While You Wait...

WHY NOT ORDER YOURSELF A “T-SHIRT”?

01 Visit: https://shirts.beachops.io/

02 Order a T-Shirt!
(Don’t worry you won’t be charged)
Cloud Scale Application Monitoring

Annie Lin
Application Platform Architect VMware (APJ)
Code → Business Value
The only thing that matters is customer happiness.
NINES don’t matter when USERS aren’t HAPPY
3D Observability
‘3D’ Observability

Metrics
# Metrics 101

## Sample Metric

<table>
<thead>
<tr>
<th>Metric</th>
<th>Data Value</th>
<th>Timestamp</th>
<th>Source Name</th>
<th>Point Tags (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>services.transactions.count</td>
<td>32584</td>
<td>1382754475</td>
<td>source=prod1_wf.com</td>
<td>dc=palo_alto service=auth</td>
</tr>
</tbody>
</table>

**Anything being measured overtime**

Lightweight data model enables real-time use case

Full flexibility to compare metrics based on time series

Easily understood (visually) and quickly identify patterns

---

©2019 VMware, Inc.
‘3D’ Observability

Histograms
Historgams 101

Sample Metric Histogram Point

Histogram Name: services.transaction.latency

{1:3; 2:4; 3:35; 5:50; 10:45...}

Full Population of Data Values in (Centroid) Bins

Time Interval: 1-minute

Source Name: source=prod1.wf.com

Point Tags (optional): dc=palo_alto service=auth

Latency distribution for=1m

Latency distribution for services.transaction.latency of 99th percentile (p99)
‘3D’ Observability

Tracing
## Tracing 101

<table>
<thead>
<tr>
<th>Application</th>
<th>Service</th>
<th>Component</th>
<th>Trace</th>
<th>Operation/Span</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Beachshirts”</td>
<td>“shopping”</td>
<td>“jersey-server”</td>
<td>“shopping.orderShirts” 4.18 seconds</td>
<td>“shopping.orderShirts” 2.44 seconds</td>
</tr>
</tbody>
</table>

Span is primary building block of a distributed trace

- Individual unit of work done; spans assemble into traces

Traces are used to profile and monitor applications

- Especially useful for microservices applications

Helps to pinpoint where failures occur and poor performance

- Accelerates application troubleshooting
Tracing 101
'3D' Observability

**Metrics**
Sequence of data points, measuring the same thing over time.

**Histograms**
Distributions, frequency of data in buckets.

**Tracing**
Profile and monitor applications, especially microservices.
Monitor a “Polyglot” microservices application

Beach Shirts App

1. Order
2. Authenticate
3. Availability
4. Pay
5. Style
6. Checkout
7. Fetch
8. Print
9. Pack
10. Dispatch
11. Notify
12. Email

Authentication

$\text{Springboot /JAX-RS}$

Notification

$\text{Springboot /Jersey}$

Delivery

$\text{Dropwizard /Jersey}$

Shopping

$\text{Dropwizard /Jersey}$

Styling

$\text{GRPC}$

Inventory

$\text{GO}$

Payments

$\text{Microsoft .NET}$

Warehouse

Printing

Packing
Modern Application Landscape
Beach Shirts App in production

#developers

on-call #SRE
#BeachOps done wrong

1. Don’t know where to start!
2. **OrderShirts** API → What’s the request call flow?
3. What are the service dependencies?
4. Health of the services involved in the request?
5. How many customers having problems?
6. What’s the median, avg, p90, p95, p99 latency?
7. Cannot identify & isolate the bottleneck service.
8. Cannot diagnose & remediate the problem.
INFO [2019-07-26 03:44:10,502] queryserver.QueryingRpcServerImpl: Running query: max((ts("build.version", tag=longboard) as xx) - lag(1h, $xx)) in context: QueryContext(startTime=1564111980, endTime=1564122880, realStartTime=1564112280, realEndTime=1564112580, sampleSeconds=60, lookback=false, includeObsoletedMetrics=false, counters=Counters(queries=0, droppedQueries=0, keys=0, points=0, summaries=0, droppedSummaries=0, bufferKeys=0, compactedKeys=0, cachedCompactedKeys=0, skippedCompactedKeys=0, compressedPoints=0, s3Keys=0, missingS3Keys=0, cpuNs=0, latency=0), running=RunState(tickets=1, cancelled=false, allStreamsPrepared=false), strategy=MEAN, queryTasks=queryserver.query.QueryTasksTracker79450df9, now=1564112659582, isAlertQuery=true, alertID=1527110125998, keysOnly=false, batchPriority=0, startForSpans=1564112290, endForSpans=1564112590)
INFO [2019-07-26 03:44:10,544] queryserver.QueryingRpcServerImpl: [collector] alert: max((ts("build.version", tag=longboard) as xx) - lag(1h, $xx)) Counters(queries=858, droppedQueries=1716, keys=1243, points=1357, summaries=16690, droppedSummaries=0, bufferKeys=1538, compactedKeys=249, cachedCompactedKeys=247, skippedCompactedKeys=95, compressedPoints=16609, s3Keys=0, missingS3Keys=0, cpuNs=159086612, latency=24); cpu_seconds: 0.038037462
INFO [2019-07-26 03:44:10,610] queryserver.QueryingRpcServerImpl: Non-serving side alert query got invoked for customer=collector, query=default(60m, 10m, 0, ts("telegraf.system.uptime", source=sonarqube*)) = 0, startAt=1564112290
INFO [2019-07-26 03:44:10,610] queryserver.QueryingRpcServerImpl: Running query: default(60m, 10m, 0, ts("telegraf.system.uptime", source=sonarqube*)) = 0 in context: QueryContext(startTime=1564111980, endTime=1564122880, realStartTime=1564112280, realEndTime=1564112580, sampleSeconds=60, lookback=false, includeObsoletedMetrics=false, counters=Counters(queries=0, droppedQueries=0, keys=0, points=0, summaries=0, droppedSummaries=0, bufferKeys=0, compactedKeys=0, cachedCompactedKeys=0, skippedCompactedKeys=0, compressedPoints=0, s3Keys=0, missingS3Keys=0, cpuNs=0, latency=0), running=RunState(tickets=1, cancelled=false, allStreamsPrepared=false), strategy=MEAN, queryTasks=queryserver.query.QueryTasksTracker3f0f0d599, now=1564112659582, isAlertQuery=true, alertID=1527173532014, keysOnly=false, batchPriority=0, startForSpans=1564112290, endForSpans=1564112590)
INFO [2019-07-26 03:44:10,612] queryserver.QueryingRpcServerImpl: [collector] alert: default(60m, 10m, 0, ts("telegraf.system.uptime", source=sonarqube*)) = 0: Counters(queries=3, droppedQueries=6, keys=60, points=60, summaries=62, droppedSummaries=0, bufferKeys=61, compactedKeys=1, cachedCompactedKeys=1, skippedCompactedKeys=0, compressedPoints=62, s3Keys=0, missingS3Keys=0, cpuNs=169269, latency=0); cpu_seconds: 8.28945E+4
INFO [2019-07-26 03:44:10,638] queryserver.QueryingRpcServerImpl: [collector] alert: (sum(rate(ts(serviceclient.*_call_failures, tag="*-primary" or tag="*-secondary" and not (tag=eval or service="anomaly"))), hosttags, metrics, service) > 2: Counters(queries=229, droppedQueries=4458, keys=7352, points=10347, summaries=32708, droppedSummaries=0, bufferKeys=8038, compactedKeys=430, cachedCompressedKeys=287, skippedCompactedKeys=287, compressedPoints=32708, s3Keys=0, missingS3Keys=0, cpuNs=58582202, latency=190000)
INFO [2019-07-26 03:44:10,647] queryserver.QueryingRpcServerImpl: [collector] alert: (sum(rate(ts(serviceclient.*_call_failures, tag="*-primary" or tag="*-secondary" and not (tag=eval or service="anomaly"))), hosttags, metrics, service) > 2: Counters(queries=229, droppedQueries=4458, keys=7352, points=10347, summaries=32708, droppedSummaries=0, bufferKeys=8038, compactedKeys=430, cachedCompressedKeys=287, skippedCompactedKeys=287, compressedPoints=32708, s3Keys=0, missingS3Keys=0, cpuNs=58582202, latency=190000)
INFO [2019-07-26 03:44:10,647] queryserver.QueryingRpcServerImpl: [collector] alert: (sum(rate(ts(serviceclient.*_call_failures, tag="*-primary" or tag="*-secondary" and not (tag=eval or service="anomaly"))), hosttags, metrics, service) > 2: Counters(queries=229, droppedQueries=4458, keys=7352, points=10347, summaries=32708, droppedSummaries=0, bufferKeys=8038, compactedKeys=430, cachedCompressedKeys=287, skippedCompactedKeys=287, compressedPoints=32708, s3Keys=0, missingS3Keys=0, cpuNs=58582202, latency=190000)
INFO [2019-07-26 03:44:10,647] queryserver.QueryingRpcServerImpl: [collector] alert: (sum(rate(ts(serviceclient.*_call_failures, tag="*-primary" or tag="*-secondary" and not (tag=eval or service="anomaly"))), hosttags, metrics, service) > 2: Counters(queries=229, droppedQueries=4458, keys=7352, points=10347, summaries=32708, droppedSummaries=0, bufferKeys=8038, compactedKeys=430, cachedCompressedKeys=287, skippedCompactedKeys=287, compressedPoints=32708, s3Keys=0, missingS3Keys=0, cpuNs=58582202, latency=190000)
INFO [2019-07-26 03:44:10,647] queryserver.QueryingRpcServerImpl: [collector] alert: (sum(rate(ts(serviceclient.*_call_failures, tag="*-primary" or tag="*-secondary" and not (tag=eval or service="anomaly"))), hosttags, metrics, service) > 2: Counters(queries=229, droppedQueries=4458, keys=7352, points=10347, summaries=32708, droppedSummaries=0, bufferKeys=8038, compactedKeys=430, cachedCompressedKeys=287, skippedCompactedKeys=287, compressedPoints=32708, s3Keys=0, missingS3Keys=0, cpuNs=58582202, latency=190000)
INFO [2019-07-26 03:44:10,647] queryserver.QueryingRpcServerImpl: [collector] alert: (sum(rate(ts(serviceclient.*_call_failures, tag="*-primary" or tag="*-secondary" and not (tag=eval or service="anomaly"))), hosttags, metrics, service) > 2: Counters(queries=229, droppedQueries=4458, keys=7352, points=10347, summaries=32708, droppedSummaries=0, bufferKeys=8038, compactedKeys=430, cachedCompressedKeys=287, skippedCompactedKeys=287, compressedPoints=32708, s3Keys=0, missingS3Keys=0, cpuNs=58582202, latency=190000)
#BeachOps done right => “3D Observability Platform”
Traces (100):

Matching Spans

- beachshirts.shop... (11.66s)
- beachshirts.shop... (10.48s)
- beachshirts.shop... (10.05s)
- beachshirts.shop... (10.02s)
- beachshirts.shop... (9.87s)
- beachshirts.shop... (9.72s)
- beachshirts.shop... (9.72s)
- beachshirts.shop... (9.33s)
- beachshirts.shop... (9.2s)
- beachshirts.shop... (9.2s)

shopping: ShoppingWebResource.orderShirts 07-26-2019 1:28:24 pm (12.06s)

- shopping ShoppingWebResource.orderShirts 11.87s
  - shopping GET 146ms
  - inventory available
    - inventory inventoryAsync 497ms
  - shopping POST 200ms
    - payments Pay
      - payments ProcessPayment 140ms
      - payments AuthorizePayment 51ms
      - payments FinishPayment 47ms
    - payments UpdateAccountAmount 1.22s
  - shopping GET-style/(id)/make 11.4s
  - styling StylingWebResource.make 11.4s
  - style POST 3.79s
shopping

Overview

See All shopping Traces

Request Rate

Error Rate

Duration (P95)

Top Requests

- GET-style/cd)/make: 120.00 Req.
- GET: 120.00 Req.
- ShoppingWebResource.orderShirts: 120.00 Req.
- POST: 120.00 Req.
- POST-delivery/(orderNum): 109.00 Req.

Top Failed Requests

- ShoppingWebResource.orderShirts: 16.00 Req.
- GET-style/cd)/make: 11.00 Req.
- POST-delivery/(orderNum): 5.00 Req.
- ShoppingWebResource.cancelShirtsOrder: 4.00 Req.
- POST-delivery/Cance/(orderNum): 4.00 Req.

Slowest Request

- ShoppingWebResource.orderShirts: 3.54 s
- GET-style/cd)/make: 3.09 s
- POST: 0.22 s
- ShoppingWebResource.updateInventory: 0.14 s
- ShoppingWebResource.getShoppingMenu: 0.13 s
Queries

- Source

100 * ts(jvm.memory.heap.usage, $replica) and (service-engine or service-query) and not tag-eval

- % heap / max_heap

((max(max(30m, max(ts(jvm.memory.heap.used, $replica) and (service-engine or service-query) and not tag="clover" and not tag="africa"), hosttags))) / (max(ts(jvm.memory.heap.max, $replica) and (service-engine or service-query), hosttags))) * 100
Cluster pps is below Target Sustained by 15% - 1:00 AM 05/05/19

- **Target Sustained by 15%**
  - we may need to decrease cluster size because the point rate on the cluster is too low compared to the threshold.
  - ID: 1496883449906

**Query:**

```sql
mavg(10m, 100 * (sum(rate(
  ts(dataingester.report-points, (tag=retired or tag=prod)) and ((tag="*-primary" or tag="*-secondary" or tag="*-tertiary")))) )
, hosttags
) + 0 * sum(rate(ts(dataingester.report-points, (tag="*-primary" or tag="*-secondary" or tag="*-tertiary"))), hosttags)) / ($(max(
  last(2d, ts(telegraf.config.globalAllowedRate, role=app and ((tag="*-primary" or tag="*-secondary" or tag="*-tertiary"))) and not tag=retired and not (tag=try or tag=brkt or tag=volatile-pps or tag=lyft or tag=nimba or tag=africa or classifier=q3))
) , hosttags
) * 2/3 )
) < 85
```

- **Points:** 19249
- **Query Time:** 2243 ms
‘3D’ Observability

The only thing that matters is customer happiness.

Metrics
immediate understanding.

Histograms
high-velocity data & outlier analysis (latency).

Tracing
Super-detailed, pinpoint failures and poor performance.