NoSQL to PostgreSQL
Adventures in Migrations

Phil Hildebrand
Moz
Where we’ll be going

1. A little background of how we got here
2. Look at the few things we did right along the way in our adventure
3. Dive into a few of the fun things we learned (some the hard way)
4. Talk a little bit about what’s next and what we’ll do different
How’d we get here?

A desire for the best thing for the job led to

- Location and customer reviews in json documents
- NodeJS, RethinkDB, MySQL, SortDB, Elastic Search

...the ever expanding technology stack
How’d we get here?

A need for simplicity drove

- PostgreSQL for JSON, Transactions, and Materialized views
- Easier integration for business analytics
- Ease for onboarding new devs

...reduce technology stack
Eyes wide open?

- PostgreSQL is not Oracle, and it’s certainly not MySQL

- Command line client and the art of backslashology
PostgreSQL is not Oracle, and it's certainly not MySQL

- Command line client and the art of backslashology
- Reviewing the core architecture
Eyes wide open?

PostgreSQL is not Oracle, and it's certainly not MySQL

- Command line client and the art of backslashology
- Reviewing the core architecture
- Configuration Options (PGTune by Alexey Vasiliev)
Eyes wide open?

PostgreSQL is not Oracle, and it's certainly not MySQL

- Command line client and the art of backslashology
- Reviewing the core architecture
- Configuration Options ([PGTune by Alexey Vasiliev](https://gist.github.com/phil-hildebrand/c59f9739bbd745f70b6c1e513931873b))
- `PG_ <underscore>` system view foo
- Inheritance? That's a thing?

```sql
-- View all roles with attributes and parent role using query
/*
 * equivalent query for /du command, found using psql -E
 */

SELECT r.rolname, r.rolsuper, r.rolinherit, r.rolcreaterole, r.rolcreatedb, r.rolcanlogin, r.rolcanospecial, r.rolvaliduntil
FROM pg_catalog.pg_auth_members m
JOIN pg_catalog.pg_roles b ON (m.roleid = b.oid)
WHERE m.member = r.oid
    AS memberof,
    r.rolreplication, r.rolbypassris,
FROM pg_catalog.pg_roles r
WHERE r.rolname LIKE 'pg_'
ORDER BY 1;
/*
List of roles
<table>
<thead>
<tr>
<th>Role name</th>
<th>Attributes</th>
<th>Member of</th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td>Create DB</td>
<td>{}</td>
</tr>
<tr>
<td>backups</td>
<td>Replication</td>
<td>{}</td>
</tr>
<tr>
<td>datalog</td>
<td></td>
<td>{mon}</td>
</tr>
<tr>
<td>dba</td>
<td>Superuser, Create role, Create DB</td>
<td>{}</td>
</tr>
<tr>
<td>dev</td>
<td></td>
<td>{}</td>
</tr>
<tr>
<td>devadmin</td>
<td>Create DB</td>
<td>{}</td>
</tr>
</tbody>
</table>
```

[https://gist.github.com/phil-hildebrand/c59f9739bbd745f70b6c1e513931873b](https://gist.github.com/phil-hildebrand/c59f9739bbd745f70b6c1e513931873b)
Eyes wide open?

**PostgreSQL is not Oracle, and it’s certainly not MySQL**

- Reviewing the core architecture
- Command line client and the art of backslashology
- PG_ <underscore> system view foo
- Inheritance? That’s a thing?

**But it’s still a relational database**

- Memory allocations
- Fragmentation
- Archive logging
- Replication
Outage #1: Materialized View Refresh

Historical sets of Reviews

- Partitioned by year
- Accessed by materialized views
- New reviews trickle in nightly

- Key functionality depends on a single materialized view
Outage #2: Out of Space

- 3X wal log space?
Outage #2: Out of Space

- 3X wal log space?
- It’s only 400 tps at peak
- 150k tps/day (over an 8 hour period)
- 200GB / day
- 1.5 k / transaction.
Outage #3: Torn Page

Corruption? But it’s a database...

- I Love active forums!!!!

The following bug has been logged on the website:

Bug reference: 15570
Logged by: Phil Hildebrand
Email address: phil(dot)hildebrand(at)moz(dot)com
PostgreSQL version: 10.6
Operating system: Ubuntu 16.04.5 LTS (Linux)
Description:

While running vacuum analyze before a materialized view refresh we started seeing the following error in the logs:

ERROR: MultiXactId 1483293027 has not been created yet -- apparent wraparound in the error logs

Both selects against and pg_dump of the table fail with the same error.

We were able narrow down the selects to at least one row / page that seems to have problems:

```
select ctid,id from reviews_2018 where ctid = '(382604,16)':::tid;
---
  ctid | id
-----------------------------
(382604,16) | 00ec91e42d21ce0b818fe8f63ab94c66
```

```
select ctid,id from reviews_2018 where id = '00ec91e42d21ce0b818fe8f63ab94c66';
ERROR: MultiXactId 1483293027 has not been created yet -- apparent wraparound
```

We would not expect to see this without some kind of system issue, but there is not indication there were any system or disk issues/errors at the time.

As a workaround, we restored the table from a previous backup.
Outage #3: Torn Page

Corruption? But it’s a database...

- I Love active forums!!!!
- Special call out to these four forum members that jumped into the muck with us:
  - Alvaro Herrera
  - Andrew Gierth
  - Michael Paquier
  - Stephen Frost

The following bug has been logged on the website:

**Bug reference:** 15570  
**Logged by:** Phil Hildebrand  
**Email address:** phil(dot)hildebrand(at)moz(dot)com  
**PostgreSQL version:** 10.6  
**Operating system:** Ubuntu 16.04.5 LTS (Linux)  
**Description:**

While running vacuum analyze before a materialized view refresh we started seeing the following error in the logs:

```
ERROR: MultiXactId 1483293027 has not been created yet -- apparent wraparound in the error logs
```

Both selects against and pg_dumps of the table fail with the same error.

We were able narrow down the selects to at least one row / page that seems to have problems:

```
select ctid, id from reviews_2018 where ctid = '(382604,16)':::tid;  
<table>
<thead>
<tr>
<th>ctid</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
(382604,16) | 00ec91e42d21ce0b818fe8f63ab94c66
```

```
select ctid, id from reviews_2018 where id = '00ec91e42d21ce0b818fe8f63ab94c66';  
ERROR: MultiXactId 1483293027 has not been created yet -- apparent wraparound
```

We would not expect to see this without some kind of system issue, but there is not indication there were any system or disk issues/errors at the time.

As a workaround, we restored the table from a previous backup.
Outage #3: *Torn Page*

Corruption? But it's a database...

- I Love active forums!!!!
- Replication didn't help
- Finally got to use those backups

---

The following bug has been logged on the website:

**Bug reference:** 15570  
**Logged by:** Phil Hildebrand  
**Email address:** phil(h dot)hildebrand(at)moz(dot)com  
**PostgreSQL version:** 10.6  
**Operating system:** Ubuntu 16.04.5 LTS (Linux)

**Description:**

While running vacuum analyze before a materialized view refresh we started seeing the following error in the logs:

```
ERROR: MultiXactId 1483293027 has not been created yet -- apparent wraparound in the error logs.
```

Both selects against and pg_dumps of the table fail with the same error.

---

000000fc0 69 6f 6e 20 76 65 72 79 20 73 65 72 6f 75 73 65 67 69 67 68 74 68 65 6d 6c 20 70 72 6f 74 20 73 75 21 78 73 74 75 73 74 75 73

Outage #3: Torn Page

- Validation Walk Through

The following bug has been logged on the website:

Bug reference: 15570
Logged by: Phil Hildebrand
Email address: phil@hildebrand@moz.com
PostgreSQL version: 10.6
Operating system: Ubuntu 16.04.5 LTS (Linux)
Description:

While running vacuum analyze before a materialized view refresh we started seeing the following error in the logs:

ERROR: MultiXactId 1483293027 has not been created yet -- apparent wraparound in the error logs

Both selects against and pg_dumps of the table fail with the same error.
If we did it all over again...

*What was helpful:*

- Read the Docs
- Creating `pg_*` cheat sheets
- Automating installation, configuration, replication
- Command client cheat sheets
- Migrating smaller/simpler data sets first

*What was maybe not so helpful:*

- Skipping load tests
- Waiting to get involved till migration time
Next Up...

Migrating the next piece into postgres

- Chose a MySQL based service to migrate
- Much better understanding of replication and materialized views
- Looking into changing the wal segment size
- Easier do to all the automation we now have in place

First new service with GIS features:

- Postgres 11
- More load testing up front
Helpful Resources

**PostgreSQL Docs**: [https://www.postgresql.org/docs/current/](https://www.postgresql.org/docs/current/)

**Percona PostgreSQL Blog**: [https://www.percona.com/blog/category/postgresql/](https://www.percona.com/blog/category/postgresql/)

**Several Nines Become a PostgreSQL DBA Blog**: [https://severalnines.com/blog?series=690](https://severalnines.com/blog?series=690)

**PG Tune**: [https://pgtune.leopard.in.ua/#/](https://pgtune.leopard.in.ua/#/)

**PG Bench**: [https://www.postgresql.org/docs/current/pgbench.html](https://www.postgresql.org/docs/current/pgbench.html)

**DBLoader**: [https://github.com/phil-hildebrand/dbloader](https://github.com/phil-hildebrand/dbloader)

**Gist Queries**: [https://gist.github.com/phil-hildebrand/c59f9739bbd745f70b6c1e513931873b](https://gist.github.com/phil-hildebrand/c59f9739bbd745f70b6c1e513931873b)

**Gist Page Inspection**: [https://gist.github.com/phil-hildebrand/27e2a5029f1bca725eea27f995edde20](https://gist.github.com/phil-hildebrand/27e2a5029f1bca725eea27f995edde20)

**Forum Issue**: [https://www.postgresql.org/message-id/flat/15570-d920421b445027cc%40postgresql.org](https://www.postgresql.org/message-id/flat/15570-d920421b445027cc%40postgresql.org)
Questions?

phil.hildebrand@moz.com
Rate My Session

Introducing gh-ost: triggerless, painless, trusted online schema migrations

Rate & Review

Feedback (optional)

SUBMIT