



# Deep dive into Query Performance

Peter Zaitsev

Founder at Percona

5 Feb 2023



nightb

CODE2040

BLACK GIRLS CODE

I AM THE FUTURE

Twitter

GOD IS

technovation

© Percona

# Database is a Black Box





**You can connect to the Database Service Point, Quickly**



**Run Queries you need to run**

**Meaning**

# Queries

1

**Run them  
without  
errors**

2

**Run them  
with correct  
results**

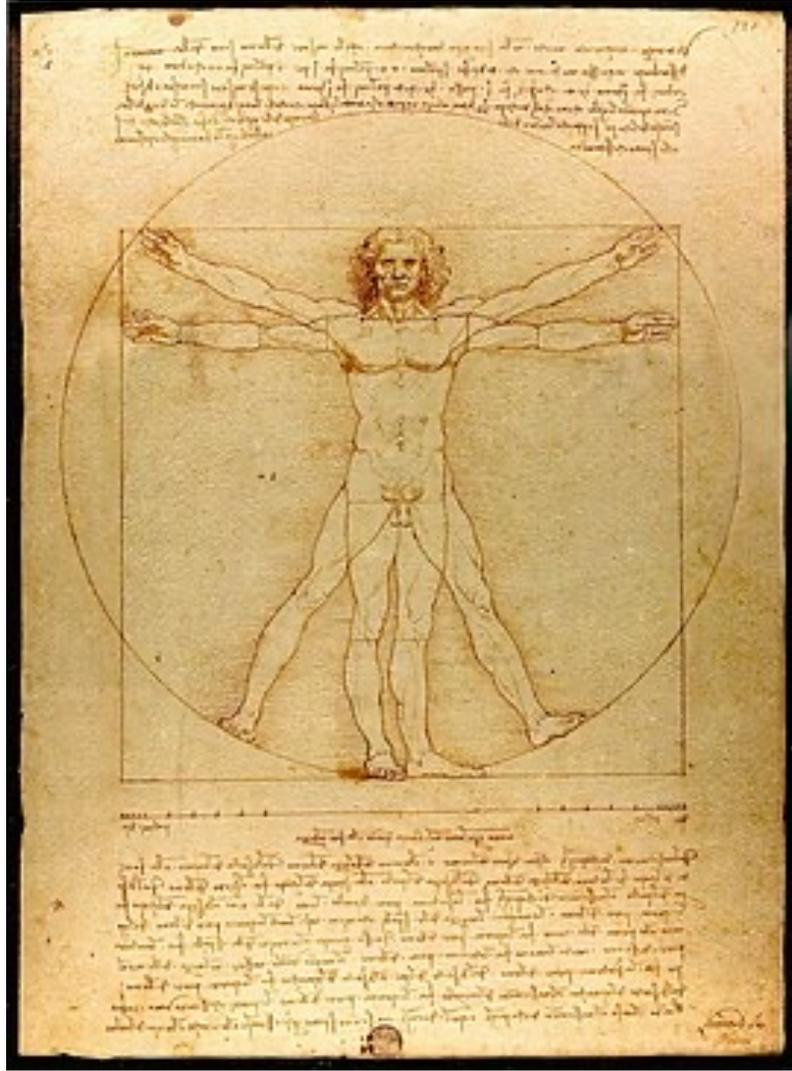
3

**Run them  
with required  
response  
time**



# Performance

**Performance is about Response Time  
you get for your Queries**



# Great design is not only about Performance

- Security
- Availability
- Costs
- Maintability
- Impact on other users

# Response Time - Database View

**"I see database responds to queries in 5ms in average"**



# Response Time – Business View

**All Users have outstanding  
performance experience with all their  
application interactions**



# Downtime

**Very Bad Performance is  
indistinguishable from downtime**

An aerial photograph of a river meandering through a dense, lush green forest. The river is dark and reflects the surrounding trees. The forest is thick and covers the entire landscape.

# Forget averages

There once lived a man who tried to cross a river,  
in average one meter deep

PMM Query Analytics ▾

Reset All



#	Query ▾	Search by...	🔍	Load	Progress	QPS	Latency
	TOTAL						7.44 ms
1	update warehouse1 set w_ytd = w_ytd + ? where w_id = ?		①				231.41 ms
2	select i_price, i_name, i_data from item1 where i_id = ?		①				18.76 ms
3	select c from sbtest1 where id=?		①				1.87 ms
4	select d_next_o_id, d_tax from district1 where d_w_id = ? and d_id ...		①				69.65 ms
5	update district1 set d_ytd = d_ytd + ? where d_w_id = ? and d_id= ?		①		1.25 load	25.49 QPS	48.95 ms
6	insert into order_line1 (ol_o_id, ol_d_id, ol_w_id, ol_number, ol_i_id, ...		①		0.61 load	255.30 QPS	2.40 ms
7	update stock1 set s_quantity = ? where s_i_id = ? and s_w_id= ?		①		0.53 load	255.30 QPS	2.07 ms
8	select count(distinct (s_i_id)) from order_line1, stock1 where ol_w_...		①		0.34 load	2.56 QPS	132.84 ms
9	commit		①		0.28 load	58.43 QPS	4.74 ms
10	insert into new_orders1 (no_o_id, no_d_id, no_w_id) values(?,?)		①		0.27 load	25.57 QPS	10.50 ms
11	update order_line1 set ol_delivery_d = now() where ol_o_id = ? and ...		①		0.18 load	25.34 QPS	7.26 ms

Query Time

---

Per query : 7.44 ms

---

Sum : 10 days, 0:59:57

---

From total : 100.00 %

---



---

Max : 32.19 sec

---

Avg : 7.44 ms

---

99% : 83.47 ms



**99 percentile does not translate in 99% users having great performance**



**If every user interaction has 10 database queries**



**User in average has 10 interactions**



**Roughly 50% of session will have query with p99 response time**

# Percentile

# Errors

- **Look at Response time of Successful Queries, do not let “fast errors” to screw up your data**
- **Measure response time of “slow errors” as it contributes to user experience**

# Over Time



LOOK AT RESPONSE TIME  
TRENDS OVER TIME



MINOR SLOWDOWN OFTEN  
HAPPENS BEFORE POOR  
PERFORMANCE  
“DOWNTIME”



PERFORMANCE CAN BE  
WORSE AT CERTAIN TIMES –  
BACKUPS, BATCH JOBS,  
MAINTENANCE

Search by...	Q	Query_Time	Query_Count	Lo
			7.44 ms	2.70k QPS
w_ytd = w_ytd + ? where w_id = ?	(i)		231.31 ms	25.52 QPS
data from item1 where i_id = ?	(i)		18.76 ms	255.92 QPS
where id=?	(i)		1.87 ms	1.00k QPS
x from district1 where d_w_id = ? and d_id ...	(i)		69.66 ms	25.61 QPS
d = d_ytd + ? where d_w_id = ? and d_id= ?	(i)		48.95 ms	25.52 QPS
l_o_id, ol_d_id, ol_w_id, ol_number, ol_i_id, ...	(i)		2.40 ms	255.67 QPS
ntity = ? where s_i_id = ? and s_w_id= ?	(i)		2.07 ms	255.67 QPS
i_id)) from order_line1, stock1 where ol_w_...	(i)		132.77 ms	2.56 QPS
	(i)		4.73 ms	58.52 QPS
(no_o_id, no_d_id, no_w_id) values(?+)	(i)		10.50 ms	25.61 QPS
l_delivery_d = now() where ol_o_id = ? and ...	(i)		7.26 ms	25.38 QPS

# Database or Application ?

- Instrument on Application side to understand what drives performance of user interactions

Instrument of Database Size to understand what causes queries to be slow and what can be done about it



# Response Time – Business View

**All Users have outstanding performance  
experience with all their application interactions**

# Enhancing Query Meta Data

**SQL Commenter project by Google**

<https://per.co.na/SQLcommenter>

Query Meta  
Data  
Possibilities

---

**Actual User/Tenant**

---

**Application/Functionality**

---

**Version Information (A/B Testing)**

---

**Responsible Engineer/Team**

# Query

**Different Queries have different performance profile**

**They also correspond to different “user actions”**

**And may have different acceptable level of Performance**

# Schema and Database

- Different Applications/Services may be using different ones
- In sharded environment can correspond to application "tenant"

#	Database	Search by...	Q	Load	Query Count	Query Time
	TOTAL				38.80 load	2.47k QPS 15.74 ms
1	tpcc1				36.89 load	449.93 QPS 81.98 ms
2	sbtest				1.28 load	1.56k QPS 815.27 µs
3	tpcc2				0.58 load	437.09 QPS 1.34 ms
4	postgres				0.06 load	11.01 QPS 5.13 ms
5	tpcc3				<0.01 load	1.00 QPS 12.31 µs
6	tpcc4				<0.01 load	0.99 QPS 11.17 µs
7	tpcc5				<0.01 load	1.06 QPS <9.80 µs

# Database view in Percona Monitoring and Management

# Table/ Collection

- Can help identify “problematic data”
- Indexing changes impact queries hitting object
- Maintenance often impacts specific table



**IDENTIFY  
SERVICE/APPLICATION**



**FIND HUMAN TROUBLE  
MAKERS WITH  
INTERACTIVE ACCESS**

**Database User**

[Copy Link](#) [+ Add column](#)

#	User Name <input type="text" value="User Name"/>	Search by... <input type="text"/>	Q	<u>Load</u>	<u>Query_Count</u>	<u>Query_Time</u>
	TOTAL			 38.83 load	2.47k QPS	15.73 ms
1	app1			 36.92 load	450.62 QPS	81.92 ms
2	app3			 1.28 load	1.57k QPS	815.02 μs
3	app2			 0.59 load	437.34 QPS	1.34 ms
4	pmm			 0.06 load	14.64 QPS	3.86 ms

## Database Host

---

**Sharded environments often have multiple hosts handling the same traffic**

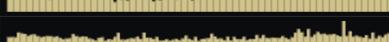
---

**Yet Problems often can be limited to some hosts**

---

**Data/Traffic Balance, configuration, invisible differences**

# Database Instances

#	Service Name	Search by...	Q	Load	Query Count	Query Time
	TOTAL			 38.93 load	2.47k QPS	15.74 ms
1	pg4-postgresql			 36.66 load	236.16 QPS	155.23 ms
2	pg2-postgresql			 1.90 load	999.10 QPS	1.90 ms
3	pg1-postgresql			 0.33 load	1.02k QPS	325.14 µs
4	pg3-postgresql			 0.04 load	221.80 QPS	161.17 µs

**App Server/  
Web Server/  
Service  
Instance**

---

**You may expect all instances of  
the same type causing same  
even load**

---

**It may not be the case**

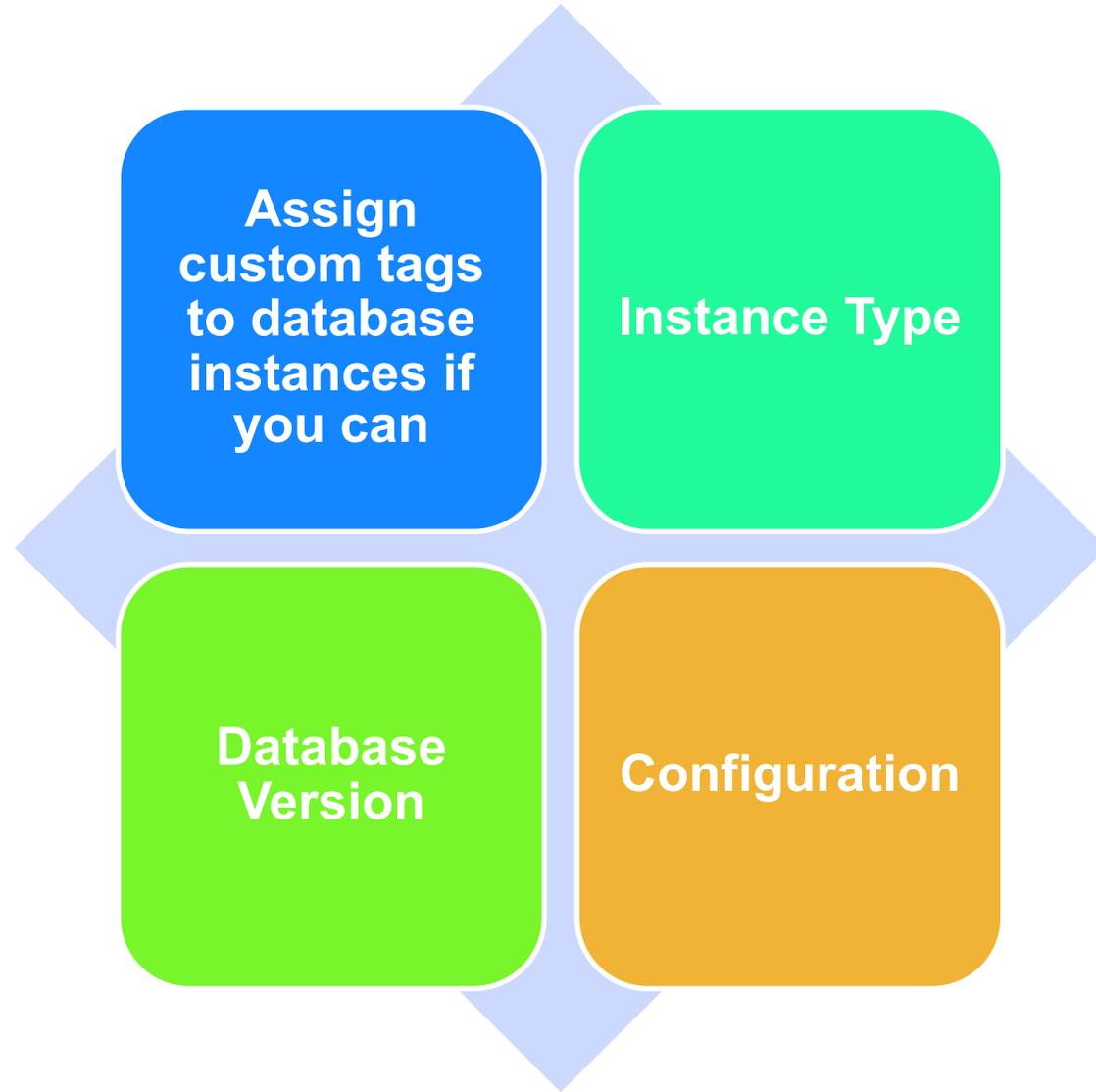
---

**Code versions, configuration,  
load balancer behavior,  
security incidents**

# Client Hosts

#	Client Host	Search by...	Q	Load	Query_Count	Query_Time
	TOTAL			 38.97 load	2.47k QPS	15.75 ms
1	139.144.169.65			 14.36 load	623.02 QPS	23.05 ms
2	139.144.169.80			 13.26 load	627.49 QPS	21.13 ms
3	139.144.169.84			 11.29 load	1.21k QPS	9.33 ms
4	127.0.0.1			 0.06 load	14.65 QPS	3.86 ms

# Custom Tags



# Query Plan

- One Query Can have Multiple Different Query Plans
- Sometimes it is good, in other cases it is a problem
- Measure Query Performance by Query Plan
- Can take action to correct query plan if this is the issue

# Where Response Time Comes From ?



**Data  
Crunching/CPU**



**Waits on CPU  
Availability**



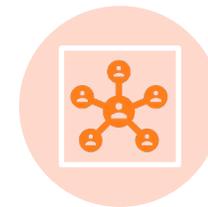
**Disk IO**



**Row Locks**



**Contention**



**Network**

# pg\_stat\_monitor

## Deep Query Performance Insights for PostgreSQL

[https://github.com/percona/pg\\_stat\\_monitor](https://github.com/percona/pg_stat_monitor)

[https://www.percona.com/blog/2020/10/14/announcing-pg\\_stat\\_monitor-tech-preview-get-better-insights-into-query-performance-in-postgresql/](https://www.percona.com/blog/2020/10/14/announcing-pg_stat_monitor-tech-preview-get-better-insights-into-query-performance-in-postgresql/)

# Why not improve pg\_stat\_statements

- Move fast, experiment new approaches to data capture
- Focus on data being constantly consumed by monitoring system (hence held on the instance for short term)
- Focusing on change over time
- pg\_stat\_statements view provided for compatibility in v 2.0

# Other Things to Consider





## “Bad Queries” vs Victims

- Query might be slow because it is heavy on its own
- Or it might be victim of other queries or their volume

**Do not  
forget  
currently  
running  
queries**

- **Response time is measured when query completes**
- **You can write queries which “never” complete**
- **Consider killing runaway queries and whitelisting queries which need to run long**

# Do not Ignore “Invisible”

- **Database Background Activities**
- **Maintenance Operations**
- **Cloud Noise**

## Avoid Biased Sampling

“Let’s Look only on slow queries”

Focus on Outliers

Likely to ignore queries causing most load, typical impact

**Good Luck**

**Get your query performance under control  
Do not over-do scaling by Credit Card**

Let's Connect!

<https://www.linkedin.com/in/peterzaitsev/>

<https://twitter.com/PeterZaitsev>

<http://www.peterzaitsev.com>



**THANK YOU!**

[percona.com](http://percona.com)