Scalable Metrics Collection using Prometheus and Thanos

Jonathan Eichelberger and Shawn Hall
Lustre Webinar – Sept 9 2020
Metrics Collection Wishlist

- Scalable to meet our needs
- Easy to implement
- Easy to manage
- Reasonable storage requirements
- Can handle high cardinality
- Single pane of glass
History of Metrics Collection at BP

- Have used several toolkits
- Some were more successful than others
- Only the simplest or most useful have survived
Ganglia
• Designed for clusters and grids
• Works well for aggregating cluster information into top level views
• RRD format works well for summaries, but inherently loses information
• Handles compute metrics, but required customization for Lustre monitoring
• Nothing was wrong, but our installation rotted away when the maintainer left our group
Splunk

- Awesome if you have a campus wide, unlimited license
- Not so awesome if you have to pay for a license yourself
- Shines at log collection and analysis, but also works well for metrics
- Never could show enough value to justify the cost
Telegraf + InfluxDB + Grafana

- Simple to install and configure
- Can parse Lustre jobstats
- Worked great initially, but...
No matter what time window you use, Grafana + InfluxDB should display an appropriate number of data points (1 hour window = 300 data points, 24 hour window = 300 data points)
  — But instead, amplitude was also scaled (1 hour window – 1 GB/s, 24 hour window - 24 GB/s)
  — Had to manually set resolution instead, meaning it was impossible to view data over large time windows
Telegraf + InfluxDB + Grafana Problems

- If data resolution not fixed, huge spikes sometimes appear at beginning of graphs making them unreadable
  - Still no real fix – just workarounds -
    https://github.com/influxdata/influxdb/issues/6451
  - Issue is 4 years, 4 months, 18 days old today- but who’s counting!
• Jobstats cardinality kills InfluxDB
  — Function of a number of jobs, but we don’t have a ton of jobs
  — Horizontal scaling requires InfluxDB Enterprise
  — InfluxDB Enterprise requires money
Current Metrics Collection at BP

- **Lustre Monitoring Tool**
  - Condensed view of server-side Lustre activity
  - First thing we put on a new file system
  - No historical data

```
Filesystem: lc1
Inodes: 446,432m total, 52,729m used (12%), 393,703m free
Space: 172,188t total, 138,933t used (81%), 33,255t free
Bytes/s: 0.000g read, 0.294g write,
MDops/s: 314 open, 156 close, 533 getattr, 6 setattr
        4 link, 196 unlink, 434 mkdir, 335 rmdir
        1 statfs, 3 rename, 0 getxattr

<table>
<thead>
<tr>
<th>OST</th>
<th>S</th>
<th>OSS</th>
<th>Exp</th>
<th>CR</th>
<th>rMB/s</th>
<th>wMB/s</th>
<th>IOPS</th>
<th>LOCKS</th>
<th>LGR</th>
<th>LCR</th>
<th>%cpu</th>
<th>%mem</th>
<th>%spc</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 F</td>
<td>tycho1</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>382</td>
<td>5</td>
<td>8</td>
<td>1</td>
<td>99</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>0001 F</td>
<td>tycho2</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>431</td>
<td>12</td>
<td>23</td>
<td>6</td>
<td>99</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>0002 F</td>
<td>tycho3</td>
<td>148</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>430</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>84</td>
<td>1</td>
<td>84</td>
</tr>
<tr>
<td>0003 F</td>
<td>tycho4</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>855</td>
<td>8</td>
<td>14</td>
<td>3</td>
<td>99</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>0004 F</td>
<td>tycho5</td>
<td>148</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>428</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>99</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>0005 F</td>
<td>tycho6</td>
<td>148</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>478</td>
<td>6</td>
<td>9</td>
<td>2</td>
<td>82</td>
<td>1</td>
<td>82</td>
</tr>
<tr>
<td>0006 F</td>
<td>tycho7</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>369</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>49</td>
<td>2</td>
<td>49</td>
</tr>
<tr>
<td>0007 F</td>
<td>tycho8</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>398</td>
<td>4</td>
<td>9</td>
<td>0</td>
<td>99</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>0008 F</td>
<td>tycho9</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>417</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>99</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>0009 F</td>
<td>tycho10</td>
<td>148</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>415</td>
<td>8</td>
<td>11</td>
<td>6</td>
<td>99</td>
<td>1</td>
<td>82</td>
</tr>
<tr>
<td>000a F</td>
<td>tycho11</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>425</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>84</td>
<td>1</td>
<td>81</td>
</tr>
<tr>
<td>000b F</td>
<td>tycho12</td>
<td>148</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>421</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>99</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>000c F</td>
<td>tycho13</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>446</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>99</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>
```
Current Metrics Collection at BP

- xltop
  - Gives critical relationship between jobs and file system performance
  - No historical data
  - TACC’s updates aren’t publicly available - we’re using 2012 code 😞

<table>
<thead>
<tr>
<th>FILESYSTEM</th>
<th>MDS/T</th>
<th>LOAD1</th>
<th>LOAD5</th>
<th>LOAD15</th>
<th>TASKS</th>
<th>OSS/T</th>
<th>LOAD1</th>
<th>LOAD5</th>
<th>LOAD15</th>
<th>TASKS</th>
<th>NIDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ranger-work</td>
<td>1/1</td>
<td>1.52</td>
<td>3.48</td>
<td>4.41</td>
<td>609</td>
<td>14/84</td>
<td>2.74</td>
<td>2.08</td>
<td>2.09</td>
<td>1347</td>
<td>4212</td>
</tr>
<tr>
<td>ranger-scratch</td>
<td>1/1</td>
<td>0.13</td>
<td>0.20</td>
<td>0.54</td>
<td>584</td>
<td>50/300</td>
<td>2.52</td>
<td>1.94</td>
<td>1.52</td>
<td>1348</td>
<td>4213</td>
</tr>
<tr>
<td>ranger-share</td>
<td>1/1</td>
<td>0.93</td>
<td>1.20</td>
<td>1.72</td>
<td>544</td>
<td>6/36</td>
<td>3.55</td>
<td>1.37</td>
<td>0.90</td>
<td>1203</td>
<td>3960</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JOB</th>
<th>FS</th>
<th>WR_MB/S</th>
<th>RD_MB/S</th>
<th>REQ/S</th>
<th>OWNER</th>
<th>NAME</th>
<th>HOSTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2526717</td>
<td>ranger-scratch</td>
<td>321.557</td>
<td>5.994</td>
<td>3556.133</td>
<td>tg803155</td>
<td>NST3.28-r0</td>
<td>20</td>
</tr>
<tr>
<td>login4</td>
<td>ranger-scratch</td>
<td>38.489</td>
<td>55.054</td>
<td>469.943</td>
<td>NONE</td>
<td>NONE</td>
<td>1</td>
</tr>
<tr>
<td>2530927</td>
<td>ranger-scratch</td>
<td>16.526</td>
<td>0.000</td>
<td>39.942</td>
<td>dkcira</td>
<td>Parametric</td>
<td>1</td>
</tr>
<tr>
<td>2529449</td>
<td>ranger-work</td>
<td>11.754</td>
<td>0.000</td>
<td>24.088</td>
<td>bealing</td>
<td>PE-OH</td>
<td>4</td>
</tr>
<tr>
<td>2530975</td>
<td>ranger-work</td>
<td>11.108</td>
<td>0.007</td>
<td>23.620</td>
<td>vishnam2</td>
<td>batch</td>
<td>16</td>
</tr>
</tbody>
</table>
Prometheus is a pull-based metric collecting / monitoring framework.

- a multi-dimensional data model (timeseries defined by metric name and set of key/value dimensions)
- a flexible query language to leverage this dimensionality
- no dependency on distributed storage; single server nodes are autonomous
- timeseries collection happens via a pull model over HTTP
- pushing timeseries is supported via an intermediary gateway
- targets are discovered via service discovery or static configuration
- multiple modes of graphing and dashboarding support

https://github.com/prometheus/prometheus
Thanos

- Thanos is a helper framework that allows Prometheus to be a highly available and scalable solution for monitoring large datacenters.
  - Global querying view across all connected Prometheus servers
  - Deduplication and merging of metrics collected from Prometheus HA pairs
  - Seamless integration with existing Prometheus setups
  - Downsampling historical data for massive query speedup
  - Simple gRPC "Store API" for unified data access across all metric data

https://github.com/thanos-io/thanos
Easily Add More Prometheus Servers

**template.yml**

```yaml
global:
  scrape_interval: 1m
  scrape_timeout: 30s
  evaluation_interval: 1m

  external_labels:
    shard: $SHARD

scrape_configs:
  - job_name: ipmi

    relabel_configs:
      - source_labels: [__address__]
        modulus: 4
        target_label: __tmp_hash
        action: hashmod
      - source_labels: [__tmp_hash]
        regex: ^$SHARD$
        action: keep
      - source_labels: [__address__]
        regex: ^([\^].*)$.*$
        target_label: instance
        replacement: ${i}

    file_sd_configs:
      - files:
        - ./targets/ipmi.yml
        refresh_interval: 5m
```

**generate_configs.sh**

```bash
config_dir=/hpc/sysadmin/prometheus/etc/configs

for i in {01..04}; do
  SHARD=$(( 10#$i - 1 ))
  envsubst < ${config_dir}/template.yml > ${config_dir}/hpcprom$i.yml
done
```

Number of Prometheus servers
In order to associate jobs with host metrics, a "flag" needs to be set on all compute nodes for the associated job.

--- Prolog

```bash
pdsh -t 60 -u 60 -f 128 -S "/hpc/SGE/bp/job-stats-start $JOB_ID"
```

--- Epilog

```bash
pdsh -t 60 -u 60 -f 128 -S "/hpc/SGE/bp/job-stats-stop $JOB_ID"
```

Without job running

```
# HELP node_jobsched_running_job Whether a scheduled batch job is currently running. Only valid for jobs with exclusive resource allocation.
# TYPE node_jobsched_running_job gauge
node_jobsched_running_job 0
```

With job running

```
# HELP node_jobsched_running_job Whether a scheduled batch job is currently running. Only valid for jobs with exclusive resource allocation.
# TYPE node_jobsched_running_job gauge
node_jobsched_running_job 107412540
```
Lustre Overview Dashboard

Lustre Overview (Compact)

Lustre J
- IO bandwidth for Lustre J
- Metadata operations for Lustre J

Lustre K
- IO bandwidth for Lustre K
- Metadata operations for Lustre K

Lustre L
- IO bandwidth for Lustre L
- Metadata operations for Lustre L
Lustre Overview Dashboard
Lustre Detail Dashboard
Lustre Detail Dashboard

Graphs showing read and write bandwidth for Lustre L per OSS and per OST.
Lustre Detail Dashboard
Lustre Detail Dashboard
Lustre Top Jobs Dashboard
Lustre Top Jobs Dashboard
Lustre Job Detail Dashboard
Lustre Job Detail Dashboard
Lustre Job Detail Dashboard

Data IOPS for hp0sk104

Average IO size for hp0sk104
Word of warning

- Precompute what you want to visualize into new metric series to reduce burden on Prometheus servers when trying to respond to complex queries
- Everything in this software stack is healthy except the Lustre Exporter
- HPE is no longer going to support the Lustre Exporter
- Join us in supporting the open source Lustre Exporter
Questions?