@gitbisect
Technical Writer/Evangelist
“Docs & Talks”
Travel Hacker & Whiskey Hunter

@datadoghq
SaaS-based monitoring
Trillions of data points per day
http://jobs.datadoghq.com
Our new monitoring product just watches Twitter and IRC for our name + "down".
COLLECTING DATA IS CHEAP; NOT HAVING IT WHEN YOU NEED IT CAN BE EXPENSIVE

SO INSTRUMENT ALL THE THINGS!
ANALOG FTW!

OUCH.

WAT?!

British Airways Union Blames Massive IT Failure On Outsourcing IT Jobs To India

The carrier cancelled hundreds of flights from London yesterday.

28/05/2017 19:51 GMT 28/05/2017 12:59 PM BST

LONDON — British Airways GMB union has blamed the airline's 2016 decision of outsourcing IT jobs to India as the reason behind cancelling all Saturday flights from
4 Qualities of Good Metrics

Not all metrics are equal
1. MUST BE WELL UNDERSTOOD
2. SUFFICIENT GRANULARITY

<table>
<thead>
<tr>
<th>RANK</th>
<th>PARTICIPANT</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Anthony ERVIN (USA)</td>
<td>21.40</td>
</tr>
<tr>
<td>2.</td>
<td>Florent MANAUDOU (FRA)</td>
<td>21.41</td>
</tr>
<tr>
<td>3.</td>
<td>Nathan ADRIAN (USA)</td>
<td>21.49</td>
</tr>
<tr>
<td>4.</td>
<td>Ben PROUD (GBR)</td>
<td>21.68</td>
</tr>
<tr>
<td>5.</td>
<td>Andrii GOVOROV (UKR)</td>
<td>21.74</td>
</tr>
<tr>
<td>6.</td>
<td>Bruno FRATUS (BRA)</td>
<td>21.79</td>
</tr>
<tr>
<td>6.</td>
<td>Bradley Edward TANDY (RSA)</td>
<td>21.79</td>
</tr>
<tr>
<td>8.</td>
<td>Simonas BILIS (LTU)</td>
<td>22.08</td>
</tr>
</tbody>
</table>

TW: @gitbisect @datadoghq
1 second
Peak 46%

1 minute
Peak 36%

5 minutes
Peak 12%
3. TAGGED & FILTERABLE
4. LONG-LIVED
RESOURCE METRICS
- Utilization
- Saturation
- Error
- Availability
P.S. - June 1! Mark your calendar!
Q: Are we losing money?
A: Can't answer that, but I can tell you what average CPU usage was 5ish mins ago.
WHAT TO PAGE ON?

PAGE ON
SYMPTOMS:
- WORK METRICS
- RESOURCE METRICS

INVESTIGATE USING
DIAGNOSTICS:
- WORK METRICS
- EVENTS
RECURSE UNTIL YOU FIND THE TECHNICAL CAUSES
SCALING & MONITORING POSTGRESQL AT DATADOG
MOAR RESOURCES!
MOAR INSTANCES!
HOW WE DO IT

REQUIREMENTS

▸ Write master is writeable, read replicas are readable!
HOW WE DO IT

REQUIREMENTS

▸ Write master is writeable, read replicas are readable!

▸ Read replicas are up to date and don’t lag
HOW WE DO IT

REQUIREMENTS

▸ Write master is writeable, read replicas are readable!

▸ Read replicas are up to date and don’t lag

▸ Additional read replicas can be provisioned quickly
SOLUTIONS

- PostgreSQL!
- WAL-E
  - https://github.com/wal-e/wal-e
Writes → Standbys → Repl → App1 Reads

App2 Reads
WHAT DO WE MONITOR AT DATADOG?
METRICS

WHAT METRICS DO WE GATHER?

connections  temp_files
commits  bgwriter.checkpoints_timed
rollbacks  bgwriter.checkpoints_requested
disk_read  bgwriterbuffers_checkpoint
buffer_hit  bgwriterbuffers_clean
rows_returned  bgwritermaxwritten_clean
rows_fetched  bgwriterbuffers_backend
rows_inserted  bgwriterbuffers_alloc
rows_updated  bgwriterbuffers_backend_fsync
rows_deleted  bgwriterwrite_time
database_size  bgwriter.sync_time
deadlocks  locks
temp_bytes  seq_scans
seq_rows_read  index_scans
index_rows_fetched  rows_hot_updated
live_rows  dead_rows
index_rows_read  index_rows_read
table_size  index_size
total_size  table.count
max_connections  percent_usage_connections
replication_delay  replication_delay_bytes
heap_blocks_read  heap_blocks_hit
index_blocks_read  index_blocks_hit
toast_blocks_read  toast_blocks_hit
toast_index_blocks_read  toast_index_blocks_hit
heap_blocks_read
index_blocks_read
max_connections
percent_usage_connections

TW: @gitbisect @datadoghq
Alert on work metrics, but resource metrics become work metrics? Alert on everything?
WHO TO ALERT?

Leadership → BUSINESS → SERVICES → EVENTS

SERVICE → COMPONENTS → EVENTS

COMPUTER → SERVERS → EVENTS

SERVERS → RESOURCES → EVENTS
WHO TO ALERT?

Developers

BUSINESS → SERVICES → EVENTS

SERVICE → COMPONENTS → EVENTS

COMPONENT → SERVERS → EVENTS

SERVER → RESOURCES → EVENTS
WHO TO ALERT?

PostgreSQL Team (Ops/SRE)
WHO TO ALERT?

BUSINESS ➔ SERVICES ➔ EVENTS

SERVICE ➔ COMPONENTS ➔ EVENTS

COMPONENT ➔ SERVERS ➔ EVENTS

SERVER ➔ RESOURCES ➔ EVENTS

Ops/SRE
POSTGRESQL WORK METRICS (AVAILABILITY)
ALERT ON WORK METRICS

WHAT ARE WE ALERTING ON?

- Base backup is too old
- Standby is missing
- Replication lag is too high
POSTGRESQL RESOURCE METRICS = OPS WORK METRICS (CAPACITY)
ALERT ON WORK METRICS

WHAT ARE WE ALERTING ON?

- Connection limit
- Disk
- Memory
- CPU
TW: @gitbisect @datadoghq
MONITORING TO IMPROVE PERFORMANCE
WHERE TO GET THE MOST PERFORMANCE GAINS?


1. Cut Activity
2. Slow Queries
3. Scale Stack
4. Fix Hardware
5. Postgresql.conf
## CUT ACTIVITY

<table>
<thead>
<tr>
<th>Name</th>
<th>Hits</th>
<th>Avg Latency</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>⭐️ SELECT key, org_id, source_typ...</td>
<td>6.70M</td>
<td>2.32 ms</td>
<td>4h 18min</td>
</tr>
<tr>
<td>⭐️ WITH sub_contexts SELECT key,...</td>
<td>6.30M</td>
<td>127 ms</td>
<td>9.3d</td>
</tr>
<tr>
<td>⭐️ WITH sub_contexts SELECT key,...</td>
<td>746k</td>
<td>38.9 ms</td>
<td>8h 4min</td>
</tr>
<tr>
<td>⭐️ WITH sub_contexts SELECT key,...</td>
<td>700k</td>
<td>15.8 ms</td>
<td>3h 3min</td>
</tr>
<tr>
<td>⭐️ WITH sub_contexts SELECT key,...</td>
<td>296k</td>
<td>130 ms</td>
<td>10h 38min</td>
</tr>
<tr>
<td>⭐️ WITH sub_contexts SELECT key,...</td>
<td>226k</td>
<td>42.6 ms</td>
<td>2h 40min</td>
</tr>
<tr>
<td>⭐️ SELECT t.oid, typarray FROM pg...</td>
<td>159k</td>
<td>23.0 ms</td>
<td>1h 1min</td>
</tr>
</tbody>
</table>
## SLOW QUERIES

<table>
<thead>
<tr>
<th>Name</th>
<th>Hits</th>
<th>Avg Latency</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>WITH sub_contexts SELECT key,</td>
<td>90</td>
<td>2.17 s</td>
<td>196 s</td>
</tr>
<tr>
<td>WITH sub_contexts SELECT key,</td>
<td>58</td>
<td>1.49 s</td>
<td>86.7 s</td>
</tr>
<tr>
<td>WITH sub_contexts SELECT key,</td>
<td>49</td>
<td>1.09 s</td>
<td>53.2 s</td>
</tr>
<tr>
<td>WITH sub_contexts SELECT key,</td>
<td>194</td>
<td>759 ms</td>
<td>147 s</td>
</tr>
<tr>
<td>WITH sub_contexts SELECT key,</td>
<td>45</td>
<td>750 ms</td>
<td>33.7 s</td>
</tr>
<tr>
<td>WITH sub_contexts SELECT key,</td>
<td>22</td>
<td>740 ms</td>
<td>16.3 s</td>
</tr>
<tr>
<td>WITH sub_contexts SELECT key,</td>
<td>70</td>
<td>559 ms</td>
<td>39.1 s</td>
</tr>
</tbody>
</table>
PERFORMANCE: LATENCY VS POTENTIAL
HOW DO YOU DEFINE PERFORMANCE?

SELECT * FROM table_x
WHERE nonindexed_col=1

SELECT * FROM table_x
JOIN table_y ON
    table_x.foo=table_y.bar
WHERE table_y.indexed_col=1
PERFORMANCE: RAM VS DISK
“Aside from shared_buffers, the most important memory-allocation parameter is work_mem... Raising this value can dramatically improve the performance of certain queries...”

ROBERT HAAS
“Aside from shared_buffers, the most important memory-allocation parameter is work_mem... Raising this value can dramatically improve the performance of certain queries, but it's important not to overdo it.”

ROBERT HAAS
FINDING **INEFFICIENT** QUERIES
EXPLAIN ANALYZE


- Explain displays the execution plan
LATENCY VS POTENTIAL

EXPLAIN ANALYZE


- Explain displays the execution plan
- Analyze runs it and gathers stats
EXPLAIN ANALYZE

Merge Right Join  (cost=25870.55..31017.51 rows=229367 width=92) (actual time=2884.501..5147.047 rows=354834 loops=1)
  Merge Cond: (a.uid = b.uid)
  ->  Index Scan using foo on bar a  (cost=0.00..537.29 rows=9246 width=27) (actual time=0.049..41.782 rows=9246 loops=1)
  ->  Materialize  (cost=25870.49..27204.80 rows=106745 width=81) (actual time=2884.413..3804.537 rows=354834 loops=1)
     ->  Sort  (cost=25870.49..26137.35 rows=106745 width=81) (actual time=2884.406..3099.732 rows=111878 loops=1)
       Sort Key: b.uid
       Sort Method: external merge  Disk: 8928kB

... Total runtime: 5588.105 ms
(14 rows)

POSTGRESQL PERFORMANCE

SUMMARY

1. Remember the 4 qualities of good metrics
   1. Well understood
   2. Sufficiently granular
   3. Tagged & filterable
   4. Long-lived
POSTGRESQL PERFORMANCE

SUMMARY

2. Understand the difference between work metrics, resource metrics & events
3. Alert on the appropriate work metrics
QUESTIONS?

JASON YEE

@gitbisect

jason.yee@datadoghq.com