



Secrets of PostgreSQL Performance

Frank Wiles

Revolution Systems

Measure Everything Before And After

**DON'T DO
DUMB
THINGS!**

Don't...

- Assume PostgreSQL is like MySQL
- Database server doing double duty
- Disk Contention
- Retrieve more data than you need

**DO SMART
THINGS!**

**Cache and cache
some more**

**No seriously, cache
the crap out of
everything**

**You probably STILL
do not have enough
caching...**

**Watch your query
counts!!!**

The “Big 3” Tuning Parameters

<p>shared_buffers</p>	<p>Set to 25% of available RAM and move up/down 5% to find sweet spot</p>
<p>effective_cache_size</p>	<p>Planning hint that tells PG how much RAM it can expect for OS disk cache. Set to 50-75% of available RAM</p>
<p>work_mem</p>	<p>Per process amount of ORDER BY space. 5MBs is a good starting point</p>

Other parameters to tune...

wal_buffers	Set to 16MB and forget it
checkpoint_segments	Increase to at least 10
maintenance_work_mem	50MB for every GB of RAM
synchronous_commit	Turn off with data loss risks

Hardware Considerations

**As much RAM as
possible...**

Disks. Lots of disks.

Not all disks are created equal. Faster disks make a huge difference.

Configuration of disks matters a lot. RAID-5 is bad. RAID-1+0 good.

Separate out the WAL onto it's own disk can 4x write performance

**Then worry about
CPU speed**

Use pgbouncer to pool connections

Tablespaces

Consider putting archive or legacy data onto slower drives/volumes

Separate your data and indexes onto different volumes if possible

This applies even on fancy SANs

SSDs

In the Cloud

- Remember: You're sharing disks and you don't even know how!
- You can get the best of both worlds with Rackspace and their Cloud Connect product
- Using EBS volumes and software RAID is best (but somewhat scary) option on AWS

OS Considerations

- Your choice of filesystem matters. Don't use a journaled filesystem for your WAL. It is a journal.
- Use XFS for maximum performance.
- *noatime*

Common Django specific problems

select_related()

SQL Queries



Time (ms)	Action	Stacktrace	Query
6.91	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "quotes_quote"."id", "quotes_quote"."author_id", "quotes_quote"."slug", "quotes_quote"."featured", "quotes_quote"."quote", "quotes_quote"."extra_content", "quotes_quote"."views", "quotes_quote"."quote_of_the_day", "quotes_quote"."enable_comments", "quotes_quote"."source_id", "quotes_quote"."created" FROM "quotes_quote" WHERE "quotes_quote"."slug" = E'resolved-experience-committee-problem-worked-difficult'</pre>
0.78	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "django_session"."session_key", "django_session"."session_data", "django_session"."expire_date" FROM "django_session" WHERE ("django_session"."session_key" = E'8f6ede4e0945c8ca7a92cb83af2c3bc6' AND "django_session"."expire_date" > E'2011-09-06 15:16:08.039900')</pre>
2.16	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "authors_author"."id", "authors_author"."first_name", "authors_author"."middle_name", "authors_author"."last_name", "authors_author"."prefix", "authors_author"."suffix", "authors_author"."display_name", "authors_author"."slug", "authors_author"."birthday", "authors_author"."died", "authors_author"."born_at", "authors_author"."died_at", "authors_author"."alive", "authors_author"."short_description", "authors_author"."description", "authors_author"."quote_count", "authors_author"."created", "authors_author"."modified" FROM "authors_author" WHERE "authors_author"."id" = 1060</pre>
6.20	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "quotes_quote"."id", "quotes_quote"."author_id", "quotes_quote"."slug", "quotes_quote"."featured", "quotes_quote"."quote", "quotes_quote"."extra_content", "quotes_quote"."views", "quotes_quote"."quote_of_the_day", "quotes_quote"."enable_comments", "quotes_quote"."source_id", "quotes_quote"."created" FROM "quotes_quote" WHERE "quotes_quote"."author_id" = 1060 ORDER BY "quotes_quote"."views" DESC LIMIT 10</pre>
0.48	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "authors_author"."id", "authors_author"."first_name", "authors_author"."middle_name", "authors_author"."last_name", "authors_author"."prefix", "authors_author"."suffix", "authors_author"."display_name", "authors_author"."slug", "authors_author"."birthday", "authors_author"."died", "authors_author"."born_at", "authors_author"."died_at", "authors_author"."alive", "authors_author"."short_description", "authors_author"."description", "authors_author"."quote_count", "authors_author"."created", "authors_author"."modified" FROM "authors_author" WHERE "authors_author"."id" = 1060</pre>
0.36	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "authors_author"."id", "authors_author"."first_name", "authors_author"."middle_name", "authors_author"."last_name", "authors_author"."prefix", "authors_author"."suffix", "authors_author"."display_name", "authors_author"."slug", "authors_author"."birthday", "authors_author"."died", "authors_author"."born_at", "authors_author"."died_at", "authors_author"."alive", "authors_author"."short_description", "authors_author"."description", "authors_author"."quote_count", "authors_author"."created", "authors_author"."modified" FROM "authors_author" WHERE "authors_author"."id" = 1060</pre>

Hide »

Time

CPU: 230.89ms (268.18ms)

Settings

HTTP Headers

Request Vars

Templates

SQL

18 QUERIES IN 24.71MS

Signals

Logging

0 MESSAGES

REVSYS

revolution systems

SQL Queries



Hide »

Time (ms)	Action	Stacktrace	Query
4.77	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "quotes_quote"."id", "quotes_quote"."author_id", "quotes_quote"."slug", "quotes_quote"."featured", "quotes_quote"."quote", "quotes_quote"."extra_content", "quotes_quote"."views", "quotes_quote"."quote_of_the_day", "quotes_quote"."enable_comments", "quotes_quote"."source_id", "quotes_quote"."created", "authors_author"."id", "authors_author"."first_name", "authors_author"."middle_name", "authors_author"."last_name", "authors_author"."prefix", "authors_author"."suffix", "authors_author"."display_name", "authors_author"."slug", "authors_author"."birthday", "authors_author"."died", "authors_author"."born_at", "authors_author"."died_at", "authors_author"."alive", "authors_author"."short_description", "authors_author"."description", "authors_author"."quote_count", "authors_author"."created", "authors_author"."modified" FROM "quotes_quote" INNER JOIN "authors_author" ON ("quotes_quote"."author_id" = "authors_author"."id") WHERE "quotes_quote"."slug" = E'resolved-experience-committee-problem-worked-difficult'</pre>
0.64	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "django_session"."session_key", "django_session"."session_data", "django_session"."expire_date" FROM "django_session" WHERE ("django_session"."session_key" = E'8f6ede4e0945c8ca7a92cb83af2c3bc6' AND "django_session"."expire_date" > E'2011-09-06 15:22:12.154459')</pre>
1.62	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "quotes_quote"."id", "quotes_quote"."author_id", "quotes_quote"."slug", "quotes_quote"."featured", "quotes_quote"."quote", "quotes_quote"."extra_content", "quotes_quote"."views", "quotes_quote"."quote_of_the_day", "quotes_quote"."enable_comments", "quotes_quote"."source_id", "quotes_quote"."created", "authors_author"."id", "authors_author"."first_name", "authors_author"."middle_name", "authors_author"."last_name", "authors_author"."prefix", "authors_author"."suffix", "authors_author"."display_name", "authors_author"."slug", "authors_author"."birthday", "authors_author"."died", "authors_author"."born_at", "authors_author"."died_at", "authors_author"."alive", "authors_author"."short_description", "authors_author"."description", "authors_author"."quote_count", "authors_author"."created", "authors_author"."modified" FROM "quotes_quote" INNER JOIN "authors_author" ON ("quotes_quote"."author_id" = "authors_author"."id") ORDER BY "quotes_quote"."views" DESC LIMIT 10</pre>
0.26	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "quotes_quote"."id", "quotes_quote"."author_id", "quotes_quote"."slug", "quotes_quote"."featured", "quotes_quote"."quote", "quotes_quote"."extra_content", "quotes_quote"."views", "quotes_quote"."quote_of_the_day", "quotes_quote"."enable_comments", "quotes_quote"."source_id", "quotes_quote"."created" FROM "quotes_quote" WHERE "quotes_quote"."id" = 14915</pre>
1.23	SELECT EXPLAIN	Toggle Stacktrace	<pre>SELECT "django_comments"."id", "django_comments"."content_type_id", "django_comments"."object_pk", "django_comments"."site_id", "django_comments"."user_id", "django_comments"."user_name", "django_comments"."user_email", "django_comments"."user_url",</pre>

Time

CPU: 84.96ms (98.20ms)

Settings

HTTP Headers

Request Vars

Templates

SQL

6 QUERIES IN 8.86ms

Signals

Logging

0 MESSAGES

REVSYS

revolution
systems

Queries in loops...

Proper indexing

COUNT() is SLOW!

Model ordering

Indexes

```
CREATE INDEX <name> ON <table> (<column>);
```

Multicolumn Indexes

```
CREATE INDEX <name> ON <table> (<column1>,  
    <column2>, <columnN>);
```

Partial Indexes

```
CREATE INDEX open_tickets_idx ON tickets (open)  
WHERE open = 't';
```

Performance Tools

Django Debug Toolbar

Slow Query Log

`log_min_duration = 1000`

pgfouine

<http://pgfouine.projects.postgresql.org/>

EXPLAIN

Sadly EXPLAIN output does not fit well on a slide....

The Dark Arts

Statistics

default_statistics_target = 10

```
ALTER <table> ALTER <column> SET STATISTICS  
    <integer>;
```

Consider Triggers

Do you have a lot of this...

```
for x in Post.objects.all():  
    if x.status == 'Active':  
        ActivePosts.objects.create(original=x.id)
```

Or overwritten save() methods that create new rows?

Table Partitioning

- Break large tables into smaller tables by date or another partitioning scheme that fits your data
- Think of it as internal sharding
- DROP <table> instead of long slow DELETE FROM <table> WHERE created <= '2000-01-01'

<http://www.postgresql.org/docs/current/static/ddl-partitioning.html>

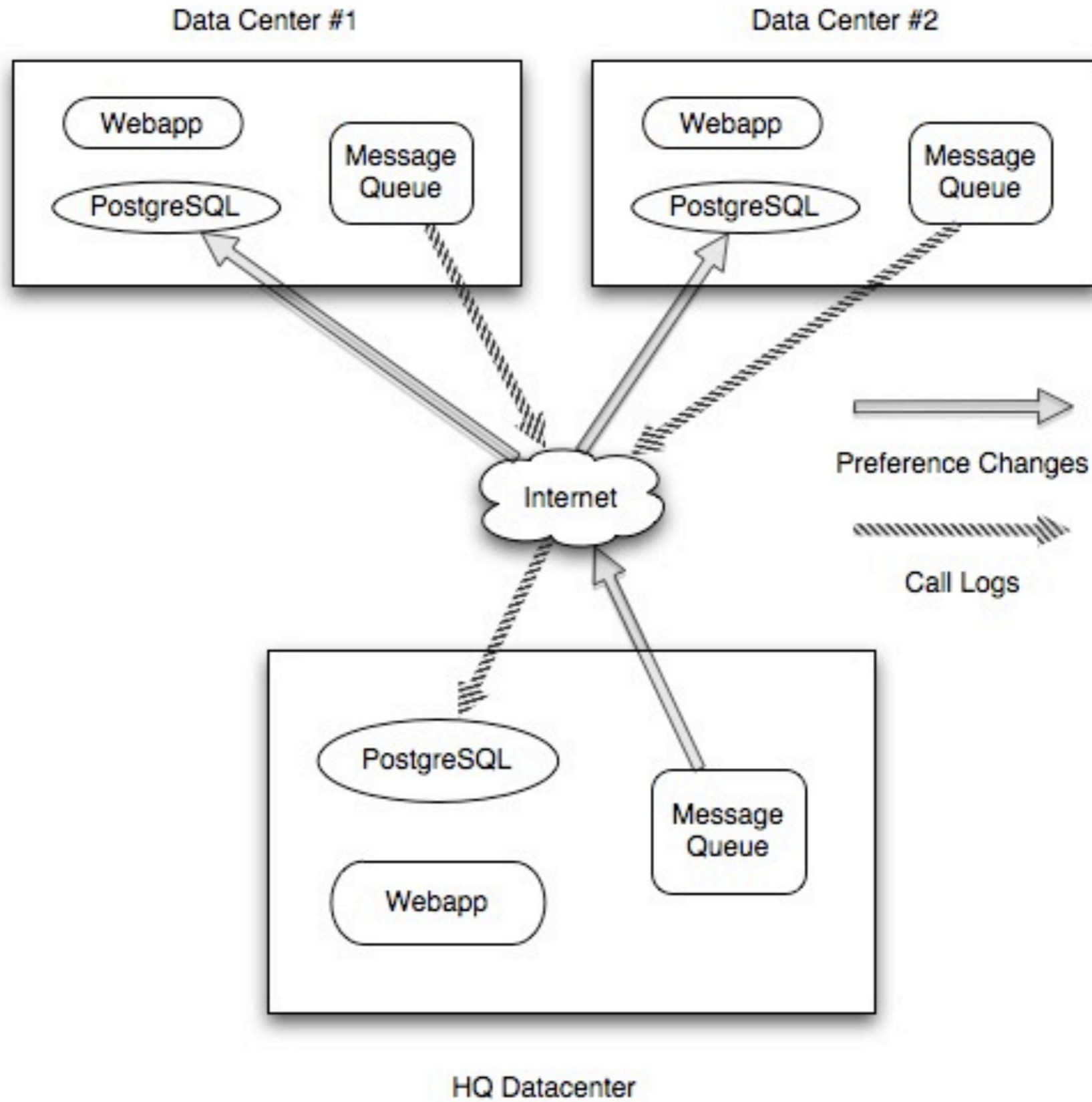
Materialized Views

Real tables populated via triggers based on actions
in other tables

Crazy Replication

The problem...

- Making phone calls from 4+ data centers all over the globe
- Web app at HQ for users to adjust preferences and view call log history/status
- Frequent connectivity issues between data centers
- HUGE legacy app that couldn't quickly be adjusted



Questions???

www.revsys.com

@fwiles / @revsys

frank@revsys.com