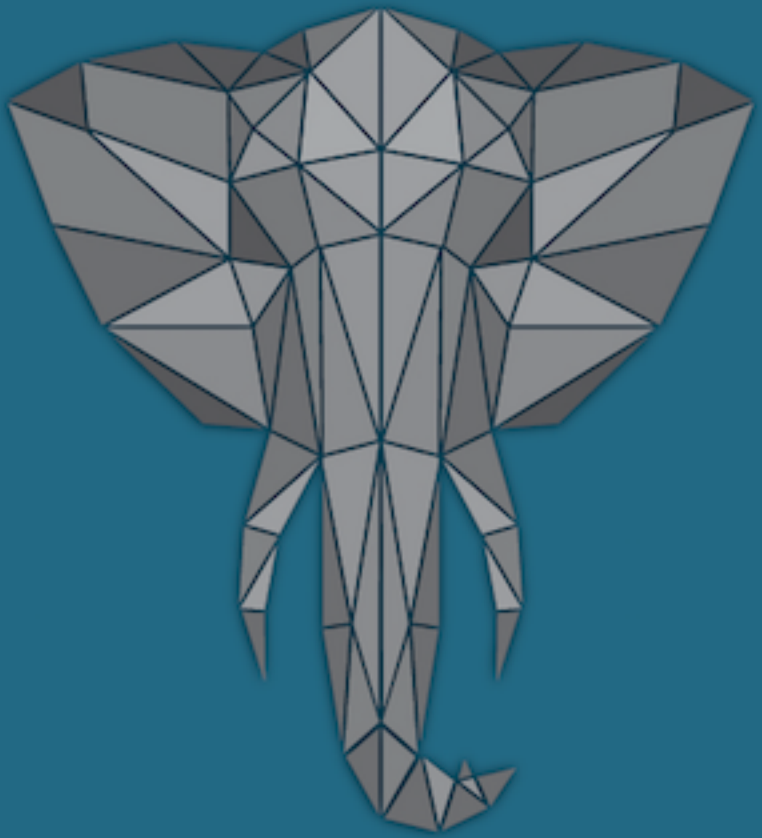
PostgreSQL Configuration for *Humans*



ONGRES

Álvaro Hernandez Tortosa

ALVARO HERNANDEZ

CEO

TELECOMMUNICATION ENGINEER
SERIAL ENTREPRENEUR (NOSYS, WIZZBILL, 8KDATA)
WELL-KNOWN MEMBER OF THE POSTGRESQL COMMUNITY
WORLD-CLASS DATABASE EXPERT (+30 PAPERS IN LAST 2 YEARS)

About ONGRES

- IT firm specialized on R&D on Databases, more specifically PostgreSQL:
  - Training
  - Consulting and development
  - PostgreSQL Support

- Developers of ToroDB (www.torodb.com), an open-source, document-store database that works on top of PostgreSQL and is compatible with MongoDB.

- Partners of www.pythian.com, reference multinational company providing database support and data services.

PostgreSQL configuration

- ✓ Mostly **postgresql.conf** (*here's most of the meat*)
- ✓Authentication: **pg_hba.conf** (and pg_ident.conf
- ✓Replicas: **recovery.conf** (may be merged soon)
- ✓Some **initdb** parametersPer-object settings (eg. *fillfactor*)

Advanced stuff:

- ✓ Developer parameters
- ✓Compile-time **#define**s

Why configure PostgreSQL?

- It only listens on localhost
- You can only replicate by default in >= 10
- To enable WAL archiving
- Application developers say they get “connection refused!”
- Set defaults for client connections
- Load extensions that required shared libraries
- Enable checksums (initdb!)Compatibility with older versions

Any other reason? ;)

Performance, performance, performance

Usual performance optimization advice:

Don't, don't, don't lag

Usual **PostgreSQL** advice:

Do, Do, Do

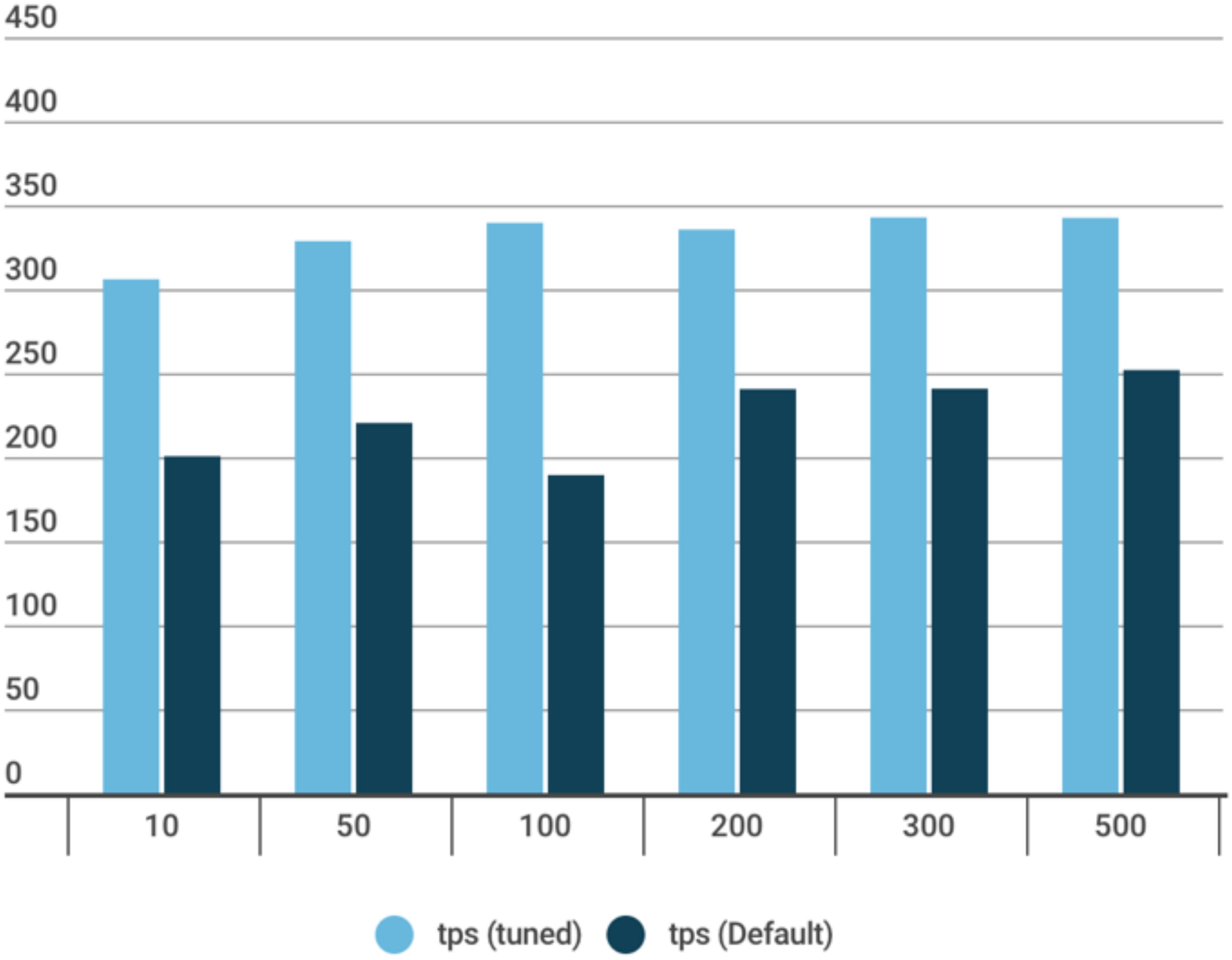
(unless you run on a Raspberry PI 1st gen)

✓ Run your own app-centric benchmarks

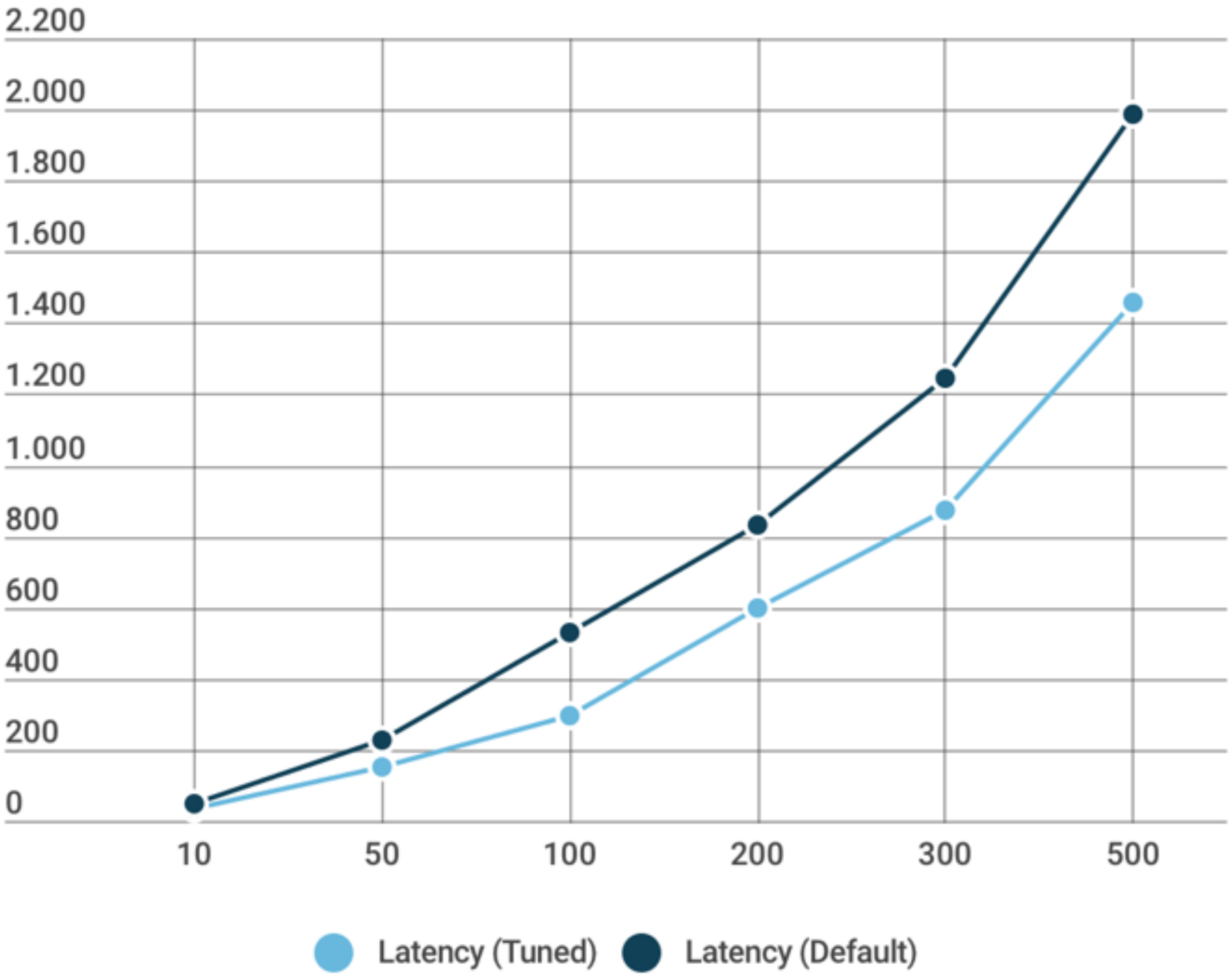
✓pg_bench is just one benchmark more

All the usual precautions about benchmarks apply

Performance, performance, performance



Performance, performance, performance



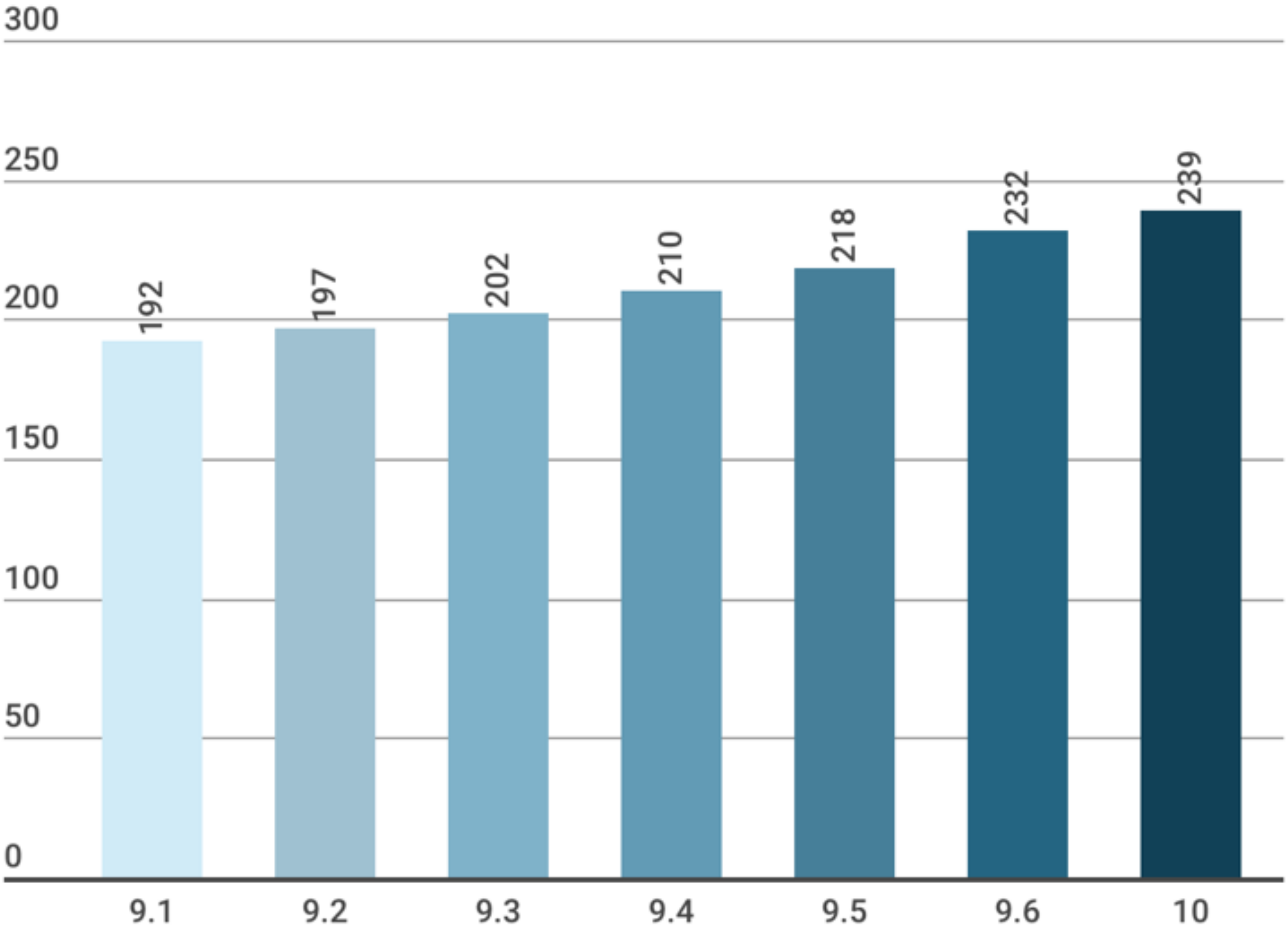
postgresql.conf parameters

- ✓ More than 200 parameters (*no kidding!*)
- ✓Classified into 40 categories / subcategories
- ✓650 lines, 23Kb *sample* config file

How many to tune? **2? 5? 10? 20? 40? 100?**

- ✓ Parameters are integer, real, string, enum, real or bool. Numeric values may have units (or are unit-less)
- ✓Some units are a bit uneasy (like "blocks of 8Kb") or too synthetic (*cpu_tuple_cost*)

postgresql.conf parameters



Some ideas about PostgreSQL tuning…

YMMV   !!

Disclaimer

✓ No, you won’t get final numbers here to write in your postgresql.conf

✓ Only a few dozens parameters discussed here

✓ Only hints provided: do your homework

✓ My opinion may differ from other’s

✓ I am probably wrong

✓ YMMV

(you got the point)

initdb

- ✓ Sometimes run on your behalf (Debian/Ubuntu), bad for selecting non defaults
- ✓ -E (encoding). Use UTF-8 unless you know what you do
- ✓ --locale, --lc_collate, –lc-ctype
- ✓ --username: if ‘postgres’ is not the superuser
- ✓ --data-checksums: enable them!

All the usual precautions about benchmarks apply

Db connections 101

✓ max_connections is a hard limit

✓PostgreSQL will reject connections over this number

✓Users not happy

✓Default is 100

✓“My app has more 100 concurrent users!

Solution is obvious: raise it!

Db connections 101

~~Solution is obvious: raise it!~~

✓ One process per connection (excl. parallelism!)

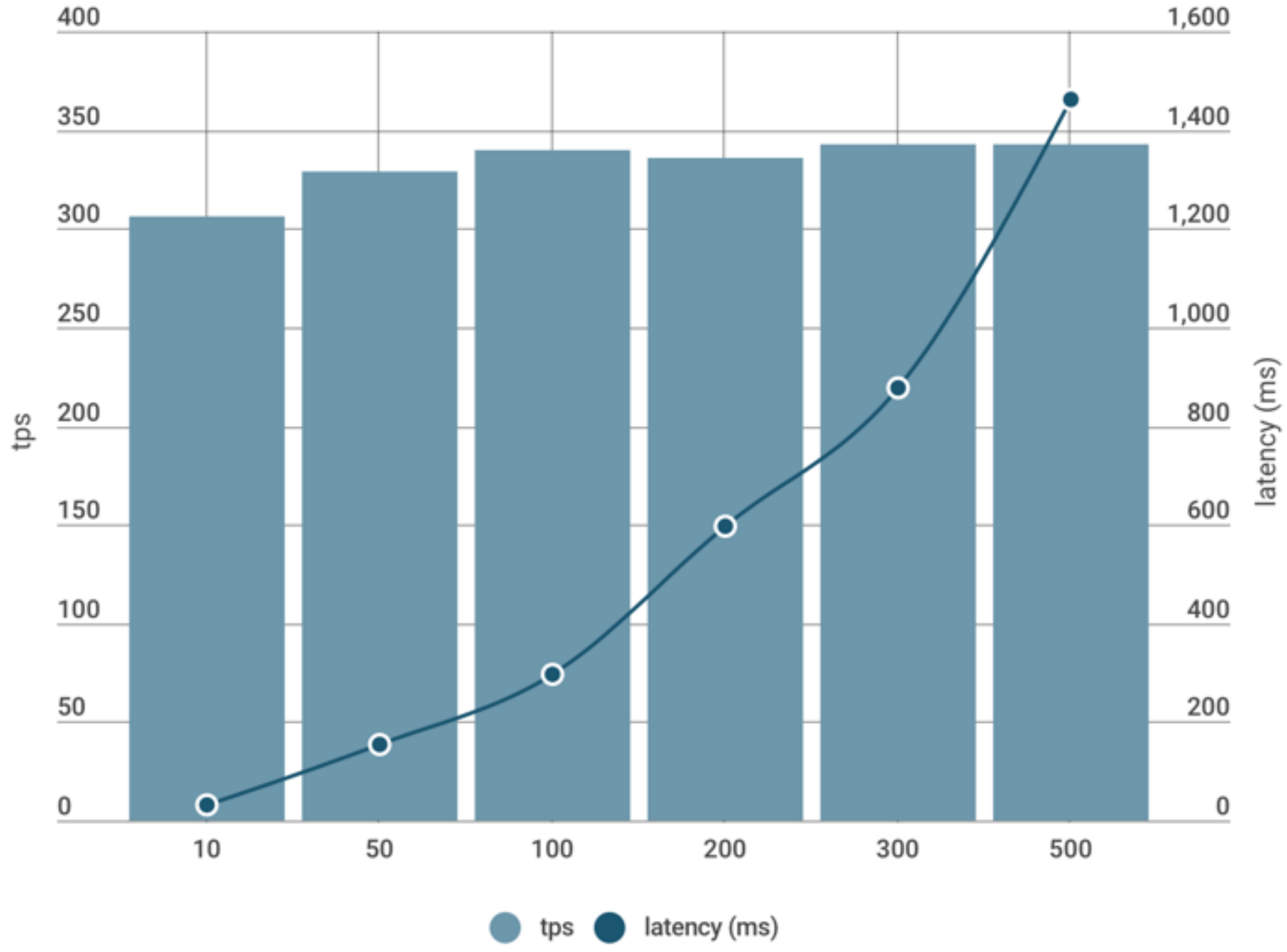
✓ One process handled by one core

✓ How many cores do you have?

✓ Sure, you have a multi-process, time-sharing OS but what is the scheduling overhead with many processes?

Much worse: cache trashing!

Db connections 101



Db connections 101

- ✓ Solution is obvious: lower it!
- ✓But how we solve the connection refused problem?
- ✓PgBouncer!
- ✓Size your connections almost 1:1 pgbouncer:max_conns
- ✓Use this formula:

$$connections = \frac{cores}{\%\,effective\,utilization\,connection} * scale\ factor$$

shared_buffers

✓ The first recommendation everybody tells you

✓ Set it to 1/4th of RAM and effective_cache_size 3/4th

✓ ~~Done!~~

✓ ¼ too low on low memory, too high on high memory

✓ **Benchmark, benchmark, benchmark**

✓ Is the db dedicated in the host? OS memory?

✓ How high is work_mem * max_connections, maintenance_work_mem, etc?

work_mem

- ✓ Max local process memory used for operations like sort and joins in queries
- ✓ Not written in stone: users can SET it overriding its value
- ✓ But if more memory is used, it spills to disk (and may use different algorithm) reducing performance
- ✓ Not the same if you are OLTP, OLAP, DW (small to very large)
- ✓ Raise it from defaults, but don't forget it could be times max_connections

Other memory/disk tunables

✓ maintenance_work_mem: vacuum, create index, check FKs…. raise it

✓{min,max}_wal_size: it’s only disk space, but too low will cause excessive checkpoints. Make min at least 1GB, max several GB up to 50-100GB.

✓stats_temp_directory: run on a RAMdisk

This requires restart, think carefully

- ✓ listen_addresses (take care with ‘*’), port
- ✓ ssl: activate only if needed, use pooling!
- ✓ huge_pages: benchmark, benchmark, benchmark (typically off)
- ✓ shared_preload_libraries: add your extensions beforehand!
- ✓ logging_collector: on
- ✓ wal_level: replica or *logical*
- ✓ archive_mode, archive_command
- ✓ cluster_name

The tyranny of max_*

✓ Most of the max_* also require restart

✓Sometimes hard to estimate, but restarting the database is pretty bad: raise limits and control usage

✓max_wal_senders: replicas, backups, make room

✓max_replication_slots

✓max_worker_processes, max_parallel_workers_per_gather, max_parallel_workers

✓autovacuum_max_workers (# cores for cleanup?)

✓max_prepared_transactions (2PC, 0 by default)

Autovacuum / vacuum

✓ Almost always too conservative

✓Bloat is one of the most frequent operational burdens

✓Hard to get it right: analyze and re-tune periodically

✓Some parameters are set to “-1” which means “look at these numbers from the vacuum parameters”

✓autovacuum_{vacuum,analyze}_scale_factor: you may override on a per-table level

Autovacuum / vacuum

General advice:

- ✓ Raise vacuum_cost_limit significantly
- ✓ Reduce autovacuum_vacuum_cost_delay
- ✓ Use more autovacuum_max_workers if you

Checkpoints and bgwriter

- ✓ You typically want to spread checkpoints a bit far apart (raise checkpoint_timeout)
- ✓ min_wal_size 1GB min
- ✓ Log checkpoints and look for warnings
- ✓ Raise checkpoint_completion_target, eg. 0.9, but study your I/O patterns, shared_buffers, wal size
- ✓ Increase bgwriter activity, very conservative default:
- ✓ Decrease bgwriter_delay
- ✓ Increase bgwriter_lru_maxpages
    - ‣ Decrease bgwriter_delay
    - ‣ Increase bgwriter_lru_maxpages

Logging

✓“Only” 24 parameters (plus some others)

✓Spend some time here, it pays off analyzing your db

✓Turn on:

- logging_collector
- log_checkpoints
- log_connections, log_disconnections

✓You typically want to spread checkpoints a bit far apart (raise checkpoint_timeout)

Other interesting parameters

✓ password_encryption use SHA-256 if possible (>= 10)

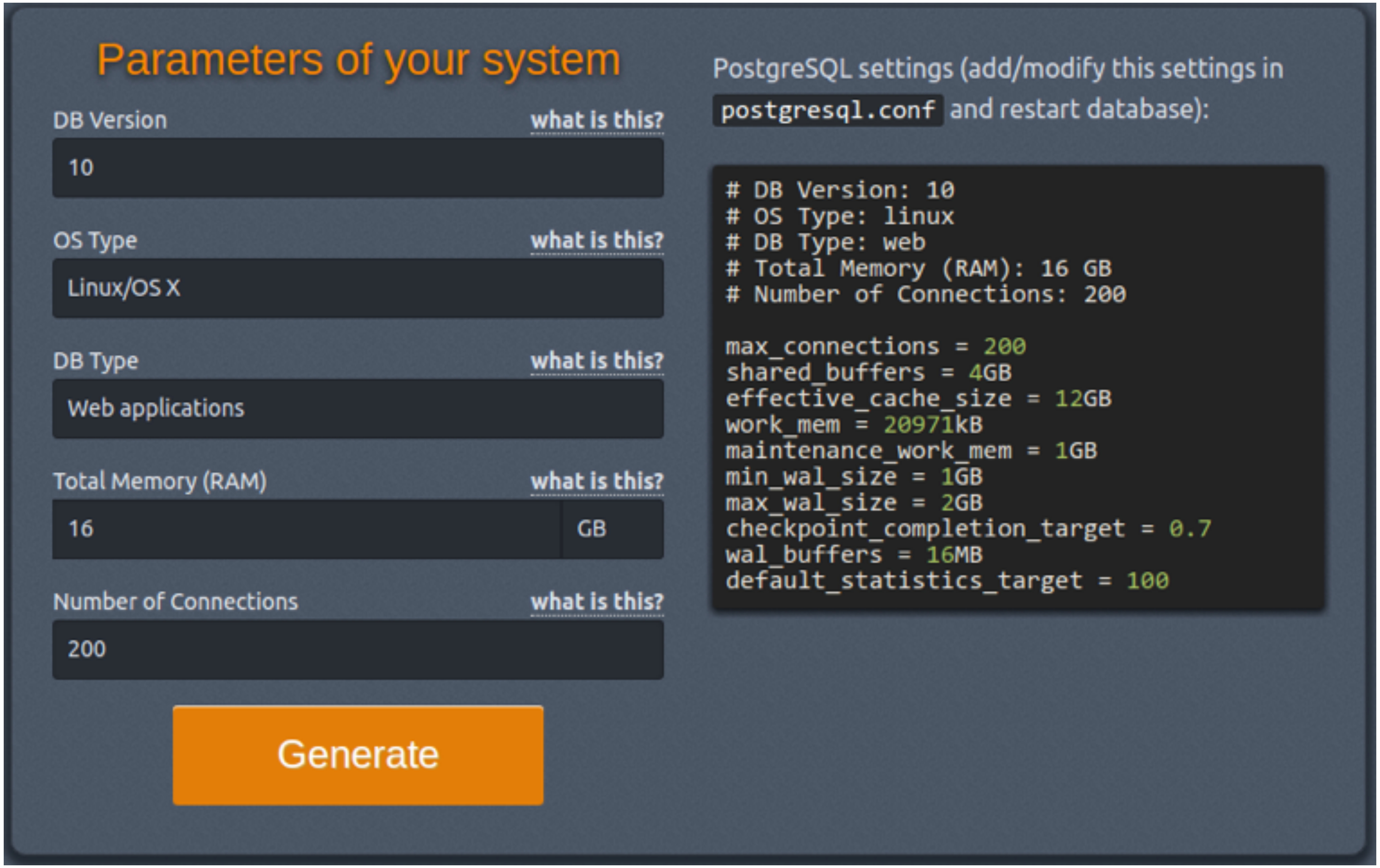
✓effective_io_concurrency (how many “spindles” your I/O has?)

✓max_standby_{archive,streaming}_delay and hot_standby_feedbak: keep replication query conflicts burden on the primary or secondaries?

✓default_statistics_target: if setting by table is not your job

Was that too much? Tools to help?

Was that too much? Tools to help?



Parameters of your system

DB Version — what is this?
10

OS Type — what is this?
Linux/OS X

DB Type — what is this?
Web applications

Total Memory (RAM) — what is this?
16 GB

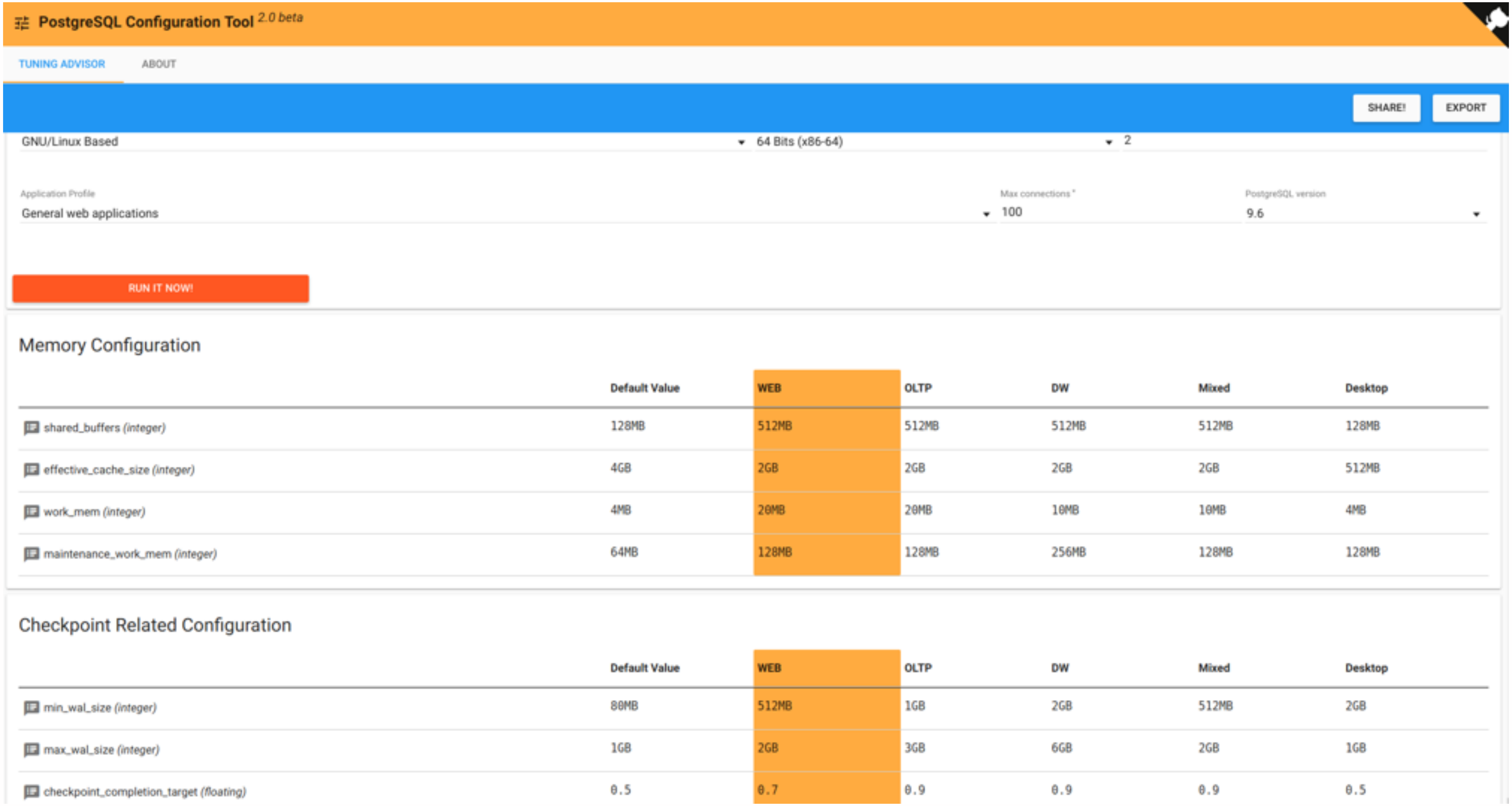
Number of Connections — what is this?
200

PostgreSQL settings (add/modify this settings in `postgresql.conf` and restart database):

```
# DB Version: 10
# OS Type: linux
# DB Type: web
# Total Memory (RAM): 16 GB
# Number of Connections: 200

max_connections = 200
shared_buffers = 4GB
effective_cache_size = 12GB
work_mem = 20971kB
maintenance_work_mem = 1GB
min_wal_size = 1GB
max_wal_size = 2GB
checkpoint_completion_target = 0.7
wal_buffers = 16MB
default_statistics_target = 100
```

Was that too much? Tools to help?



PostgreSQL Configuration Tool 2.0 beta

TUNING ADVISOR   ABOUT

SHARE!   EXPORT

GNU/Linux Based ▾   64 Bits (x86-64) ▾   2

Application Profile: General web applications ▾
Max connections *: 100
PostgreSQL version: 9.6 ▾

RUN IT NOW!

Memory Configuration

	Default Value	WEB	OLTP	DW	Mixed	Desktop
shared_buffers *(integer)*	128MB	512MB	512MB	512MB	512MB	128MB
effective_cache_size *(integer)*	4GB	2GB	2GB	2GB	2GB	512MB
work_mem *(integer)*	4MB	20MB	20MB	10MB	10MB	4MB
maintenance_work_mem *(integer)*	64MB	128MB	128MB	256MB	128MB	128MB

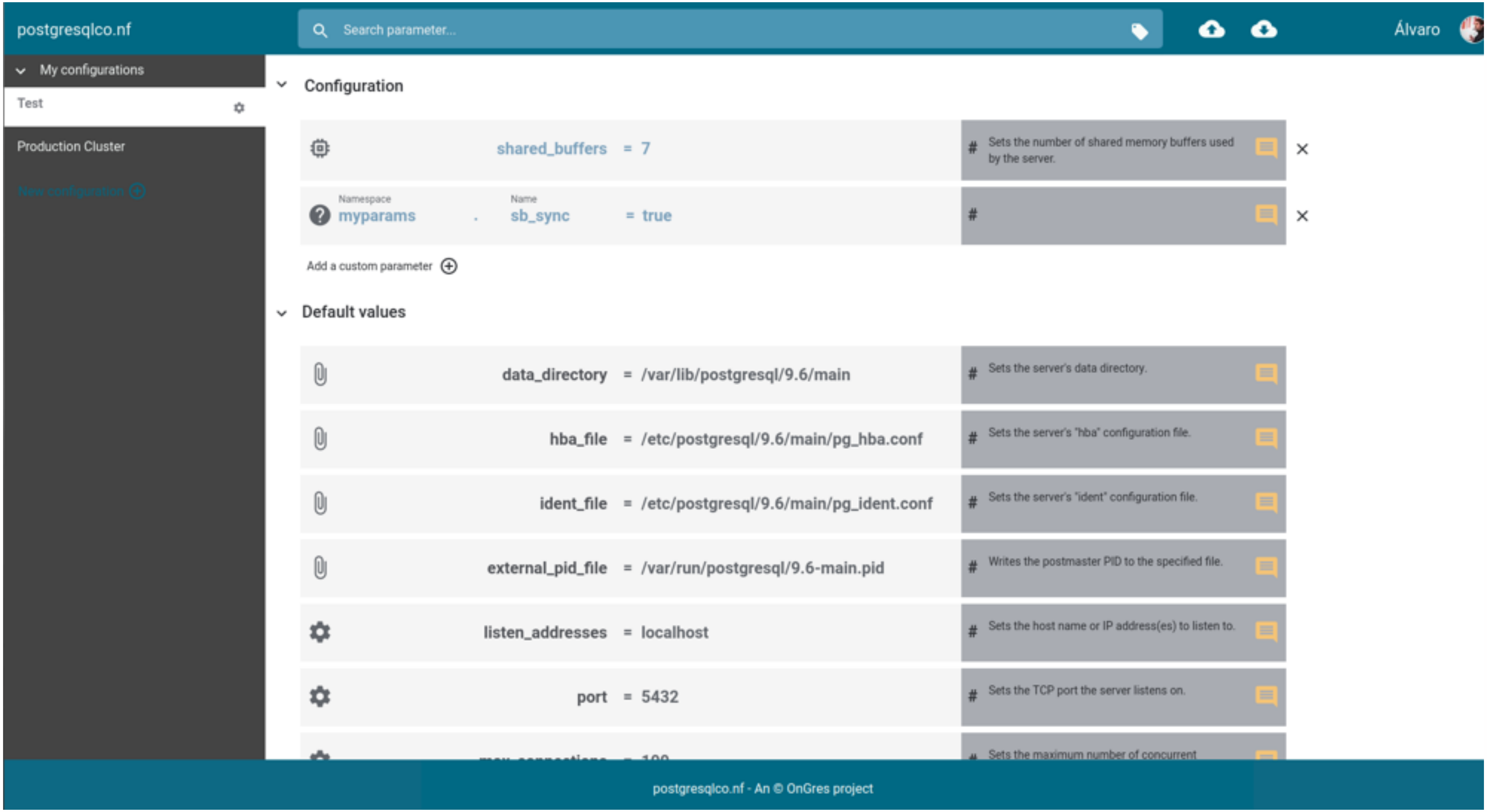
Checkpoint Related Configuration

	Default Value	WEB	OLTP	DW	Mixed	Desktop
min_wal_size *(integer)*	80MB	512MB	1GB	2GB	512MB	2GB
max_wal_size *(integer)*	1GB	2GB	3GB	6GB	2GB	1GB
checkpoint_completion_target *(floating)*	0.5	0.7	0.9	0.9	0.9	0.5

PostgreSQL wants a new configuration tool

(IMVHO)

postgresqlco.nf



Configuration

shared_buffers = 7 — # Sets the number of shared memory buffers used by the server.

Namespace myparams . Name sb_sync = true — #

Add a custom parameter

Default values

data_directory = /var/lib/postgresql/9.6/main — # Sets the server's data directory.

hba_file = /etc/postgresql/9.6/main/pg_hba.conf — # Sets the server's "hba" configuration file.

ident_file = /etc/postgresql/9.6/main/pg_ident.conf — # Sets the server's "ident" configuration file.

external_pid_file = /var/run/postgresql/9.6-main.pid — # Writes the postmaster PID to the specified file.

listen_addresses = localhost — # Sets the host name or IP address(es) to listen to.

port = 5432 — # Sets the TCP port the server listens on.

max_connections = 100 — # Sets the maximum number of concurrent

postgresqlco.nf - An © OnGres project

postgresqlco.nf

- ✓ Web configuration tool
- ✓ Drag&drop your postgresql.conf and tune it!
- ✓ Integrated help
- ✓ Recommendations
- ✓ Easy search & filtering
- ✓ Create your configurations
- ✓ Download as postgresql.conf, json, yaml, SQL
- ✓ Rest API

Questions?