



# What is the Future of PostgreSQL?

- Robert Haas | PGConf.ASIA 2018

# The Future Looks Good!

- Increasing popularity.
  - Big companies like Amazon, Microsoft, Fujitsu, and NTT are taking an interest in PostgreSQL.
  - Ranking rising quickly on db-engines.com.
- Great new features, especially in v9.6 – v11.
  - Logical Replication
  - Parallelism
  - Partitioning
  - Stored Procedures
  - Just-in-Time Compilation.
- But there are also new challenges: large-scale analytics, the cloud, database migration.

# PostgreSQL Popularity

Date	Rating	Increase vs. Prior Year	% Increase
December 2015	280.092	+26.087	+10%
December 2016	330.016	+49.924	+17%
December 2017	385.43	+55.414	+16%
December 2018	460.636	+75.206	+19%

Source: db-engines.com

- PostgreSQL is the #4 system, still far behind Oracle, MySQL, and SQL Server – but gaining quickly.
- Overall, from December 2015 – December 2018, PostgreSQL's rating increased by 64%; the ratings of all three of the top systems *decreased* by 8 – 15% during the same period.

# Major New Features: v9.6 - v11

- *Logical Replication*. An amazing new feature! Allows for partial replication and cross-version replication, and doesn't replicate database bloat or cause query cancellations.
- *Parallelism*. For the first time, PostgreSQL can use more than one CPU for the same task.
- *Partitioning*. Easier to set up than table inheritance, and much faster, especially in v11.
- *Stored Procedures*. Lets you put more logic in the database and more easily migrate code from other systems.
- *Just-in-Time Compilation*. Speeds up analytic queries, especially those with complex expressions.

# Caveat

- These are just some of the major new features.
- There are another 10+ major features listed in the release notes for these releases.
- These are my picks, but somebody else might have a different list.
- I picked these five by looking at –
  - what the feature does today, and
  - where it could go in the future.

# Refinement Needed!

- Even though these are already great features, there is room for further improvement.

# Major New Features: v9.6 - v11

- *Logical Replication*. An amazing new feature! Allows for partial replication and cross-version replication, and doesn't bloat or cause query cancellations.
  - Doesn't handle failover to standbys.
  - Difficult to avoid downtime when performing major version upgrades – see Atsushi Torikoshi's talk just after lunch.
  - Multi-master replication requires third-party tools.
- *Parallelism*. For the first time, PostgreSQL can use more than one CPU for the same task.
  - Some simple queries run much faster, but more complex queries often see little or no benefit.
  - The only maintenance command supported in current releases is CREATE INDEX – for btree indexes only.

# Major New Features: v9.6 - v11

- *Partitioning*. Easier to set up than table inheritance, and much faster, especially in v11.
  - Still can easily be slower than no partitioning.
  - Some SQL features still don't work on a partitioned table the way they do on a plain table (e.g. can't be referenced by a foreign key).
- *Stored Procedures*. Lets you put more logic in the database and more easily migrate code from other systems.
  - Still missing some related features that are present in other systems, e.g. schema variables.
- *Just-in-Time Compilation*. Speeds up analytic queries, especially those with complex expressions.
  - Could be applied more widely and optimized better.



# Work is in Progress!

- Improvements in most of these areas – and many others which I didn't have time to mention – are already in progress.
- It's too early to say exactly how these developments will affect PostgreSQL 12, 13, and beyond, but expect progress on many of them.
- There will be talks later in the conference about many of these topics, covering at least what is in the current releases and in some cases future work as well, so please attend those talks to learn more.

# New Development Drivers

- *Large-Scale Analytics.* PostgreSQL's core strength is often considered to be OLTP, but many users turn to PostgreSQL for analytic workloads, and data sets are growing.
- *Cloud.* The traditional way to deploy software – including PostgreSQL – is on a dedicated server in the data center, but users increasingly want to deploy virtual machines, containers, or in the cloud.
- *Database Migration.* The PostgreSQL community has sometimes been skeptical of compatibility features, but there is new emphasis on helping users move away from proprietary database engines.

# Large-Scale Analytics: It All Takes Longer

- *Slower Queries.* Parallel query and just-in-time compilation help, but they are not sufficient.
- *Slower Maintenance Operations.* B-tree index creation runs 2-3x faster with parallelism, but other maintenance operations such as backup can't use parallelism yet.
- *Slower Backup and Restore.* Copying a lot of data with `pg_dump` takes a very long time. Even with hot backup, it's still slow if your database is big enough.

# Cloud: Non-Stop Chaos

- *Unattended Operation.* There is no DBA.
- *Flash Mobs.* Load can increase and decrease very quickly, and may exceed what a single server can bear.
- *Zero Downtime.* Even at night, it's daytime somewhere.
- *Global Reach.* Must be able to run transactions quickly from both Tokyo and New York.

# Database Migration: Old Idea, New Trend

- There are still some good reasons to run Oracle.
- But for a large and growing percentage of use cases, PostgreSQL is an option.
- Big companies like Amazon are getting involved.

# Current Work: Pluggable Storage

- Hope to see pluggable storage in PostgreSQL 12.
- Allows for innovation at the storage layer.
  - Try new things without breaking the existing heap.
  - Add special-purpose storage formats such as a WORM (write-once read-many), columnar, in-memory, non-transactional.
- Hope to see first version of zheap in PostgreSQL 13.
  - In-place update with undo means that bloat goes away automatically in most cases.
  - So, better for unattended operation + large data sets.
  - Smaller on disk and fewer writes, too.

# Current Work: Sharding/Clustering

- Two-phase commit for FDWs.
- Cluster-wide MVCC.
- Asynchronous execution.

# Current Work: Compatibility

- MERGE.
- Schema variables.



# Future: Cluster Management

- Graceful failover without risk of transaction loss.
- Easy cluster reconfiguration, including automatically demoting a master to a standby.
- Automatic topology discovery.
- Built-in load-balancing.

# Future: Processes → Threads

- Scalability to thousands of connections, especially when most are idle.
- Better and faster parallel query.
- Better for some procedural languages – avoids a separate interpreter per backend.

# Thanks

- Any Questions?