

# A look at the elephants trunk



[http://www.flickr.com/photos/aussy\\_greg/255942923/](http://www.flickr.com/photos/aussy_greg/255942923/)

P2D2 2011  
Prague, Czech Republic

Magnus Hagander  
magnus@hagander.net

# PostgreSQL 9.1

- Is not yet done!
- Feature set still uncertain
- Many things are done
  - Some completed
  - Some partial
- We give no guarantees, sorry...
- Many “invisible” changes, not included here



# Development Schedule

- July 9, 2010: 9.0 branched, HEAD opened
- July 15-Aug 15: Commitfest #1
- Sep 15-Oct 15: Commitfest #2
- Nov 15-Dec 15: Commitfest #3
- **Jan 15-Feb 15: Commitfest #4**
- Beta releases
- Release candidates (before summer?)
- Release (after summer?)



# Current state of tree

- CF4 in progress
- Alpha4 will be next release
- 2845 files changed,  
116926 insertions(+),  
46235 deletions(-)
- Less than 9.0 – but no small release!



# Many new features

- **Replication and Backup**
- Security
- Performance
- SQL and application functionality



# Most probable app-breaker

- *standard\_conforming\_strings* is now **on by default**

- This means:

```
postgres=# select '0\'Brien';  
postgres' #
```

- Make sure you use:

```
postgres=# SELECT '0' 'Brien', E'0\'Brien';  
0'Brien | 0'Brien
```



# Replication and backup

- Biggest features in 9.0 were:
  - Streaming Replication
  - Hot Standby
- Many rough edges
- Management and monitoring functionality based on real world experience



# Replication and backup

- Replication is now a separate permission
- Superuser not required/recommended

```
postgres=# CREATE USER replica WITH REPLICATION;  
CREATE ROLE
```

- Superusers granted replication permission by default
  - Can be revoked





# Replication monitoring

- View all replication sessions from master

```
postgres=# SELECT * FROM pg_stat_replication;  
... | 2011-02-07 12:52:20.141376+01 | STREAMING | 0/13000B70
```

- View transaction replay *timestamp* on slave

```
postgres=# SELECT pg_last_xact_replay_timestamp();  
2011-02-07 12:47:36.608706+01
```



# Hot Standby monitoring

- Query conflicts are the big issue with HS

- Total per database

```
postgres=# select datname, conflicts FROM pg_stat_database;
```

```
postgres |          79
```

- In full detail

```
postgres=# SELECT * FROM pg_stat_database_conflicts;
```

```
datname          | postgres
```

```
confl_tablespace | 1
```

```
confl_lock        | 3
```

```
confl_snapshot   | 74
```

```
confl_bufferpin  | 1
```

```
confl_deadlock   | 0
```



# Streaming base backup

- Used for backups
  - No need to set `archive_command` in small deployments
  - No need for complicated scripts
  - No need for SSH/rsync/whatever access
  - Write to directory or native tarfiles
  - Just:

```
pg_basebackup -D /some/where -x
```



# Streaming base backup

- Used for deploying replicas:
  - No need to use backups/log archiving
  - Single command deployment of slave
  - Just create *recovery.conf*
- Runs over libpq protocol
- Supports all authentication and encryption options
- Requires *REPLICATION* privilege and walsender



# Concurrent base backups

- Run more than one backup in parallel
- Only valid for streaming backups
  - One regular + unlimited streaming
  - Uses one walsender slot each
- E.g. cloud replication deployment



# Detailed recovery control

- Ability to pause during recovery
- With hot standby, data can be reviewed
- Addition of “named restore points” during normal operation

```
SELECT pg_create_restore_point('before_stupid')
```



# pg\_dump directory format

- One object – one file
- Allows parallel generation and later collection
- Mainly useful for parallel pg\_dump
  - ... not included yet, but hopefully before 9.1



# Many new features

- Replication and Backup
- **Security**
- Performance
- SQL and application functionality





# Host names in pg\_hba.conf

- Removes need to hard-code IPs
  - Reverse lookup followed by forward
- Domain matching using `.domain.com`
- Use a local resolver cache!



# Server auth on unix sockets

- Previously, peer could only be verified from server to client
- Now we can specify

dbname=foo **requirepeer=postgres**

- Avoids local attacks
- For TCP, use SSL certificate validation



# SE-PGSQL

- Integrates with SE-Linux
- Label based security
- Umm. Yeah, go try it.



# Many new features

- Replication and Backup
- Security
- **Performance**
- SQL and application functionality



# More monitoring points

- `pg_stat_*_tables` added counters
  - Number of vacuum
  - Number of analyze
  - Differentiated by regular and background processes
  - Helps tuning autovacuum



# More monitoring points

- `pg_stat_bgwriter` counts fsync requests
  - Detect when background processes aren't keeping up
  - fsync() by backends is *very* bad



# Detailed checkpoint data

- Instrument many details around checkpoints
  - Individual files, not just totals
  - Sync time, longest sync, avg sync
- `log_checkpoints=on`



# Auto-tuning wal\_buffers

- What the headline says
- Still possible to override manually
- Auto-tunes between 64kB and 16MB





# Unlogged tables

- Create tables without writing to WAL
  - Considerable performance increase for large loading or changes
- Truncate on crash recovery
- *Not* included in log based replication
- No way (yet) to convert between logged and unlogged



# KNN-GiST

- “ORDER BY for GiST”
- Fast, **indexed**, “K-Next-Neighbour” search
- For example, “the 10 graphical objects closest to this point”

```
SELECT * FROM t
ORDER BY pos <-> myposition
LIMIT 10
```



# Many new features

- Replication and Backup
- Security
- Performance
- **SQL and application functionality**



# Serializable Snapshot Isolation

- True **SERIALIZABLE** transactions
  - Current is really **SNAPSHOT** isolation
- Very little overhead
- Reduces need for **SELECT FOR UPDATE**
- Old behavior still there as **REPEATABLE READ**



# PK functional dependencies

- Functional dependencies on **PRIMARY KEYS** are recognized for **GROUP BY**
- No more

```
postgres=# SELECT uid,first,last FROM users GROUP BY uid;
```

```
ERROR: column "users.first" must appear in the GROUP BY  
clause or be used in an aggregate function at character 12
```

- Only recognizes PRIMARY KEY, not UNIQUE constraints or indexes



# Add values to enum

- No more ugly hacks with pg\_enum
- Just:

```
ALTER TYPE foo ADD VALUE 'bar' BEFORE 'baz'
```



# Per column collation

- Before 8.4, collation was per cluster
- Since 8.4, it's per database
- Now moving to per column
- One column English, another Danish
- Controls sort order and upper/lower

```
CREATE TABLE t (  
  a text,  
  b text COLLATE "sv_SE"  
)
```



# Triggers on VIEWS

- INSTEAD OF triggers only
- Can be used to implement UPDATEable views
- Much nicer to work with than RULEs
- Gets the whole modified view row, figures out the rest





# Attach WITH to DML

- Nicer way to write “subqueries” for DML
- Unfortunately, not Writable CTEs just yet

```
WITH t AS (  
    SELECT ...  
)  
UPDATE x SET a=t.a+20, b=t.b  
RETURNING t.c
```



# Extensions

- Wrap extensions (contrib, postgis etc)
- Distinct objects containing schema items
- Controlled dump/reload/upgrade
- (Only partial so far)



# Not quite there yet...

- Some features still pending review/commit
- Hopefully in 9.1, but less certain
- Please help to review ;)



# SQL/MED

- “Managed External Data”
- Table-like access to external data
  - Other PostgreSQL servers (“dblink”)
  - CSV files (without COPY)
  - Any other data sources (“Foreign Data Wrappers”)



# Synchronous replication

- Current solutions are asynchronous
- Sync often wanted for data security
- “Semi-sync” for decent performance
- Controllable *per transaction*
- Mix of sync and async fully supported



# Range data types

- Arbitrary “range” datatype wrapper
  - Integer ranges
  - Timestamp ranges
  - `<any>` ranges
- Very useful for exclusion constraints



# Thank you!

## Questions?

Twitter: @magnushagander  
<http://blog.hagander.net/>  
magnus@hagander.net

