17 Things Developers should know about Databases

Peter Zaitsev
CEO, Percona
MySQL Belgian Days

January 30, 2025

30 preFOSDEM
MySQL™ Belgian Days 2025

PERCONA

Devs vs Ops Conflict

Devs

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

Ops

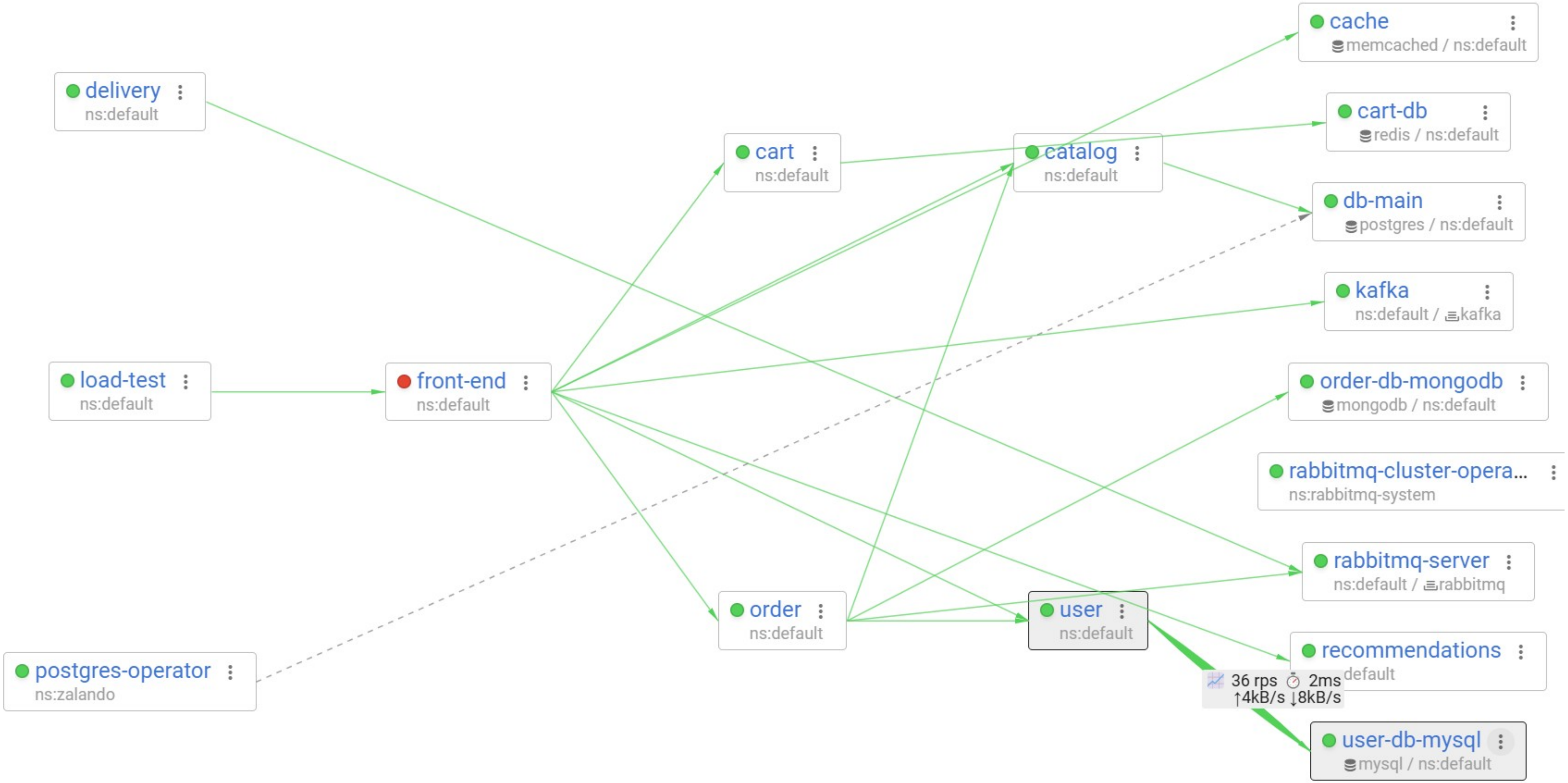
- Why do not learn schema design
- Why do not you write optimized queries
- Why do not you think about capacity planning

Database Responsibility

Shared Responsibility for Ultimate Success

Top Recommendations for Developers

Is it even MySQL ?



Learn Database Basics

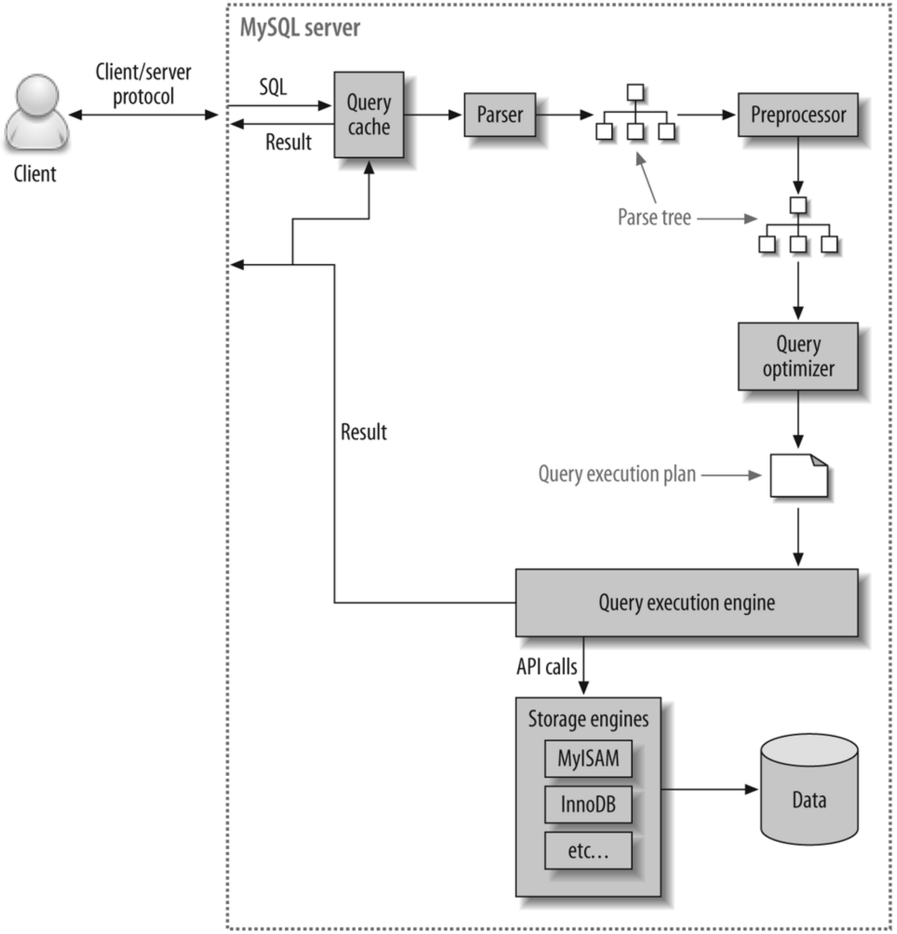
You can't build great database powered applications if you do not understand how databases work

Schema Design

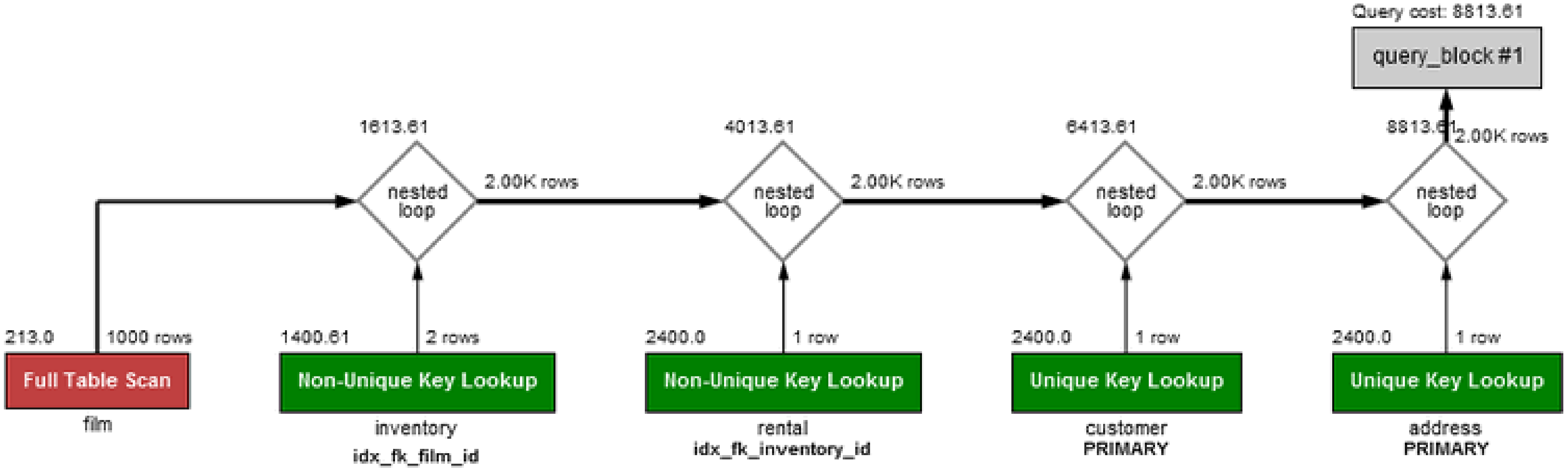
Power of the Database Language

How Database Executes the Query

Query Execution Diagram

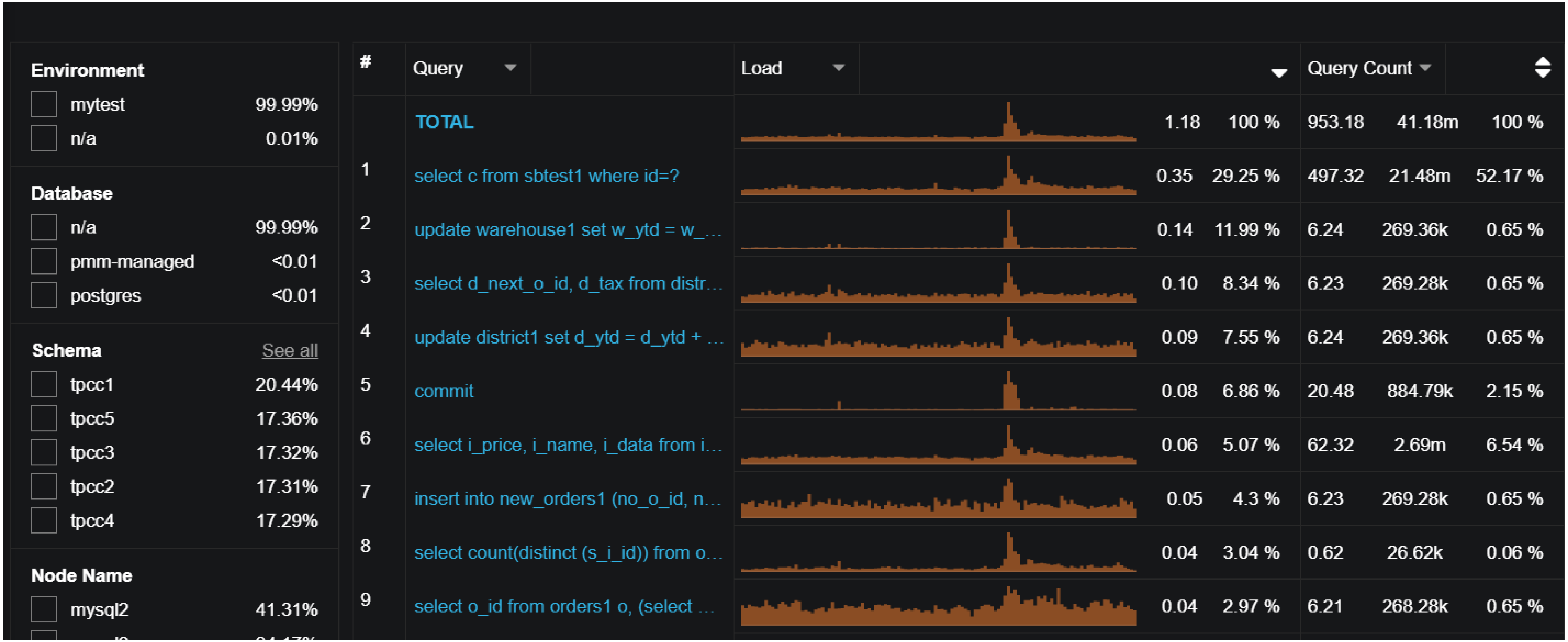


EXPLAIN



https://dev.mysql.com/doc/refman/8.0/en/execution-plan-information.html

Which Queries are Causing the Load



Environment

□	mytest	99.99%
□	n/a	0.01%

Database

□	n/a	99.99%
□	pmm-managed	<0.01
□	postgres	<0.01

Schema See all

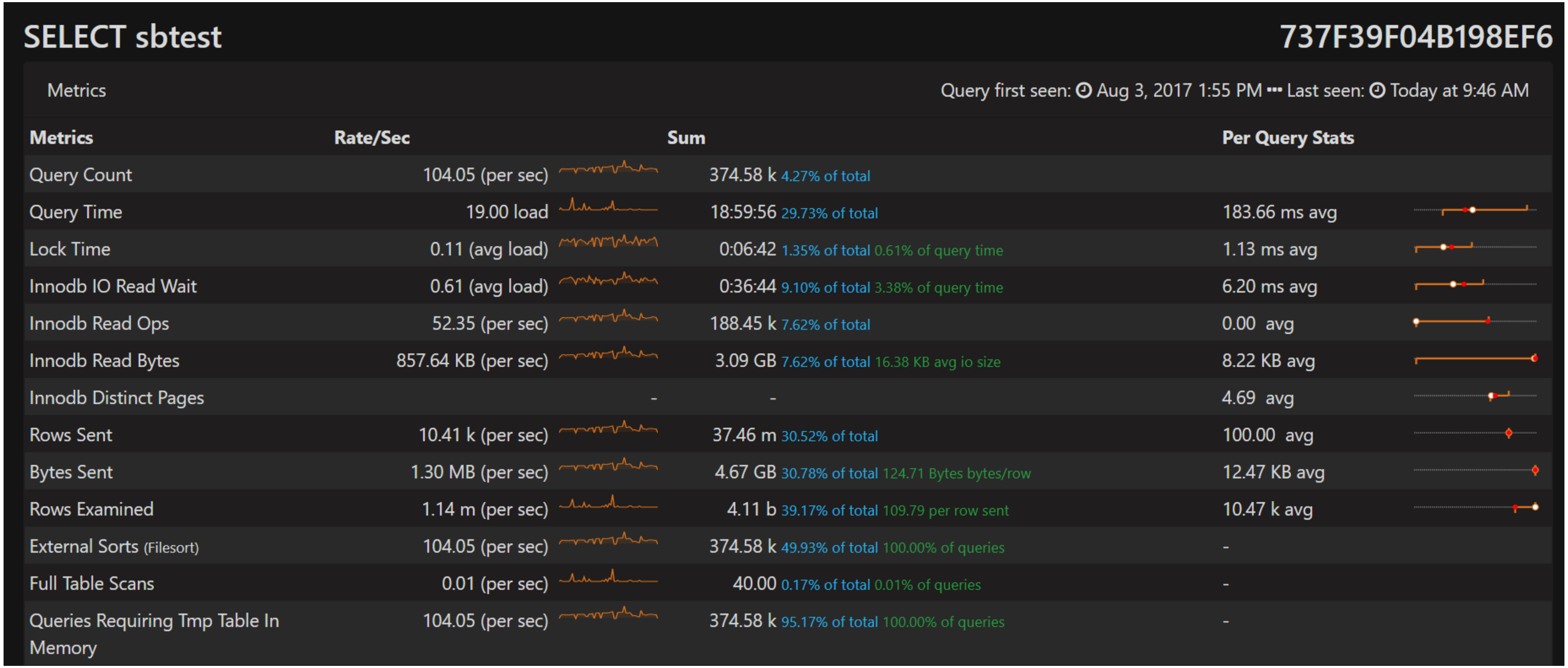
□	tpcc1	20.44%
□	tpcc5	17.36%
□	tpcc3	17.32%
□	tpcc2	17.31%
□	tpcc4	17.29%

Node Name

□	mysql2	41.31%

#	Query	Load			Query Count		
	TOTAL		1.18	100 %	953.18	41.18m	100 %
1	select c from sbtest1 where id=?		0.35	29.25 %	497.32	21.48m	52.17 %
2	update warehouse1 set w_ytd = w_…		0.14	11.99 %	6.24	269.36k	0.65 %
3	select d_next_o_id, d_tax from distr…		0.10	8.34 %	6.23	269.28k	0.65 %
4	update district1 set d_ytd = d_ytd + …		0.09	7.55 %	6.24	269.36k	0.65 %
5	commit		0.08	6.86 %	20.48	884.79k	2.15 %
6	select i_price, i_name, i_data from i…		0.06	5.07 %	62.32	2.69m	6.54 %
7	insert into new_orders1 (no_o_id, n…		0.05	4.3 %	6.23	269.28k	0.65 %
8	select count(distinct (s_i_id)) from o…		0.04	3.04 %	0.62	26.62k	0.06 %
9	select o_id from orders1 o, (select …		0.04	2.97 %	6.21	268.28k	0.65 %

Why Are they Causing this Load

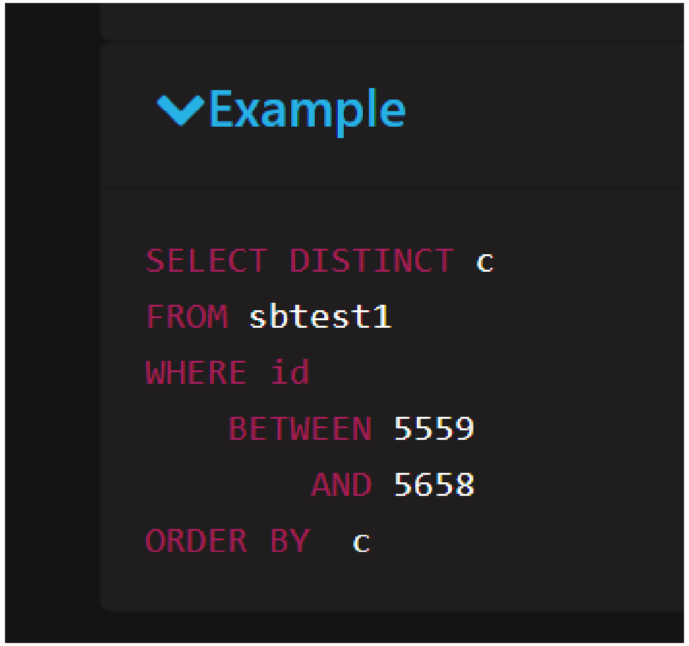


SELECT sbtest 737F39F04B198EF6

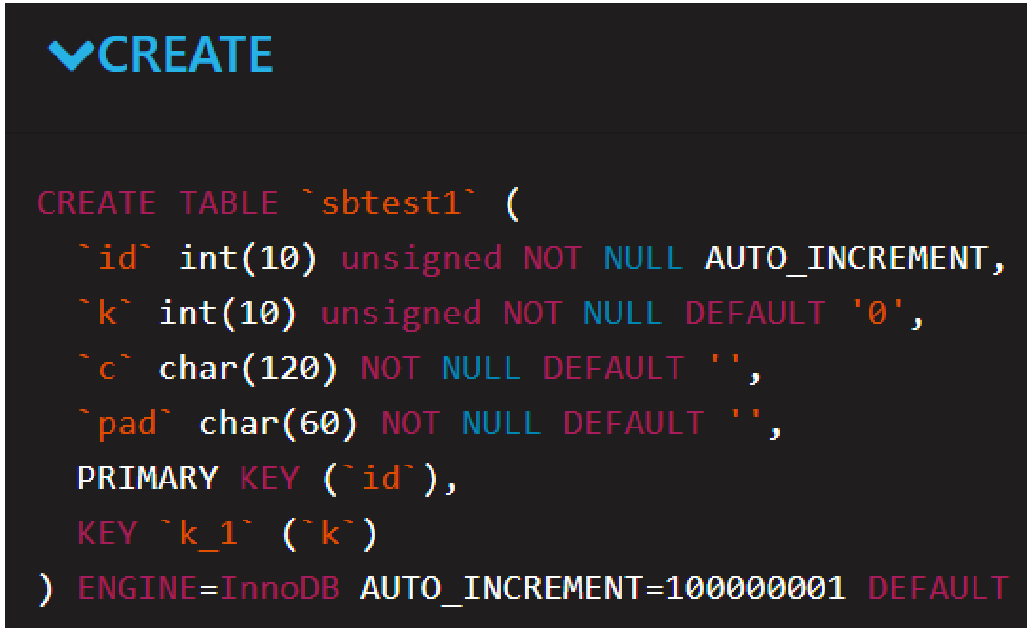
Metrics Query first seen: Aug 3, 2017 1:55 PM ••• Last seen: Today at 9:46 AM

Metrics	Rate/Sec	Sum	Per Query Stats
Query Count	104.05 (per sec)	374.58 k 4.27% of total	
Query Time	19.00 load	18:59:56 29.73% of total	183.66 ms avg
Lock Time	0.11 (avg load)	0:06:42 1.35% of total 0.61% of query time	1.13 ms avg
Innodb IO Read Wait	0.61 (avg load)	0:36:44 9.10% of total 3.38% of query time	6.20 ms avg
Innodb Read Ops	52.35 (per sec)	188.45 k 7.62% of total	0.00 avg
Innodb Read Bytes	857.64 KB (per sec)	3.09 GB 7.62% of total 16.38 KB avg io size	8.22 KB avg
Innodb Distinct Pages	-	-	4.69 avg
Rows Sent	10.41 k (per sec)	37.46 m 30.52% of total	100.00 avg
Bytes Sent	1.30 MB (per sec)	4.67 GB 30.78% of total 124.71 Bytes bytes/row	12.47 KB avg
Rows Examined	1.14 m (per sec)	4.11 b 39.17% of total 109.79 per row sent	10.47 k avg
External Sorts (Filesort)	104.05 (per sec)	374.58 k 49.93% of total 100.00% of queries	-
Full Table Scans	0.01 (per sec)	40.00 0.17% of total 0.01% of queries	-
Queries Requiring Tmp Table In Memory	104.05 (per sec)	374.58 k 95.17% of total 100.00% of queries	-

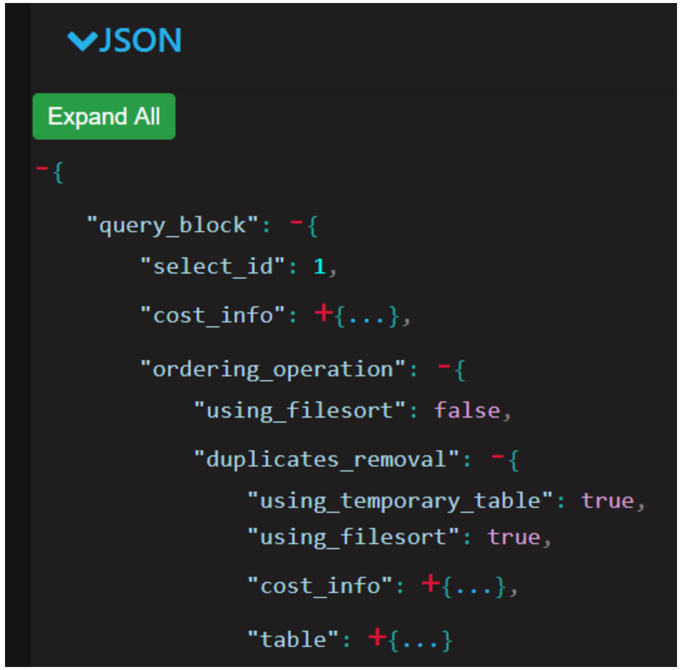
How to Improve their Performance



```
SELECT DISTINCT c
FROM sbtest1
WHERE id
    BETWEEN 5559
        AND 5658
ORDER BY  c
```

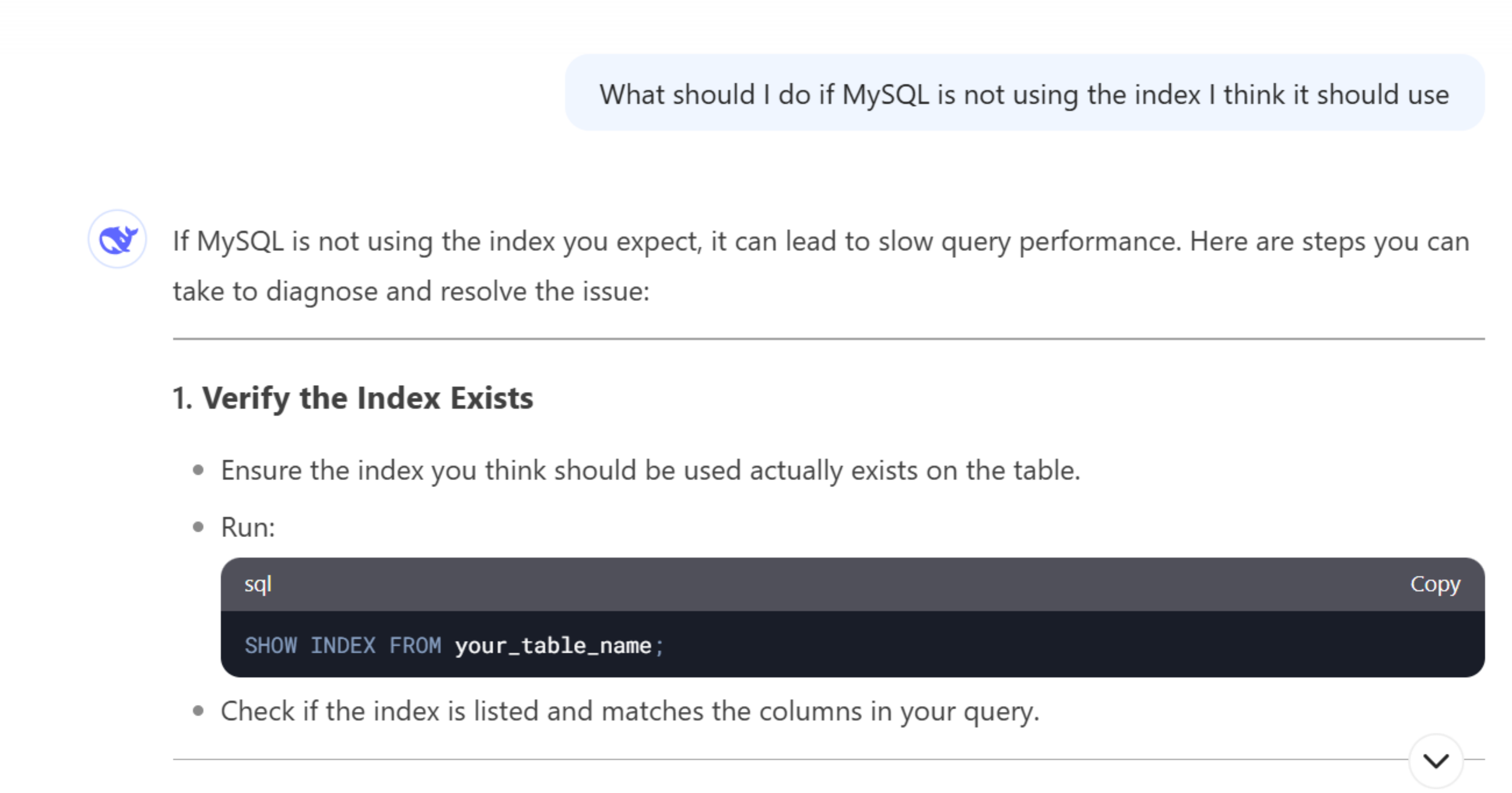


```
CREATE TABLE `sbtest1` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `k` int(10) unsigned NOT NULL DEFAULT '0',
  `c` char(120) NOT NULL DEFAULT '',
  `pad` char(60) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`),
  KEY `k_1` (`k`)
) ENGINE=InnoDB AUTO_INCREMENT=100000001 DEFAULT
```



```
-{
   "query_block": -{
      "select_id": 1,
      "cost_info": +{...},
      "ordering_operation": -{
         "using_filesort": false,
         "duplicates_removal": -{
            "using_temporary_table": true,
            "using_filesort": true,
            "cost_info": +{...},
            "table": +{...}
```

Get proficient talking to AI



What should I do if MySQL is not using the index I think it should use

If MySQL is not using the index you expect, it can lead to slow query performance. Here are steps you can take to diagnose and resolve the issue:

1. Verify the Index Exists

- Ensure the index you think should be used actually exists on the table.
- Run:

```sql
SHOW INDEX FROM your_table_name;
```

- Check if the index is listed and matches the columns in your query.

How are Queries Executed ?

Single Threaded

Single Node

Distributed

Indexes

Indexes are Must

Indexes are Expensive

Capacity Planning

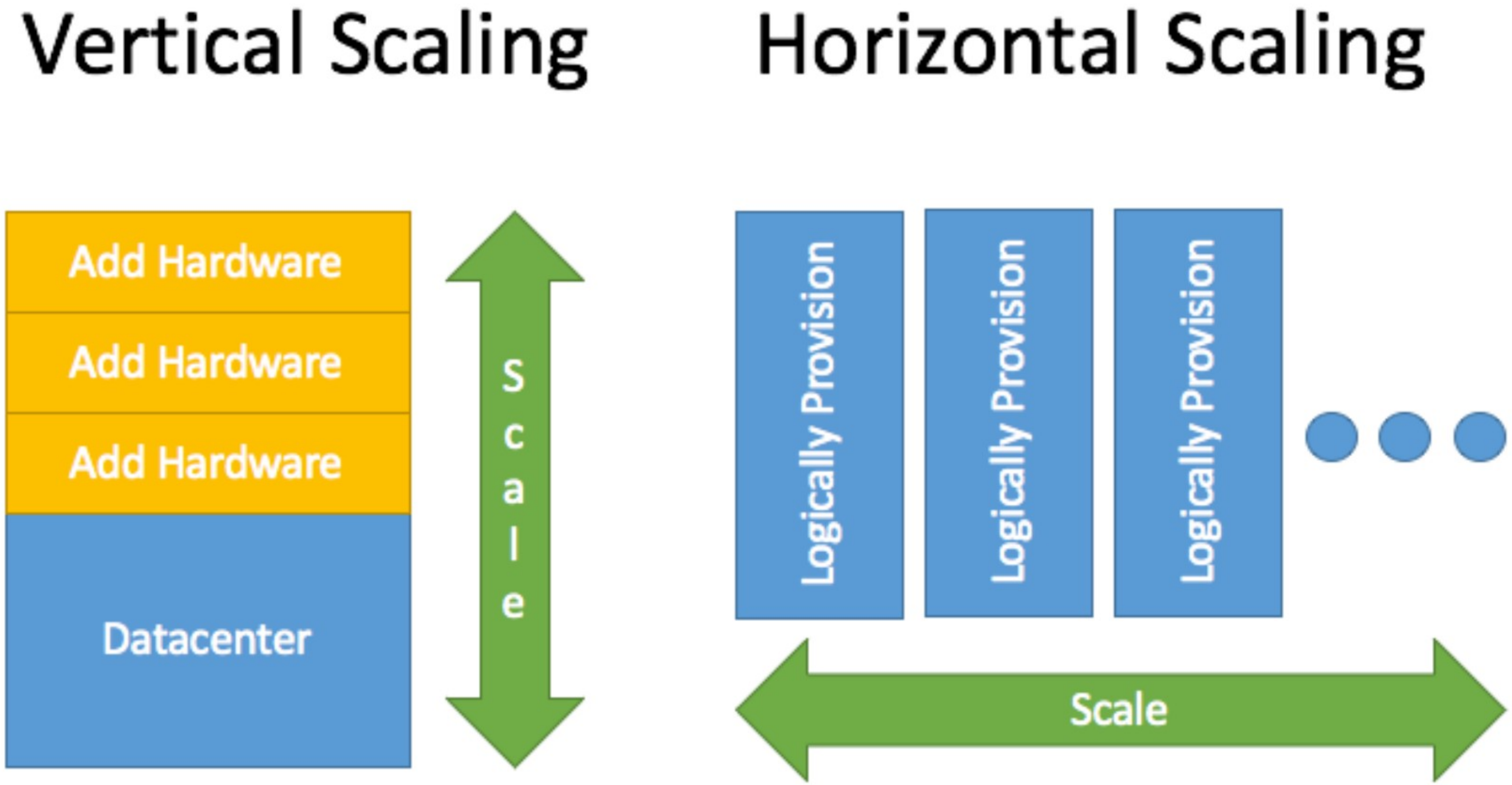
No Database can handle "unlimited scale"

Scalability is very application dependent

Trust Measurements more than Promises

Can be done or can be done Efficiently ?

Vertical and Horizontal Scaling



Scalable != Efficient

Scalable systems can be less efficient

Hadoop, Cassandra, TiDB, Vitess are great examples

Do not choose Scalability if what you need Is Performance

Throughput != Latency

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

PERCONA

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

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

PERCONA

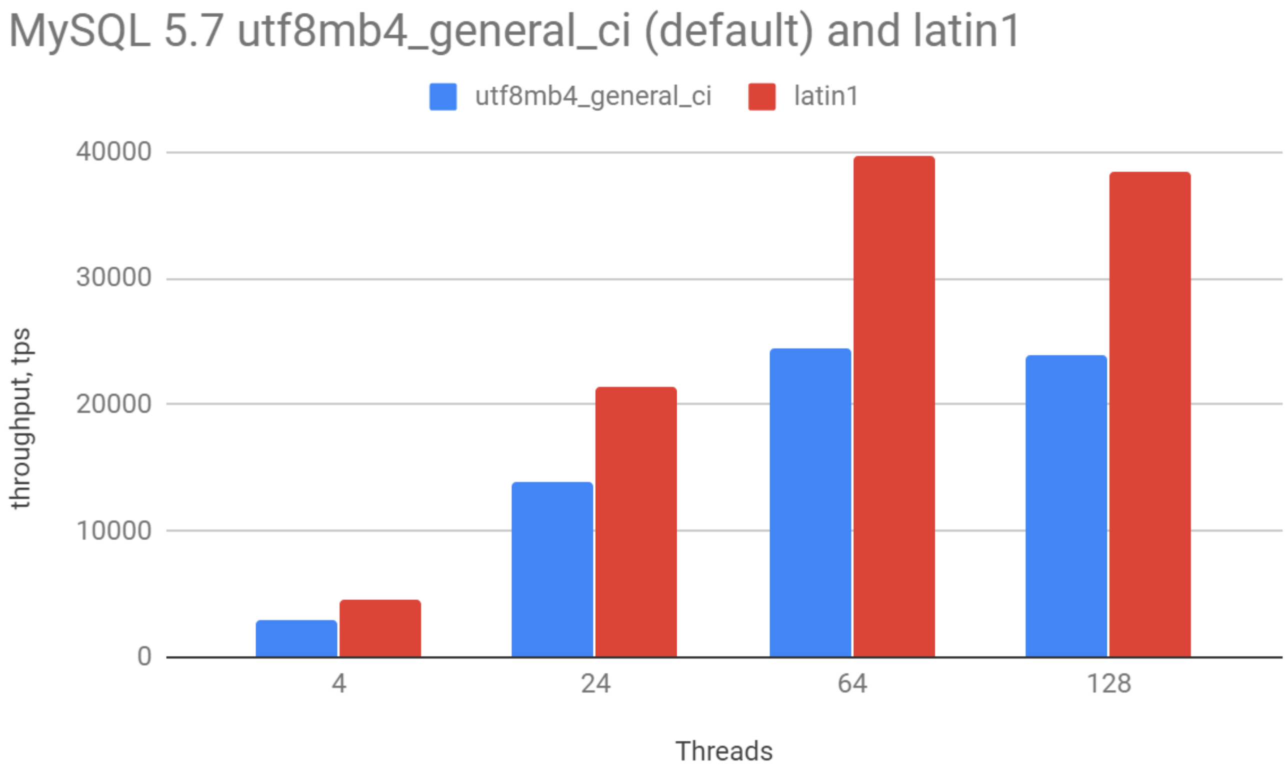
Character Sets

Performance Impact

Pain to Change

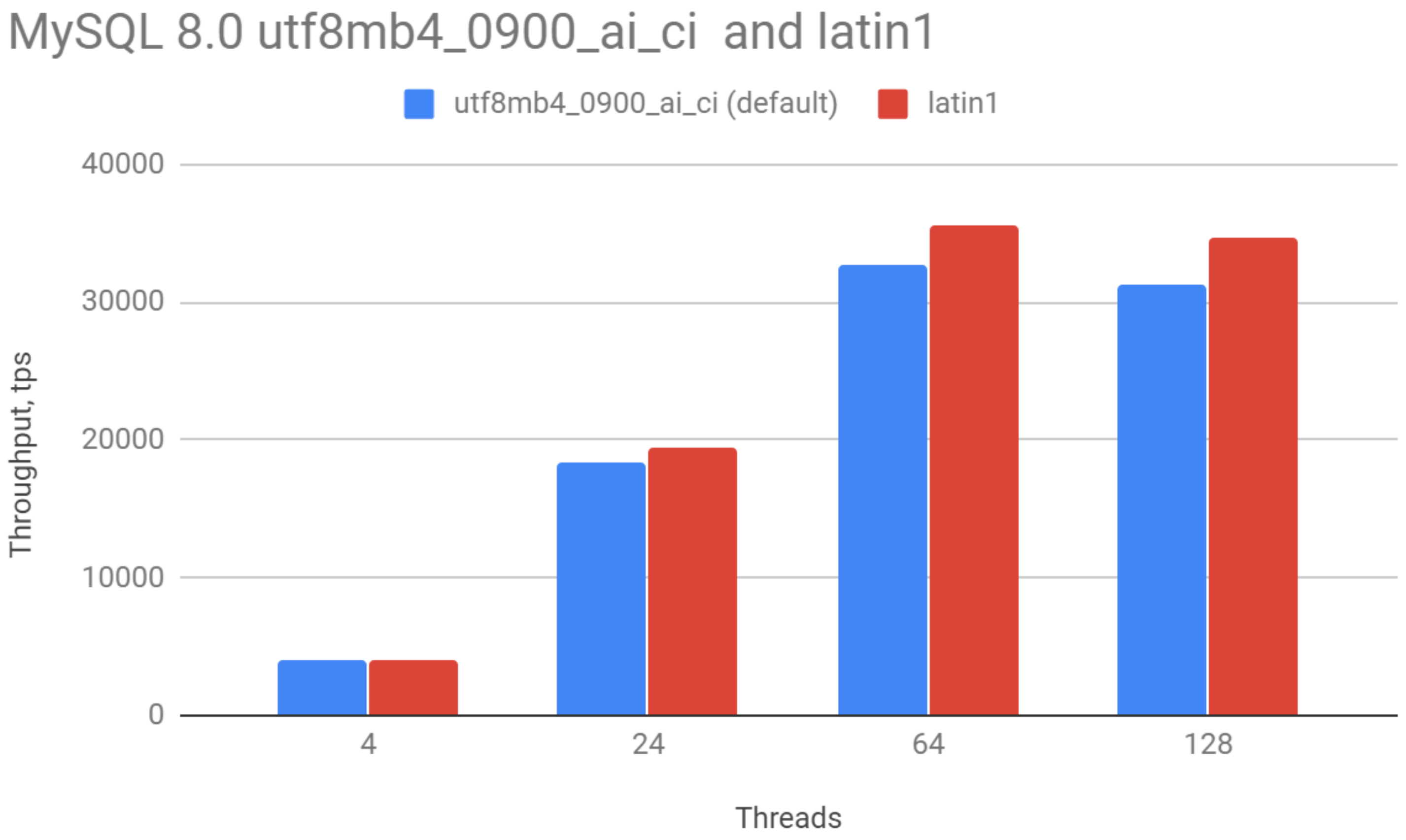
Wrong Character Set can cause Data Loss

Character Sets



https://per.co.na/MySQLCharsetImpact

Less impact In MySQL 8



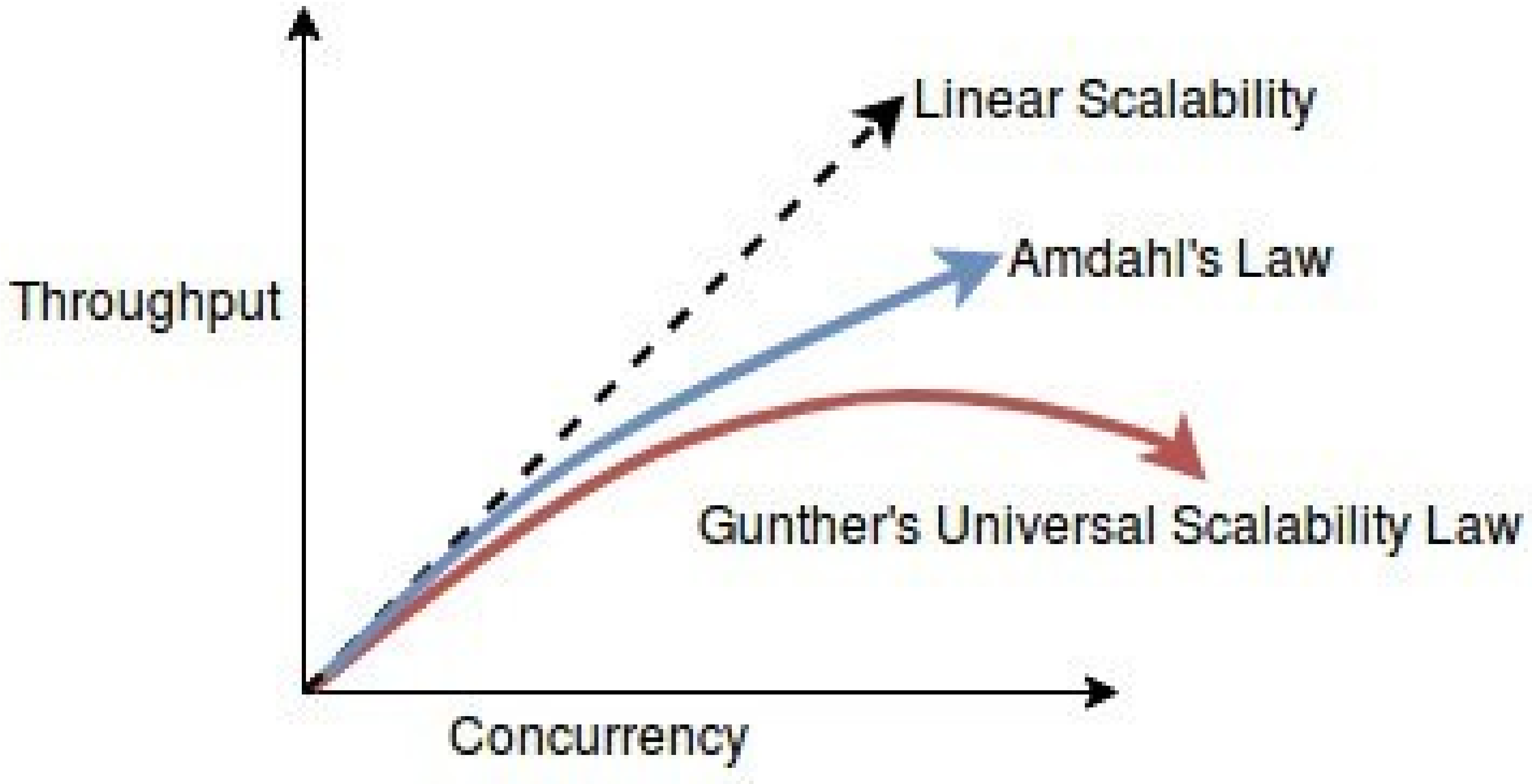
Do not Leave Transactions Open

- Open Connection is very inexpensive
- Transaction open for Long Time can get very expensive
- SET AUTOCOMMIT=0 - Any SELECT query will Open Transaction
- COMMIT/ROLLBACK closes connection

ORM (Object-Relational-Mapping)

- Allows Developers to query the database without need to understand SQL
- Can create SQL which is very inefficient
- Learn SQL Generation “Hints”, Learn   JPQL/HQL advanced features
- Be ready to manually write SQL if there is no other choice

Understanding Optimal Concurrency



Queueing

Request Queueing is Normal

With requests coming at “Random Arrivals” some queueing will happen with any system scale

Should not happen to often or for very long

Queueing is “Cheaper” on Higher Level

Benefits of Connection Pooling

1

Avoiding Connection Overhead, especially TLS/SSL

2

Avoiding using Excessive Number of Database Connections

3

Multiplexing/Load Management

Configuring Connection Pool

- Default and Maximum Connection Pool Size
- Scaling Parameters
- Combined Connection Pool Max Size should be smaller than number of connections database can support
- Waiting for free connection to become available is OK

Operational Overhead

Operations Take Time, Cost Money, Cause Overhead

10TB Database Backup ?

Adding The Index to Large Table ?

PERCONA

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

What Else

What Would you Add ?

“What is the next thing you’re climbing ?”

https://geeksgopeaks.com

Thank You!

www.linkedin.com/in/peterzaitsev/