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 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
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)
bpsc_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


#redhat #rhsummit
Checking instance-0034e2c7 ...
   drive-virtio-disk0 is RBD, current throttle: 250/104857600 ... All good :)
Checking instance-0034e3b7 ...
   drive-virtio-disk0 is RBD, current throttle: 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.

[globals]
runtime=30
direct=1
bs=16k
randrepeat=0
rw=randread
iodepth=2
numjobs=4
cpus_allowed=0-3
cpus_allowed_policy=split
group_reporting

[ioengine]
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 | \n  awk '{print $3}' | sort -n | uniq -c | sort -rn
```

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

#redhat #rhsummit
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   |
|-----|---------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 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   | 0   |
After the reboot of 021
All osds show signs of having issues
During the reboot of 021, all slow requests stopped.

Second run

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

### Rack c01 (OpenStack)
- **R620 - director**
- **R620 - controller**
- **R620 - controller**
- **R620 - Compute**
- **R620 - Compute**

### Rack c02 (OpenStack)
- **R620 - Compute**
- **R620 - Compute**
- **R620 - Compute**

### Rack c04 (Ceph)
- **6018r MON**
- **6018r MON**
- **6048r OSD**
- **6048r OSD**
- **6048r OSD**
- **7 OSD hosts**

### Rack c05 (Ceph)
- **6018r MON**
- **6018r MON**
- **6048r OSD**
- **6048r OSD**
- **6048r OSD**
- **7 OSD hosts**

### Rack c06 (Ceph)
- **6048r OSD**
- **6048r OSD**
- **6048r OSD**

### Rack c07 (Ceph)
- **6048r OSD**
- **6048r OSD**
- **6048r OSD**

**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

<table>
<thead>
<tr>
<th>Bucket group</th>
<th>Bucket 1 max latency</th>
<th>Bucket 2 max latency</th>
<th>...</th>
<th>Bucket 64 ( (2^6) ) max latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>2</td>
<td>65</td>
<td>66</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>3</td>
<td>130</td>
<td>132</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>8519680</td>
<td>8650752</td>
<td></td>
<td>16777216+</td>
</tr>
</tbody>
</table>
Histogram logging enables merging histograms from N fio threads.

- Emit histogram log records every 60 seconds.
- Example: for interval $[60, 120]$ seconds.

<table>
<thead>
<tr>
<th>Histogram bucket</th>
<th>Thread 1</th>
<th>...</th>
<th>Thread N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bucket N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster Histogram</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUM</td>
</tr>
<tr>
<td>SUM</td>
</tr>
<tr>
<td>SUM</td>
</tr>
<tr>
<td>Bucket 1</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Bucket N</td>
</tr>
</tbody>
</table>
Converting histograms to latency percentiles

<table>
<thead>
<tr>
<th>Bucket number</th>
<th>Sample count</th>
<th>Last max</th>
<th>Sample count</th>
<th>Sample count</th>
<th>Last max</th>
<th>Sample count</th>
<th>Sample count</th>
<th>Last max</th>
<th>Sample count</th>
<th>Sample count</th>
<th>Last max</th>
<th>Sample count</th>
<th>Sample count</th>
<th>Last max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OSD node power-down - impact on I/O latency
Simulated OSD failure and recovery

RHOSP 10 on RHCS 2.0, Random Read/Write IOPS & Latencies During Ceph OSD Recovery (tuned)
RHEL 7.3, 22 RHCS servers (787 OSDs), 20 Computes, 512 Instances, Max 35 IOPS-instance,
17s latency ceiling, 1 thread-instance, 95G filesz, 4K recsz, DIO

- IOPS
- 95th %ile
- 99th %ile
- max

Latency in Seconds (logarithmic)

Test Time in Minutes
Big improvement from RHCS 1.3.2 = Ceph Hammer (see 95% spikes for single-OSD failure/recovery)
Node loss not so happy

RHOSP 10 on RHCS 2.0, Random Read/Write IOPS & Latencies During Ceph OSD Node Loss (tuned)
RHEL 7.3, 22 RHCS servers (787 OSDs), 20 Computes, 512 Instances, Max 35 IOPS/instance,
1 thread/instance, 95G filesz, 4K recsz, DIO, 17s latency ceiling

IOPS

Latency in Seconds (logarithmic)

Backfill Start

Test Time in Minutes
Node power down and recovery - monitor DB size

monitor database size during a node power down and node recovery

- monitor DB size (MB)
- hour

- Off
- On
- HEALTH OK

Lines:
- c07-h29
- c07-h30
- c06-h29
- c05-h33
- c04-h33
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.
THANK YOU

- plus.google.com/+RedHat
- linkedin.com/company/red-hat
- facebook.com/redhatinc
- twitter.com/RedHatNews
- youtube.com/user/RedHatVideos
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.
OpenStack should enforce KVM throttling on Nova guests
(from http://ceph.com/planet/openstack-ceph-rbd-and-qos/)

```bash
$ 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         |
+---------------------+--------------------------------------+