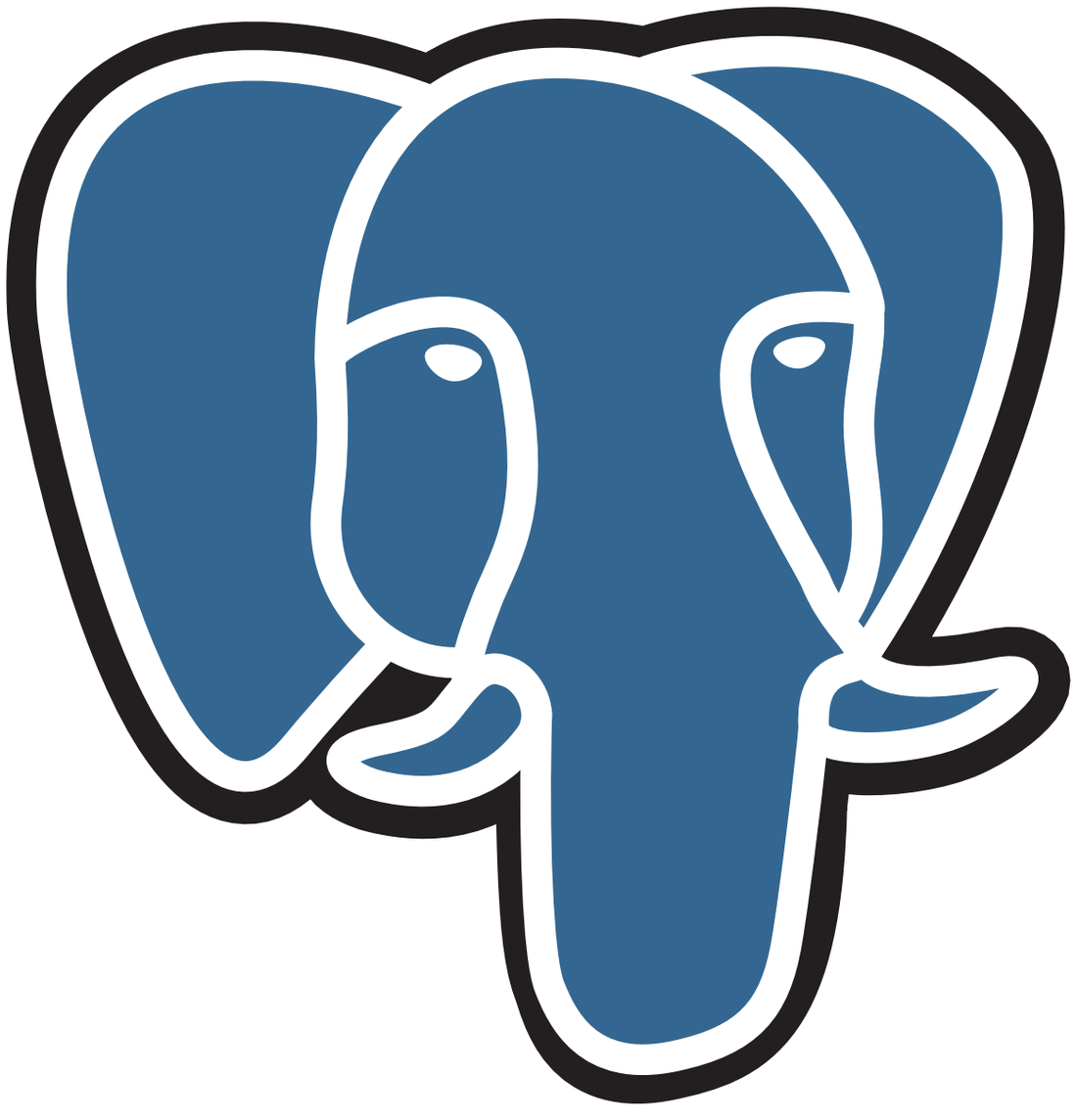
PostgreSQL for Servoy Developers



Christophe Pettus
PostgreSQL Experts, Inc.
christophe.pettus@pgexperts.com

Hello.

- Database Architect/DBA with PostgreSQL Experts, Inc.
- PostgreSQL user for 13 years.
- Servoy user for 3 years.
- Consultant with PostgreSQL Experts, Inc.
  - http://pgexperts.com

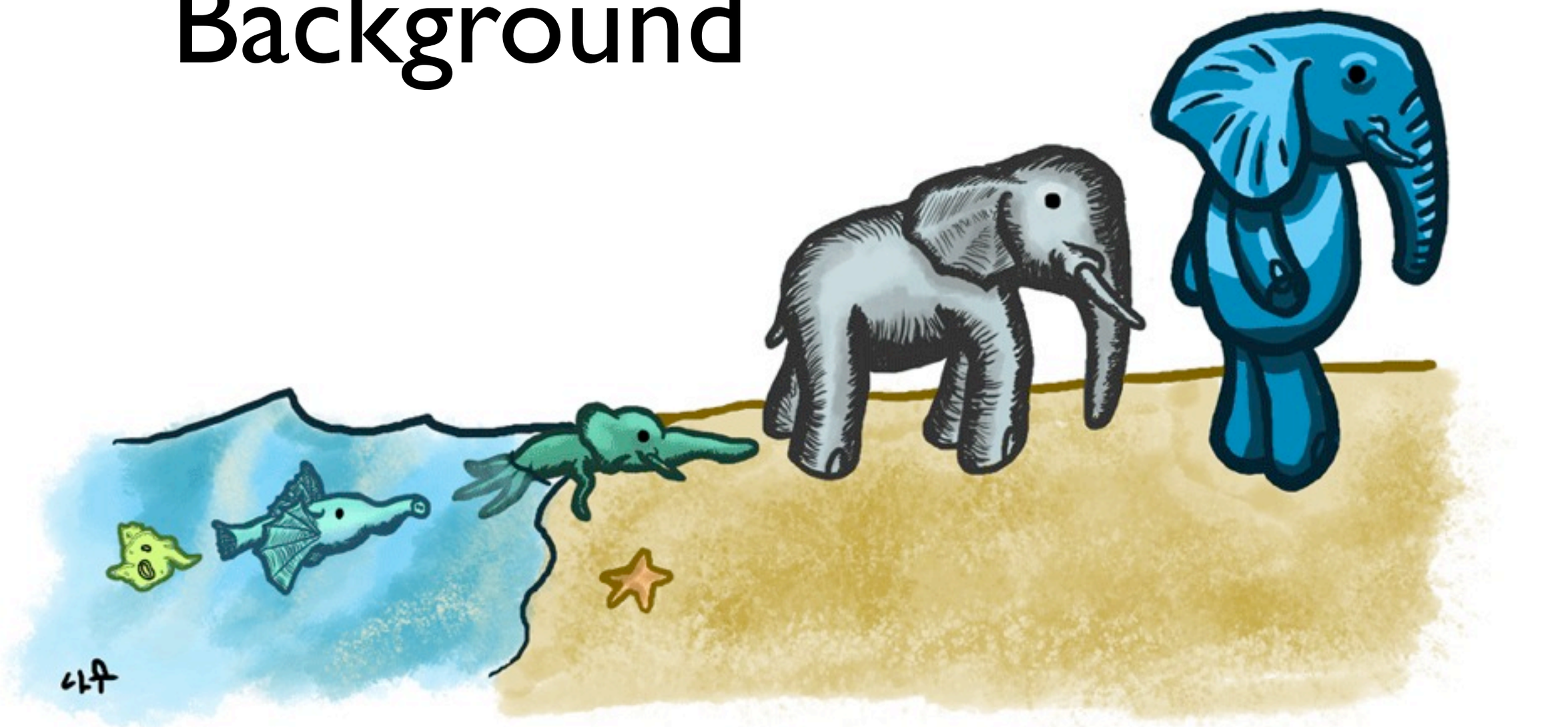
Agenda

- A quick introduction to PostgreSQL.
- Orientation, not exhaustive.
- A little bit of database knowledge assumed.
- … but not too much.

What's on the Menu?

- A Little Bit of Background
- Features Overview
- Basic Configuration
- Pitfalls and Gotchas
- Performance
- Maintenance, Monitoring and Tools
- Advanced Features
- The Community

Background



So, What is PostgreSQL?

- Relational Database Management System
- Features ≥ to Commercial RDBMSes
- Emphasis on Data Integrity and Extensibility
- Open Source, BSD/MIT License

Quick History

- Derives from the 1986 POSTGRES project at UC Berkeley.
- Also the origin of Illustra, thence Informix.
- And thence Sybase, SQL Server.
- Open Source Since 1995.

Platforms

- Any *nix you can imagine
  - Linux, FreeBSD, Solaris, etc.
- Windows
  - Natively since 8.0
- Scales nicely from laptops to large clusters

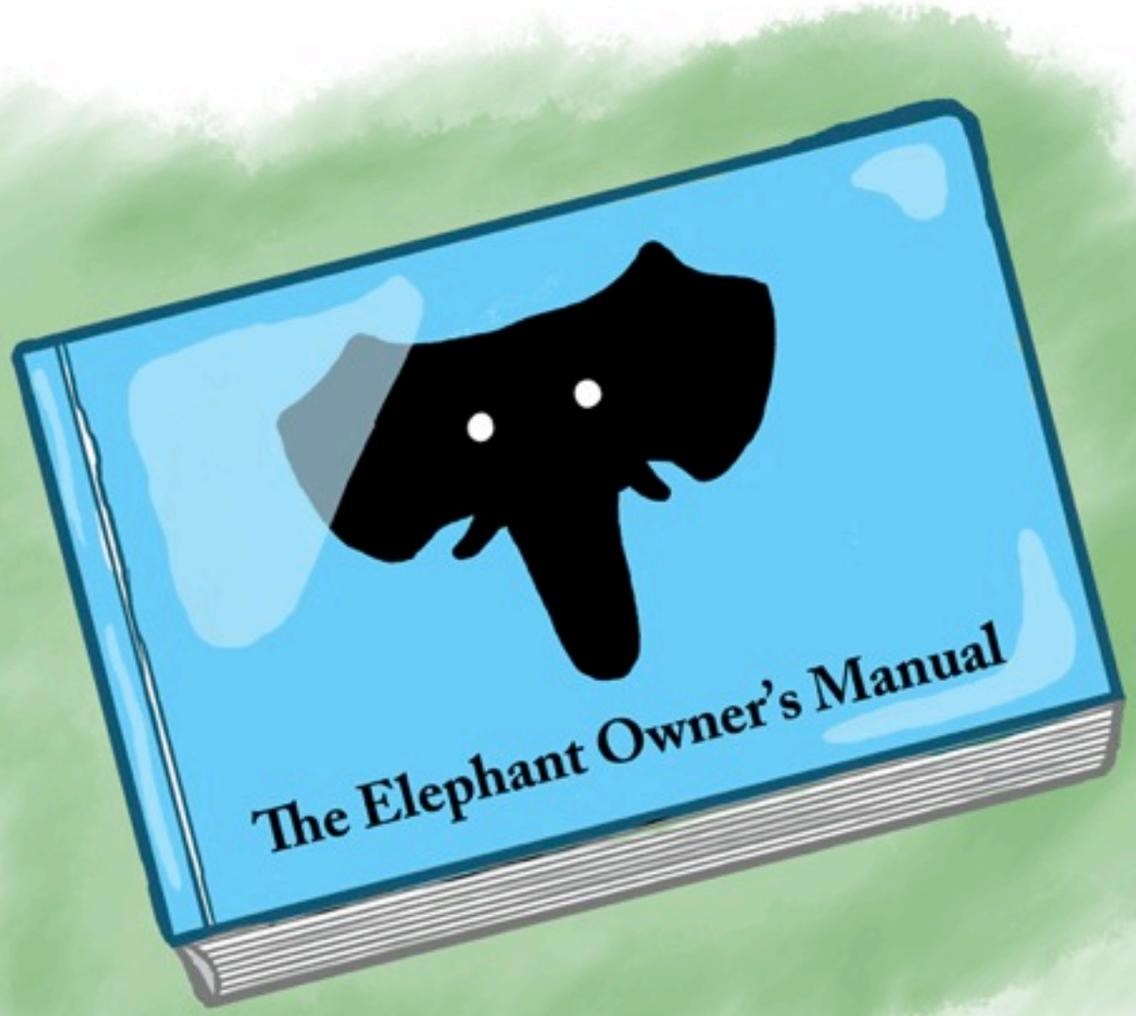
License

- BSD/MIT-style license
- Proprietary derivatives welcome
- Not owned by a single company

Packaged with Servoy

- Version 8.4.3 as of the latest download
- Installed in application_server/postgres_db
- Just Works
  - Catch for multiple installs: listens on default port

Features Overview



A Quick Spin Around the Elephant

- PostgreSQL is the most feature-rich open source database. Full stop.
  - Focus on “big database” features
  - High-rate OLTP, Data warehousing…
  - Easily comparable with proprietary RDBMSes.
- Far more features than we can mention here.

So, let's look at…

- Data Integrity
- Security
- Multi-Version Concurrency Control
- Transactional DDL

What Gets Committed, Stays Committed

- Community has an intense focus on data integrity.
  - Not "the first car to the ten-car pileup"
- Near-bulletproof crash recovery.
- Only compromised by (major) hardware failures.

Putting the C in ~~K~~Consistency.

- Data Integrity includes data quality.
- Sophisticated constraints.
  - Single-Column
  - Multi-Column
  - Foreign-Key
  - And more…

Transactional DDL

```
BEGIN;

DROP TABLE employee_compensation;

ROLLBACK;
```

Security

- Far and away the most secure open-source database.
  - One (1) exploit in the last six years.
- Sophisticated users, roles, groups and permissions.
- Useful even with primitive web framework user management.

Multi-Version Concurrency Control

- Uses versioning and snapshots to isolate concurrent queries and updates.
- Queries see the state of the database at the start of the query or transaction.
- READ COMMITTED or SERIALIZABLE isolation levels.

That Means…

- Readers don't block writers.
- Writers don't block readers (much).
- Writers only block writers if they are writing the same row/object.

MVCC and You!

- PostgreSQL can handle many concurrent clients.
- Very little locking required for most applications.
- Transactions are not inherently expensive to open, commit or roll back.

Basic Configuration

A Not-Uncommon Reaction:

196 CONFIGURATION PARAMETERS?
WE'RE ALL GOING TO DIE!

Relax.

- For experimenting and basic development, apply straight out of the tin.
- 90% of all installations need to adjust <12 configuration parameters.
- Production use does require some parameter tweaking.

Why Tweak?

- The standard parameter values are very conservative.
- Like, “party like it’s 1999” conservative.
- Fine for development on your laptop.
- But you probably have more than 128MB in your production machine.

Memory-Related

- PostgreSQL expects the OS to do a lot of the heavy cache-lifting.
- Give PostgreSQL plenty of memory, but don't overdo it.
- shared_buffers, work_mem, maintenance_work_mem

shared_buffers

- PostgreSQL internal data cache
- Set to 1/4 to 1/3rd of total server memory
- Much over 16GB is usually a waste
- Remember, the OS is caching, too
- Make sure the OS can support the allocation.

work_mem

- Controls memory used for sorts, joins, etc.
- The server can use many times more than the parameter value
- connections / (1-3 * available memory) is a good starting place

maintenance_work_mem

- Controls the memory used for VACUUM and related.
- Available RAM / 8.
- Almost never makes sense to go over 2GB.

Integrity-Related

- Improve performance: checkpoint_segments, checkpoint_timeout, wal_buffers
- Accept some risk: synchronous_commit
- Pure madness: fsync

WAL Background

- Write-Ahead Log of database activity.
- Continuous stream of data changes.
- A *checkpoint* is a point at which the in-memory changes have been flushed to the data pages
  - A checkpoint is where recovery starts, if required.

checkpoint_segments

- The WAL is written in segments of fixed size (by default, 16MB).
- A checkpoint is done every time "checkpoint_segments" segments have been written.
- Increasing improves performance, at the expense of recovery time.
- 16-128 is typical.

checkpoint_timeout

- A checkpoint is automatically done each "checkpoint_timeout" seconds.
- Default is usually fine.
- Increase if you start seeing warnings in the logs.

synchronous_commit

- PostgreSQL waits until all data is flushed before returning from a COMMIT.
- This parameter can defeat that.
- You can lose committed data with this "off."
- You will not corrupt your database, however.

fsync

- Do not turn this off.
- No, not for that reason.
- No, not for that reason either.
- Leave it on.
- Always.
- Yes, **always**.

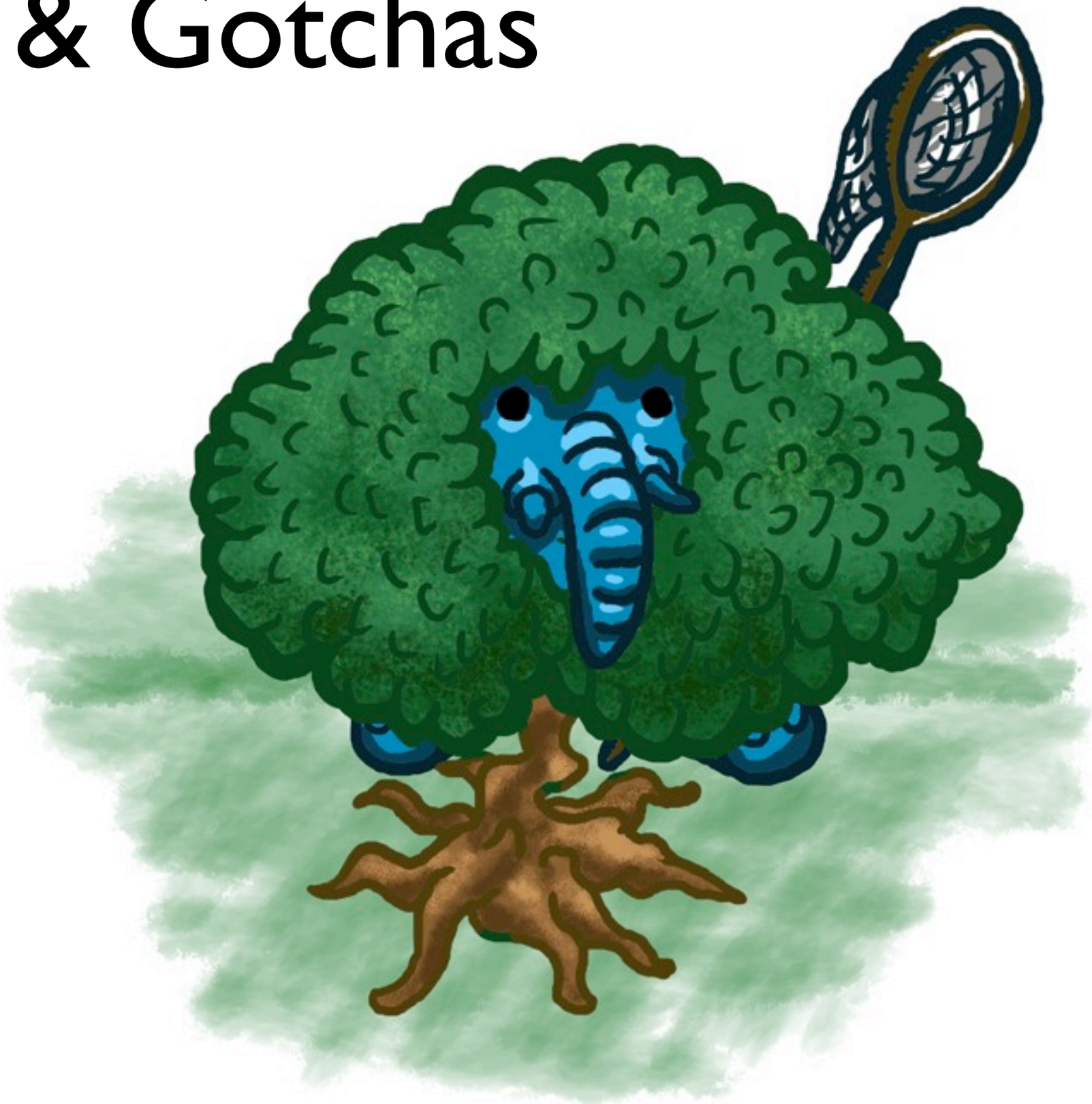
Connection-Related

- max_connections
  - If you are setting this to huge values (>200), it's time to look into connection pooling.
- pgbouncer, pg_pool

Logging

- PostgreSQL can log just about any db activity.
- Pick a log level and log format that makes sense for your application.
- CSV is handy for analysis tools, such as
  - pgFouine

Pitfalls & Gotchas



SELECT COUNT(*);

- PostgreSQL implements this as a full table scan.
- So, don't do that.
- Any qualification accelerates it appropriately.
- Does your application really need to do this all the time?

VACUUM

- Required to find and remove "dead" tuples.
  - That MVCC thing? Yeah, OK, it's not free.
- autovacuum will usually take care of you.
- Manual VACUUM recommended after large data changes.
  - Don't forget to ANALYZE, too!

In-Place Upgrade

- There isn't any yet.
- Minor versions (9.0 to 9.0.1) don't require dump/reload.
- Don't stick with back versions for this reason alone. Plan for this.

Spec Compliance

- More serious than most databases.
  - Even commercial ones.
- Some bad habits may not carry over.
- Some flexibilities may be missing.
  - Consider it a personal growth opportunity.

Performance

PostgreSQL Performance

- Hardware Choices and Configuration
  - Way beyond the scope of this talk
- Schema & App Design
- Connection Management

Schema & App Design

- Transactions are cheap.
- Do not fear joins.
  - Do not denormalize unless you *know* you need to.
- Use indexes, but only when they help.
- Wide rows and BLOBs can be slow.

Connection Management

- Connections are not free.
- Establishing a connection is definitely not free.
- Use pooling unless you have relatively few clients.
- Look for bad connection situations.
  - <IDLE>, <IDLE IN TRANSACTION>

Query Analysis

- ANALYZE
- EXPLAIN, EXPLAIN ANALYZE
- Check for missing indexes
- Check for unused indexes
  - http://it.toolbox.com/blogs/database-soup/finding-useless-indexes-28796

Maintenance, Monitoring and Tools.

Low-Maintenance

- Low-growth systems can run on automatic.
- Multi-year uptime (except for version upgrades) entirely possible.
- A little bit of setup goes a long way.

Basics

- VACUUM & ANALYZE regularly
  - autovacuum will usually take care of you.
- Rotate and process logs.
  - Look for errors and pathological queries.

Backups

- What is not backed up, you do not truly possess.
- pgdump: Hot full backup.
- WAL log shipping / Warm Standby
  - Enables point-in-time recovery
- Hot Standby introduced in 9.0!

Development Tools

- psql
- pgAdmin
- phpPgAdmin

Monitoring

- Connection usage.
- Disk usage.
- Pathological queries.
- check_postgres
- pgFouine

Advanced Features

Extensible

- Data Types
- Indexes
- Operators
- Functions / Triggers
- Languages

Out of the Box…

- PL/pgSQL
- PL/Tcl
- PL/Perl
- PL/Python

Or Define Your Own.

```
CREATE FUNCTION LOL_MAIN_TEST(TEXT)
RETURNS BOOLEAN
LANGUAGE PLLOLCODE
AS $$
HAI
     VISIBLE INFO LOL1
     FOUND YR WIN
KTHXBYE
$$;

SELECT LOL_MAIN_TEST('IM IN YR DATABUKKIT');
```

Full-Text Search

- In the PostgreSQL Core
- Multiple languages
- Dictionary / stemming / tokenizing all customizable
- Fully integrated with the database

Other Extensions

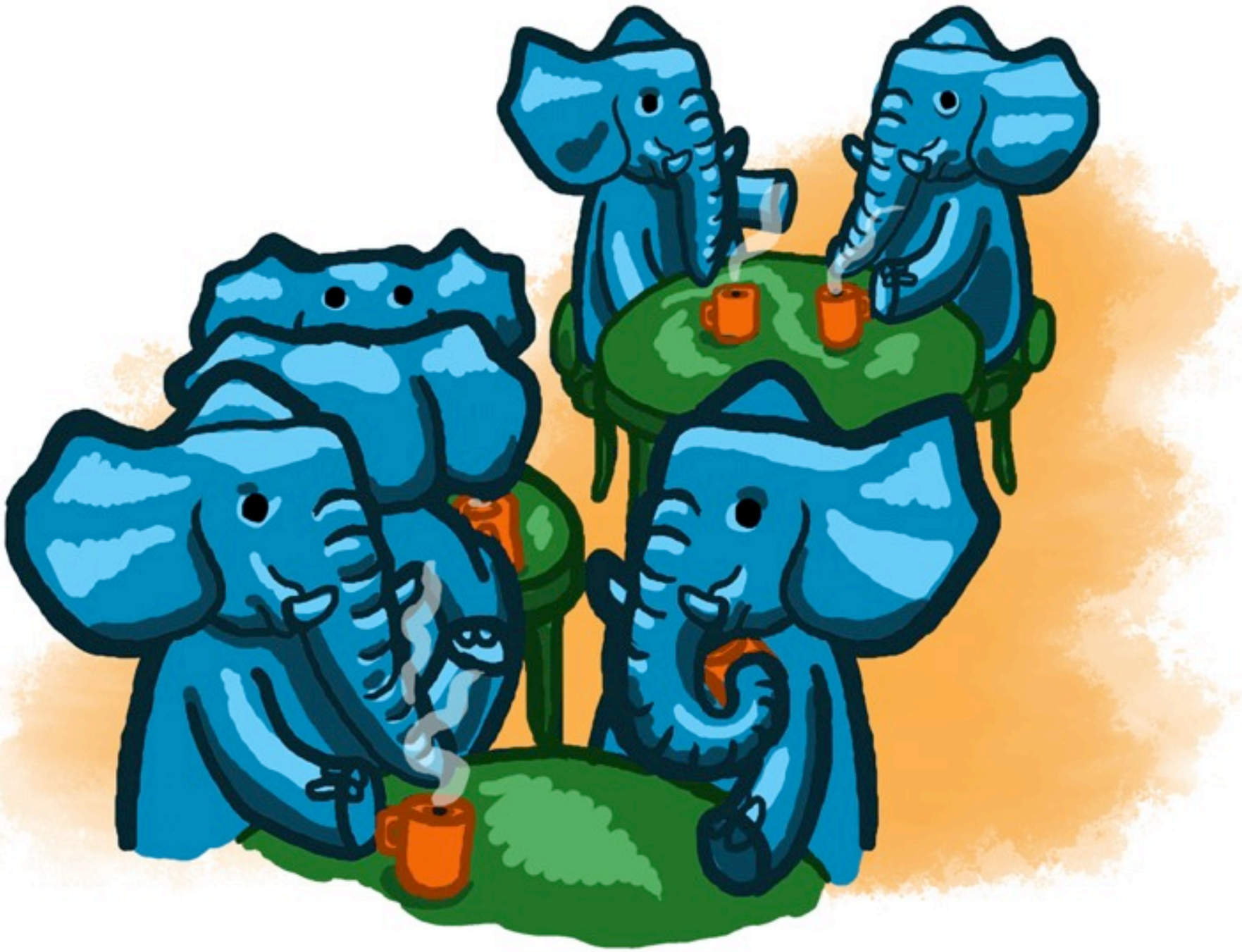
- Cryptography
- Hierarchical Data Storage
- Inter-Database Communication
- Check out the contrib/ directory; it rocks.

PostGIS

- Open-Source Geographic Information System.
- http://postgis.refractions.net/

Replication

- As of 9.0, hot standby!
  - Read-only slaves.
- Third-party solutions for master/slave replication.
  - Slony



The Community

Elephants are Everywhere.

- World-wide community of developers and users.
- Not company-centric.
- Extremely supportive.
- Intense focus on the quality of PostgreSQL.
  - Strong meritocratic culture.

Mailing Lists

- Very active developer and user lists.
- Listen, contribute, do some research first.
- http://www.postgresql.org/community/lists/

IRC

- #postgresql on irc.freenode.net
- Language-specific channels, too.
- http://www.postgresql.org/community/irc

Conferences

- Dedicated PostgreSQL conferences world-wide.
- Strong presence at other open-source events, too.

Commercial Support

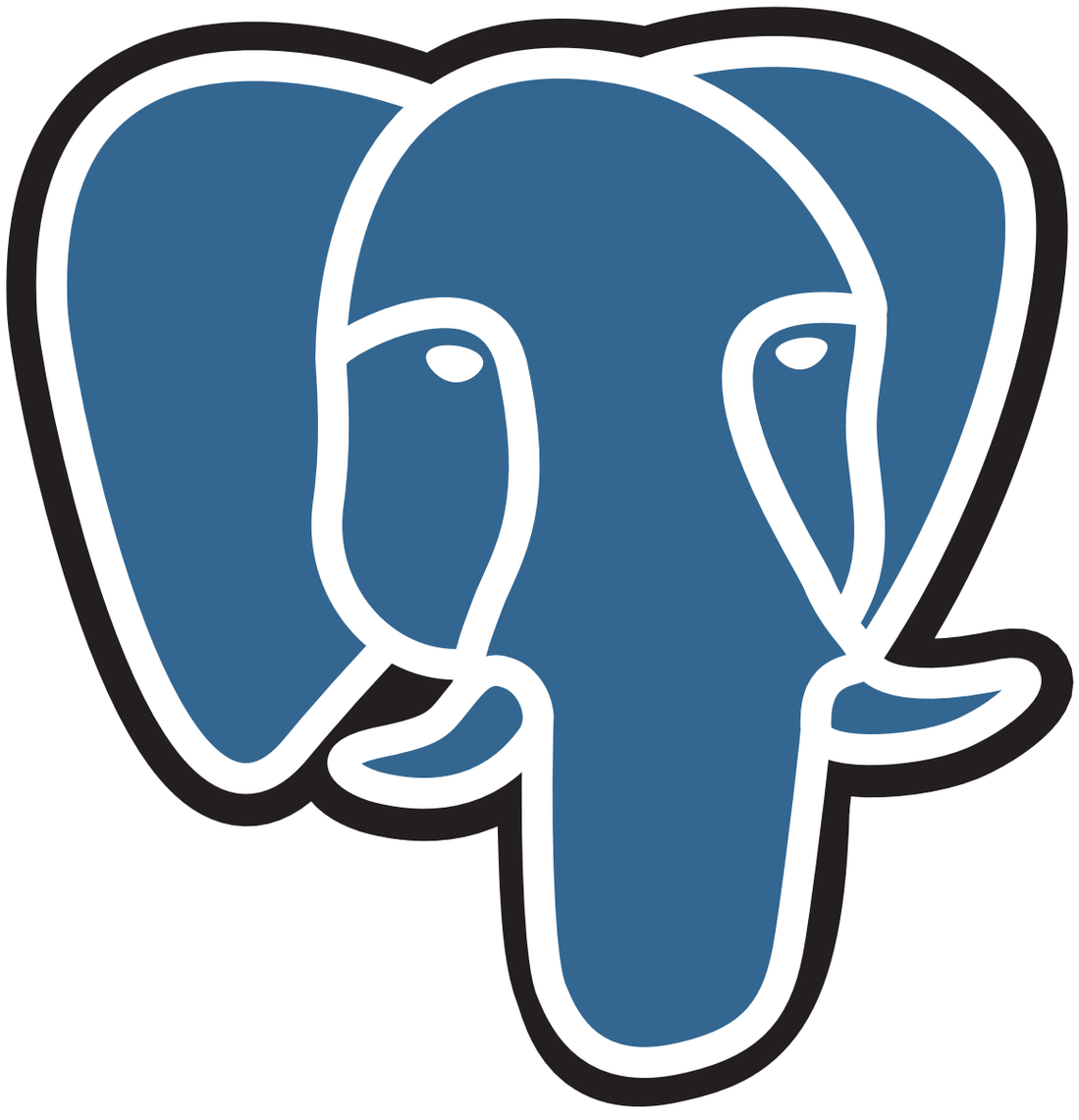
- Many commercial support organizations.
  - http://www.postgresql.org/support/
  - Allow me to recommend PGExperts, Inc.
- Proprietary versions and extensions to PostgreSQL, too.

Questions?

More Info.

- pgfoundry.org
- www.pgexperts.com/documents.html

- Cartoons by Chris Lowrance
  - http://chrislowrance.net
- Presentation available for download at:
  - http://thebuild.com



Thank you!

Christophe Pettus
PostgreSQL Experts, Inc.
christophe.pettus@pgexperts.com