A MySQL DBA's journey to support and implement PostgreSQL

By: CJ Estel, DBA, CoverMyMeds

Github

github.com/cjestel

Company Tech Blog

http://bit.ly/2yzui2b

https://techblog.covermymeds.com/

Henry Ford

None of our men are 'experts.' We have most unfortunately found it necessary to get rid of a man as soon as he thinks himself an expert because no one ever considers himself expert if he really knows his job...The moment one gets into the 'expert' state of mind a great number of things become impossible.

About the Speaker

- 15+ years of DBA + Linux Systems Administration
- MySQL experience is mostly prior to 5.5
- Working with Postgres for about 3.5 years
- Work more at architecture level than query level

About this talk?

- Moving around, getting started (commands)
- Replication
- Backup differences and similarities
- Locking
- User preferences
- Other observations
- Q&A

Moving Around

MySQL

Connecting:
mysql

Connection File:
~/.pgpass

Exiting:
exit
quit
[ctrl]+c
\q
[ctrl]+d

PostgreSQL

Connecting:
psql

Connection File:
~/.my.cnf

Exiting:
\q
[ctrl]+d

Moving Around (continued)

MySQL

Seeing what's going on:

innodb_top

show full processlist;

show variables;

show engine innodb status\G

PostgreSQL

Seeing what's going on:

pg_top

select * from pg_stat_activity;

show all;

SELECT * FROM
- pg_stat_activity;
- pg_stat_database;
- pg_stat_user_tables;
- pg_stat_user_indexes;
- pg_stat_*
- pg_locks;

Moving Around (continued)

MySQL	PostgreSQL
<u>Seeing what's going on (cont):</u>	<u>Seeing what's going on (cont):</u>
show slave status\G	select * from pg_stat_replication;
	select now() - pg_last_xact_replay_timestamp( ) AS replication_delay;
Show commands	Slash commands \?

Moving Around (continued)

MySQL

Toggle Output:

\G instead of ;

Slave management:

stop slave;
start slave;

PostgreSQL

Toggle Output:

\x before query to toggle feature

Slave management:

select pg_xlog_replay_pause();
select pg_xlog_replay_resume();

Define Schema

MySQL

Schema = Database

Cross-database queries supported

PostgreSQL

Schema = Name space within the database

Cross-database queries not supported, cross schema is

Permission Management

MySQL

Users and networks created and managed in the mysql database

No role support, coming in 8.0

Ownership of objects isn't a thing. Stored procedures, views, triggers, and events can have a definer

PostgreSQL

User and network access granted in pg_hba.conf

Role's can be used to manage user permissions

User or role owns objects

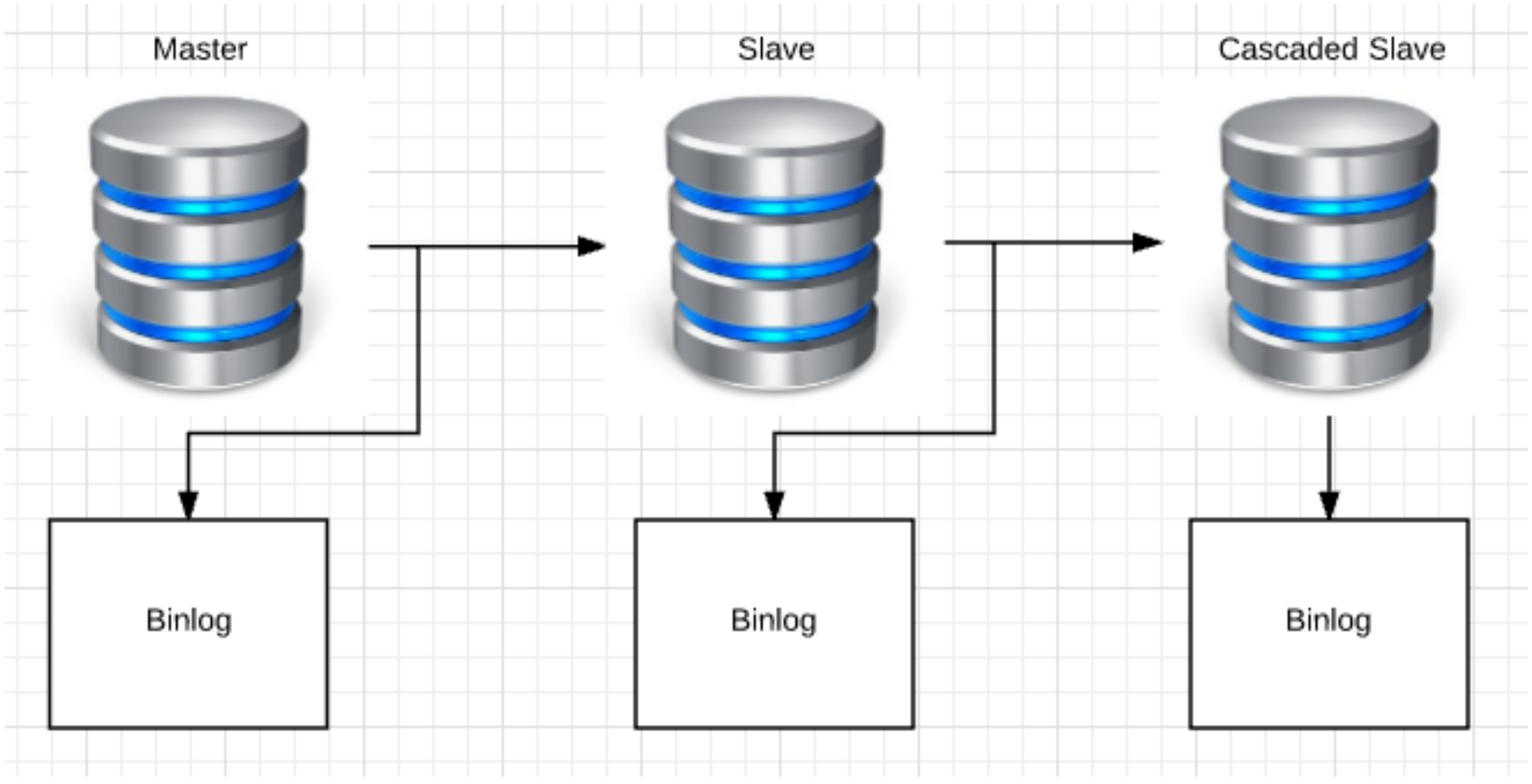
Memory Management

MySQL	PostgreSQL
Allocate about 80% of memory for buffer pool for MySQL to manage memory	Allocate about 25% of memory for shared buffers and allow OS to manage memory
Manage max memory per connection: key_buffer_size + (read_buffer_size + sort_buffer_size) * max_connections	work_mem is the main tunable and is allocated based on number of joins and sorts
max_connections overhead only when in use	max_connections has overhead

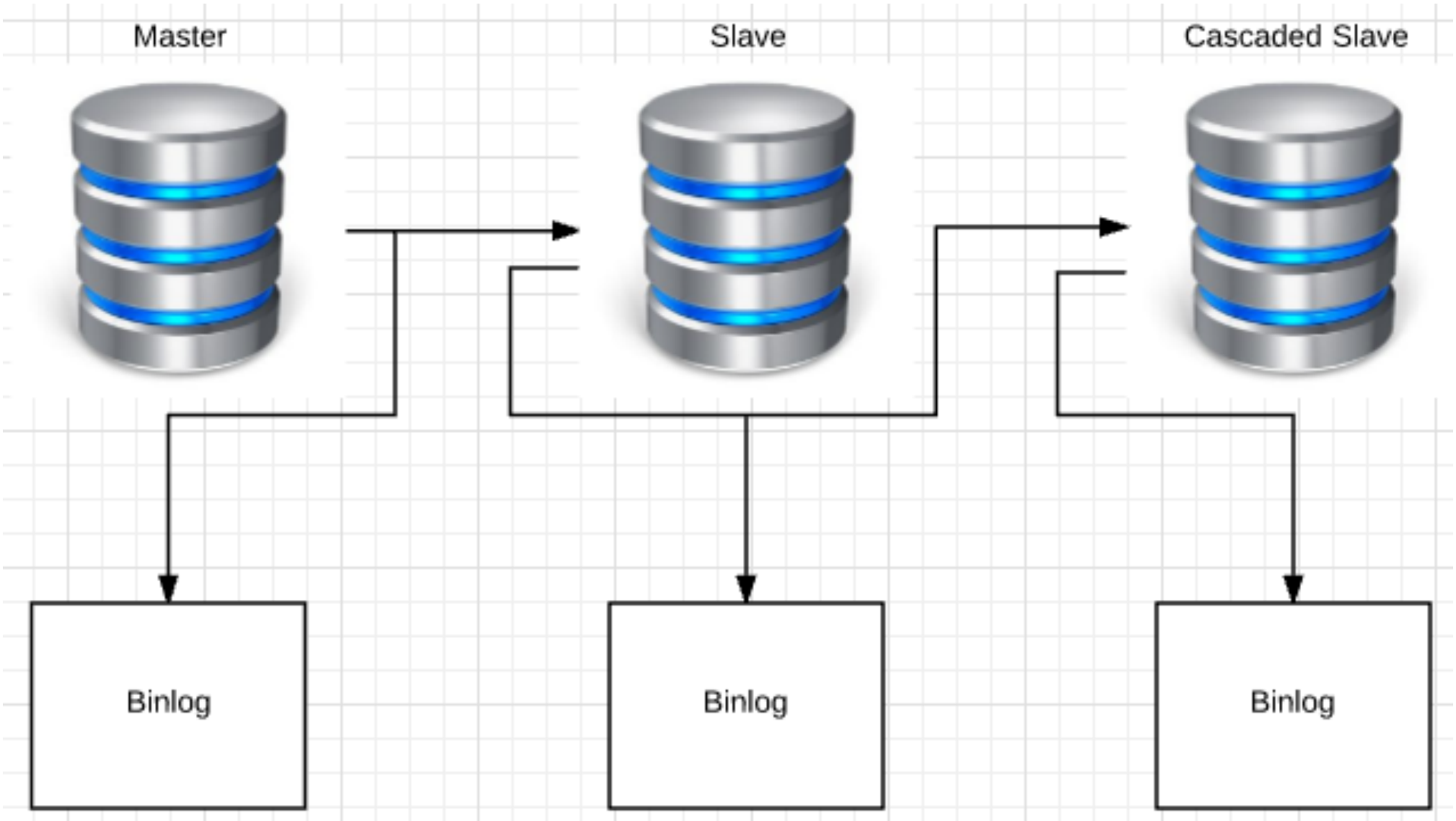
Performance/Monitoring

MySQL	PostgreSQL
performance_schema	pg_stat_statements
long_query_time	log_min_duration_statement
show global counters; show variables; show engine innodb status\G show full process list;	pg_stat_*
	autovacuum
	transaction wrap around

Repl in MySQL



Streaming Repl PostgreSQL



Replication Pro/Con

MySQL (Native)

Pro:

- Manual setup / position mgmt
- Intentional differences in schema
- Follow at table or db
- Supports Master-Master

Con:

- Easy to shoot self in foot
- Difficult to point to new master
- Difficult to setup (comparatively)

PostgreSQL (Streaming)

Pro:

- Slave consistency
- Easy to setup (pg_basebackup)
- Easy to repoint to new master
- Replication Slots (9.4+)

Con:

- No Master-Master support
- No writes on slaves (tmp tables)
- Follow only at server level

Backup Differences

MySQL

mysqldump:

Offers position information for PITR with flag

Takes a global write lock for duration of dump

~~Takes a long time for large databases~~

PostgreSQL

pg_dump/pg_dumpall:

Doesn't allow for PITR

Due to MVCC doesn't take a global write lock while giving consistent dump

~~Takes a long time for large databases~~

Backup Differences (cont)

MySQL

Incremental:

Flush logs and copy

Native block level backup:

Not aware of any*, use 3rd party

- XtraBackup (percona)
- LVM Snapshot
- *MySQL Enterprise Backup

PostgreSQL

Incremental:

Setup to write out to archive location

Native block level backup:

pg_basebackup

Locking MySQL

*Note: experience from 5.5 and earlier

- *DDL changes are very expensive
- *Index creation is very expensive
- MVCC - Keep latest version and reconstruct older versions using undo

Locking PostgreSQL

*Note: experience from 9.3+

- Most DDL changes are a metadata change
- Option to create index concurrently
- MVCC - Store multiple versions of record in database and garbage collect or vacuum dead tuples

Making Postgres Look Like MySQL

User Configuration File: ~/.psqlrc

\timing

\pset border 2

\set pager off

Other Encounters

MySQL	PostgreSQL
Limit varchar's in size	varchar and text are same
GIS new in 5.7	PostGIS has been around
Common Table Expressions (CTE) coming in 8.0	Common Table Expressions (CTE)
One trigger per ins/upd/del	Multiple triggers available
Check constraints ignored	Check constraint supported
Auto increment stored by table	Sequence has own table

Other Encounters (Cont)

MySQL	PostgreSQL
Stored procedures	PL/pgSQL and support for many languages
No materialized view support	Materialized view support
FK's must be indexed	FK's do not need to be indexed
?	Additional index types (brin, gin, partial, etc)
?	\e opens prior query in VI
?	3rd party pooling software
?	3rd party sharding software

Questions? I have an answer, lets see if they match.