RED HAT SUMMIT

Ceph and OpenStack at Scale

Ben England, Red Hat,
Performance & Scale Engineering

Jared King, Cloud Operations Engineering
Cisco

4/12/2017 (v4)

2-Part Session

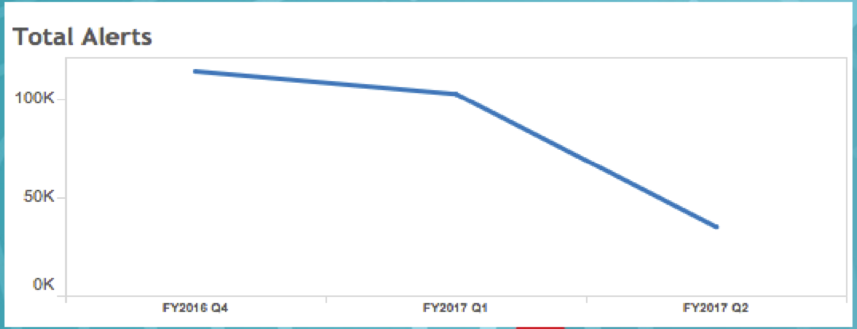
- Real-world experiences with Ceph Hammer (RHCS 1.3) – Jared King, Cisco
- Lab results with Ceph Jewel – Ben England, Red Hat
  - RHCS 2.0 external cluster
  - RHOSP 10.0 (OpenStack Newton)

Over to Jared King - Part 1

Ceph problems we encountered at scale

1. Latency spikes
2. Hard to isolate a hardware failure
3. Host failure

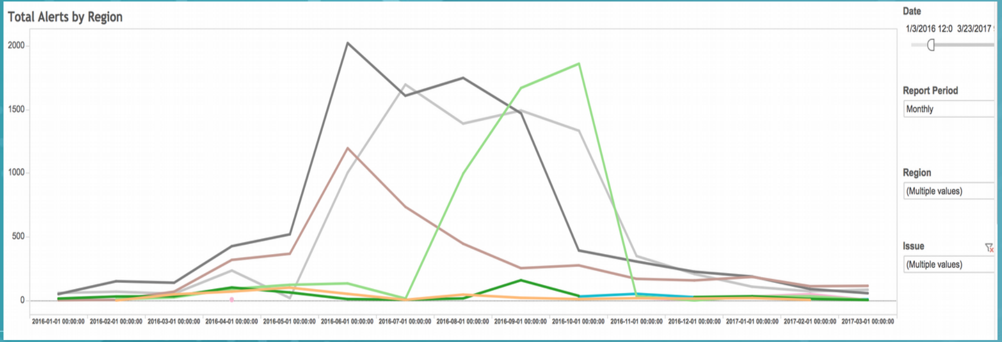
8 regions
Largest region has about 750 nova nodes


Total Alerts

8 ceph clusters
28,838 total alerts for ceph from 01-01-2016 to 03-27-2017

Largest Cluster
750 osds across 75 servers
NVME for journals
5 Mons

Highest quarter for just ceph alerts we responded to ~5k total


Total Alerts by Region

Date
1/3/2016 12:0 3/23/2017

Report Period
Monthly

Region
(Multiple values)

Issue
(Multiple values)

Disk (RBD) IO throttling in QEMU using virsh-throttle-rbd.py

Throttle tenant instances so that there is fairness for all shared storage

What can be throttled in QEMU ?

iops_wr	(IOPS Write) 250
iops_rd	(IOPS Read) 250
iops	(Total IOPS)
bps_wr	(Bits per second Write Max)
bpc_rd	(Bits per second read)
bps	(Total Bits per second)
iops_max	(Burst IOPS) 500
bps _max	(Burst Bps)120 MB/s

Run on each compute node every 5 minutes and log it.

```
*/5 * * * * python /var/nfv/virsh-throttle-rbd.py --bps 104857600 --iops 250 --doit >> /var/nfv/logs/throttle_io-$(date +\%d-\%m-\%Y).log 2>&1
```

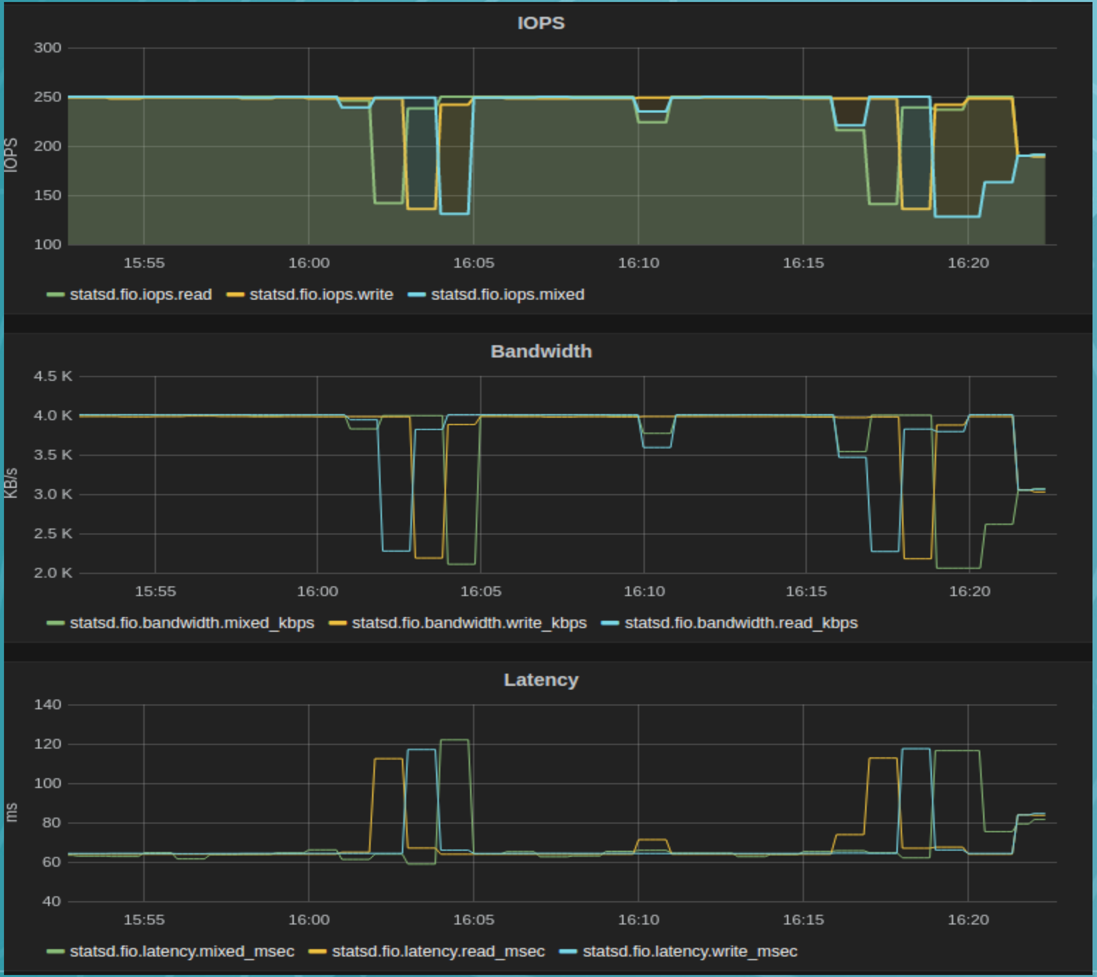
https://github.com/cernceph/ceph-scripts/blob/master/tools/virsh-throttle-rbd.py

Sample logs from virsh-throttle-rbd.py

```
Checking instance-0034e2c7 ...
        drive-virtio-disk0 is RBD, current thottle: 250/104857600 ... All good :)
Checking instance-0034e3b7 ...
        drive-virtio-disk0 is RBD, current thottle: 0/0 ...
virsh qemu-monitor-command instance-0034e3b7
  --cmd  '{"execute":"block_set_io_throttle","arguments":
     {"device":"drive-virtio-disk0","iops":250,"iops_rd":0,"iops_wr":0,"bps":104857600,"bps_rd":0,"bps_wr":0}}’
{"return":{},"id":"libvirt-13"}
```

FIO Tests

Used to catch any instance I/O issues, these feed into nagios and alert if there are any problems.



FIO Tests

The script invokes three fio operations that are executed sequentially.

1. read
2. write
3. mixed

Each fio runs for 30 seconds. These are the parameters for the read fio. For write, the value for the rw parameter is randwrite. The rw's value for the mixed fio is randrw.

```
[global]
runtime=30
direct=1
bs=16k
randrepeat=0
rw=randread
iodepth=2
numjobs=4
cpus_allowed=0-3
cpus_allowed_policy=split
group_reporting
[F1552]
ioengine=libaio
filename=/root/fio-test
size=1024M
```

Ceph log parsers

Not all issues are obvious by looking just at ceph.log or osd logs.

Issue showed up as intermittent blockedio

- No single osd seemed to be the cause

- Not able to pin point a single host

- Random blocked request in tenant console logs
  INFO: task jbd2/vda1-8:231 blocked for more than 120 seconds.

- Much more accurate than just pulling slow requests from ceph.log

```
grep -i slow /var/log/ceph/ceph.log | \
  awk {'print $3'} | sort -n | uniq -c | sort -rn
```

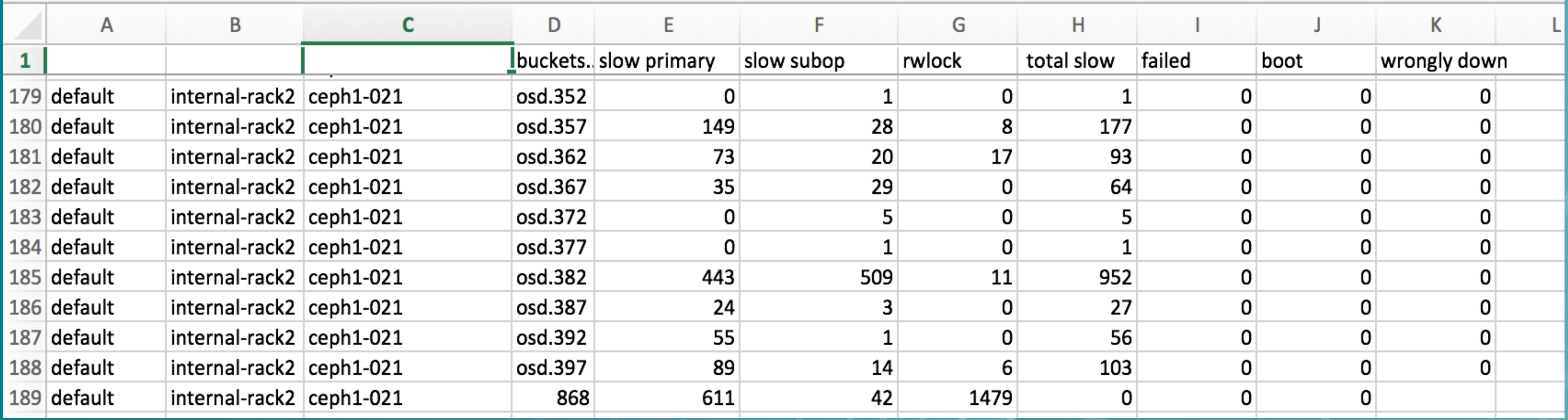
https://github.com/linuxkidd/ceph-log-parsers

To run:

map_events_to_buckets.sh /var/log/ceph/ceph.log

First run

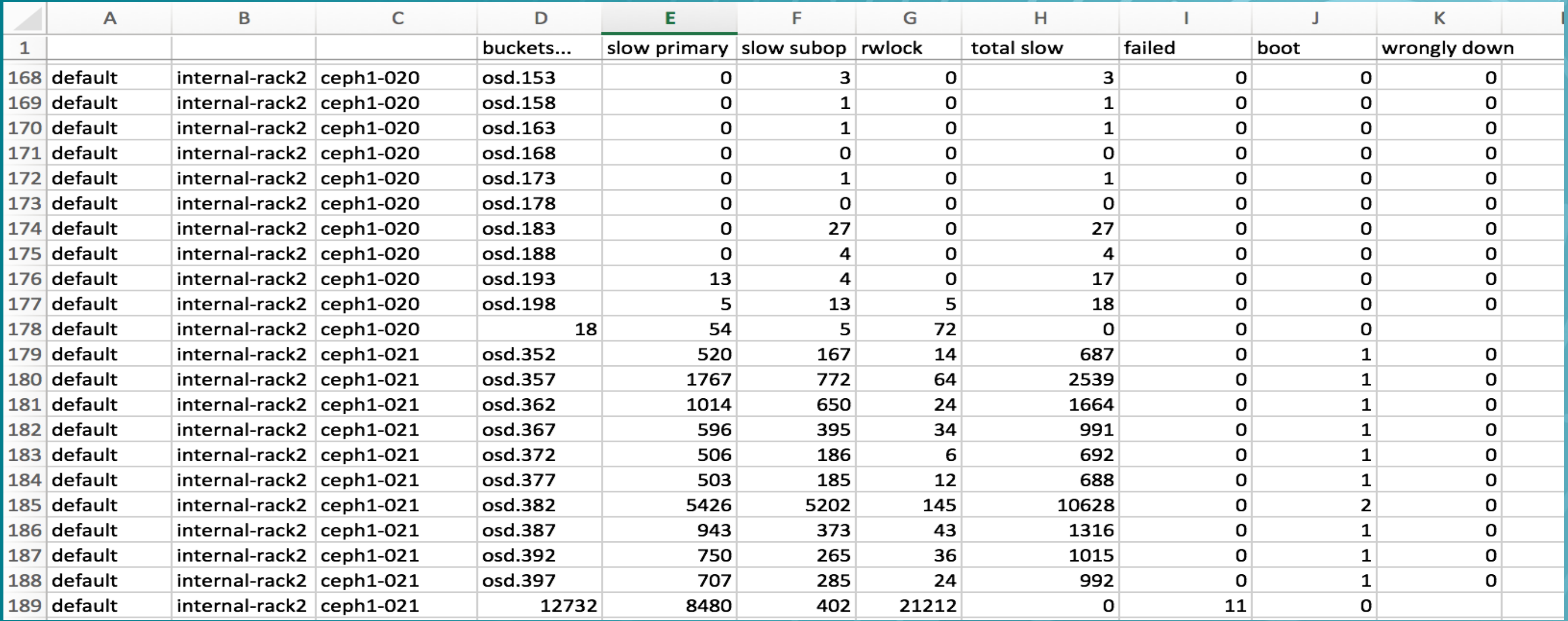
osd.382 is the top offender, but nothing else on 021 looks bad.



	A	B	C	D	E	F	G	H	I	J	K	L
1				buckets..	slow primary	slow subop	rwlock	total slow	failed	boot	wrongly down	
179	default	internal-rack2	ceph1-021	osd.352	0	1	0	1	0	0	0	
180	default	internal-rack2	ceph1-021	osd.357	149	28	8	177	0	0	0	
181	default	internal-rack2	ceph1-021	osd.362	73	20	17	93	0	0	0	
182	default	internal-rack2	ceph1-021	osd.367	35	29	0	64	0	0	0	
183	default	internal-rack2	ceph1-021	osd.372	0	5	0	5	0	0	0	
184	default	internal-rack2	ceph1-021	osd.377	0	1	0	1	0	0	0	
185	default	internal-rack2	ceph1-021	osd.382	443	509	11	952	0	0	0	
186	default	internal-rack2	ceph1-021	osd.387	24	3	0	27	0	0	0	
187	default	internal-rack2	ceph1-021	osd.392	55	1	0	56	0	0	0	
188	default	internal-rack2	ceph1-021	osd.397	89	14	6	103	0	0	0	
189	default	internal-rack2	ceph1-021	868	611	42	1479	0	0	0		

After the reboot of 021
All osds show signs of having issues
During the reboot of 021, all slow requests stopped.

Second run



	A	B	C	D	E	F	G	H	I	J	K
1				buckets...	slow primary	slow subop	rwlock	total slow	failed	boot	wrongly down
168	default	internal-rack2	ceph1-020	osd.153	0	3	0	3	0	0	0
169	default	internal-rack2	ceph1-020	osd.158	0	1	0	1	0	0	0
170	default	internal-rack2	ceph1-020	osd.163	0	1	0	1	0	0	0
171	default	internal-rack2	ceph1-020	osd.168	0	0	0	0	0	0	0
172	default	internal-rack2	ceph1-020	osd.173	0	1	0	1	0	0	0
173	default	internal-rack2	ceph1-020	osd.178	0	0	0	0	0	0	0
174	default	internal-rack2	ceph1-020	osd.183	0	27	0	27	0	0	0
175	default	internal-rack2	ceph1-020	osd.188	0	4	0	4	0	0	0
176	default	internal-rack2	ceph1-020	osd.193	13	4	0	17	0	0	0
177	default	internal-rack2	ceph1-020	osd.198	5	13	5	18	0	0	0
178	default	internal-rack2	ceph1-020	18	54	5	72	0	0	0	
179	default	internal-rack2	ceph1-021	osd.352	520	167	14	687	0	1	0
180	default	internal-rack2	ceph1-021	osd.357	1767	772	64	2539	0	1	0
181	default	internal-rack2	ceph1-021	osd.362	1014	650	24	1664	0	1	0
182	default	internal-rack2	ceph1-021	osd.367	596	395	34	991	0	1	0
183	default	internal-rack2	ceph1-021	osd.372	506	186	6	692	0	1	0
184	default	internal-rack2	ceph1-021	osd.377	503	185	12	688	0	1	0
185	default	internal-rack2	ceph1-021	osd.382	5426	5202	145	10628	0	2	0
186	default	internal-rack2	ceph1-021	osd.387	943	373	43	1316	0	1	0
187	default	internal-rack2	ceph1-021	osd.392	750	265	36	1015	0	1	0
188	default	internal-rack2	ceph1-021	osd.397	707	285	24	992	0	1	0
189	default	internal-rack2	ceph1-021	12732	8480	402	21212	0	11	0	

Limiting failure impact

mon osd down out subtree limit
Description: The smallest CRUSH unit type that Ceph will not automatically mark out.
For instance, if set to host and if all OSDs of a host are down, Ceph will not automatically mark out these OSDs.

From ceph.conf:

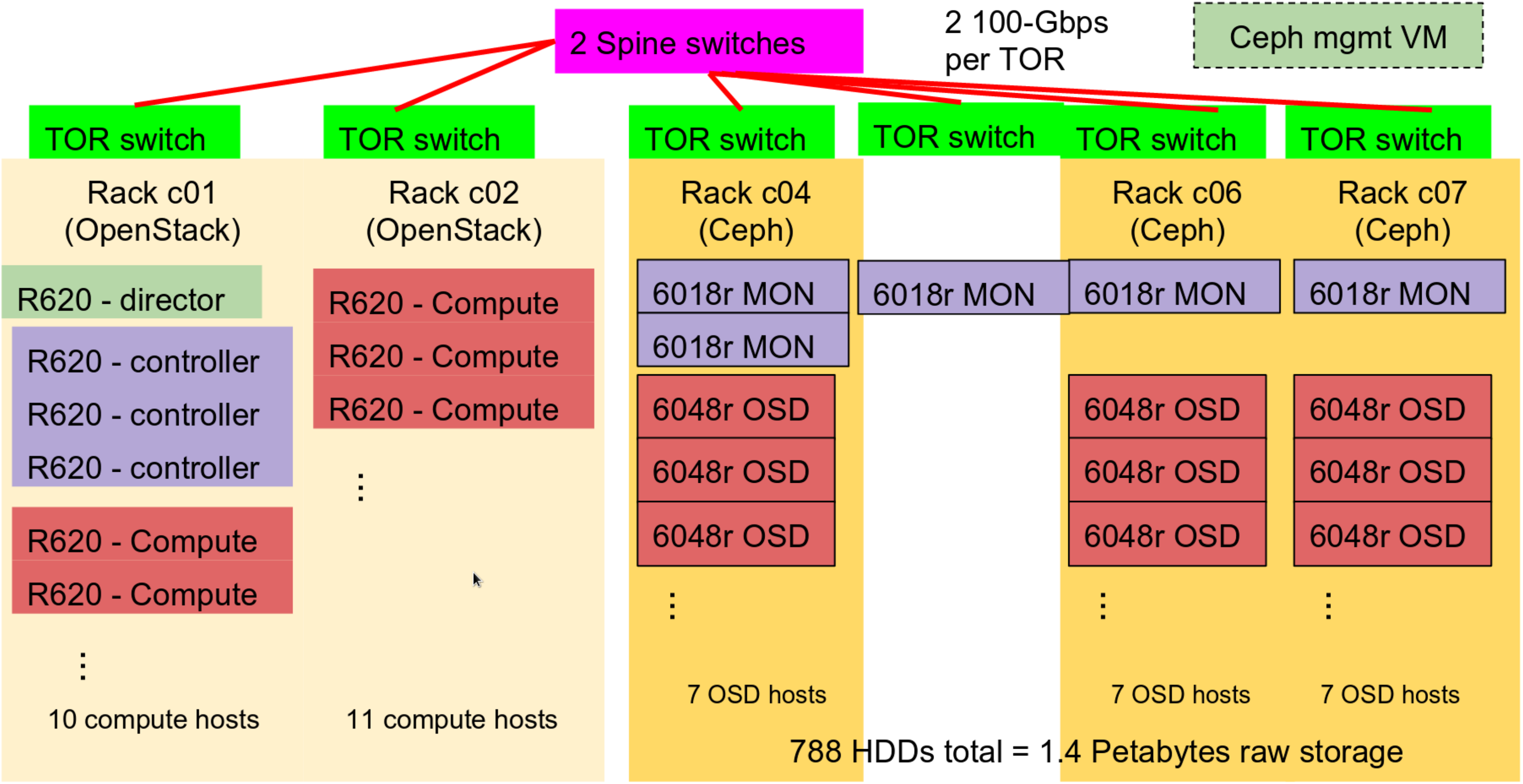
```
mon_osd_down_out_subtree_limit = host
```

If a host goes down, will not automatically trigger backfill.

http://docs.ceph.com/docs/master/rados/configuration/mon-osd-interaction/

Over to Ben England – Part 2

Physical hardware view - racks and switches



2 Spine switches

2 100-Gbps
per TOR

Ceph mgmt VM

TOR switch

Rack c01
(OpenStack)

R620 - director

R620 - controller

R620 - controller

R620 - controller

R620 - Compute

R620 - Compute

⋮

10 compute hosts

TOR switch

Rack c02
(OpenStack)

R620 - Compute

R620 - Compute

R620 - Compute

⋮

11 compute hosts

TOR switch

Rack c04
(Ceph)

6018r MON

6018r MON

6048r OSD

6048r OSD

6048r OSD

⋮

7 OSD hosts

TOR switch

6018r MON

TOR switch

Rack c06
(Ceph)

6018r MON

6048r OSD

6048r OSD

6048r OSD

⋮

7 OSD hosts

TOR switch

Rack c07
(Ceph)

6018r MON

6048r OSD

6048r OSD

6048r OSD

⋮

7 OSD hosts

788 HDDs total = 1.4 Petabytes raw storage

Deploy Time for Ceph

- Ignore OS deployment time
- RHCS “pre-flight” script < 5 min
- ceph-ansible finished in ~1 hour (using ansible -f 30)
- Early RHOSP 10 deploy had difficulty with external Ceph cluster
  - Fixed in RHOSP 10 z-stream and upcoming RHOSP 11
- 2 months ago with RHOSP 10 hyperconverged using director to deploy Ceph
  - Entire openstack deployment on 3 controllers + 8 computes
  - Completed in 54 min, repeated it 3x to be sure
  - Required /dev/disk/by-path naming for server with 2 disk controllers

Tests performed

- Create and populate storage pool that consumes > 50% of physical space
- Create and populate 100-GB Cinder volume for each of 512 guests
- Use 4-KB **fio** read/write random I/O mix (**fio** = “flexible input/output” benchmark)
- Measure throughput and latency as a function of guest count
- Throttle back workload to 80% capacity and:
  - remove and replace: monitor, HDD, OSD host while workload runs
  - measure effect on latency percentiles over time

New methodology – **fio** histogram logging

- Developed last summer by Perf & Scale team
- `fio —write-hist-log=1 —log-hist-msec=60000`
- Periodically logs latency histogram to fio_clat_hist.*.log
- **fio/tools/hist/fiologparser_hist.py** computes latency percentiles
  - As a function of time, for entire cluster
- More scalable and accurate because we are merging histograms

fio already had histograms within each process (fio/stat.h for details)

- 64 (2^6) buckets within each group are equal in latency range
- 19 bucket groups
- Each bucket group has double the latency range of preceding group
- Last bucket catches all latencies > preceding bucket

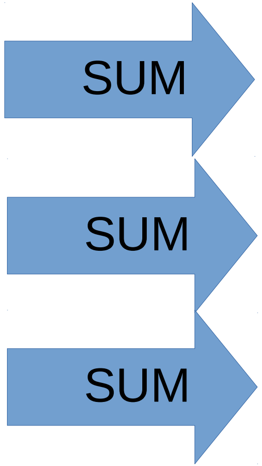
Bucket group	Bucket 1 max latency	Bucket 2 max latency	...	Bucket 64 (2^6) max latency
1	1	2		64
2	65	66		128
3	130	132		256
...				
19	8519680	8650752		16777216+

Latencies in microsec

Histogram logging enables merging histograms from N fio threads

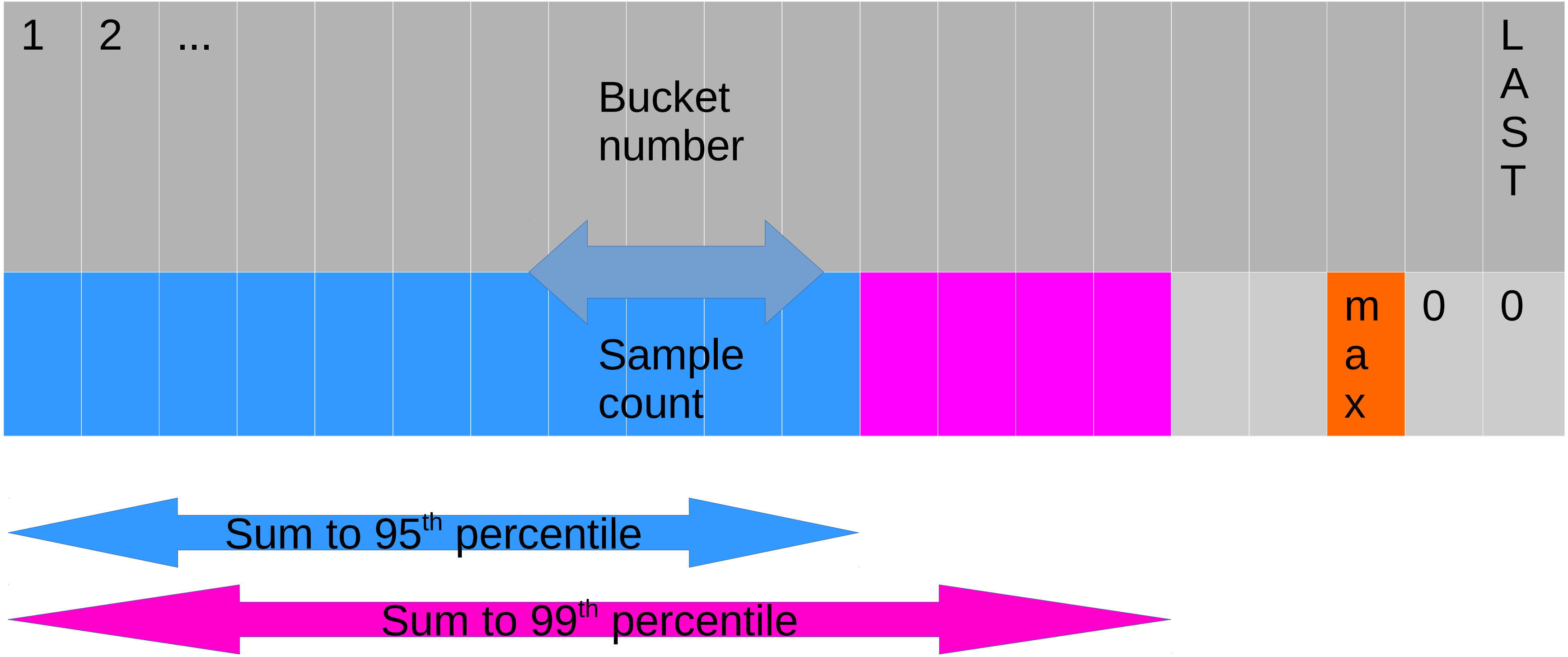
emit histogram log records every 60 seconds
example: for interval **[60, 120]** seconds

Histogram bucket	Thread 1	...	Thread N
Bucket 1			
...			
Bucket N			

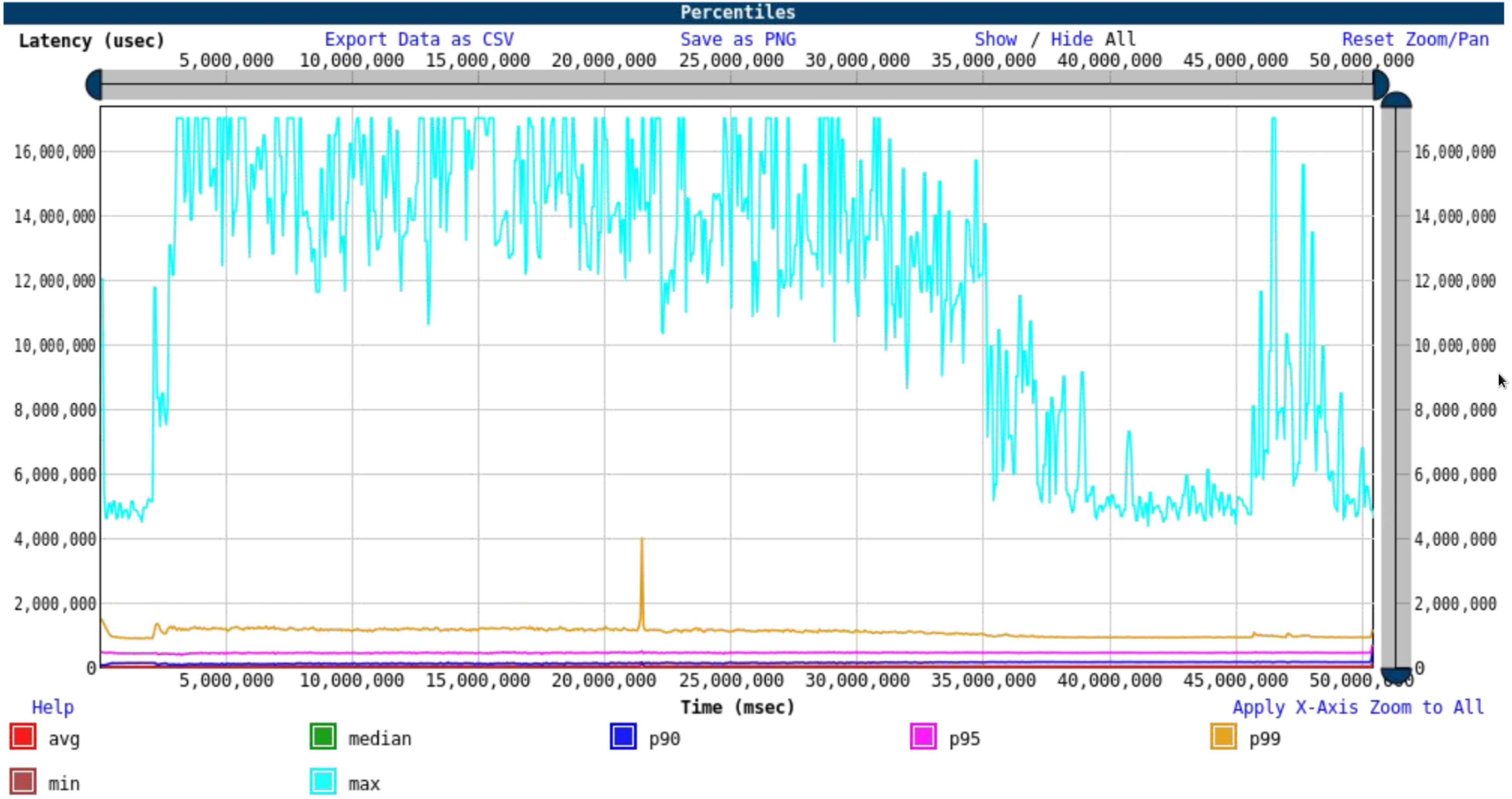


Cluster Histogram
Bucket 1
...
Bucket N

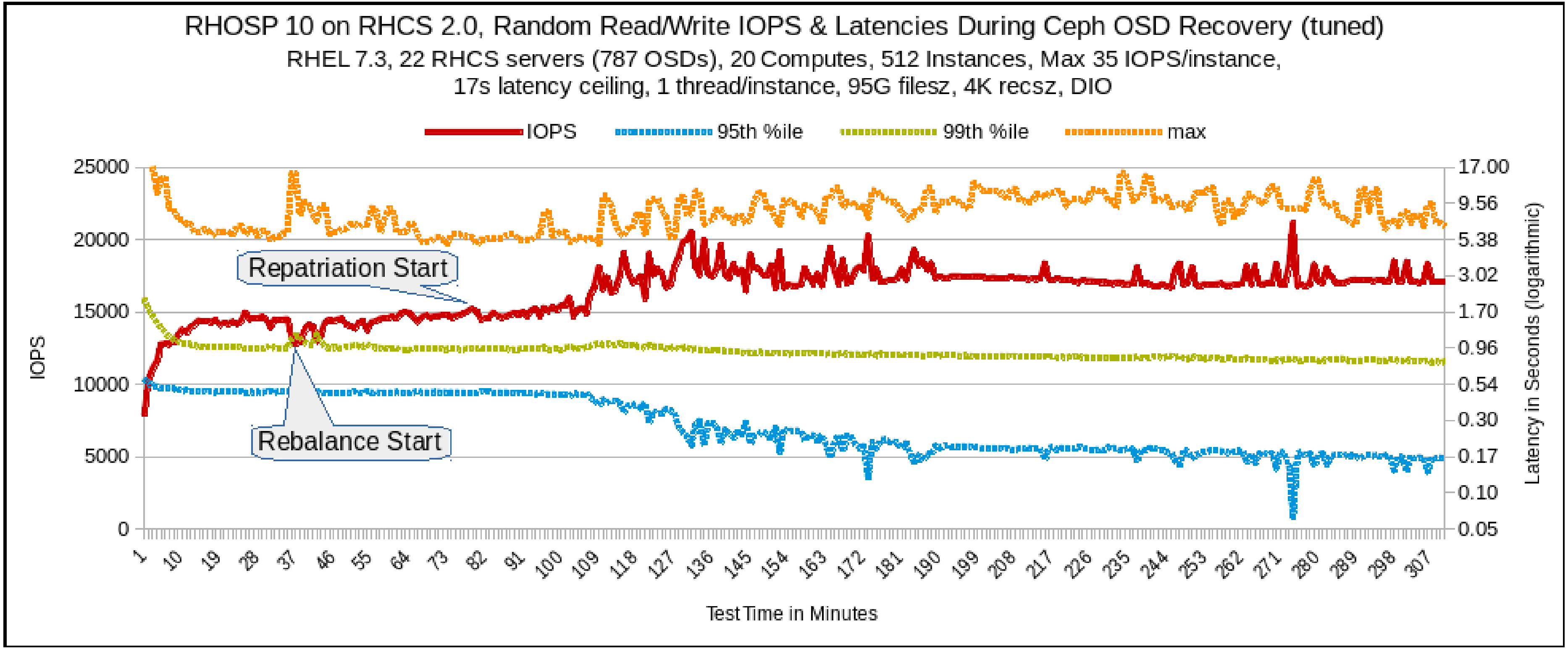
Converting histograms to latency percentiles



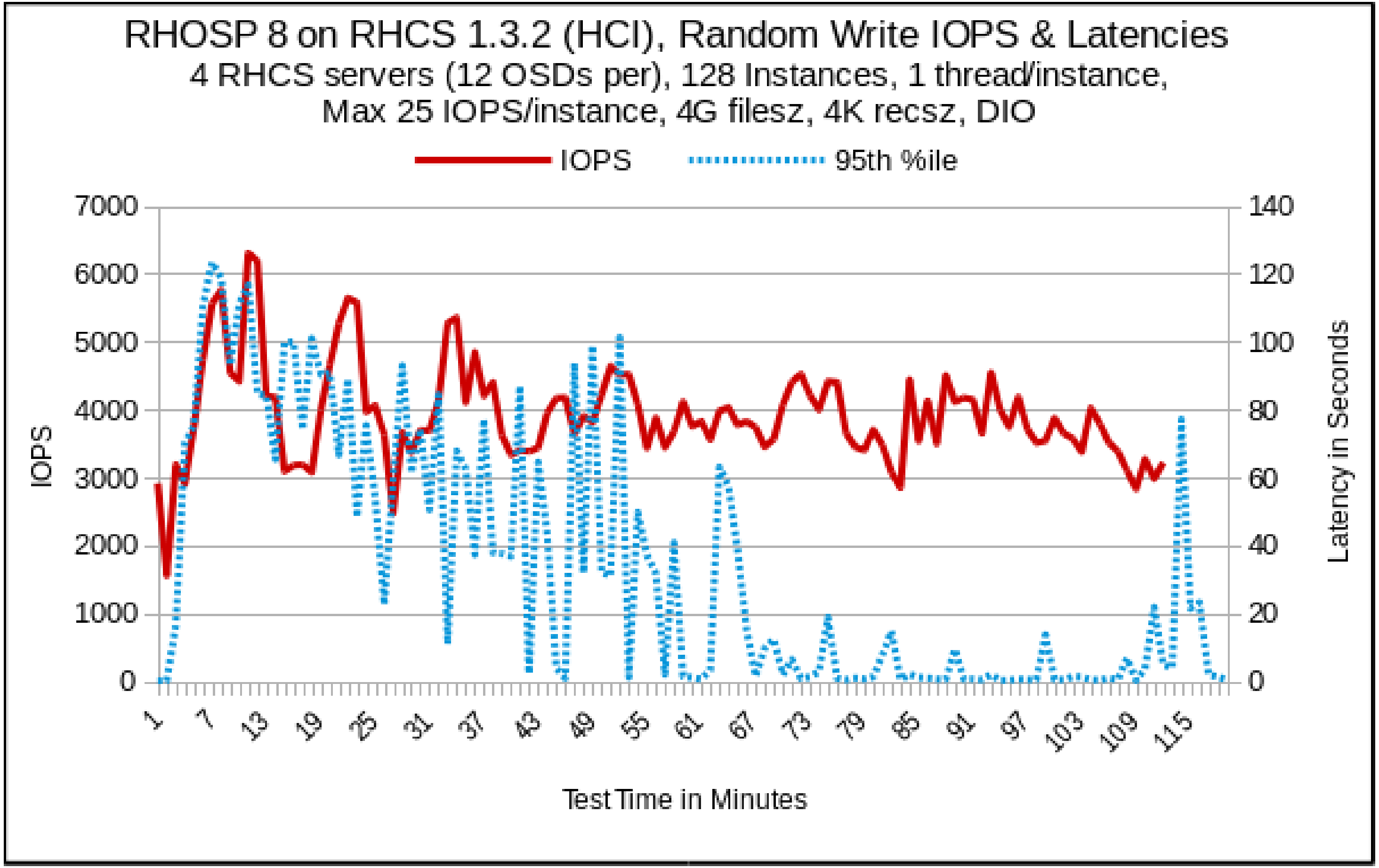
OSD node power-down - impact on I/O latency



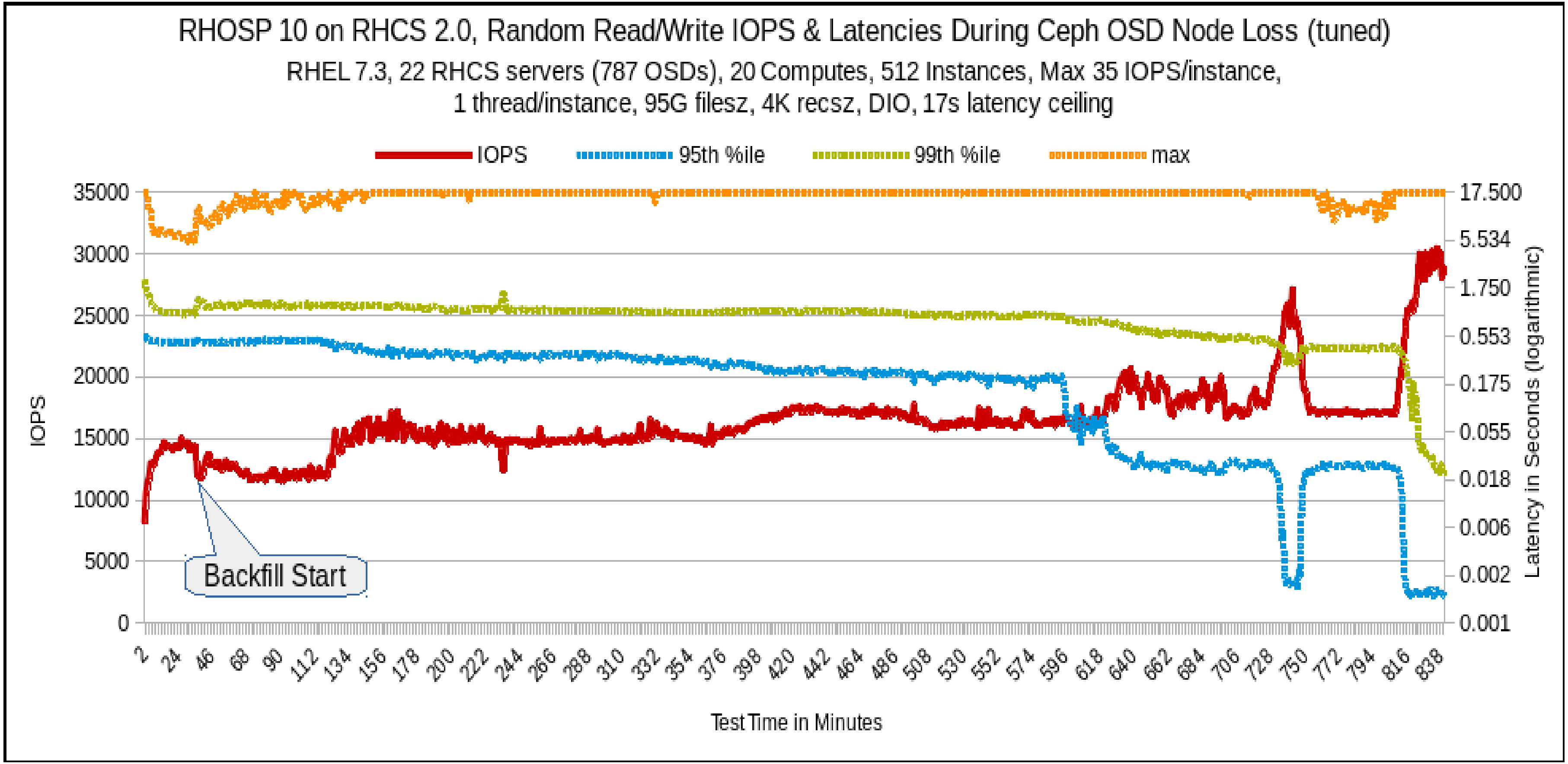
Simulated OSD failure and recovery



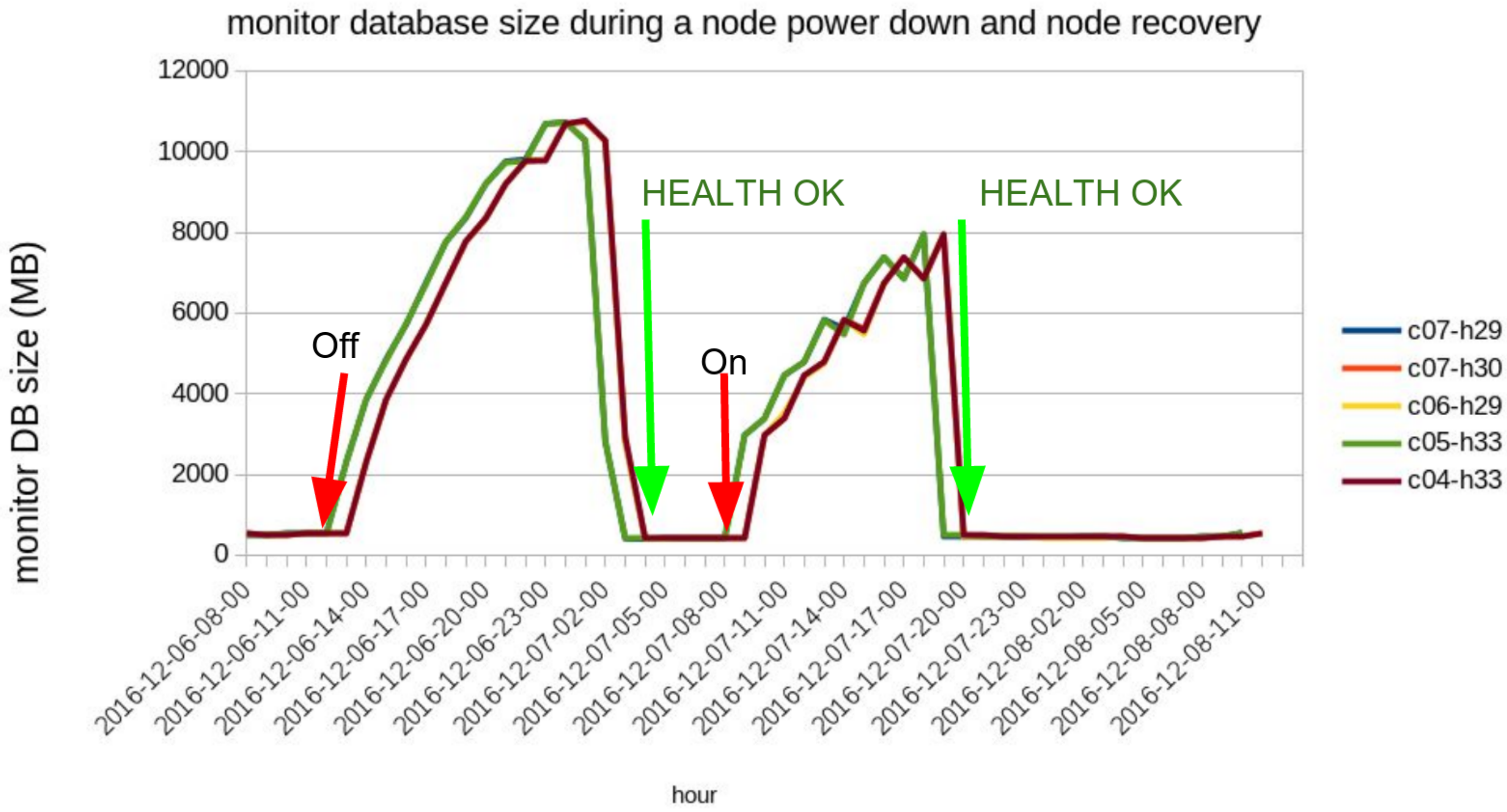
Big improvement from RHCS 1.3.2 = Ceph Hammer (see 95% spikes for single-OSD failure/recovery)



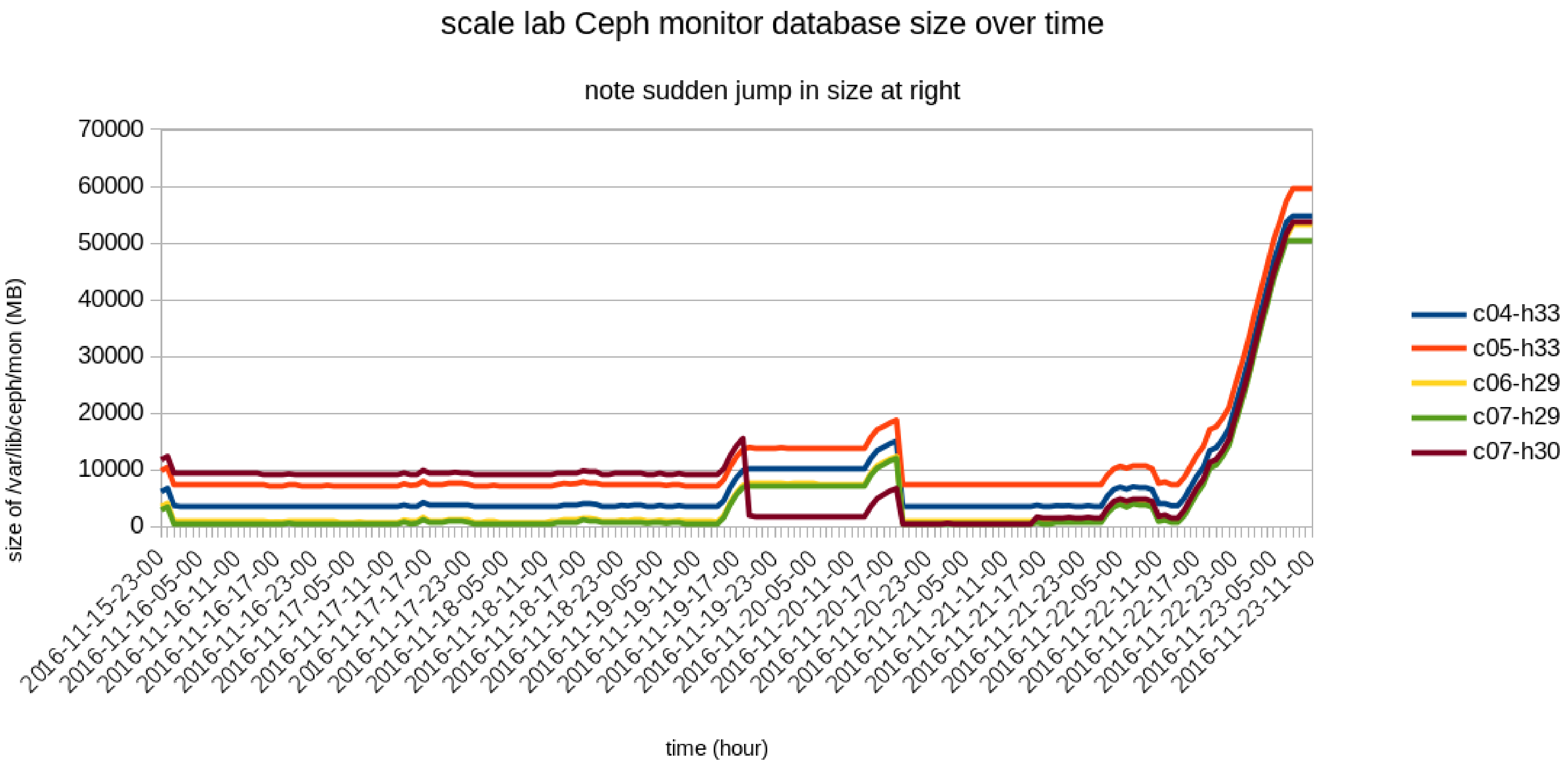
Node loss not so happy



Node power down and recovery - monitor DB size



Result of network partitioning due to switch fabric problem



Possible resolutions for node-failure issues

- If you know it’s happening
  - **ceph osd set noout**, fix node(s), then **ceph osd unset noout**
- Prevent automatic node-level backfill in ceph.conf:
  - **mon osd down out subtree limit = host**
  - Do one OSD at a time, takes longer but less disruption
- Long-term solution to this problem? My tentative hypothesis:
  - Throttle per-PG backfilling
  - backfill more PGs in parallel.

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

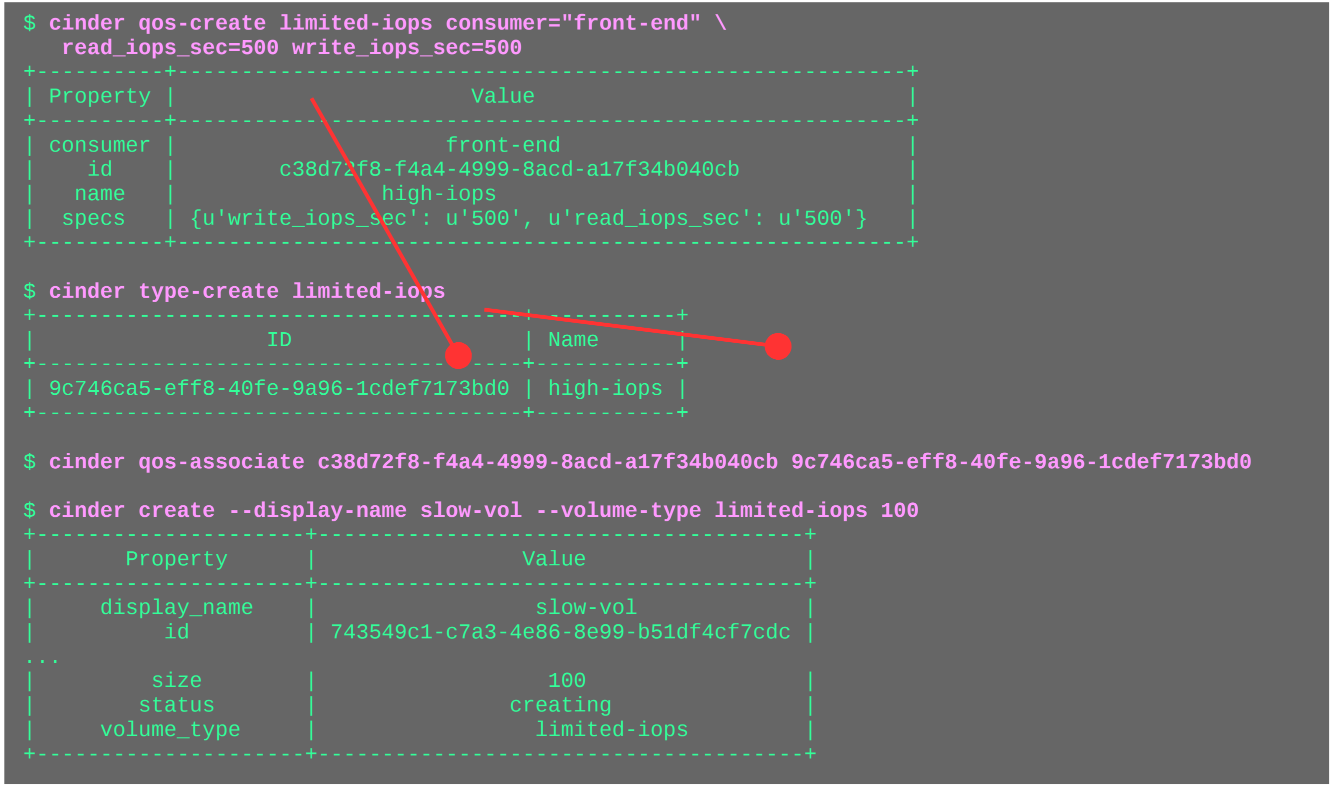
twitter.com/RedHatNews

youtube.com/user/RedHatVideos

RED HAT SUMMIT

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.

OpenStack should enforce KVM throttling on Nova guests
(from http://ceph.com/planet/openstack-ceph-rbd-and-qos/)



```
$ cinder qos-create limited-iops consumer="front-end" \
   read_iops_sec=500 write_iops_sec=500
+----------+---------------------------------------------------------+
| Property |                       Value                             |
+----------+---------------------------------------------------------+
| consumer |                     front-end                           |
|    id    |        c38d72f8-f4a4-4999-8acd-a17f34b040cb             |
|   name   |                high-iops                                |
|  specs   | {u'write_iops_sec': u'500', u'read_iops_sec': u'500'}   |
+----------+---------------------------------------------------------+

$ cinder type-create limited-iops
+--------------------------------------+-----------+
|                  ID                  | Name      |
+--------------------------------------+-----------+
| 9c746ca5-eff8-40fe-9a96-1cdef7173bd0 | high-iops |
+--------------------------------------+-----------+

$ cinder qos-associate c38d72f8-f4a4-4999-8acd-a17f34b040cb 9c746ca5-eff8-40fe-9a96-1cdef7173bd0

$ cinder create --display-name slow-vol --volume-type limited-iops 100
+---------------------+--------------------------------------+
|       Property      |                Value                 |
+---------------------+--------------------------------------+
|     display_name    |                 slow-vol             |
|          id         | 743549c1-c7a3-4e86-8e99-b51df4cf7cdc |
...
|         size        |                  100                 |
|        status       |               creating               |
|     volume_type     |                 limited-iops         |
+---------------------+--------------------------------------+
```