Expand your monitoring with Graphite/Grafana

IT Professionals Conf. 2019
Alec Sheperd
Motivation

- Find a different way to monitor our Condor cluster
- Move away from Ganglia for custom metrics
- Easily create custom dashboard and views of different graphs
- Use multiple datasource backends
- Spoiler: Graphite and Elasticsearch
  - Running job metrics in Graphite
  - Condor history in Elasticsearch
What is it?

● **Graphite - Time series data collection**
  - Carbon: collector/aggregator daemon
  - Whisper: Time-series database
  - Graphite-Web/API: HTTP endpoint for querying metrics and rendering graphs

● **Grafana - Graphing and visualization**
  - Front facing web application
  - Proxy requests from time-series data sources
    - (Graphite, Elasticsearch, Prometheus…)
  - Build graphs and other displays for visualization
Graphite

https://github.com/graphite-project/carbon
Graphite - Daemons

- **Carbon-cache**
  - Accepts and writes metrics to disk

- **Carbon-relay**
  - Forward metrics to multiple carbon-caches via rules or consistent hashing

- **Carbon-aggregator**
  - Buffer metrics in front of a carbon-cache to reduce disk IO or aggregate a matching namespace together
Graphite - Pushing in metrics

Plain-text:

PORT=2003
SERVER=carbon-server.icecube.wisc.edu
echo "my.namespace.metric 4 `date +%s`" | nc ${SERVER} ${PORT}

Python pickle:

data = ["my.namespace.metric", (timestamp, v)), ...]

payload = cPickle.dumps(data, protocol=2)
header = struct.pack("!L", len(payload))
message = header + payload
Graphite - Namespaces

- Metric identifier that corresponds to a POSIX file path
  - my.namespace.metric to $WHISPER/my/namespace/metric

- Group related metrics under similar name spaces
  - File retention
  - Ease of querying

- Best not to get too “married” to a namespace
  - Run into organization issues
  - Keep dynamic paths deeper in the namespace
Graphite - Retentions

- Set different retention policies for different namespaces
  - Aggregation methods (average, min, max, last...)

- Lower precision for historical data
  - Save disk space and query latency at the cost of CPU

```
[title]
pattern = ^my.namespace.*$
retentions = 60s:1d,300s:1yr
```

- File retentions determines total disk usage
Graphite - Pros

● Highly customizable
  ○ Tailor to fit your needs
  ○ Can scale well

● Lots of documentation and tools
  ○ Many other technologies and services have functions to export to graphite

● Simple design
Graphite - Problems & Limitations

- Double-edged: Generally easy to stand up; tweaks required for extensive use
  - Ran into performance issues when you get move toward 200k updates/minute
- No standard data collection service
  - Collectd, Collectl, Ganglia-graphite

Graphite - Clustering

- Relay metrics to multiple carbon-caches
- Carbon-cache nodes can be either distinct machines, or a single machine with multiple disks
- Helm charts for running cluster in kubernetes
Grafana

- Web app for organizing and displaying dashboards of graphs
- Authentication via OAuth, LDAP, or custom Auth Proxy
- Collection of community built dashboards and plugins
Dashboards are a collection of multiple panels
  - Histograms, line graphs, tables, single stats...

Each panel can have multiple queries to data sources
  - HTTP-API calls to a backend

Template variables allow dynamic changes to be made to queries
Grafana - Creating a new dashboard
Grafana - Building a panel
Grafana - Template variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$prefix</td>
<td>collectd</td>
</tr>
<tr>
<td>$namespace</td>
<td>$prefix.*</td>
</tr>
<tr>
<td>$server</td>
<td>$prefix.$namespace.*</td>
</tr>
<tr>
<td>$disk</td>
<td>$prefix.$namespace.$server.disk.*</td>
</tr>
</tbody>
</table>
Questions!

- Try out a Grafana dashboard at [https://play.grafana.org/d/000000056/graphite-templated-nested?orgId=1](https://play.grafana.org/d/000000056/graphite-templated-nested?orgId=1)