

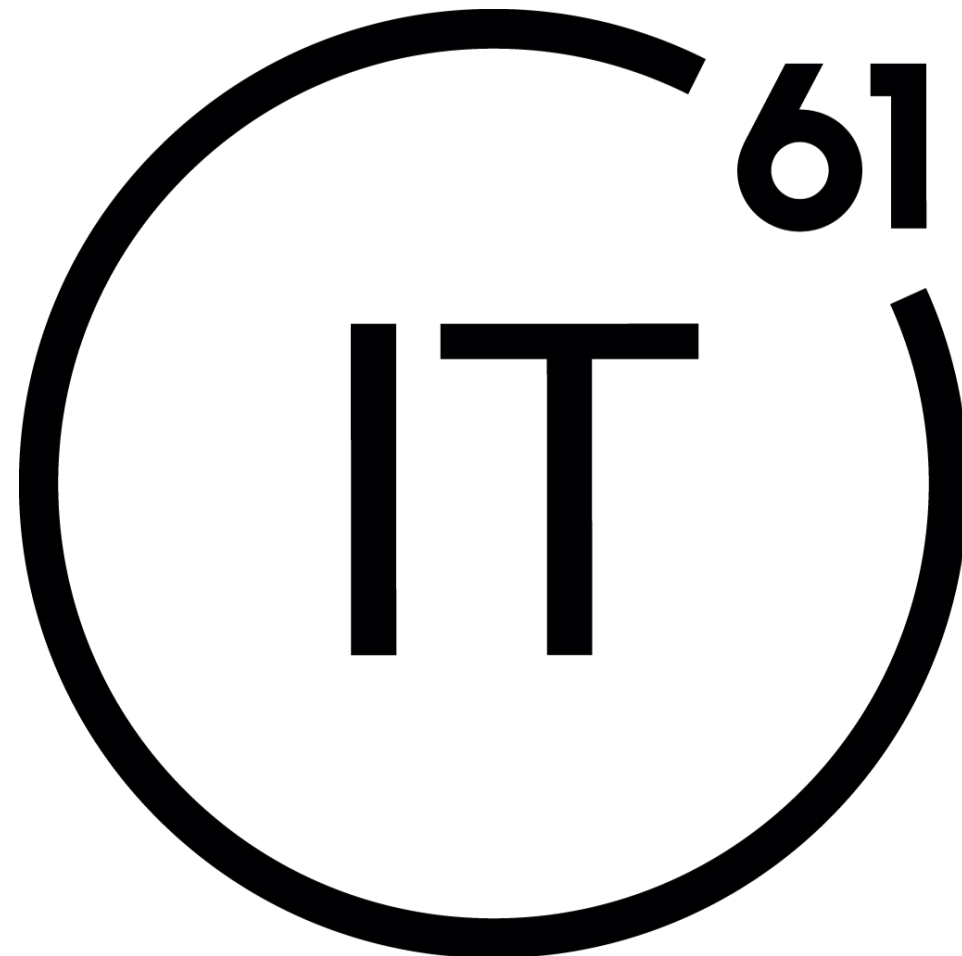
10 Things Developers should know about Databases

Peter Zaitsev
CEO, Percona

27 June 2019



Thank you



Who Are you ?

**More
Developer ?**

More OPS ?

Ops

Focused on Database Only

Generalist

Programming Language

**What Programming
Languages does your
team use ?**

Devs vs Ops

DevOps suppose to have solved it but tension is still common between Devs and Ops

Especially with Databases which are often special snowflake

Especially with larger organizations

Large Organizations

**Ops vs Ops have conflict
too**

Devs vs Ops Conflict

Devs

- Why is this stupid database always the problem.
- Why can't it just work and work fast

Ops

- Why do not learn schema design
- Why do not you write optimized queries
- Why do not you think about capacity planning

Database Responsibility

**Shared Responsibility for
Ultimate Success**

Top Recommendations for Developers

Learn Database Basics

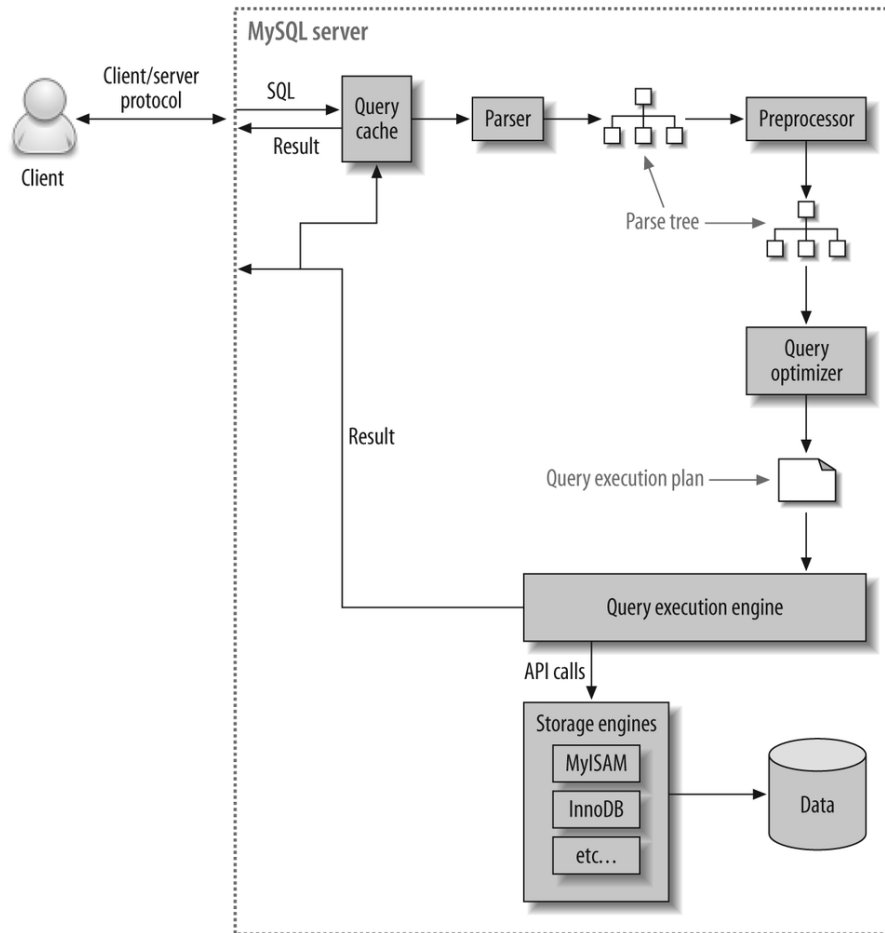
You can't build great database powered applications if you do not understand how databases work

Schema Design

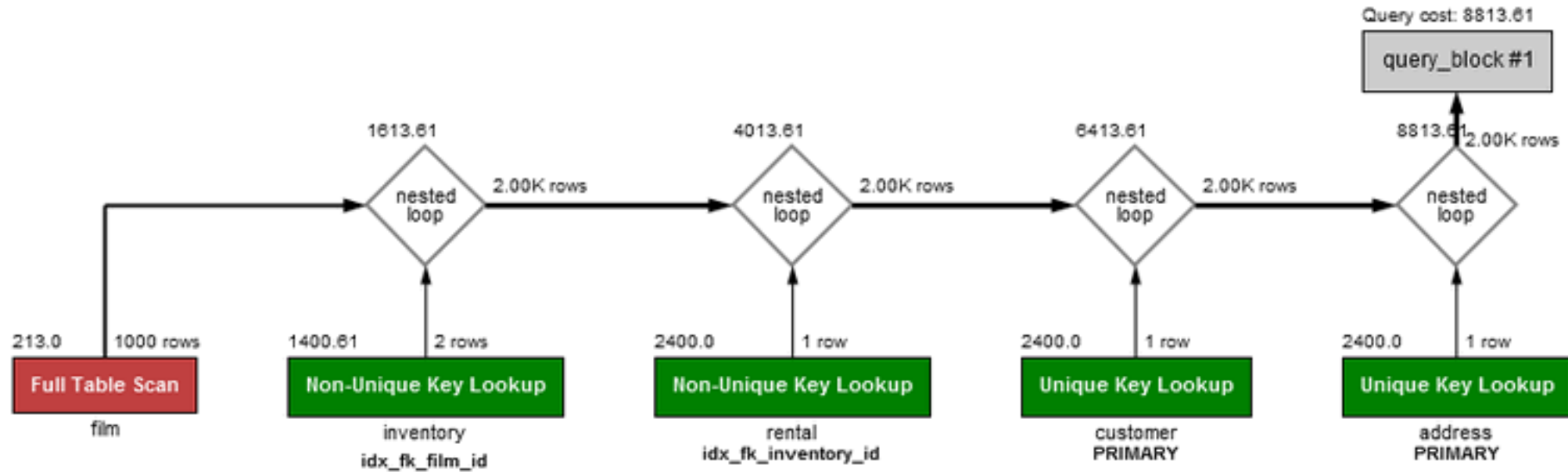
Power of the Database Language

How Database Executes the Query

Query Execution Diagram



EXPLAIN



How are Queries Executed ?

Single Threaded

Single Node

Distributed

Indexes

**Indexes are
Must**

**Indexes are
Expensive**

Capacity Planning

No Database can handle “unlimited scale”

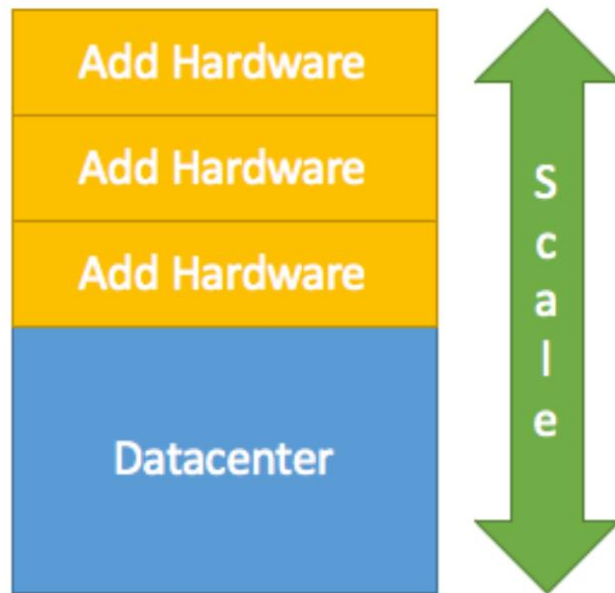
Scalability is very application dependent

Trust Measurements more than Promises

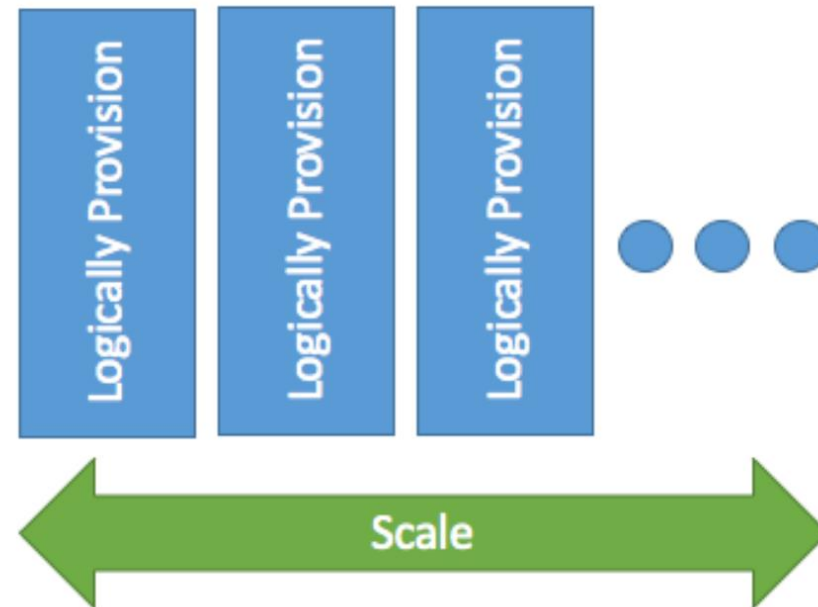
Can be done or can be done Efficiently ?

Vertical and Horizontal Scaling

Vertical Scaling



Horizontal Scaling



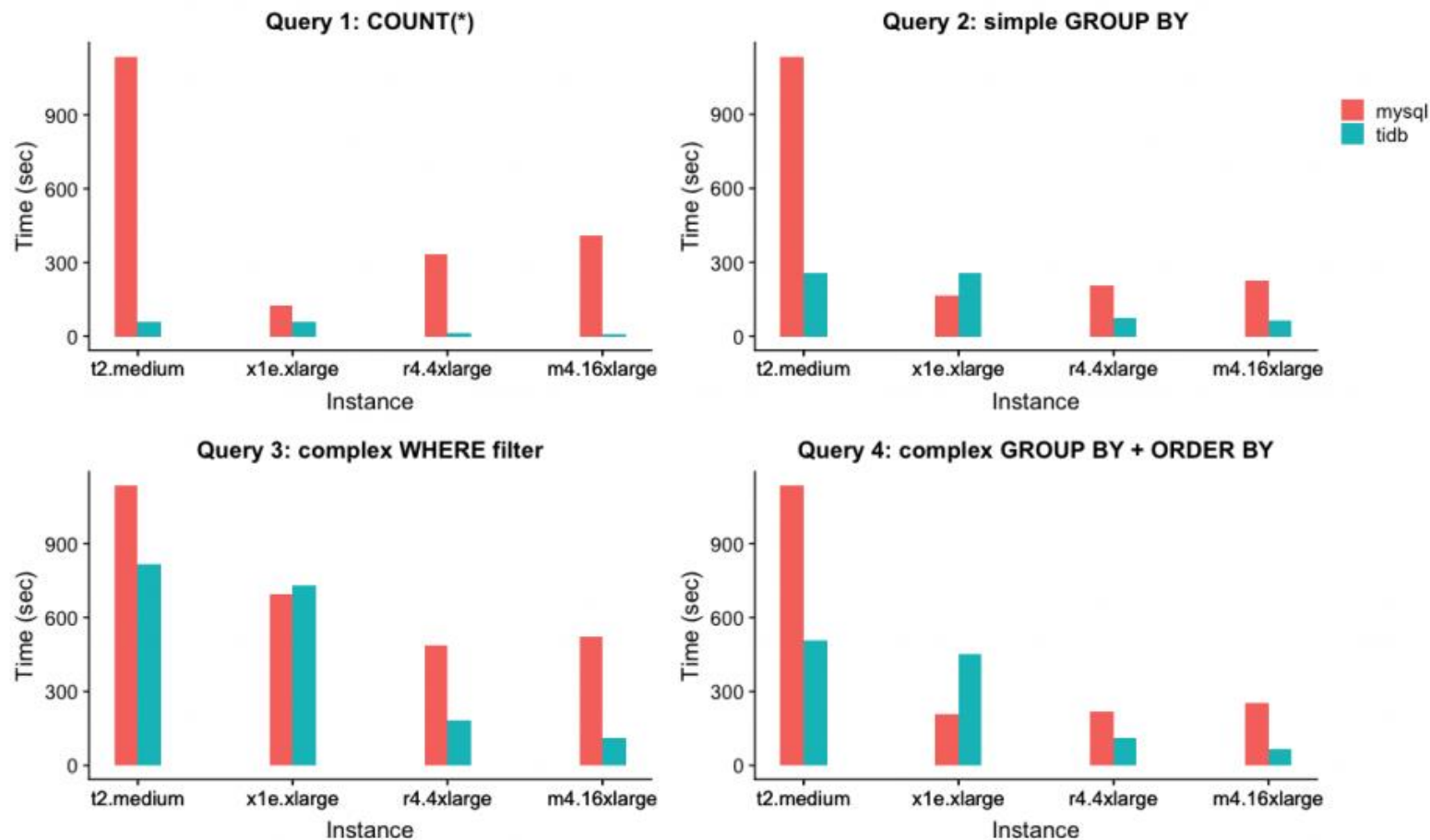
Scalable != Efficient

The Systems which promote a scalable can be less efficient

Hadoop, Cassandra, TiDB are great examples

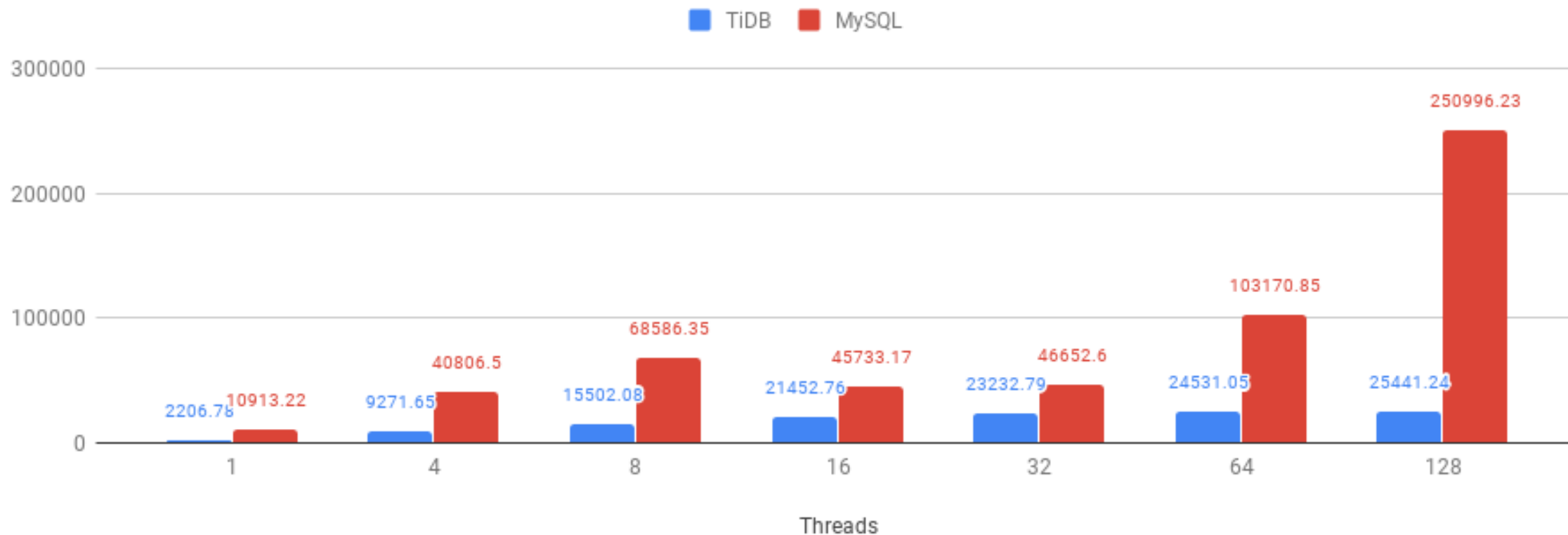
By only the wrong thing you can get in trouble

TiDB Scalability (Single Node)



TiDB Efficiency

TiDB and MySQL - point selects - sysbench



Throughput != Latency

If I tell you system can do
100.000 queries/sec
would you say it is fast ?

Speed of Light Limitations

High Availability Design Choices

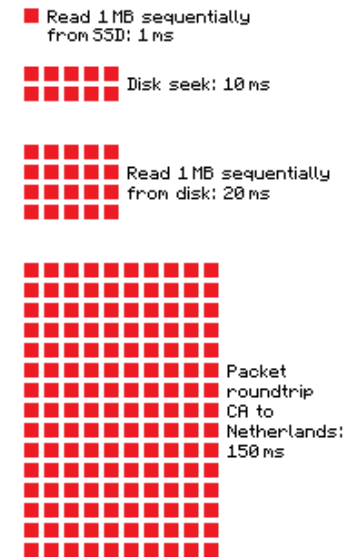
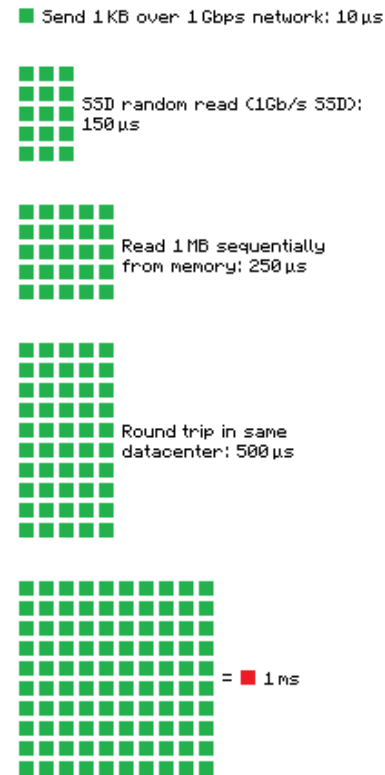
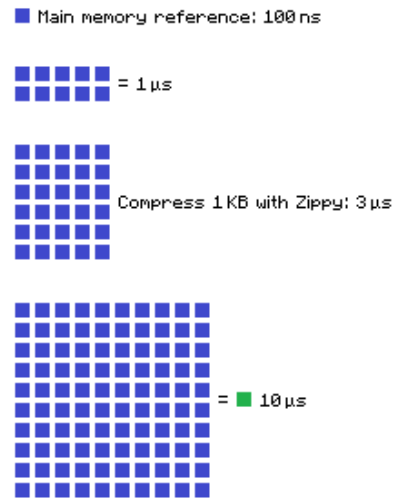
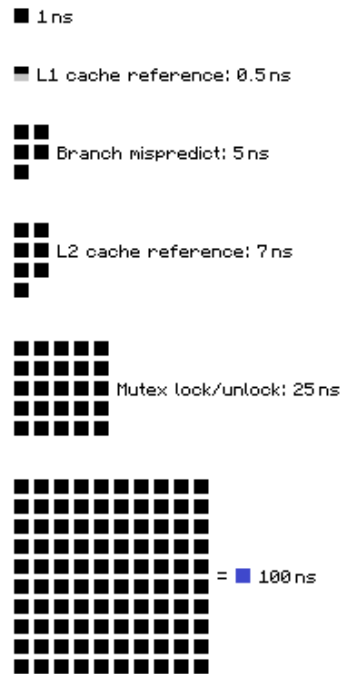
You want instant durable replication over wide geography or Performance ?

Understanding Difference between High Availability and Disaster Recovery protocols

Network Bandwidth is not the same as Latency

Mind Network Latency

Latency Numbers Every Programmer Should Know



Source: <https://gist.github.com/2841832>

Also Understand

Connections to the database are expensive

Especially if doing TLS Handshake

Query Latency Tends to Add Up

Especially on real network and not your laptop

Law of Gravity

**Shitty Application at
scale will bring down any
Database**

Scale Matters

Developing and Testing with Toy Database is risky

Queries Do not slow down linearly

The slowest query may slow down most rapidly

Memory or Disk

Data Accessed in memory is much faster than on disk

It is true even with modern SSDs

SSD accesses data in large blocks, memory does not

Fitting data in Working Set

Newer is not Always Faster

Upgrading to the
new
Software/Hardware
is not always faster

Test it out

Defaults Change are
often to blame

Upgrades are needed but not seamless

Major Database Upgrades often require application changes

Having Conversation on Application Lifecycle is a key

Character Sets

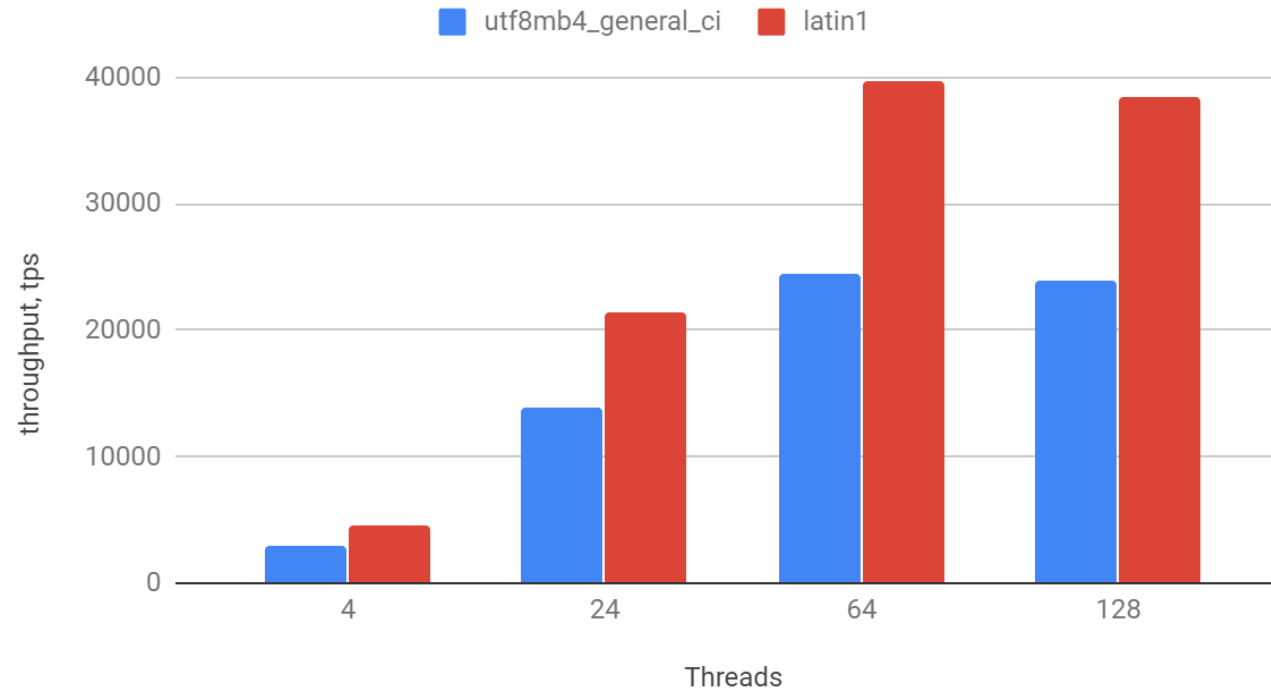
Performance Impact

Pain to Change

Wrong Character Set can cause Data Loss

Character Sets

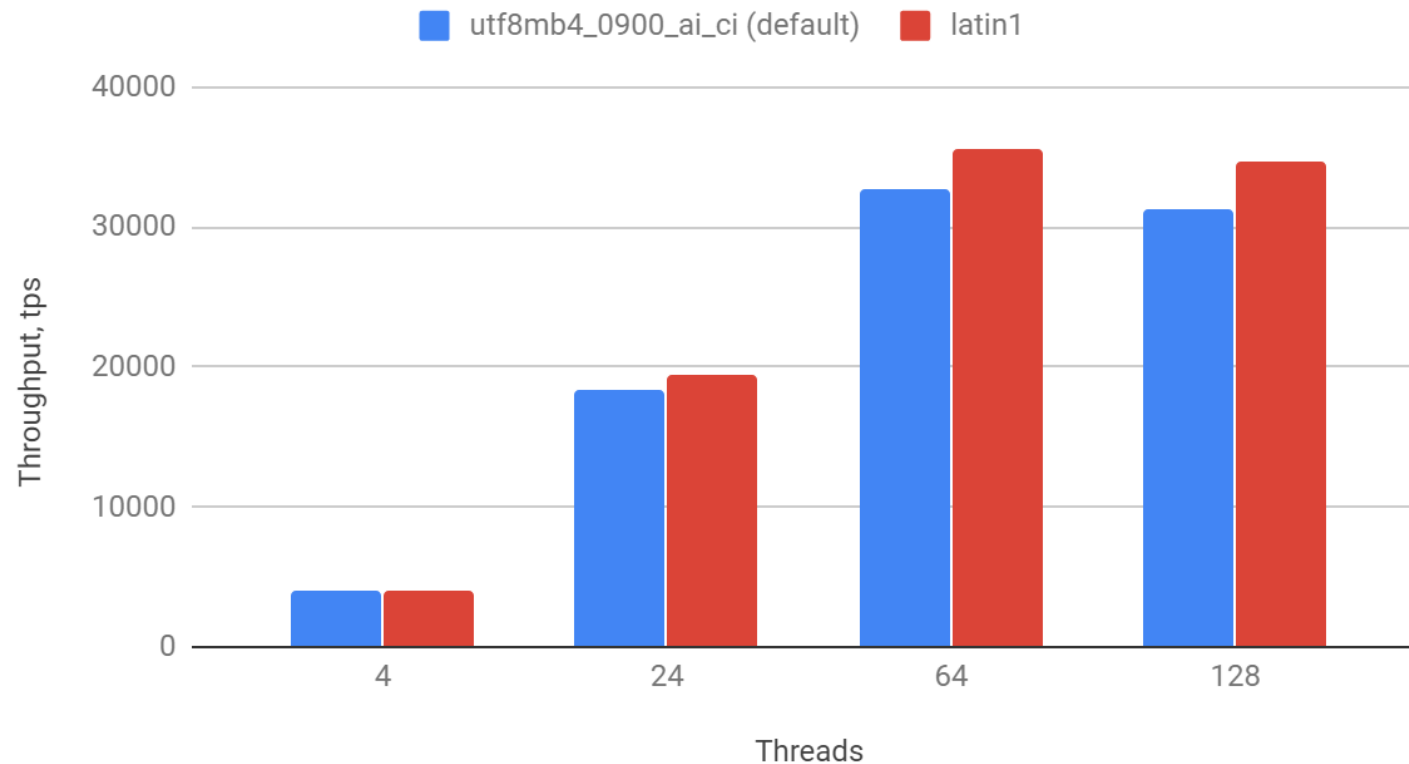
MySQL 5.7 utf8mb4_general_ci (default) and latin1



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

Less impact In MySQL 8

MySQL 8.0 utf8mb4_0900_ai_ci and latin1



Operational Overhead

Operations Take Time, Cost Money, Cause Overhead

10TB Database Backup ?

Adding The Index to Large Table ?

Distributed Systems

10x+ More Complicated

Better High Availability

Many Failure Scenarios

Test how application performs

Risks of Automation

**Automation is
Must**

**Mistakes can
destroy
database at scale**

Security

Database is where the most sensitive data tends to live

Shared Devs and Ops Responsibility

Beyond Technical Considerations

DB-Engines Ranking

347 systems in ranking, May 2019

Rank			DBMS	Database Model	Score		
May 2019	Apr 2019	May 2018			May 2019	Apr 2019	May 2018
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1285.55	+5.61	-4.87
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1218.96	+3.82	-4.38
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	1072.19	+12.23	-13.66
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	478.89	+0.17	+77.99
5.	5.	5.	MongoDB +	Document	408.07	+6.10	+65.96

Open Source

**Truly Open
Source**

**Kind of
Open Source**

**Open Source
Compatible**

Truly Open Source

OSI: <https://opensource.org/osd-annotated>

GNU: <https://www.gnu.org/philosophy/free-sw.en.html>

DEBIAN:
https://www.debian.org/social_contract#guidelines

Peter's Practical Question

What will I have to give up if I stop commercial relationship with a vendor ?

Kind of Open Source Software

Everyone loves Open Source, why do not we use it for Marketing ?

Kind of Open Source Software

Open Core

Shared Source Software

Eventually Open Source Software

Open Source Compatible Software

**Honest Proprietary
Software claiming
compatibility with Open
Source Technology**

“Hotel California Compatibility”



Benefits of DBaaS

Reduce Cost at Small and Medium Scale

Increase Agility via Self Service

Reduce Risks through providing sensible boundaries

Emerging Open Source Response

Kubernetes as Operating System for your Data Center

Operators as a way to run Autonomous Databases

Percona Operators for MySQL and MongoDB are Now Available

Types of Open Source Licenses

Copyleft

- GPL, AGPL

Permissive

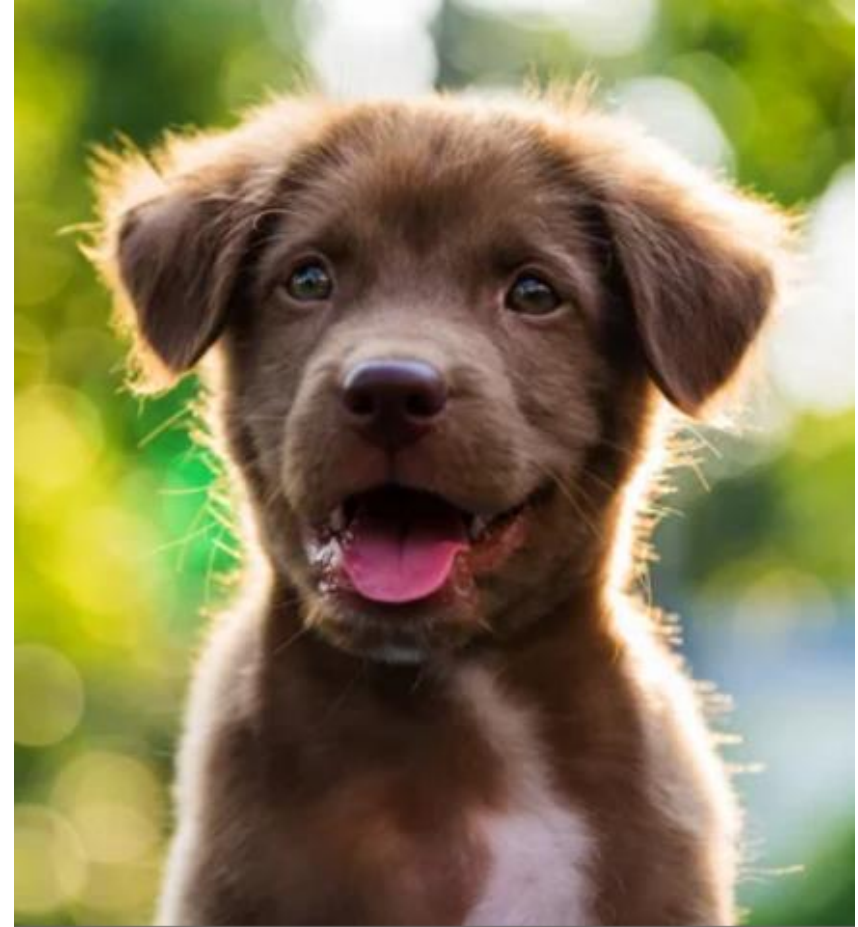
- BSD, Apache, MIT

Public Domain

- Not Really a License

“Free” in “Free Software”

Free as in
Puppy



What Else

What Would you Add ?

Check Out <http://per.co.na/careers>



**BECOME A
PERCONA SUPERHERO**

WE'RE HIRING

**CONTACT
careers@percona.com**

Percona Live Europe 2019

Open Source Database Focused Conference

Takes place in Amsterdam Sep 30 – Oct 2

Call for Papers is now Open

<https://per.co.na/ple19>

Thank You!

Twitter: @percona @peterzaitsev
