Interactive Analytics with Druid at Yahoo!

By
Himanshu Gupta
Yahoo! Inc.
Agenda

- Why Druid and What do we do with it?
- How do we manage Druid clusters at scale?
- How do we debug and solve [performance] issues?
What is Druid?

An open-source high performance column-oriented distributed data store.

- Queries for splitting by arbitrary dimensions and filtering
- Batch and Realtime Ingestion
- Very extensible - aggregation functions, parsers, filters, emitters, pluggable storage etc.
- Rich set of open source UIs available for dashboards out of the box
Druid Community

- Apache 2.0 license
- >140 contributors so far
- Used by >49 companies for interactive ad-hoc analytics
Druid Architecture Overview

Diagram showing the components and data flow in Druid architecture.
Evolution of Analytics Architectures...
Analytics about 2.5 years ago...
Analytics now... ("Lambda" Architecture)
Analytics now... ("Kappa" Architecture)
Druid use cases At Yahoo!
Druid use cases At Yahoo!

- Usage Analytics
Druid use cases At Yahoo!

- Usage Analytics
- Revenue reporting
Druid use cases At Yahoo!

- Usage Analytics
- Revenue reporting
- Spam analytics
Druid use cases At Yahoo!

- Usage Analytics
- Revenue reporting
- Spam analytics
- Ad feedback
Druid use cases At Yahoo!

- Usage Analytics
- Revenue reporting
- Spam analytics
- Ad feedback
- Flurry SDK reporting
Druid use cases At Yahoo!

- Usage Analytics
- Revenue reporting
- Spam analytics
- Ad feedback
- Flurry SDK reporting
- A/B Testing performance analytics

..... and many more
Scale

- >35 Druid Clusters
- On >2000 hosts with clusters having up to 325 nodes
- >2.5 mn "data" queries per day (excluding cached and metadata queries)
- Individual clusters indexing up to 75 bn events per day in realtime
- Hundreds of terabytes of data "loaded and queryable" on the clusters
Build and Deployment Automation
Automation: Continuous Integration

Druid Mirror -> Build -> Unit Tests -> Test Cluster Deploy -> Integration Tests -> Druid Artifacts
Automation: Continuous Integration

- 2 CI pipelines, release and master
- All druid core change PRs are sent to public github druid repository
Automation: Blue/Green Deployment

UI

DNS Resolver

1

VIP

Blue Region

Druid

VIP

Red Region

Druid
Automation: Canary Deployment

- Druid releases are backward compatible and provide a forward migration path.
Alerts and Monitoring

is something wrong?
Druid Emitters

Druid Process

emitter

Monitoring System

emitter
Druid Emitters

- We wrote a custom emitter for our internal monitoring system.
- Community provided emitter extensions are also available.
Alerts

- Historical Nodes capacity utilization using "segment/size" metric
Alerts

- Historical Nodes capacity utilization using "segment/size" metric
- Realtime ingestion events dropped using "ingest/events/thrownAway" and "ingest/events/unparsable"
Alerts

- Historical Nodes capacity utilization using "segment/size" metric
- Realtime ingestion events dropped using "ingest/events/thrownAway" and "ingest/events/unparsable"
- Percent query failure threshold using "success" dimension in "query/time" metric
- Expected loaded intervals using segmentMetadata query and coordinator endpoints
Alerts

- Historical Nodes capacity utilization using "segment/size" metric
- Realtime ingestion events dropped using "ingest/events/thrownAway" and "ingest/events/unparsable"
- Percent query failure threshold using "success" dimension in "query/time" metric
- Expected loaded intervals using segmentMetadata query and coordinator endpoints
- http://host:port/status alerts
- process alerts
- other system resource (cpu, memory, disk, bandwidth etc) alerts
Monitoring

- query latency using "query/time" at brokers
- query throughput using "query/time" event counts
- "query/bytes" at brokers
- "query/time", "query/bytes", "query/segment/time" and "query/segment/wait" from historicals
- "ingestion/*" metrics for ingestion
Also, Alerts on known fatal exceptions/errors such as "OutOfMemoryError"
Debugging Problems

- is something wrong?
- are queries are running slower than usual?
- some query is failing.
- ingestion is failing.
- ingestion successful, but data is not showing up in queries.
- ......
Debugging Performance Problems

- Why did my query run slow?
Druid Query Processing Overview

Broker

- Historical
- Historical
- Historical
- Historical
- Historical
Druid Query Processing Overview
Druid Query Processing Overview

broker merges results from historicals

1. Schedule segments
2. Scan segments parallelly
3. Segment results
4. Merge results

Zoom In
Druid Query Processing Overview

query/time

Broker

Historical

Historical

Historical

Historical

Historical

merged result

Zoom In

1

2

3

4

1. schedule segments

2. scan segments parallelly

3. segment results

4. merged result

workers

database

engine
Druid Query Processing Overview

Broker

1. Historical
2. Historical
3. Historical
4. Historical

merged result

1. Schedule segments
2. Scan segments parallelly
3. Segment results
4. Zoom In

query/node/time
query/time
Druid Query Processing Overview

- Broker
  - Historical
  - Historical
  - Historical
  - Historical
  - Historical

- Zoom In
  - Engine
  - Workers
    - Segment results
    - Scan segments parallelly
  - Schedule segments
  - Merged result

- Query/node/time
- Query/time
Druid Query Processing Overview

**Broker**
- query/node/time
- query/time
- query/time

**Historical**
- Historical
- Historical
- Historical
- Historical

**Engine**
- Schedule segments
- Segment results
- Scan segments parallelly

**Zoom In**
- Merged result

**Workers**
- query/wait/time

Diagram details:
- 1: Schedule segments
- 2: Scan segments parallelly
- 3: Segment results
- 4: Merged result
Query Processing Overview

"query/bytes" and "query/node/bytes"

metrics have rich set of dimensions like queryId, service, host, segmentId etc
Debugging Performance Problems

- Why did my query run slow?

... debug with metrics, but typical monitoring systems cannot handle high cardinality of queryId and segmentId.
Metrics Cluster (yet another Druid use case)
Metrics Cluster (yet another Druid use case)
<table>
<thead>
<tr>
<th>Server</th>
<th>Events</th>
<th>Max Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>40.0</td>
<td>41.3 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>41.3 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>6.5 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>5.4 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>4.1 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.9 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.9 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.8 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.7 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.7 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.7 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.6 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.6 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.5 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.5 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.5 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.5 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.4 k</td>
</tr>
<tr>
<td>hoo.com</td>
<td>1.0</td>
<td>3.4 k</td>
</tr>
<tr>
<td>Segment</td>
<td>Events</td>
<td>Max Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>Total</td>
<td>8.0</td>
<td>26.3 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>26.3 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>15.9 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>7.8 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>4.6 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>3.8 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>1.4 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>0.7 k</td>
</tr>
<tr>
<td>00:00:00.000Z_20</td>
<td>1.0</td>
<td>0.7 k</td>
</tr>
<tr>
<td>Segment</td>
<td>Events</td>
<td>Max Value</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>Total</td>
<td>8.0</td>
<td>16.3 k</td>
</tr>
<tr>
<td>0:00:00.000Z_20</td>
<td>1.0</td>
<td>16.3 k</td>
</tr>
<tr>
<td>0:00:00.000Z_20</td>
<td>1.0</td>
<td>15.9 k</td>
</tr>
<tr>
<td>0:00:00.000Z_20</td>
<td>1.0</td>
<td>15.6 k</td>
</tr>
<tr>
<td>0:00:00.000Z_20</td>
<td>1.0</td>
<td>15.6 k</td>
</tr>
<tr>
<td>0:00:00.000Z_20</td>
<td>1.0</td>
<td>15.2 k</td>
</tr>
<tr>
<td>0:00:00.000Z_20</td>
<td>1.0</td>
<td>15.0 k</td>
</tr>
<tr>
<td>0:00:00.000Z_20</td>
<td>1.0</td>
<td>14.4 k</td>
</tr>
</tbody>
</table>
Other Tips
Other Tips

- datasketches.github.io for unique-counts and retention analysis
Other Tips

- datasketches.github.io for unique-counts and retention analysis
- Custom dimension partitioning instead of hash based
Other Tips

- datasketches.github.io for unique-counts and retention analysis
- Custom dimension partitioning instead of hash based
- Performance slices or specialized DataSource/Tables (Don't overdo it!)
Other Tips

- datasketches.github.io for unique-counts and retention analysis
- Custom dimension partitioning instead of hash based
- Performance slices or specialized DataSources/Tables (Don't overdo it!)
- Historical Tiers to increase capacity in a cheaper way
Other Tips

- datasketches.github.io for unique-counts and retention analysis
- Custom dimension partitioning instead of hash based
- Performance slices or specialized DataSources/Tables (Don't overdo it!)
- Historical Tiers to "increase" capacity in a cheaper way
- Keeping up-to-date with releases
Recap...

- Druid's out-of-the-box support for lambda architecture
- Build and Deployment automation
- Alerts, Monitoring and Logs
- Analytics on druid metric events for tuning performance of the cluster
- Other tips for effectively using Druid