PERCONA TECH DAYS

Elephant by the pool

PostgreSQL connection poolers overview

Sergey Kuzmichev & Jobin Augustine
Senor support Engineers - Percona
Percona Tech Days

Agenda

- Why do we need connection poolers at all
- How Connection Pooler address the problem
- External connection poolers, Features & Advantages
    - PgPool II
    - PgBouncer
    - Odyssey
- Limitations of External connection pooler one should be aware

Why we need new connection poolers

Cost of connection

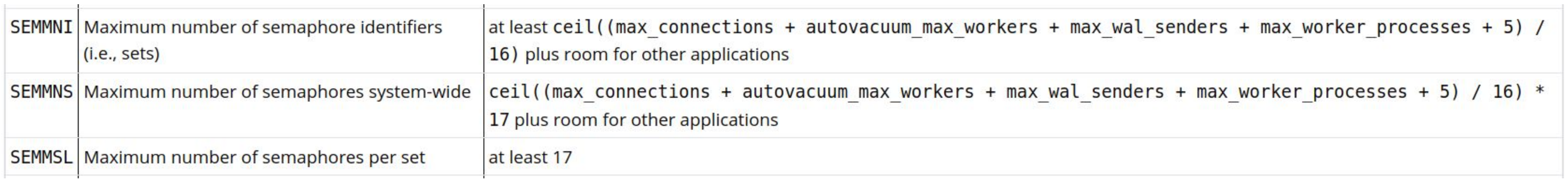
- Every client connection need a dedicated new process
- Process forking
    - memory allocation, cancel key and other housekeeping
    - child process completes all authentication and handshakes
    - OS level overheads
- Overhead of connections are higher in process based databases.

Common DBA Observations

- New server processes are created when lot are “idle” in state.
- Too many PostgreSQL processes in the host OS
- Disconnection, deallocations / clean up

Cost of idling connections

- **Memory overhead**
- **Scheduling overhead**
- **Locking overhead**
  - Semaphores



SEMMNI	Maximum number of semaphore identifiers (i.e., sets)	at least `ceil((max_connections + autovacuum_max_workers + max_wal_senders + max_worker_processes + 5) / 16)` plus room for other applications
SEMMNS	Maximum number of semaphores system-wide	`ceil((max_connections + autovacuum_max_workers + max_wal_senders + max_worker_processes + 5) / 16) * 17` plus room for other applications
SEMMSL	Maximum number of semaphores per set	at least 17

- **Requirement of higher limits for connections.**
  - Server Abuse and Outages

Impact of authentication

Query over Fresh Connection

Trusted Connection
tps = **129.153727** (including connections establishing)
tps = 179.175602 (excluding connections establishing)

Password authentication
tps = **95.778541** (including connections establishing)
tps = 147.387896 (excluding connections establishing)

Password over TLS
tps = **72.597286** (including connections establishing)
tps = 117.009713 (excluding connections establishing)

Query over Persistent connection

Password over TLS
tps = 658.220014 (including connections establishing)
tps = 658.519444 (excluding connections establishing)

PERCONA

The solution : Connection poolers

- **Application side Connection pooler**
  - Support from ORMs
  - Right-sizing challenges
- **External connection pooler**
  - When there is no support by Application frameworks / language
  - Third party applications

What external poolers do

- Initialize a pool of connection at the startup or at the first connection.
- A user and database combination defines a pool.
- Connections are released back to pool. But not disconnected from the database side.
- Limited number of connections serving larger number of application connections.
- Connection **request queueing** at pool level.
- **Routing** of the connection

PERCONA

Popular External Poolers

pgpool-II

- **First version in 2003**
- **Connection Pooler**
- **High Availability Solution for PostgreSQL**
  - New quorum and consensus based backend failover
- **SPOF of pgpool can be avoided using multiple pgpool and Watchdog, Virtual IP failover**
- **Replication by managing multiple backend PostgreSQL**
- **Connection queuing.**
- **query cache**

pgpool-II

- High complexity
- Overhead

pgbouncer

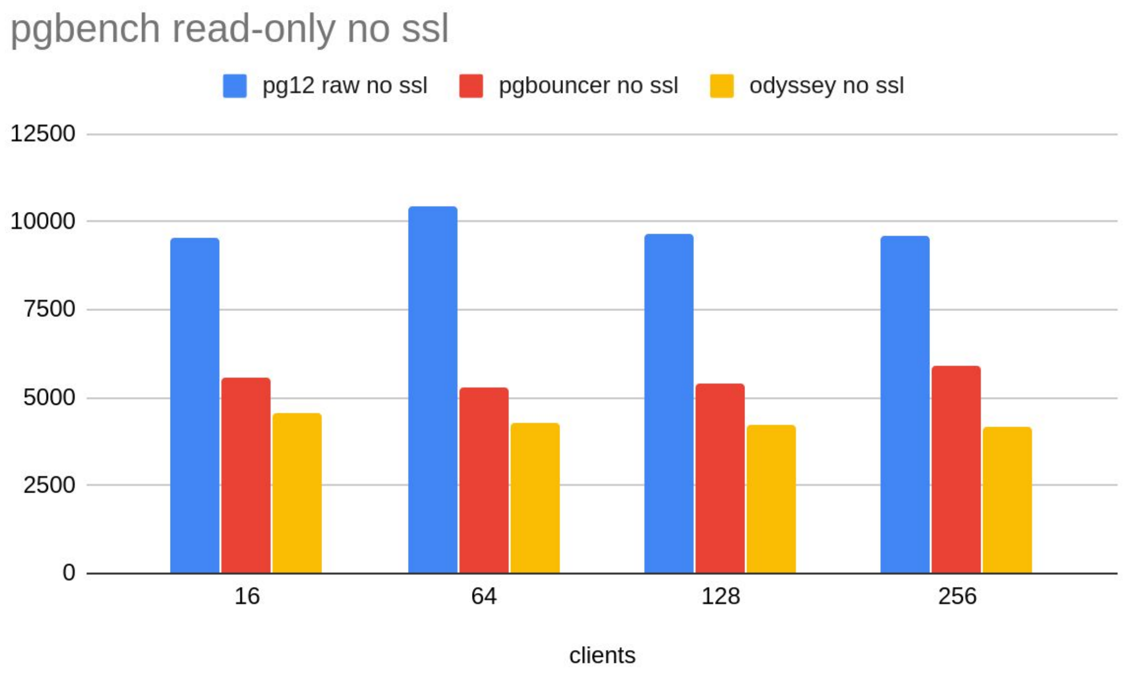
- Lightweight and Performant
- SSL support
- Popular
- Monitoring
  - https://github.com/prometheus-community/pgbouncer_exporter
- Available in almost all repositories.

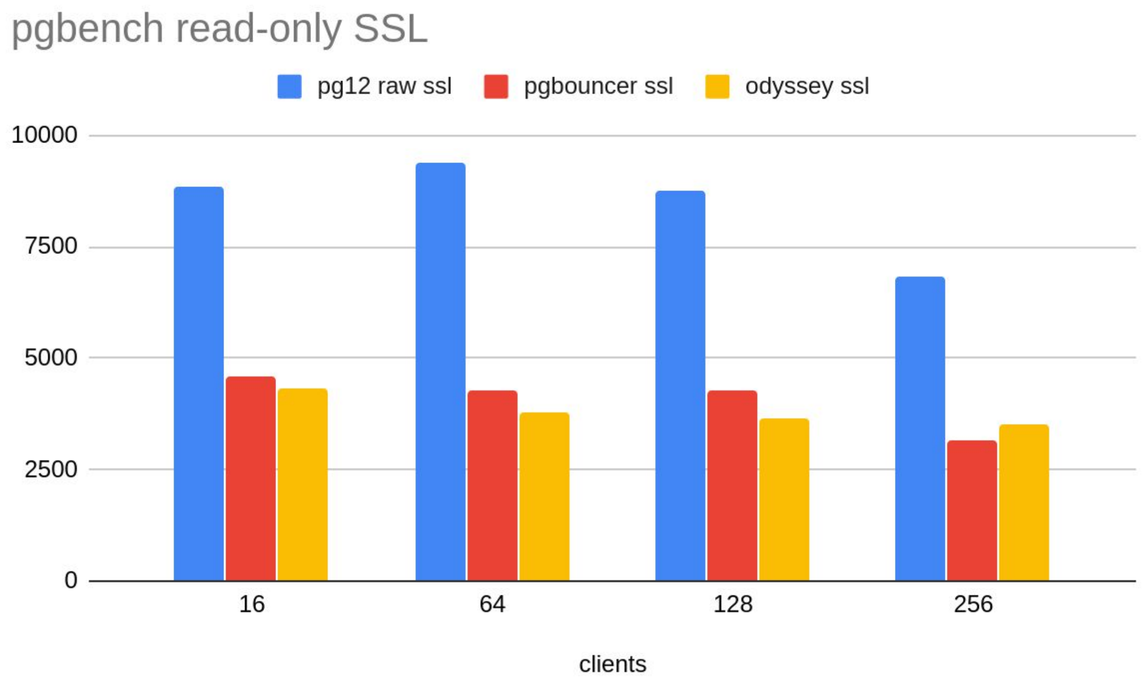
One Instance of pgbouncer uses only one CPU core.
Refer so_reuseport more details with latest kernel

Odyssey

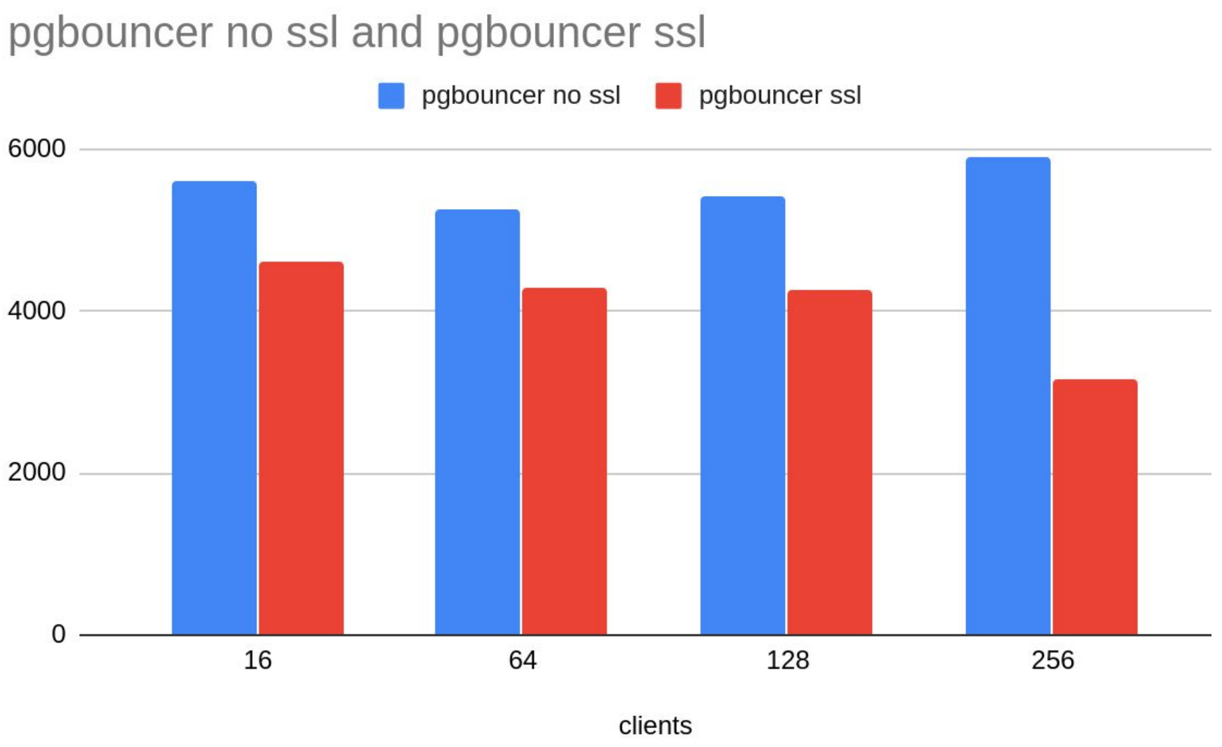
- Multi-threaded using worker threads
- Feature parity with pgbouncer
- Very new
- Requires OpenSSL 1.1 (CentOS 8 or custom build)
- Monitoring (same as pgbouncer)
  - https://github.com/prometheus-community/pgbouncer_exporter
- Improves on pgbouncer pitfalls, like performance with SSL

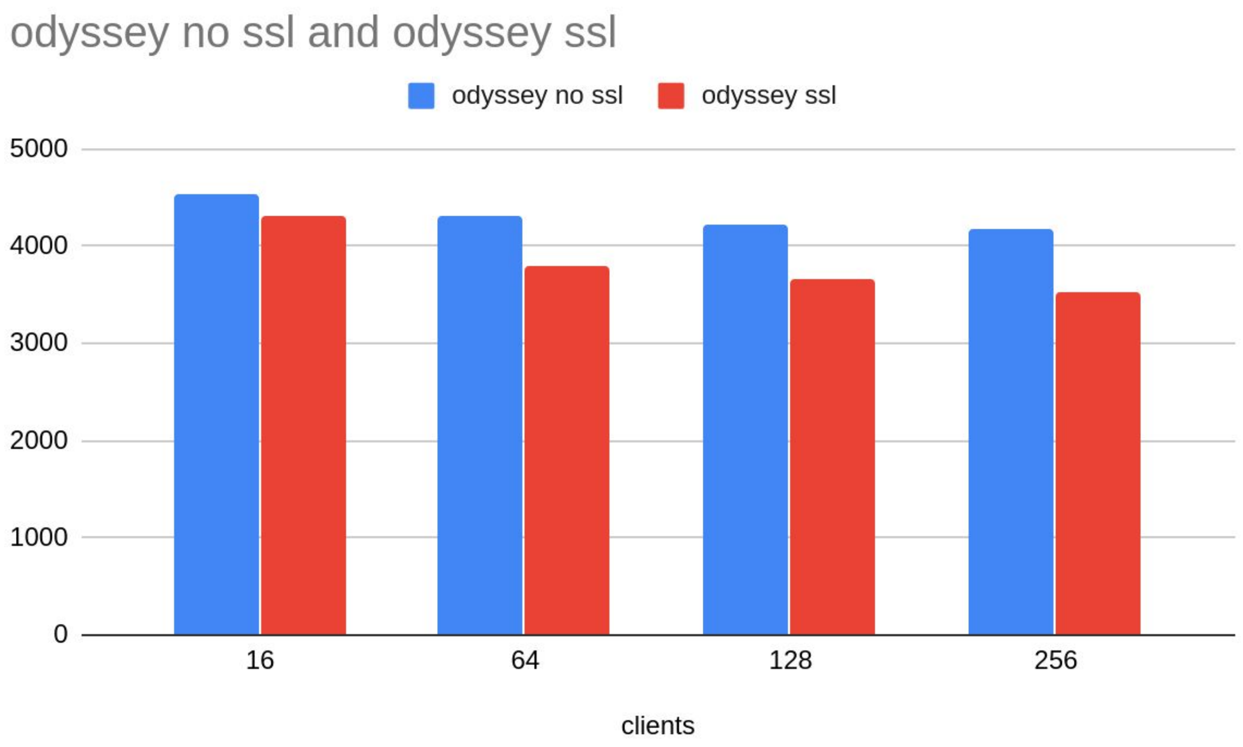
Persistence at application



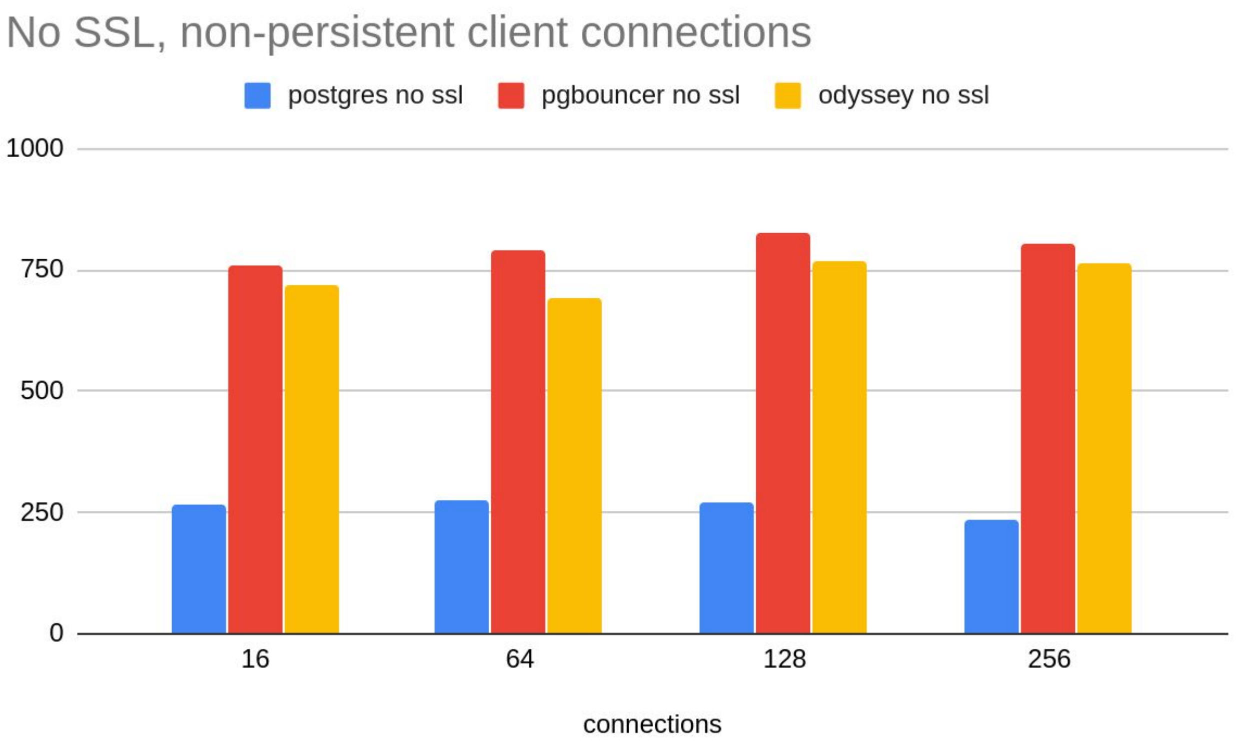


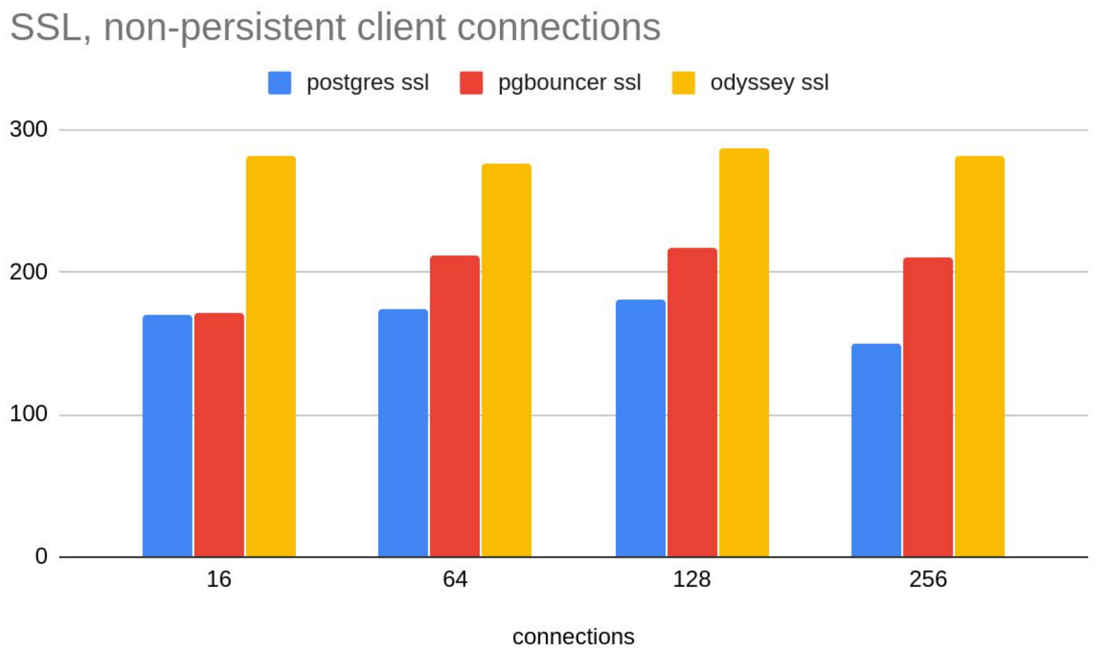
pgbouncer and odyssey





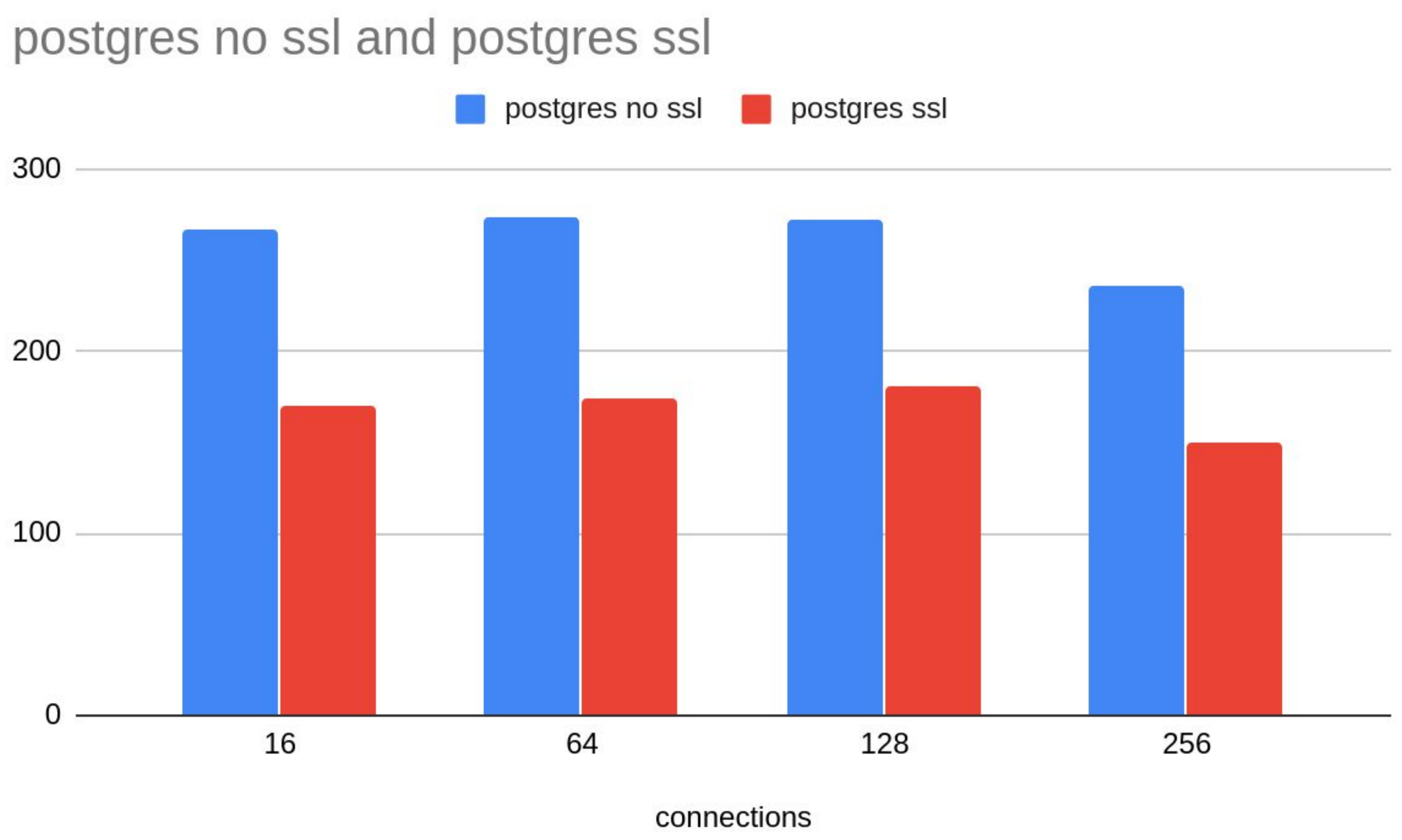
New Connections



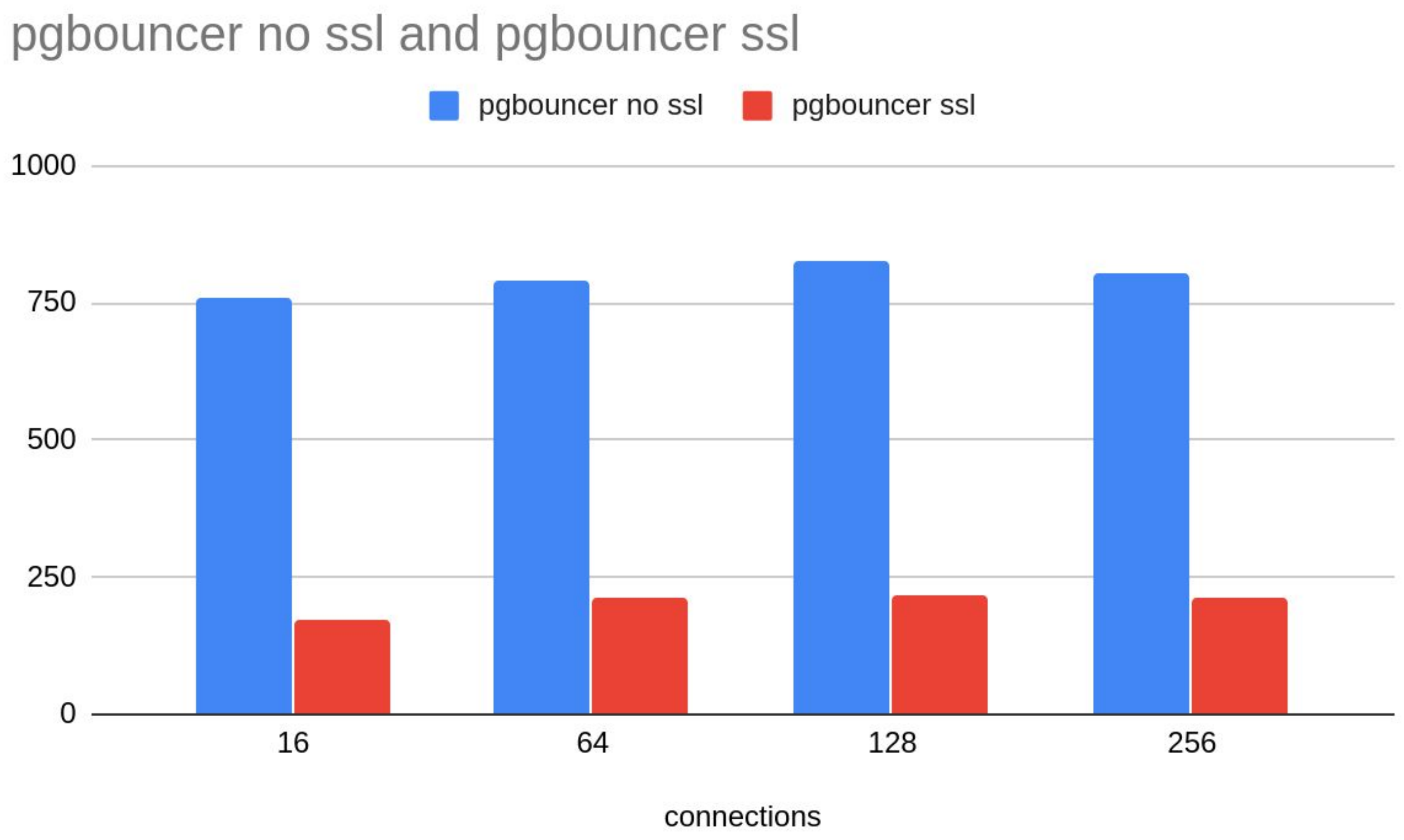


PERCONA

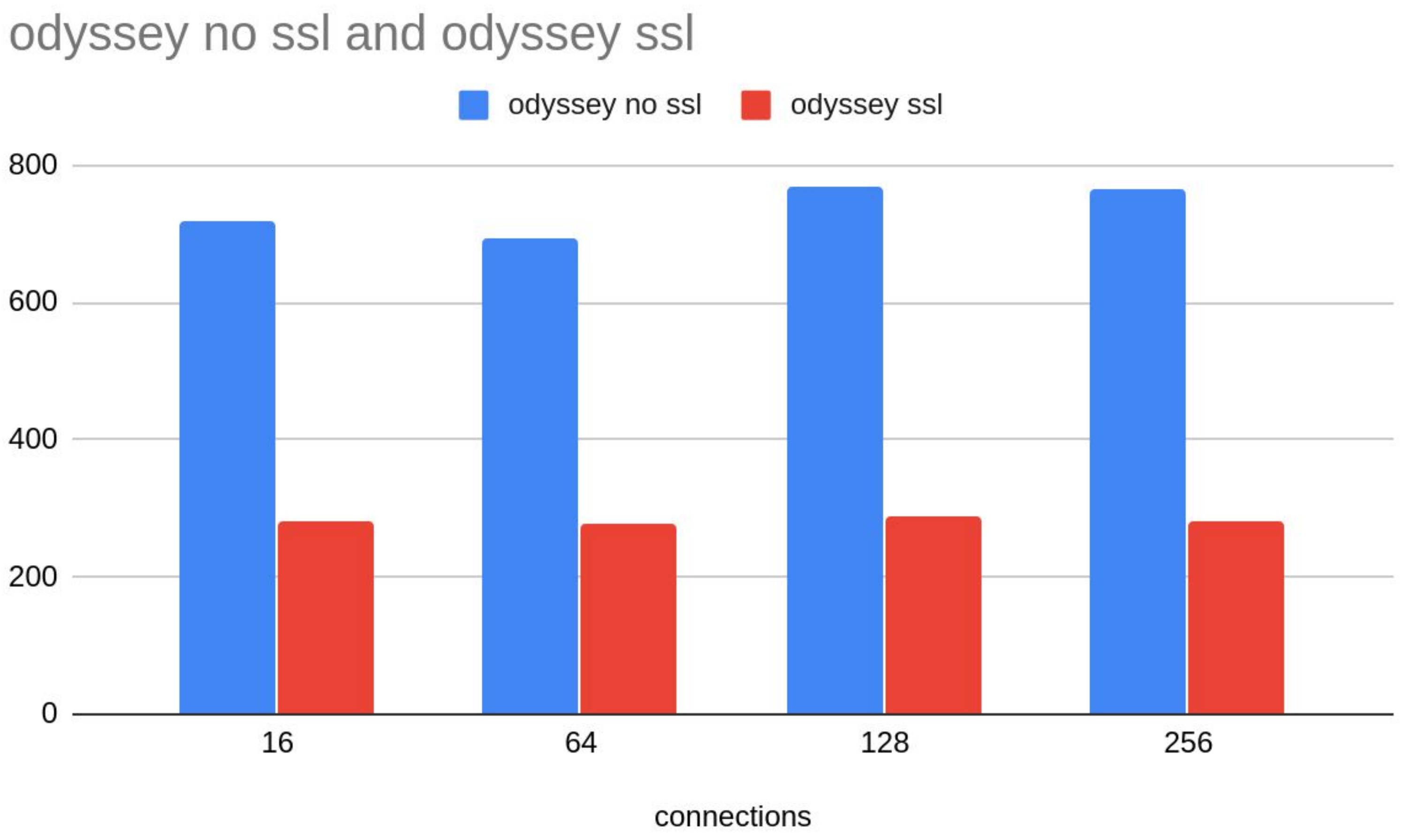
Overhead of SSL - No Pooler



Overhead of SSL - PgBouncer



Overhead of SSL - odyssey



Limitations of Connection Poolers

Limitations of External connection Poolers

Security, Auditing, Monitoring and troubleshooting Challenges

- Client address in the server logs points to pooler
- Server side audits won’t have end client information
- tcpdump of the connection stream won’t be possible
    - client may use different connection each time
- Sessions are shared

Its a man in middle

PERCONA

Limitations of External connection Poolers

Limitations On functionality

- prepared statements
- temporary tables

Limitations of External connection Poolers

Performance and Stability

- Applications which needs a persistent connection won't see any performance benefit.
  - Releasing to pool and getting it back
  - Extra hop
- I could be performance overhead
  - Eg : serve_reset_query = DISCARD ALL; or at least to DEALLOCATE ALL
- They won't remove SPOF (they become SPOF!)

PERCONA

References

prepared statements in Odyssey
https://github.com/yandex/odyssey/issues/16
https://blog.bullgare.com/2019/06/pgbouncer-and-prepared-statements/
https://pgconf.ru/media/2017/04/03/20170316H1_V.Borodin.pdf
potential prepared statements workaround
https://github.com/dimitri/preprepare
pg_hba in odyssey
https://github.com/yandex/odyssey/issues/75
cursor :
https://github.com/yandex/odyssey/issues/45
leader failover
https://github.com/yandex/odyssey/issues/44
prepared statements in pgbouncer
https://github.com/pgbouncer/pgbouncer/issues/499
application name reset in pgbouncer
https://github.com/pgbouncer/pgbouncer/issues/457
https://www.pgpool.net/docs/latest/en/html/runtime-config-connection-pooling.html
pgpool only has session-level pooling
https://github.com/yandex/odyssey/issues/3
https://github.com/kwent/pgbouncerhero
Pecrona Blog
https://www.percona.com/blog/2018/06/27/scaling-postgresql-with-pgbouncer-you-may-need-a-connection-pooler-sooner-than-you-expect/

ONLINE, AROUND-THE-WORLD OPEN SOURCE DATABASE CONFERENCE

PERCONA

LIVEONLINE

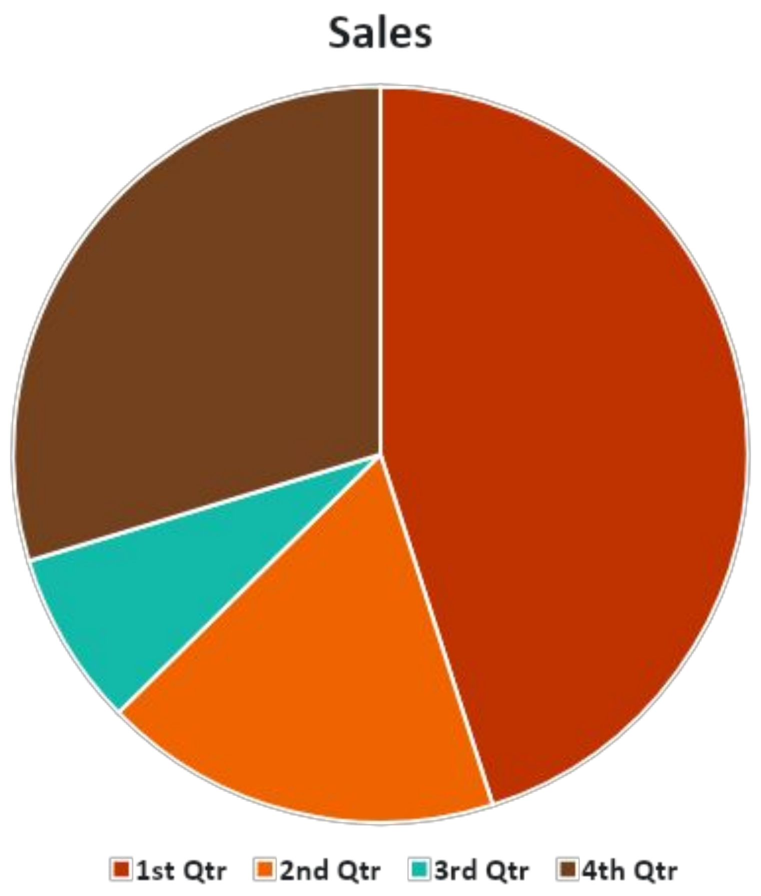
20-21 OCTOBER

2020

Two-Column Slide

This symmetrical layout is optimized for approximately square charts or graphics.

- Use only level-one bullets when possible
  - *Use level two bullets and below judiciously to maximize legibility*

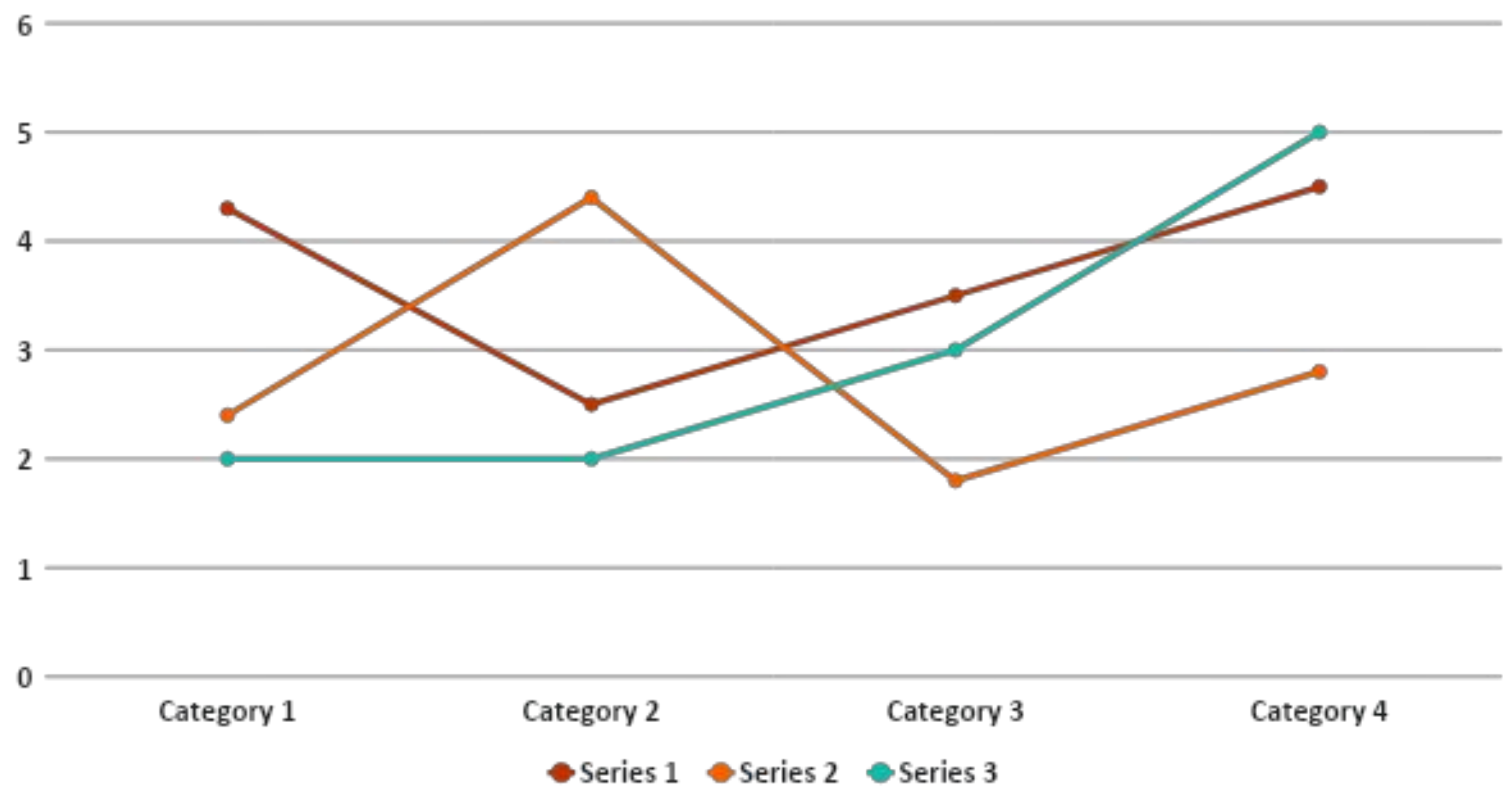


PERCONA

Two-Column Offset

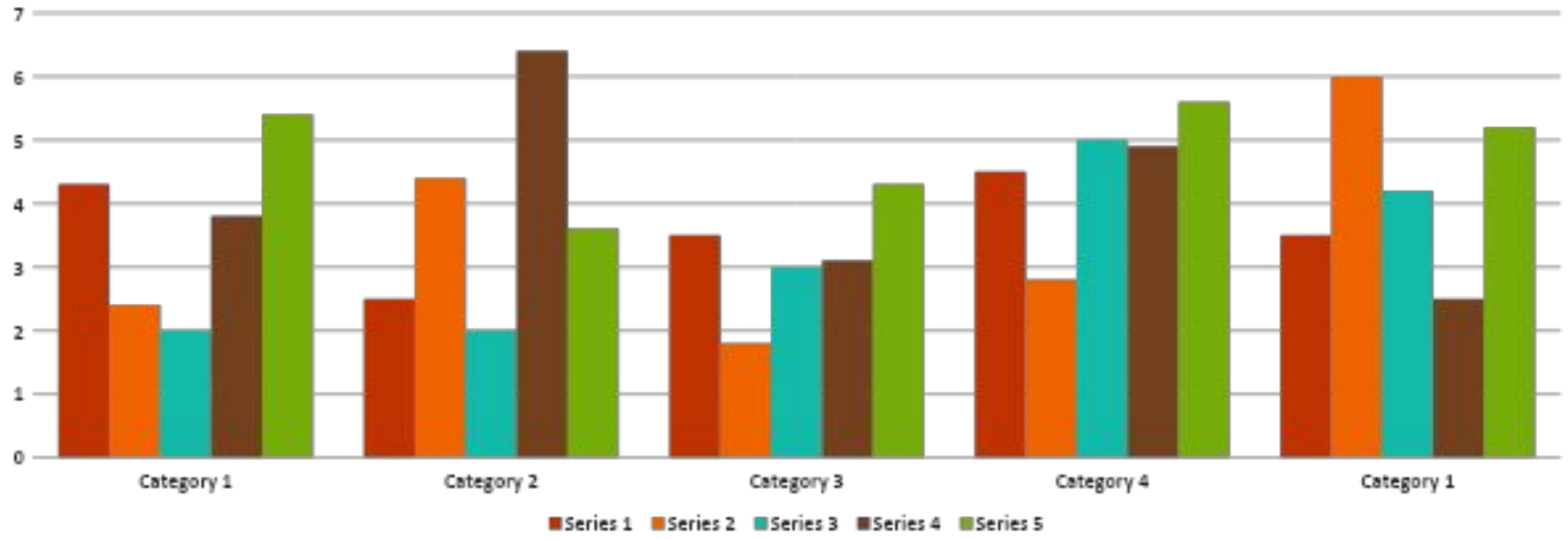
Narrowed text box accommodates wider charts and graphics

- Bullets should be short and concise
- Use of level one bullets increases legibility and maintains a clean look



PERCONA

Single-Column Slide for Horizontal Content



About Percona

- Solutions for your success with MySQL and MongoDB
- Support, Managed Services, Software
- Our Software is 100% Open Source
- Support Broad Ecosystem – MySQL, MariaDB, Amazon RDS
- In Business for 10 years
- More than 3000 customers, including top Internet companies and enterprises

PERCONA

About This Presentation

Overview of Existing Solutions and History

Discuss what we use at Percona

Show what specific things to look at

Smart Object Chevron List

Hired the Two Bobs as Consultants	They cherish the power to fire people at will	They also cherish and celebrate Michael Bolton
Peter Gibbons interviewed by the Two Bobs	Expresses loathing and disdain for his job	Immediately promoted to management
Michael Bolton interviewed by the Two Bobs	Lies about liking Michael Bolton to keep his job	Fired immediately

PERCONA

Champions of Unbiased
Open Source Database Solutions