Move over MySQL, make room for PostgreSQL

#MOMMRP, Integration and Tools

Brian Ghidinelli

www.ghidinelli.com

CF.Objective()

Why are you using MySQL?

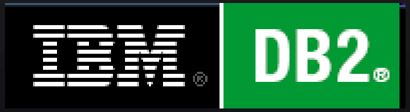
- It has a sweet dolphin for a mascot

Why are you *really* using MySQL?

- Fast
- Free
- “The most popular open source database”
- “The web database”

What if money was no object?

- Oracle?
- IBM DB2?
- MS SQL Server?



What if you could have a solid mix of both?

Hint: it’s supported out of the box in ColdFusion MX 8

Why Postgres?

- Standards-compliant ANSI SQL with ACID reliability
- About as fast as MySQL when comparing apples to apples
- Free as in speech and beer: BSD License
- “The world’s most advanced open source database”, “The open source Oracle”

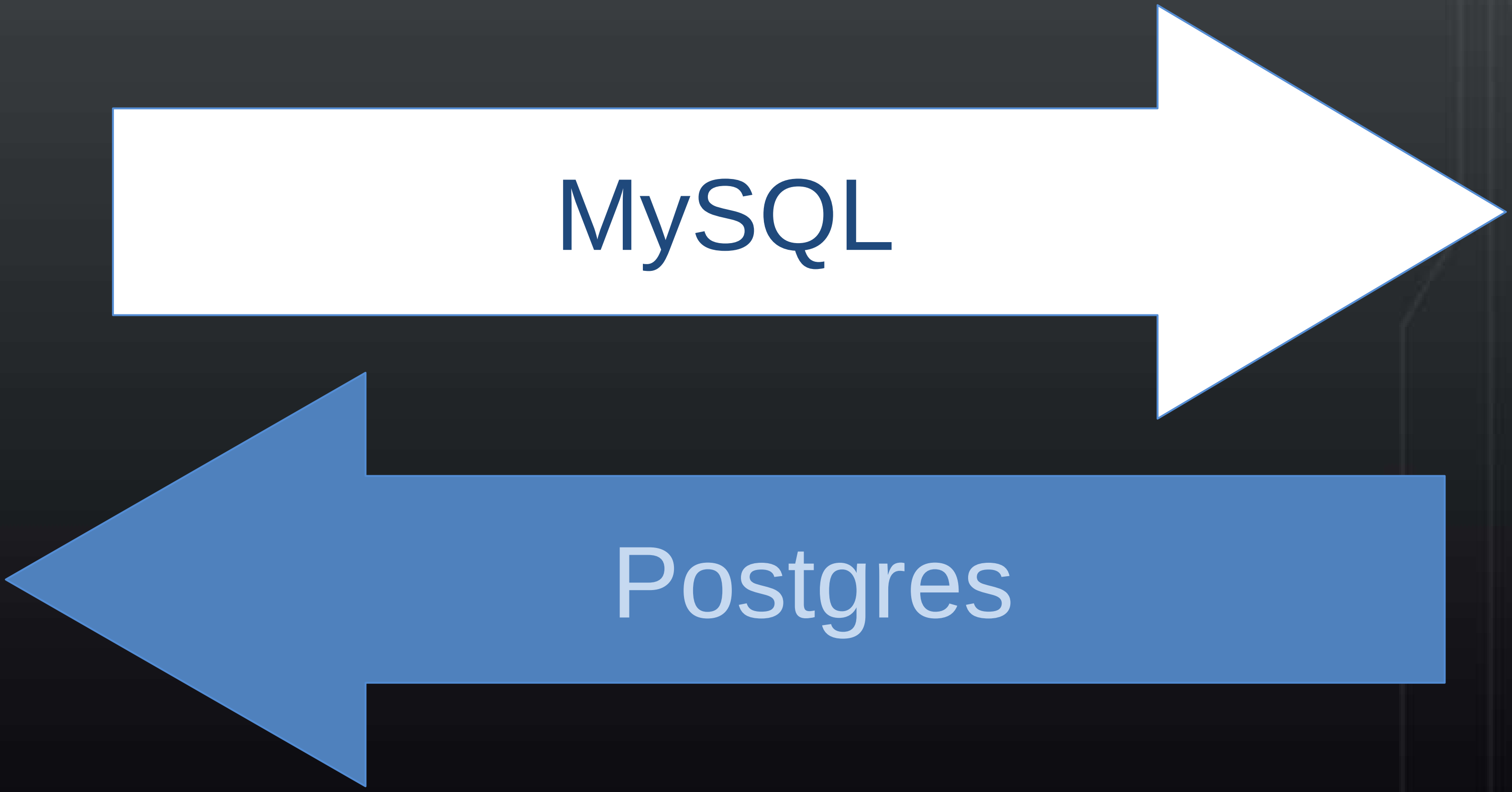
Who is using Postgres?

- Skype
- Yahoo
- Hi5
- Nextbus.com
- Sony Online Games
- Redfin

Where Postgres Came From

- Ingres @ UC Berkeley ‘77-85
- Next generation Post-gres from ’86-94
- Postgres95 in 1995
- PostgreSQL 6.0 in ’96, open source plus:
  - Multiversion concurrency control (MVCC)
  - Important SQL features
  - Improved built-in types
  - Speed

Philosophical Differences


MySQL

Postgres

Simple,
Easy to use,
Fast

Features,
Security,
Standards

Isn’t Postgres complex?

- “Oracle-like”; uses schemas
  - MySQL: db/schema -> table -> column
  - Postgres: db -> schema -> table -> column
- Access controlled by pg_hba.conf
  - By default, no TCP/IP connections
- Driver included in CF8 (JDBC < CF8)
- Native Win32 build and installer
- Clusters possible, warm standby easy

Isn’t Postgres slow?

- Idea of Postgres as turtle and MySQL as hare is antiquated
- Excellent multi-proc/core performance
- TOAST stores “bigger” data elsewhere with lightweight compression
- Postgres had major speed ups in 8.x releases

Ok, I’m vaguely interested.

How these two differ from a developers point of view?

Licensing

- Both open source, but different
- MySQL: GPL 2.0
  - Free with some very strict obligations
  - What does Oracle purchase of Sun bring?
- Postgres: BSD 2.0
  - Free with very simple obligations

No faking it

- Full ACID reliability out of the box
  - No storage engine trade offs
- Full referential integrity
  - Foreign keys with CASCADE
- Triggers
- Check constraints

Security conscious

- Client Authentication
  - Login, SSL, Kerberos, LDAP, PAM
  - Host-based authentication
- Flexible logging / user auditing
  - Helps with compliance
- Permissions support SQL ROLES
- Excellent security track record

More goodies

- Regular Expressions
  - LIKE, SIMILAR TO, POSIX, UDFs
- Partial indexes
  - CREATE INDEX … WHERE total > 0
- Table inheritance
  - CREATE TABLE capitals (state char(2)) INHERITS (cities);
- Full text search

Procedural language support

Standard Distro

- PL/SQL
- pgSQL
- TCL
- Perl
- Python

Contrib Plug-ins

- Java
- PHP
- R
- Ruby
- Scheme
- sh (shell)
- Lua
- LOLCODE
- pl/js, pl/j, pl/pgpsm…

Datatype support

Common

- char, varchar
- Large text
- Numeric
- Integers
- Floats
- Time, date, timestamp
- Bytea (binary)

Exotic

- Geometric (polygon, line)
- GIS (via PostGIS)
- Crypto
- ISN, ISBN
- XML with XPATH
- INET, CIDR
- Arrays
- Full text index
- Genome
- UUID

Strong Transaction Support

- Including schema/DDL changes!

```
$ psql mydb
mydb=# drop table foo;
  ERROR: table "foo" does not exist
mydb=# begin;
  BEGIN
mydb=# create table foo (bar int);
  CREATE TABLE
mydb=# insert into foo values (1);
  INSERT 0 1
mydb=# rollback;
  ROLLBACK
mydb=# select * from foo;
  ERROR: relation "foo" does not exist
```

Enterprise features

- Tablespaces
- Table partitioning
- BI with BizGres
- Point In Time Recovery
- Scales on big iron hardware

Developer Tools

MySQL

- phpMyAdmin =
- MySQL GUI Tools =
- mysql CLI =
- Mysqldump =

PostgreSQL

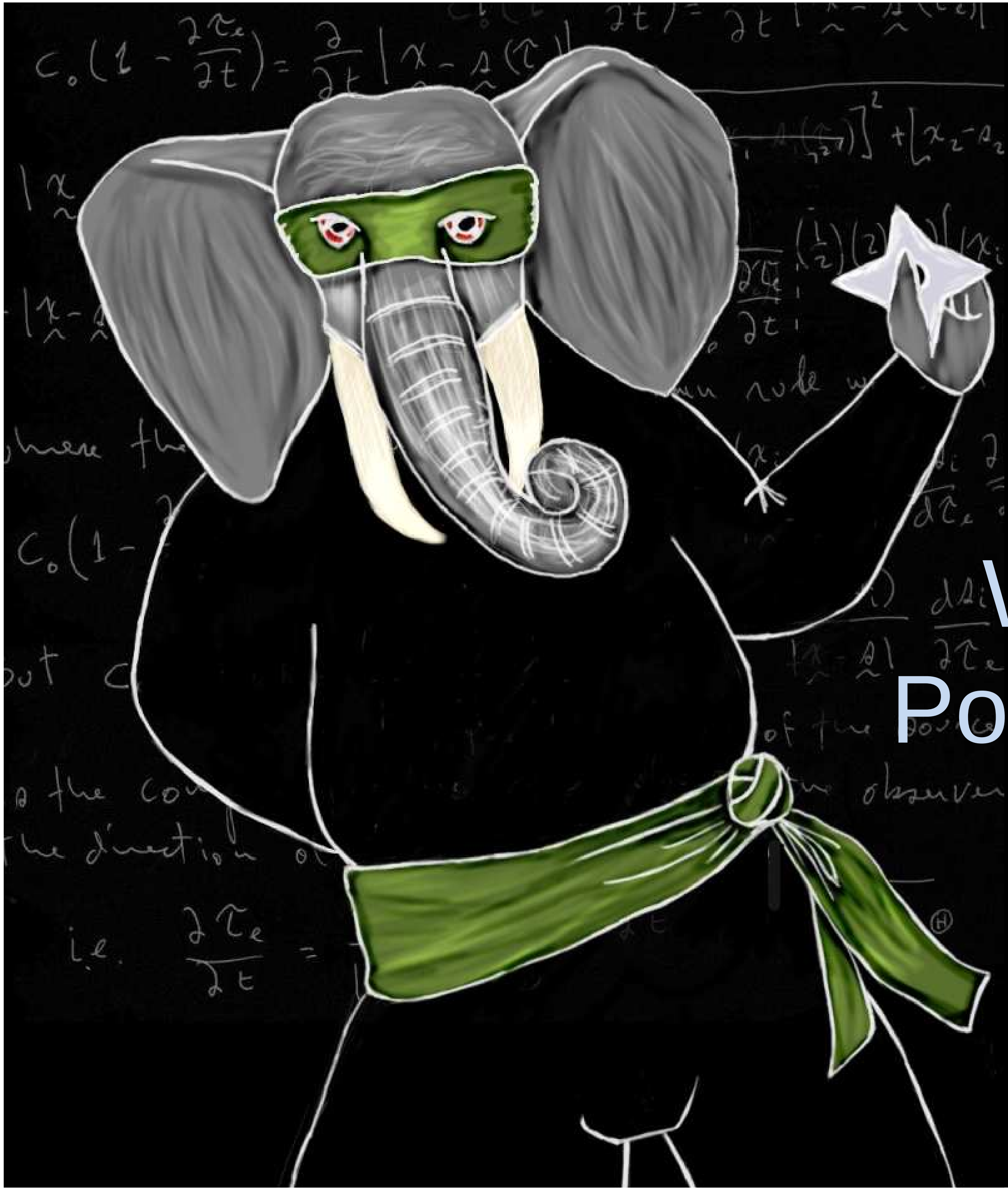
- phpPgAdmin
- pgAdmin
- psql CLI
- pg_dump / pg_restore

User community

- Community organized and operated web, documentation, mailing lists, IRC,
- Q108 - 8 Postgres-only conferences in 5 countries
- MySQL as corporation has bigger userbase but less active community

New in 8.4 (now in beta)

- Column level privileges
- CREATE OR REPLACE view syntax
- Better EXPLAIN tools
- Case insensitive module
- Default and variadic function parameters
- ColdFusion-compatible UUID data type (good for keys)
- Windowing functions
- Common Table Expressions (CTEs, WITH queries)
- RANK()
- LIMIT based on subquery
- SSL certs for user auth



Why consider Postgres for my next project?

It’s not the mascot…

- Stereotype that kept you away is outdated
- Rock-solid ACID reliability
- Generally as fast as MySQL in equal workloads
- Data type + procedural language support
- Lots of goodies and enterprise features
- Active, grassroots user community

What do I need to know to switch?

Cheat sheet time!

Datatypes

MySQL		PostgreSQL
Tinyint	=>	Smallint / bit / boolean *
Mediumint	=>	Smallint / integer *
Datetime	=>	Timestamp
Longtext	=>	Text / varchar
Blob	=>	Bytea
Auto_increment	=>	Serial
Enum(val1, val2, val3)	=>	Check constraint or enum type

* Postgres does not have unsigned integers

Functions/Syntax

MySQL	PostgreSQL
LAST_INSERT_ID()	CURRVAL()
SELECT … INTO OUTFILE ‘/tmp/…’	COPY (SELECT …) TO ‘/tmp/…’
SHOW DATABASES	\l
SHOW TABLES	\dt
DESCRIBE TABLE	\d table
REPLACE	Emulate with PL function
“help”	“\?”

Other gotchas

- Comment character (#, --)
- Case-sensitivity
  - PostgreSQL != postgresql != “PostgreSQL”
  - Use upper(), lower(), ILIKE for MySQL style
  - 8.4 has case-insensitive module
- No backticks (`) to quote system objects

The Million-Dollar Question:

Should I migrate an existing app?

Things to consider…

- Despite tools, migration isn't point/click
- Consider financial impact
  - Life of app, length of migration
- If you have problems with MySQL, you might have problems with Postgres too
  - Consider which project philosophy aligns best with your needs
- Try it for a new project instead?
  - Let it gel – takes a couple of months

Quick look at tools…

- PgAdmin III (GUI)
- CF Administrator datasource
- DDL change in Transaction

Q&A / Resources

CFO Surveys: http://www.cfobjective.com/surveys.cfm!

- http://www.postgresql.org
- http://en.wikibooks.org/wiki/Programming:Converting_MySQL_to_PostgreSQL
- http://jochem.vandieten.net/
- http://www.ghidinelli.com/

- Thanks to Josh Berkus of PostgreSQL for slide assistance
- Elephants from fundraw.com, fc07.deviantart.com and worth1000.com