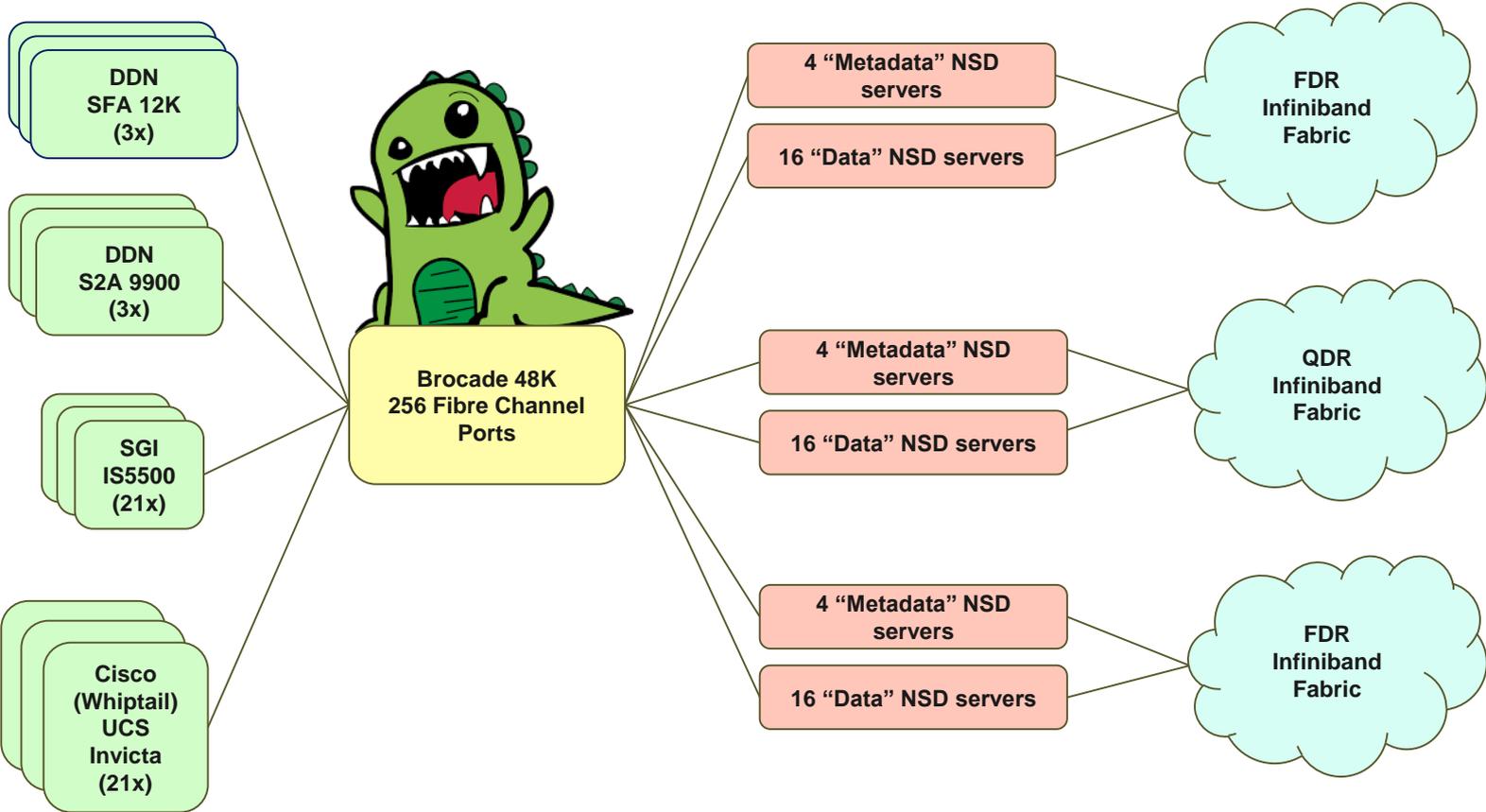




Discover GPFS Highlights

- **GPFS 3.5.0.31 (4.1 coming soon)**
- **60 “I/O” Nodes (NSD Servers)**
 - 20 Per- InfiniBand Fabric
 - » 4 “MDS” Nodes Per Fabric
 - » 16 “NSD” Nodes Per Fabric
- **3500+ GPFS clients**
- **25 Filesystems**
- **500M inodes**
- **All SSD Metadata**
 - Observed sustained 200K IOPS in GPFS (mmillenniumfac- recursive getfacl()/setfacl())
 - Hardware can go up to 1.2M IOPS
- **88GB/s aggregate data NSD read bandwidth**

Discover GPFS Architecture



Sure, blame the filesystem.



- **Desire for high resolution real time and historical filesystem performance data**
- **Goal was 10s resolution**
- **Quickly realized magnitude of the problem**
 - 8,664 LUN Presentations (1,44 NSDs, 6 Presentations)
 - For each of reads and writes, per-presentation
 - » 5 Attributes/Tags (NSD, Server, /dev/*, Filesystem, Pool)
 - » 5 Metrics/Measurements
(Operations, Bytes, Total/Min/Max Wait Time)
- **Need to store 173,280 fields (measurements + tags) every 10s!**
 - **Also need to get those fields from GPFS**

“A supercomputer is a device for turning compute-bound problems into I/O-bound problems.”
Ken Batcher

We have ways of making you talk



- **How to get this data out of GPFS?**

- **Tried the SNMP agent**

- But it exploded
- (No admins or servers were injured)
- Overwhelmed a static buffer

- **The solution? mmpmon**

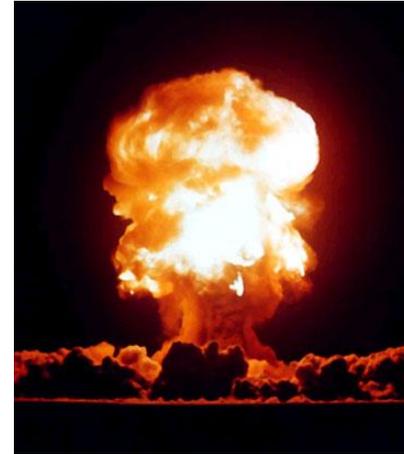
```
nlist new borgnsd01 borgnsd02 borgnsd03
```

```
nlist add borgnsd04 borgnsd05 borgnsd06
```

```
nsd_ds
```

- **But how do I get NSD pool and filesystem information?**

- Scraping mmlsnsd seems kludgy
- No -Y option
- `mmsdrquery sdrq_nsd_info sdrq_nsd_name:sdrq_fs_name:sdrq_storage_pool all norefresh`





Where do I put all this data?

- **Time Series Database (TSDB)**

- Optimized for storing colossal volumes of data indexed by time
- Many have built-in aggregation queries
- Great open source visualization tools

- **Using InfluxDB 1.0**

- Examined others, InfluxDB appeared to be simplest to install and maintain
- Some growing pains
- Intend to explore others in the future

- **Sample insert record**

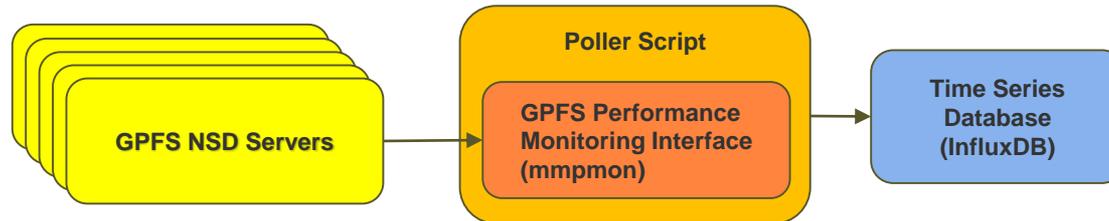
- `write_stats,nsd=d10_10_010,nsd_server=borgnsd01 bytes=19384 ops=31245 1478914388`





Putting it all together

- **Perl poller script fires mmpmon every 10 seconds**
- **Data enqueued to internal queue**
- **Separate thread batch-feeds writes (1000 inserts per batch) to InfluxDB**
- **Database lives on ZFS filesystem**
 - 4 disk RAIDZ2
 - 2x SSDs for log and cache

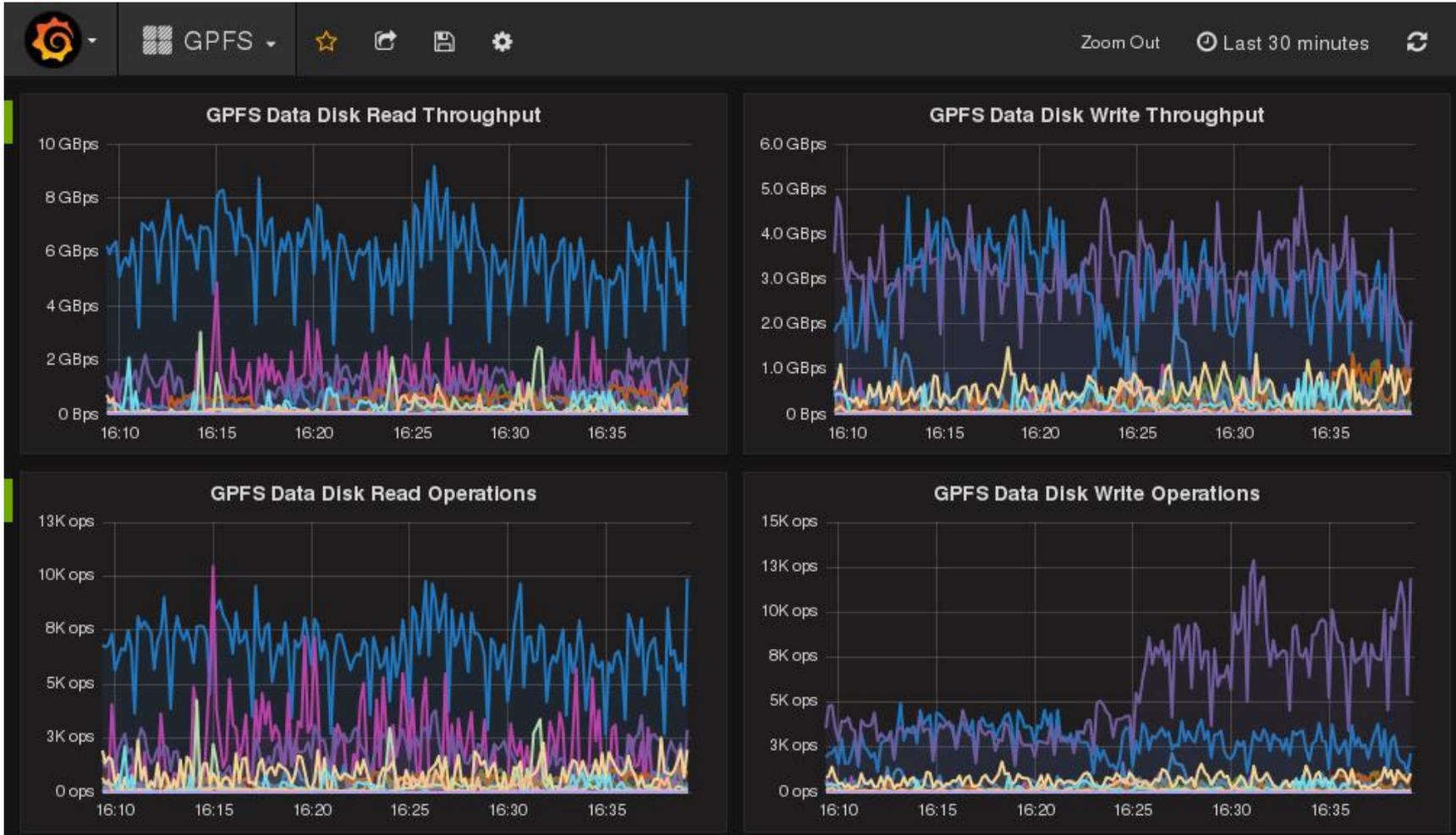


How do I get the data out?

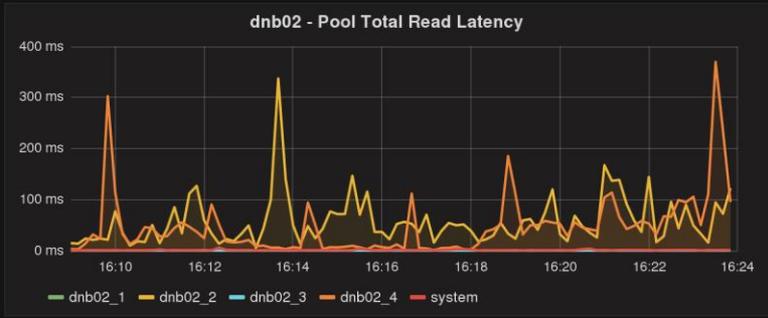
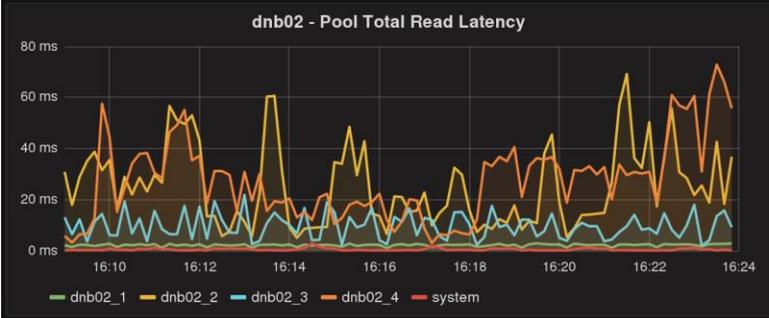
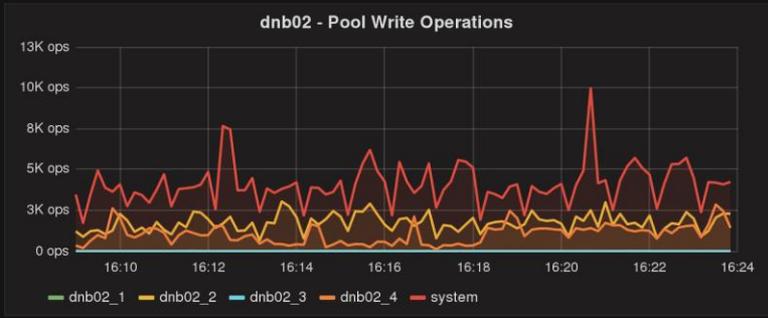
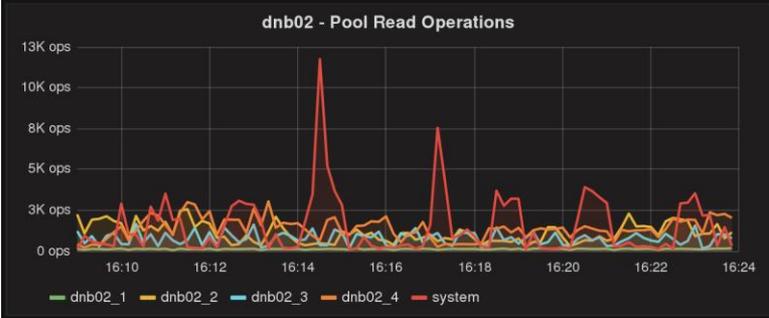
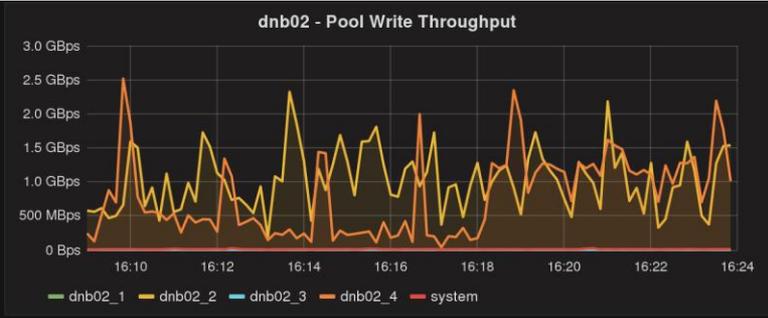
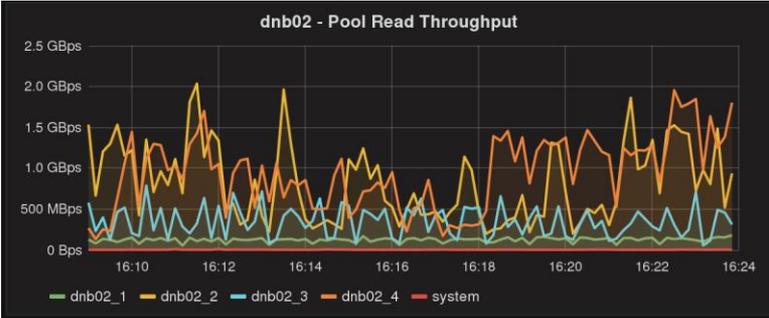


- **CLI**
 - Queries via InfluxQL
- **GUI**
 - Grafana

Grafana Dashboard



Single filesystem dashboard



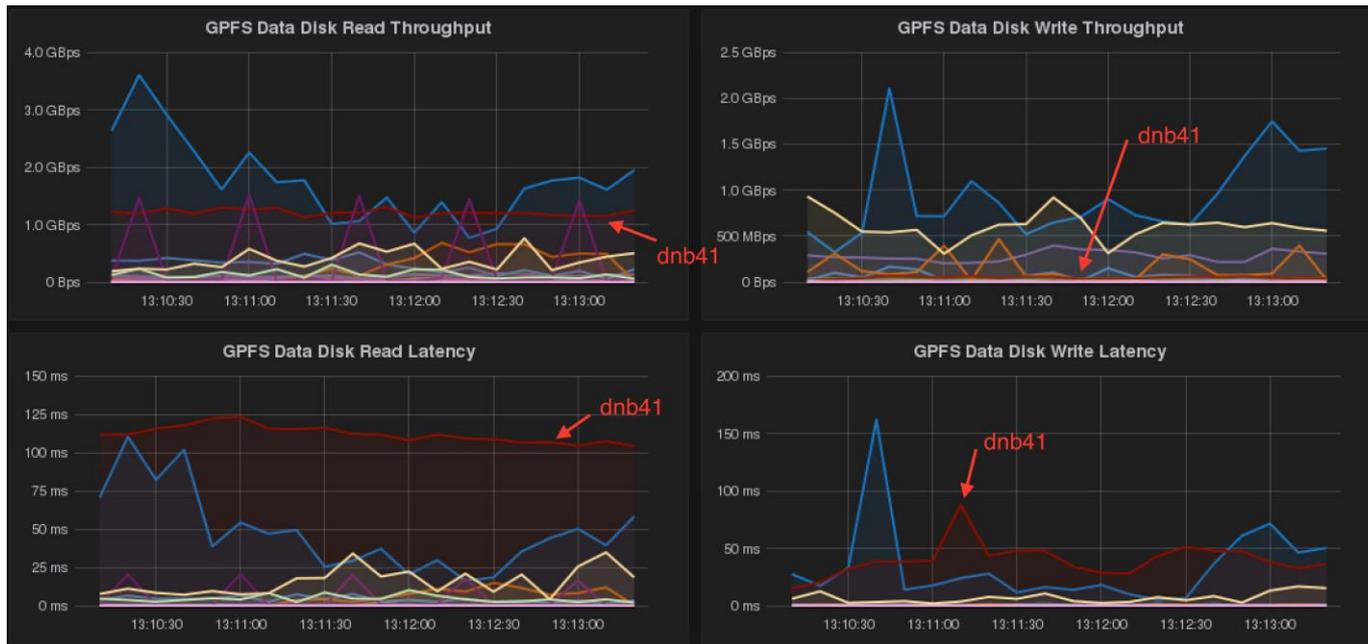
Real-World Examples





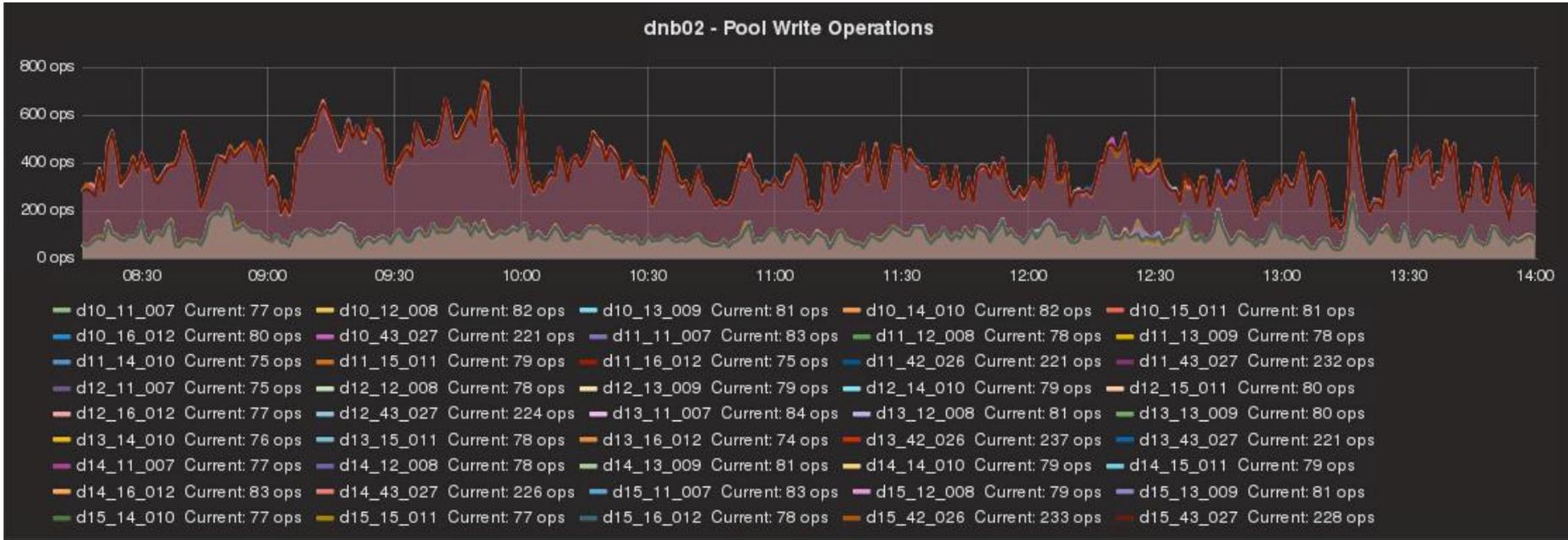
Troubleshooting slow job

- Reports of slow filesystem performance starting several days back on “dnb41” filesystem



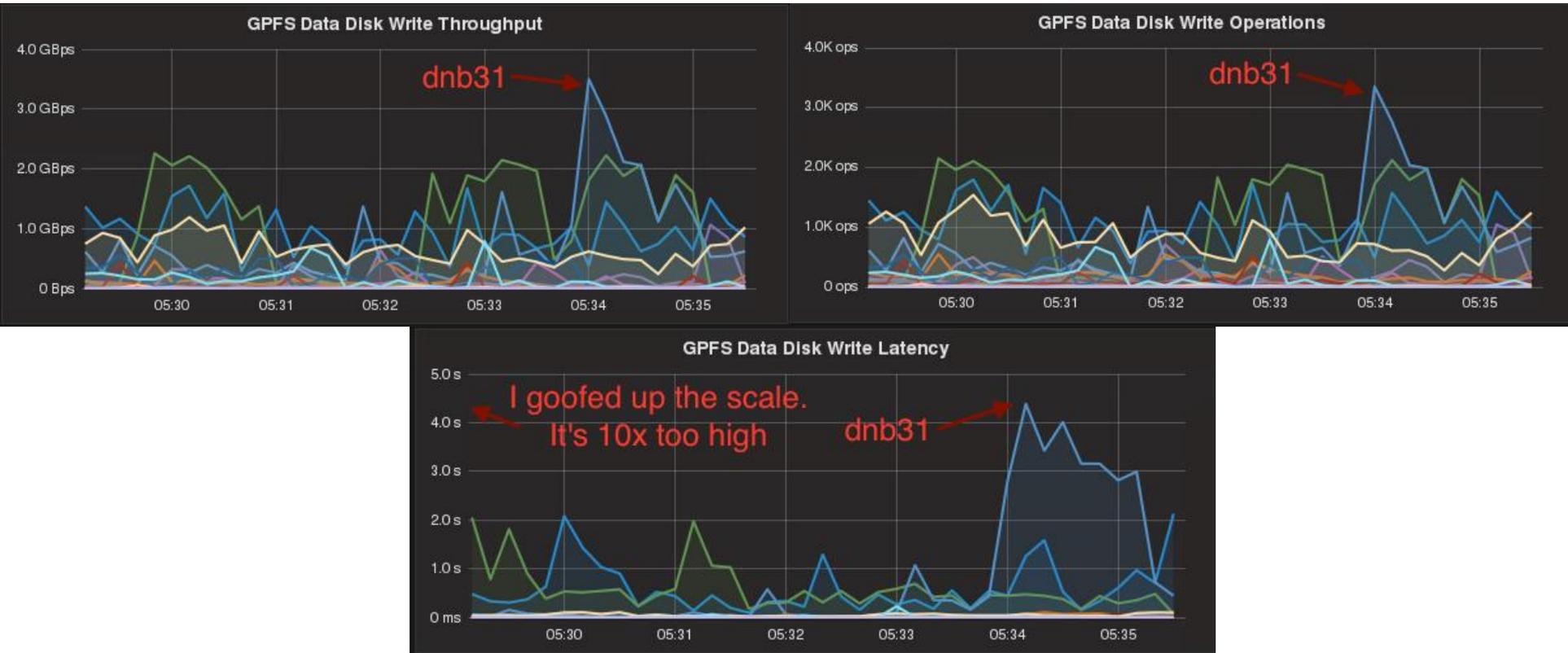
- High disk latency identified as cause stemming from disk contention

High NSD server load



- **Uneven load distribution caused by newly added disks taking brunt of I/O load due to differences in free space**

Benefits of high resolution



Future Work

- **AMQP-based transport between poller and database**
 - Allows other databases to be tried without re-writing poller code
- **Node-based performance aggregation (from private mmpmon gfis interface)**
 - Slightly daunting
 - 17 Metrics per-filesystem
 - 25 Filesystem
 - 3500 Nodes
 - 1 Million data points!
 - Actually ~1.5 Million
- **Effort to open source poller code**



Any Questions?

