MySQL Performance Tuning

DOAG K+A 2024, Nürnberg

Oli Sennhauser

CTO, FromDual GmbH

https://www.fromdual.com/presentations

Über FromDual GmbH

www.fromdual.com

Support

Beratung

remote-DBA

Schulung

Inhalt

MySQL Performance Tuning

- Hardware: RAM, I/O, CPU
- Betriebssystem (O/S)
- Datenbank Server Variablen
- Query Tuning
- Architektur & Design
- Was heisst Performance?
- Performance Degradation

Hardware – RAM

- **Viel RAM hilft viel!!! → Haupts. beim Lesen!**

```
Latency Comparison Numbers (2020)

L1 CPU cache reference                           1 ns -
L2 CPU cache reference                           4 ns -
RAM reference                                  100 ns -
Read 1 MB sequential read from RAM           3'000 ns -      3 µs
SSD random read                             16'000 ns -     16 µs
Read 1 MB sequentially from SSD             49'000 ns -     49 µs
Read 1 MB sequential read from HDD         825'000 ns -    825 µs
Disk seek (HDD)                          2'000'000 ns - 2'000 µs - 2 ms
```

- **Wie viel RAM?**
  1. **Antwort: So viel wie DB gross ist!**
  2. **Antwort: So viel wie heisses Datenset gross ist...**
- **Quelle:** https://colin-scott.github.io/personal_website/research/interactive_latency.html

Hardware – I/O

- HDD → SSD (NVMe)
- direkt verbaut, KEIN SAN (SAN sucks!)
- RAID 10, kein RAID 5/6 (short reads/writes)
- Messt mal Eure I/O Latenz! (`iostat -xk 1`)

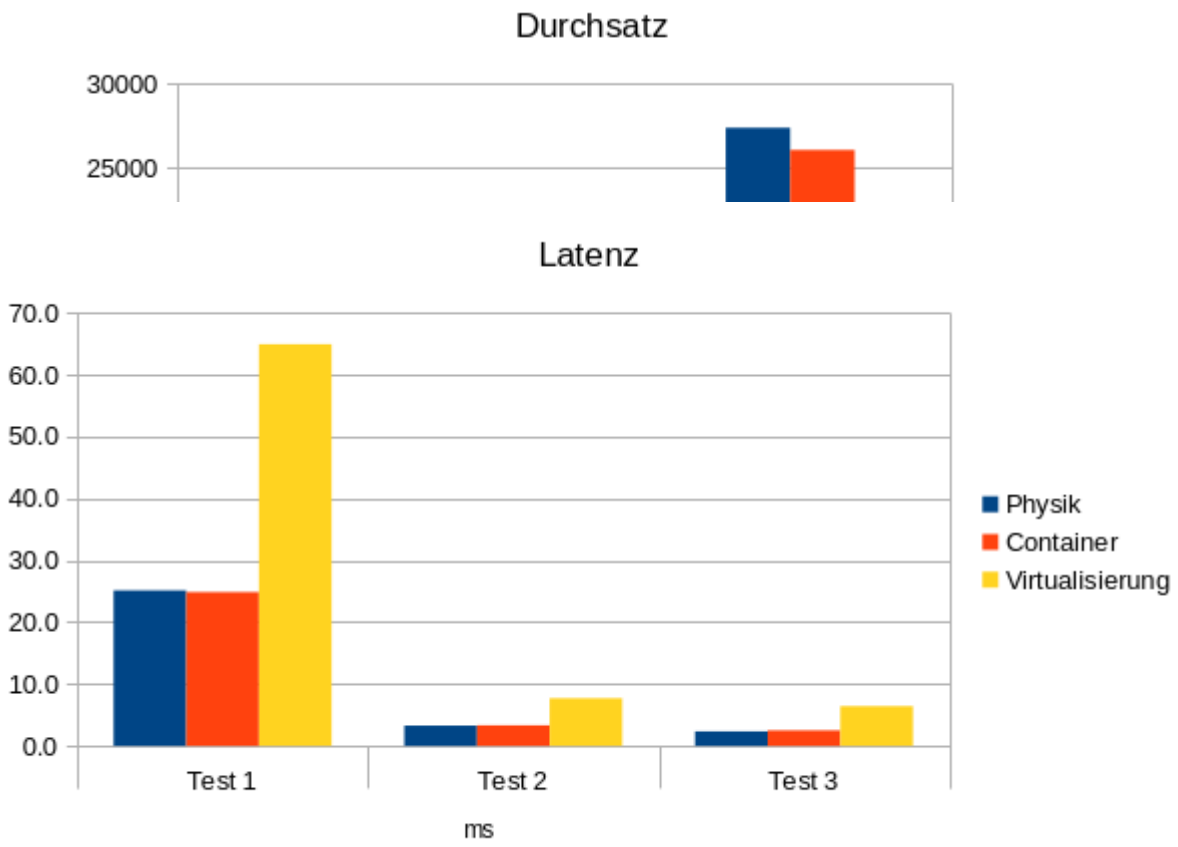
```
Latency Comparison Numbers (2020)

L1 CPU cache reference                           1 ns -
L2 CPU cache reference                           4 ns -
RAM reference                                  100 ns -
Read 1 MB sequential read from RAM           3'000 ns -     3 µs
SSD random read                             16'000 ns -    16 µs
Read 1 MB sequentially from SSD             49'000 ns -    49 µs
Read 1 MB sequential read from HDD         825'000 ns -   825 µs
Disk seek (HDD)                          2'000'000 ns - 2'000 µs - 2 ms
```

Hardware – CPU

- Antwortzeit vs. Durchsatz?
- Schnelle (clockrate) vs. viele Cores (threads)!
- Intel vs. AMD vs. ARM (single thread performance!)
- NUMA vermeiden
- Energiesparmodus vermeiden
- Virtualisierung kostet!

Hardware vs. Container vs. VM



- **Test1: heavy I/O, Test 2: moderate I/O, Test 3: read-only**

Betriebssystem (O/S)

- Modernes O/S
  - malloc, Virtualisierung, Container, I/O Scheduler
- Filesystem xfs oder ext4 vs. fancy Filesystem oder Cluster FS
- Filehandles:
  - SystemD: `LimitNOFILE=16384`
- `swappines` auf 0 – 10, aber Swap NICHT abschalten!!!
- Swap kann auch NUMA Hardware kommen
  - mehrere CPU-Sockel
  - `# lscpu | grep -i numa`
  - `SystemD: NUMAPolicy=interleave`
- HugePages: Kommt drauf an, ausprobieren, Lohnt nicht?
- Transparent HugePages: NICHT tun!
- Mountpoints (1!), aber mehrere Verzeichnisse(data, binlog, tmp, ...?)

Server Variablen I

- Zur Zeit knapp 600 (8.4) → relevant ca. 10
- SQL Schicht
  - Table Open Cache (`table_open_cache`) (r/(w))
    - Multi-mandanten Systeme (viele Schemata) → zu klein sonst OK
  - Table Definition Cache (`table_definition_cache`) (r/(w))
    - Multi-mandanten Systeme (viele Schemata) → zu klein sonst OK
  - Thread Cache (`thread_cache_size`) (connection)
    - Persistente (Java) oder nicht-persistente (PHP) Verbindungen?
    - Bei heftigen PHP Anwendungen (kein Connection-pooling) ggf. zu klein
- Binary Logging
  - Binary Logs seit 8.0 per default eingeschaltet!
  - `syny_binlog = 1 → 0`? (nur write!)
    - Signifikanten Einfluss auf die Schreibperformance

Server Variablen II

- Storage Engine: InnoDB
  - `innodb_buffer_pool_size` → 75% RAM (r/w)
  - `innodb_log_buffer_size` → 1M → 64M (nur w)
  - `innodb_flush_log_at_trx_commit = 1` (nur w)
    - → 2 aber Achtung: Datenverlust bei Crash möglich!
  - `innodb_redo_log_capacity = 100M` zu klein? (nur w)
    - 512M oder grösser je nach Schreibaufkommen...
  - `innodb_flush_method = O_DIRECT`
    - für lokale Platten (SSD, was bei SAN?)!, (nur w)
  - `innodb_flush_neighbors = 0`
    - für lokale SSD (was bei SAN?)
  - `innodb_dedicated_server = ON`???
    - `innodb_buffer_pool_size, innodb_redo_log_capacity, innodb_flush_method` (8.0 only)

Query Tuning

- Wie Abfragen finden?
  - Jemand kommt jammern... (reaktiv)
  - Slow Query Log (proaktiv)
  - `sys` Schema (proaktiv?)
  - `PEFORMANCE_SCHEMA` (`P_S`) (proaktiv?)

Slow Query Log


```
Ein- und Ausschalten:

slow_query_log                          = OFF
log_output                              = FILE (oder mysql.slow_log)
slow_query_log_file                     = $datadir/<hostname>_slow.log


Ausgabe beeinflussen:

log_timestamps                          = UTC / SYSTEM
log_slow_extra                          = OFF
log_raw                                 = OFF


Filtern:

min_examined_row_limit                  = 100
long_query_time                         = 10.0 → 0.1?
log_slow_admin_statements               = OFF
log_slow_replica_statements             = OFF
log_queries_not_using_indexes           = OFF
log_throttle_queries_not_using_indexes = 0 (per minute)
```

Slow Log: log_slow_extra


```
SQL> SET GLOBAL slow_query_log = ON;
SQL> SET GLOBAL long_query_time = 0.05;

# Time: 2024-11-15T10:11:55.815981Z
# User@Host: root[root] @ localhost []  Id:    72
# Query_time: 0.125410  Lock_time: 0.000003 Rows_sent: 262144 ⏎
  Rows_examined: 262144
use test;
SET timestamp=1731665515;
SELECT * FROM test;
```

```
SQL> SET GLOBAL log_slow_extra = ON;

# Time: 2024-11-15T10:12:43.468664Z
# User@Host: root[root] @ localhost []  Id:    73
# Query_time: 0.122279  Lock_time: 0.000003 Rows_sent: 262144 ⏎
  Rows_examined: 262144 Thread_id: 73 Errno: 0 Killed: 0 Bytes_received: 30 ⏎
  Bytes_sent: 10676014 Read_first: 1 Read_last: 0 Read_key: 1 Read_next: 0 ⏎
  Read_prev: 0 Read_rnd: 0 Read_rnd_next: 262145 Sort_merge_passes: 0 ⏎
  Sort_range_count: 0 Sort_rows: 0 Sort_scan_count: 0 Created_tmp_disk_tables: 0 ⏎
  Created_tmp_tables: 0 Start: 2024-11-15T10:12:43.346385Z End: ⏎
  2024-11-15T10:12:43.468664Z
SET timestamp=1731665563;
SELECT * FROM test;
```

Was dann?


```
# mysqldumpslow -s t slow.log > slow.profile
```

```
Count: 67  Time=32.83s (2199s)  Lock=0.00s (0s)  Rows_sent=996.3 (66752), ⏎
Rows_examined=69631694.3 (4665323520), Rows_affected=0.0 (0), user[user]@[IP] ⏎
  SELECT id FROM cdr_in WHERE anonymize=N AND date_time <= DATE_SUB(NOW(), ⏎
  INTERVAL N DAY) ORDER BY id LIMIT N

Count: 47  Time=2.09s (98s)  Lock=0.00s (0s)  Rows_sent=32.0 (1502), ⏎
Rows_examined=2053273.1 (96503834), Rows_affected=0.0 (0), user[user]@[IP] ⏎
  SELECT id, path_filename FROM customer_files WHERE to_delete IS NULL AND ⏎
filesize IS NULL AND date_time <= DATE_SUB(NOW(), INTERVAL N SECOND) LIMIT N,N

Count: 4  Time=0.73s (2s)  Lock=0.00s (0s)  Rows_sent=59980.2 (239921), ⏎
Rows_examined=119960.5 (479842), Rows_affected=0.0 (0), user[user]@3hosts ⏎
  SELECT id, gender, company, position, department, name_first, name_last, ⏎
private, note_count, owner_type, owner_id, UPPER( SUBSTRING( CONCAT(name_last, ⏎
name_first, company) , N, N ) ) AS sort FROM ab_entries WHERE kid = 'S' AND ⏎
ab_id = 'S' AND private=N ORDER BY id

...
```

- **EXPLAIN SELECT ...**

EXPLAIN

- explain_format = {TRADITIONAL|JSON|TREE}

```
SQL> EXPLAIN SELECT * FROM test AS t
JOIN documents AS d ON d.test_id = t.id
WHERE t.data = 'bla' ORDER BY t.ts;

+----+-------------+-------+------------+------+---------------+------+---------+
| id | select_type | table | partitions | type | possible_keys | key  | key_len |
+----+-------------+-------+------------+------+---------------+------+---------+
|  1 | SIMPLE      | t     | NULL       | ref  | PRIMARY,data  | data | 515     |
|  1 | SIMPLE      | d     | NULL       | ALL  | NULL          | NULL | NULL    |
+----+-------------+-------+------------+------+---------------+------+---------+

+-------+------+----------+----------------------------------------------+
| ref   | rows | filtered | Extra                                        |
+-------+------+----------+----------------------------------------------+
| const |    1 |   100.00 | Using temporary; Using filesort              |
| NULL  |    6 |    16.67 | Using where; Using join buffer (hash join)   |
+-------+------+----------+----------------------------------------------+
```

EXPLAIN FORMAT=JSON

- explain_json_format_version = {1|2} (8.4)

```
SQL> EXPLAIN FORMAT = JSON SELECT * FROM test AS t
JOIN documents AS d ON d.test_id = t.id
WHERE t.data = 'bla' ORDER BY t.ts;
```

```
"query_block": {
  "select_id": 1,
  "cost_info": {"query_cost": "1.20"},
  "ordering_operation": {
    "using_temporary_table": true,
    "using_filesort": true,
    "nested_loop": [{
      "table": {
        "table_name": "t",
        "access_type": "ref",
        "possible_keys": [
          "PRIMARY","data"],
        "key": "data",
        "used_key_parts": [data"
        "key_length": "515",
        "ref": ["const"],
...
```

```
"inputs": [
  {
    "alias": "d",
    "operation": "Table scan on d",
    "table_name": "documents",
    "access_type": "table",
    "schema_name": "test",
    "used_columns": ["id",...],
    "estimated_rows": 6.0,
    "estimated_total_cost": 0.35099
  },
  {
    "alias": "t",
    "heading": "Hash",
    "covering": false,
    "operation": "Index lookup on t
...
```

EXPLAIN FORMAT=TABLE

- **Auf Oracle Art:**

```
SQL> EXPLAIN FORMAT = TREE SELECT * FROM test AS t
JOIN documents AS d ON d.test_id = t.id
WHERE t.data = 'bla' ORDER BY t.ts;

-> Sort: t.ts
  -> Stream results  (cost=1.2 rows=1)
      -> Inner hash join (d.test_id = t.id)  (cost=1.2 rows=1)
          -> Table scan on d  (cost=0.351 rows=6)
          -> Hash
              -> Index lookup on t using data (data='bla')  (cost=0.35 rows=1)
```

`sys` Schema (auch `P_S`)

- Besteht aus Tabellen, Views, Functions, Procedures und Triggers
- Basiert auf dem `PERFORMANCE_SCHEMA`
- 2 Tabellen-Typen:
  - `x$...` (Rohdaten)
  - und aufbereite Daten

```
root@mysql-84 [sys] SQL> show tables;
+-----------------------------------------------+
| Tables_in_sys                                 |
+-----------------------------------------------+
| host_summary_by_file_io                       |
| host_summary_by_file_io_type                  |
| host_summary_by_statement_latency             |
| innodb_buffer_stats_by_table                  |
| innodb_lock_waits                             |
| io_by_thread_by_latency                       |
| io_global_by_wait_by_bytes                    |
| io_global_by_wait_by_latency                  |
| latest_file_io                                |
| memory_by_user_by_current_bytes               |
| memory_global_total                           |
| metrics                                       |
| processlist                                   |
```

`sys` Schema Themen

- `host_*` → Aktivitäten pro Host
- `innodb_*` → InnoDB Informationen
- `io_*` → I/O pro File, Bytes und Latenz
- `memory_*` → Memory Nutzung pro Host, User
- `schema_*` → Informationen über Schemata
- `statement_*` → Statistiken zu Statements
- `user_*` → Informationen pro User
- `waits_*` → Wait Event Informationen

`sys` Schema – Ansicht

- MySQL Workbench (8.0) – nicht mehr supportet!

  → Visual Studio Code?

- FromDual Ops Center

Top Memory by User

Performance for Instance mysql-80

- **Memory Usage**
  - Total Memory
  - Top Memory by Event
  - Top Memory by User
  - Top Memory by Host
  - Top Memory by Thread
- **Hot Spots for I/O**
  - Top File I/O Activity Report
  - Top I/O by File by Time
  - Top I/O by Event Category
  - Top I/O in Time by Event Categories
  - Top I/O Time by User/Thread
- **High Cost SQL Statements**
  - Statment Analysis
  - Statements in Highest 5% by Runtime
  - Using Temporary Tables
  - With Sorting
  - Full Table Scans
  - Error or Warnings
  - Recent Statements
- **Database Schema Statistics**
  - Schema Object Overview (High Overhead)
  - Schema Index Statistics
  - Schema Table Statistics
  - Schema Table Statistics (with InnoDB buffer)
  - Tables with Full Table Scans
  - Unused Indexes
- **Wait Event Times (Expert)**
  - Global Waits by Time
  - Waits by User by Time
  - Wait Classes by Time
  - Wait Classes by Average Time
- **InnoDB Statistics**

Top Memory by User

Shows Users consuming the most Memory.

User	Count (#)	Mem (Bytes)	Avg Mem (Bytes)	Max Mem (Bytes)	Tot Mem (Bytes)
background	9516	2'186'755	230	598'920	114'250'340
event_scheduler	12688	1'201'796	95	639'481	91'507'691
focmm	24	1'195'905	49'829	1'048'608	13'756'769
root	0	0	0	0	1'275'843

Export... Copy Selected Copy Query

TOP I/O by File by Time

Performance for Instance mysql-80

Top I/O by File by Time

Show highest I/O usage by File and Latency.

File	Tot I/Os (#)	Tot Time (µs)	Read Requests (#)	Read Time (µs)	Write Requests (#)	Write Time (µs)	Misc Requests (#)	Misc Time (µs)
$datadir/#innodb_redo/#ib_redo1659	6619	17337157.1	6	45.5	2534	93730.0	4079	17243381.6
$datadir/#ib_16384_0.dblwr	1926	10055395.1	1	386.5	961	49342.7	964	10005665.9
$datadir/ibdata1	1353	5341075.8	10	598.0	669	15299.9	674	5325177.9
$datadir/undo_002	2111	3352510.2	404	101113.0	1163	24620.9	544	3226776.3
$datadir/mysql.ibd	1923	2755416.9	534	101094.7	703	13067.1	686	2641255.1
$datadir/undo_001	2186	2597081.5	420	139052.3	1156	29730.6	610	2428298.6
$datadir/ibtmp1	147	52580.7	0	0.0	143	6390.4	4	46190.3
$datadir/#innodb_redo/#ib_redo1664_tmp	3	38840.9	0	0.0	0	0.0	3	38840.9
/home/mysql/product/mysql-8.0.37-linux-glibc2.28-x86_64/share/charsets/Index.xml	3	21696.4	1	21669.0	0	0.0	2	27.4
/home/mysql/database/mysql-80/binlog/mysql-80-binlog.000524	6	20447.8	0	0.0	2	13.6	4	20434.2
$datadir/#innodb_temp/temp_2.ibt	6	19831.0	0	0.0	2	39.0	4	19792.0
/home/mysql/product/mysql-8.0.37-linux-glibc2.28-x86_64/share/english/errmsg.sys	5	17864.8	3	16660.5	0	0.0	2	1204.3
$datadir/#ib_16384_1.dblwr	4	17247.9	1	17226.1	0	0.0	3	21.8
$datadir/#innodb_redo/#ib_redo1665_tmp	3	15762.2	0	0.0	0	0.0	3	15762.2
$datadir/#innodb_temp/temp_4.ibt	6	15671.4	0	0.0	2	34.8	4	15636.5
$datadir/#innodb_temp/temp_8.ibt	6	15389.4	0	0.0	2	27.5	4	15361.9

Statement Analysis

www.fromdual.com

Performance for Instance mysql-80

- **Memory Usage**
  - Total Memory
  - Top Memory by Event
  - Top Memory by User
  - Top Memory by Host
  - Top Memory by Thread
- **Hot Spots for I/O**
  - Top File I/O Activity Report
  - Top I/O by File by Time
  - Top I/O by Event Category
  - Top I/O in Time by Event Categories
  - Top I/O Time by User/Thread
- **High Cost SQL Statements**
  - Statment Analysis
  - Statements in Highest 5% by Runtime
  - Using Temporary Tables
  - With Sorting
  - Full Table Scans
  - Error or Warnings
  - Recent Statements
- **Database Schema Statistics**
  - Schema Object Overview (High Overhead)
  - Schema Index Statistics
  - Schema Table Statistics
  - Schema Table Statistics (with InnoDB buffer)
  - Tables with Full Table Scans
  - Unused Indexes
- **Wait Event Times (Expert)**
  - Global Waits by Time
  - Waits by User by Time
  - Wait Classes by Time
  - Wait Classes by Average Time
- **InnoDB Statistics**

Statement Analysis

List statements with various aggregated statistics.

Query	FTS	Executed (#)	Errors (#)	Warnings (#)	Tot Time (ms)	Max Time (µs)	Avg Time (µs)	Rows Sent (#)	Avg Rows Sent (#)	Rows Scanned (#)	Avg Rows Scanned (#)	Tmp Tables (#)	Tmp Disk Tables (#)	Rows Sorted (#)	Sort Merge Passes (#)	Digest
SELECT * FROM `sys` . `x$memory_by_threa	*	1	0	0	45.9	45944.1	45944.1	25	25	25	25	2	0	34	0	6553...bccd
SELECT SCHEMA_NAME AS SCHEMA_NAME FROM `		13	0	0	43.8	35542.7	3369.6	13	1	52	4	0	0	0	0	aa20...765c
SELECT `VARIABLE_NAME` AS `variable_name		13	0	0	21.5	3285.0	1653.4	13	1	13	1	0	0	0	0	ae04...ec89
SELECT * FROM `sys` . `statements_with_t	*	1	0	0	14.1	14090.4	14090.4	7	7	24	24	0	0	7	0	1ec8...1158
SELECT SYSTEM_USER ( )		1	0	0	13.6	13552.5	13552.5	1	1	1	1	0	0	0	0	c987...b163
SELECT * FROM `sys` . `x$memory_by_user_	*	2	0	0	11.9	7488.1	5949.9	8	4	8	4	4	0	8	0	c078...fe7a
SHOW GLOBAL VARIABLES	*	1	0	0	8.8	8847.9	8847.9	603	603	603	603	1	0	0	0	c006...c749
SELECT * FROM `sys` . `x$io_by_thread_by	*	1	0	0	6.0	5984.9	5984.9	15	15	15	15	2	0	15	0	8205...fe94
SELECT * FROM `sys` . `x$memory_global_t	*	1	0	0	4.2	4193.7	4193.7	1	1	499	499	1	0	0	0	e418...961e
SELECT * FROM `sys` .	*	1	0	0	4.1	4075.1	4075.1	3	3	3	3	2	0	3	0	ee96...a57a

Architektur & Design

- RDBMS sind grundsätzlich eine SEHR langsame Technologie → "andere" sind schneller...
- Query Cache, gibt's nicht mehr...
  - Memcached, Valkey (ex. Redis)
- Volltext Suche, ja, können wir auch...
  - Apache Solar, Apache Lucene
- BLOBs in der Datenbank, ja können wir auch... :-(
  - Filesystem
- Spatial Data/Indices, Geo Daten (`GEOMETRY`, `POINT`, `LINESTRING`, ...), ja können wir...
  - Es gibt andere, die können das wahrscheinlich besser! Pg? :-|
- JSON, ja können wir auch...
  - Document Store: MongoDB?
- Zeitreihen, ja können wir recht gut...
  - Time Series Database: RRD, InfluxDB, Prometheus
- AI / KI, ja können wir auch...
  - Bin ja mal gespannt, was MySQL als Vector Store taugt...

Was heisst Performance?

- Durchsatz (throughput)
  - Wie viel Operationen kriege ich pro Zeit durch? → Anzahl/Zeit
  - Abfragen pro Sekunde (QPS)
  - Mehr gleich besser!
- Antwortzeit (response time)
  - Wie lange dauert meine Operation? → Laufzeit
  - Laufzeit einer SQL-Abfrage (Zeit)
  - Weniger gleich besser!

Diese Grössen sind tendenziell gegenläufig!!!

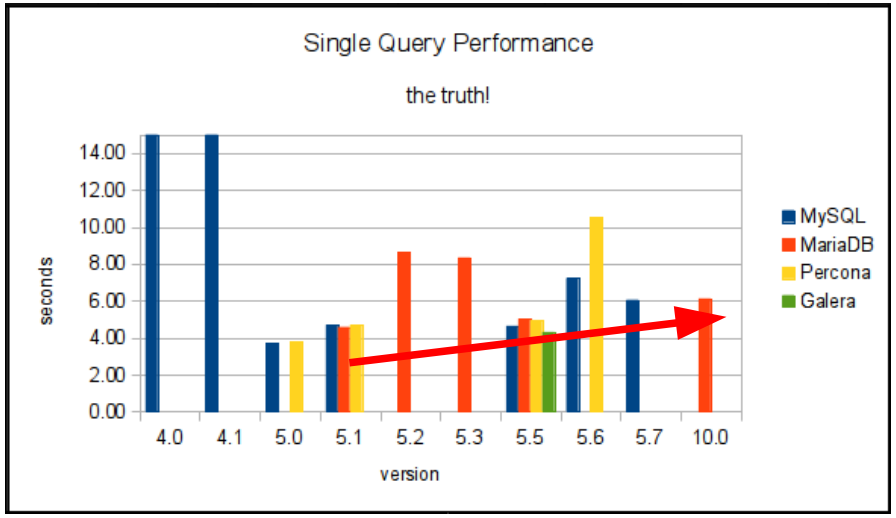
Durchsatz (throughput)

- Grad der Parallelität eines Systems
  - Concurrency, Nebenläufigkeit
  - Design und Architektur einer Applikation
- Neue DB Versionen haben tendenziell mehr Durchsatz
- Tummelplatz von Marketing!!!

Dimitri Kravtchuk, MySQL Performance Architect @ Oracle

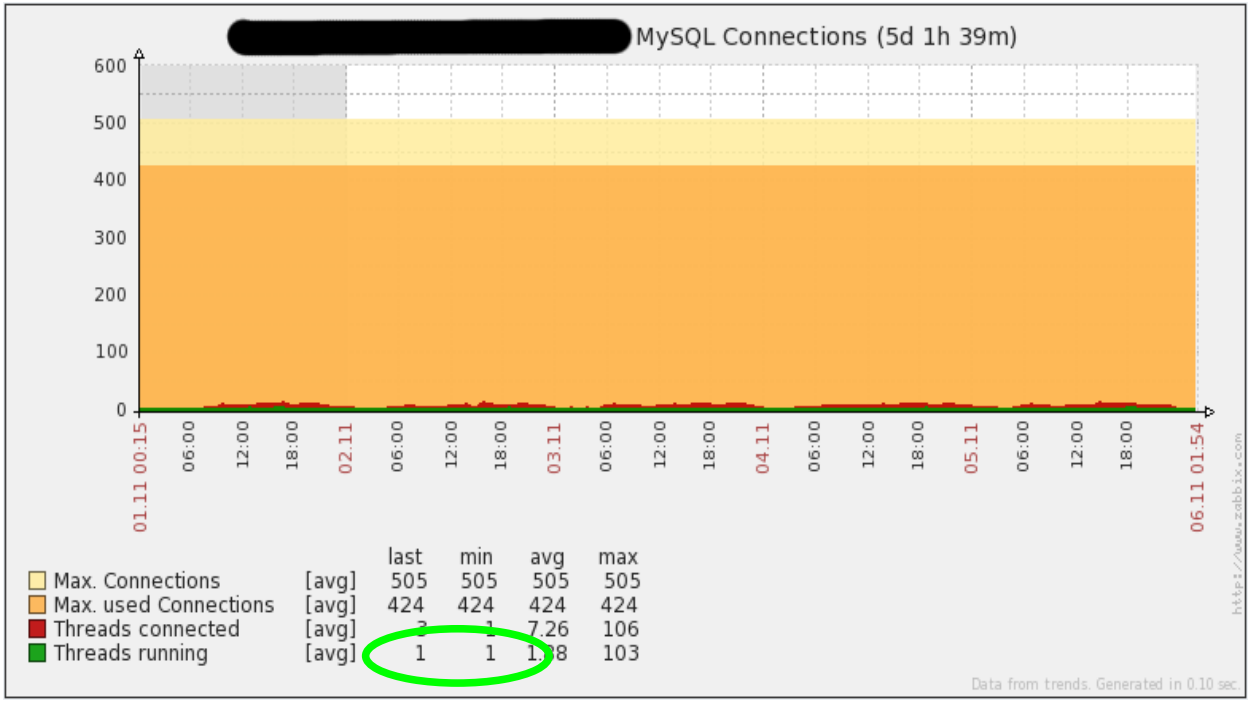
Antwortzeit (response time)

- Hängt ab von
  - der Implementierung und
  - der Datenmenge
- Neue DB Versionen haben tendenziell längere Antwortzeiten
- Durchsatz geht auf Kosten der Antwortzeit (längerer Codepath)
- Tummelplatz des DBA!



Durchschnittliche DB-Nutzung

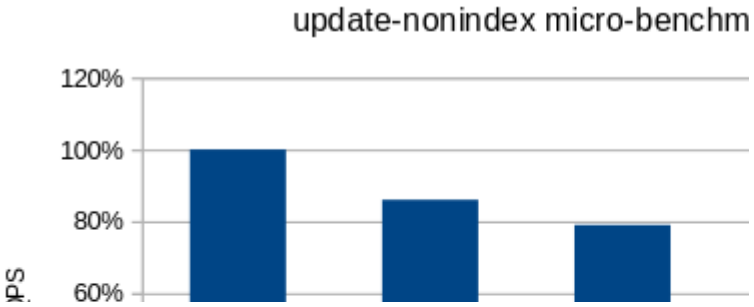
- Wo liegen nun die durchschnittliche DB-Nutzung?
- Wir sind nicht
  - Facebook,
  - Twitter,
  - LinkedIn...?

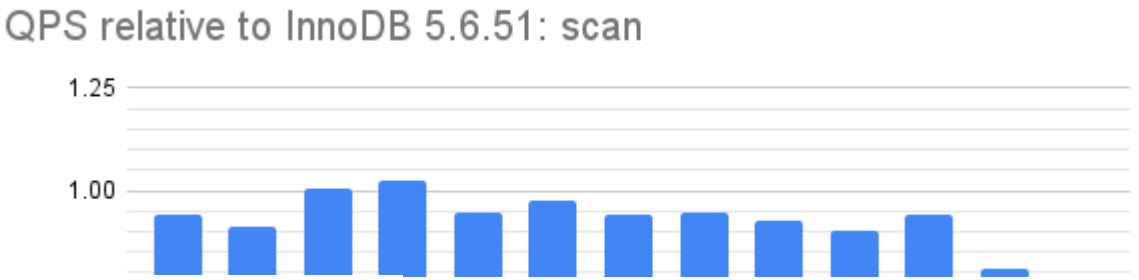


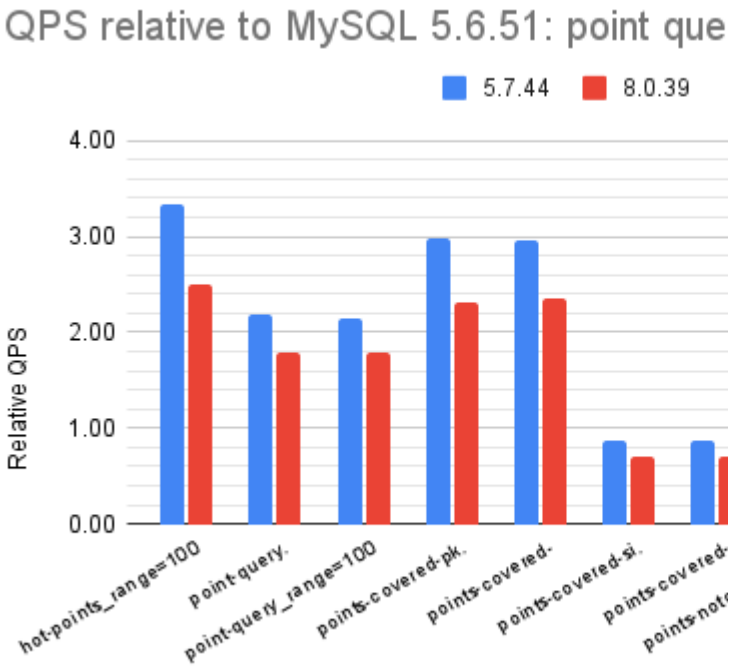
- `SHOW GLOBAL STATUS LIKE 'threads_running';`

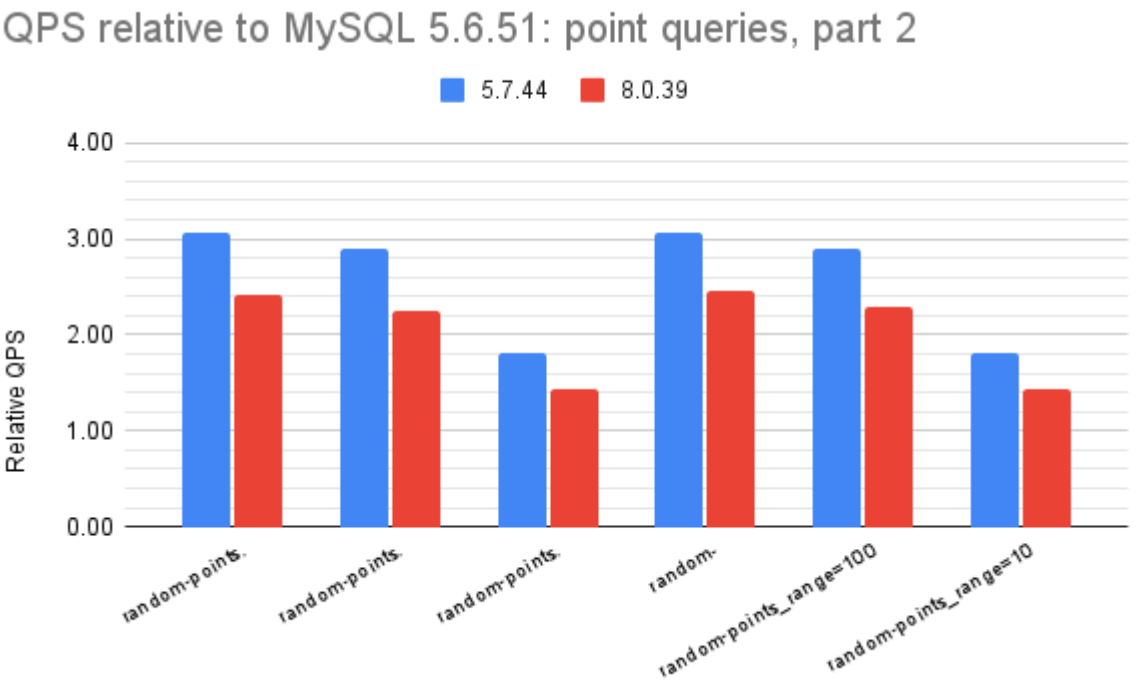
Performance Degradation

www.fromdual.com









- **Quelle: https://smalldatum.blogspot.com/**

Q & A

Fragen ?

Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

- **FromDual bietet neutral und unabhängig:**
  - **Beratung**
  - **Remote-DBA**
  - **Support für MySQL, Galera, Percona Server und MariaDB**
  - **Schulung**

www.fromdual.com/presentations