 PostgreSQL monitoring with *pgwatch2*

Kaarel Moppel / PostgresConf US 2018
www.cybertec-postgresql.com
Why to monitor

- Failure / Downtime detection
- Slowness / Performance analysis
- Proactive predictions
- Maybe wasting money?
Different levels of Database monitoring

- Service availability level
- System level monitoring
- Database
- Application level would be nice also
PostgreSQL land

- Log analysis
- Stats Collector
  - Dynamic views
    - `pg_stat_activity`, `pg_stat_(replication|wal_receiver)`, `pg_locks`, `pg_stat_ssl`, `pg_stat_progress_vacuum`
  - Cumulative views
    - Most `pg_stat_*` views
    - Long uptimes cause “lag” for problem detection
- NB! Not all `track_*` parameters enabled by default
PostgreSQL Monitoring Tools
No shortage of tools

https://wiki.postgresql.org/wiki/Monitoring
Approaches

- Ad hoc
- Continuous monitoring frameworks
  - Cloud / SaaS / APM / Generic
  - DIY / Open Source
Postgres specific Open Source continuous monitoring frameworks
Postgres specific

- pghero
- PoWa (server side, quite advanced - pg_qualstats, pg_stat_kcache)
- PgObserver (client side + ad hoc)
- pgwatch2 (client side)
- ...

Kaarel Moppel / PostgresConf US 2018
www.cybertec-postgresql.com
pgwatch2
Main principles - why another tool?

- 1-minute setup
  - Docker first
- User defined visuals / Dashboarding
- Non-invasive
  - No extensions for main functionality
- Easy extensibility
  - SQL defined metrics
- Do minimal work needed, use existing good software
Architecture components

- Web UI for administration
  - Python / Bootstrap
- Metrics gathering daemon
  - Go
- Config database
  - Postgres
- Metrics storage layer
  - InfluxDB (Graphite possible)
- Easy dashboarding with intuitive data discovery
  - Grafana
Features (1)

- “Ready to go”
  - Defaults cover almost all pg_stat* views
- Supports Postgres 9.0+ (older versions also possible)
- Security
- SSL between all components possible
- Admin / normal user separation
- Custom metrics via SQL, also for business domain!
- Reuse of existing Postgres, Grafana, InfluxDB installation possible
- Can be integrated with Kubernetes/OpenStack
Features (2)

- Possible to monitor all databases of a cluster automatically
- Change detection
  - Added/changed/deleted table/index/sproc/config events
- PgBouncer statistics support
- AWS CloudWatch support
- Alerting easily possible with Grafana
  - Kapacitor (“K” from InfluxData’s TICK stack)
- Extensible data presentation - Grafana has plugins!
1. `docker run -d -p 3000:3000 -p 8080:8080 --name pw2 cybertec/pgwatch2`
2. Wait some seconds and open browser at localhost:8080
3. Insert your DB connection strings and wait some minutes
4. Start viewing/editing dashboarding!
### Top queries by total runtime

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>49884409</td>
<td>UPDATE pgbench_accounts SET abalance = abalance + 7 WHERE aid = 7;</td>
</tr>
<tr>
<td>35135818</td>
<td>UPDATE pgbench_tellers SET balalance = balalance + 7 WHERE tid = 7;</td>
</tr>
<tr>
<td>2037868801</td>
<td>select (extract? from now()) * 7?int8 as epoch_ms, load_1min, load_5min, load_15min from public.get_load_avg(); needs the psql tool proc from 'metric_fetching_helpers' folder</td>
</tr>
</tbody>
</table>

### Top queries by avg runtime

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>811055313</td>
<td>select (extract? from now()) * 7?int8 as epoch_ms, approx_free_percent, approx_free_space as approx_free_space, b from public.get_table_bloat_approx() where approx_free_space &gt; 7</td>
</tr>
<tr>
<td>2037868801</td>
<td>select (extract? from now()) * 7?int8 as epoch_ms, load_1min, load_5min, load_15min from public.get_load_avg(); needs the psql tool proc from 'metric_fetching_helpers' folder</td>
</tr>
</tbody>
</table>

### Top queries by calls

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>2351822814</td>
<td>SELECT abalance FROM pgbench_accounts WHERE aid = 7;</td>
</tr>
<tr>
<td>2775512216</td>
<td>INSERT INTO pgbench_history (txid, bid, aid, delta, mixtime) VALUES (?, ?, ?, ?), CURRENT_TIMESTAMP;</td>
</tr>
<tr>
<td>49884409</td>
<td>UPDATE pgbench_accounts SET abalance = abalance + 7 WHERE aid = 7;</td>
</tr>
</tbody>
</table>

### Top queries by IO

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>811055313</td>
<td>select (extract? from now()) * 7?int8 as epoch_ms, approx_free_percent, approx_free_space, b from public.get_table_bloat_approx() where approx_free_space &gt; 7</td>
</tr>
<tr>
<td>49884409</td>
<td>UPDATE pgbench_accounts SET abalance = abalance + 7 WHERE aid = 7;</td>
</tr>
</tbody>
</table>
Alerting
Grafana

- Easy setup, point and click
- Most important alerting services covered
  - Email
  - Slack
  - PagerDuty
  - Web hooks
  - Kafka
  - ...
- Graph panel only currently
Graph

Alert Config

Name: Site Logins Too Low
Evaluate every: 10s

Conditions

WHEN: avg()
OF: query (A, 5m, now)
IS BELOW: 20

If no data points or all values are null, SET STATE TO: No Data
Anomaly detection
Kapacitor

Part of the InfluxData’s TICK stack

- Harder to get going but very powerful!
- Features
  - Extensive math/string processing support
  - Statistical data mangling
  - UDF-s
  - Alert topics - pub/sub
  - Stream caching (e.g. last 10min moving average)
  - Stream redirection - store transformed data back into InfluxDB
stream
  |from()
    .measurement('cpu')
  |alert()
    .crit(lambda: "usage_idle" < 70)
    .log('/tmp/alerts.log')
    .email()
|from()
  |.measurement('cpu')
|groupBy('service', 'datacenter')
|window()
  |.period(1m)
|percentile('load_1min', 95.0)
|eval(lambda: sigma("percentile"))
  |.as('sigma')
>alert()
  |.id('{{ .Name }}/{{ index .Tags "service" }}/{{ index .Tags "datacenter"}}')
  |.message('{{ .ID }} is {{ .Level }} cpu-95th:{{ index .Fields "percentile" }}')
  |.crit(lambda: "sigma" > 3.0)
pgwatch2 - What’s next?
Improvement areas

- More system level metrics
- Log parsing
- Per metric timeouts and activity times
- Fully automatic Docker updates
- ???

User input very much appreciated!
github.com/cybertec-postgresql/pgwatch2

Austria / Switzerland / Estonia
Web: www.cybertec-postgresql.com
Github: github.com/cybertec-postgresql
Twitter: @PostgresSupport