Výkonové optimalizace MySQL

Anna Janáčková, Petr Medonos

etnetera

O čem to bude?

./mysqld --run-fast

etnetera
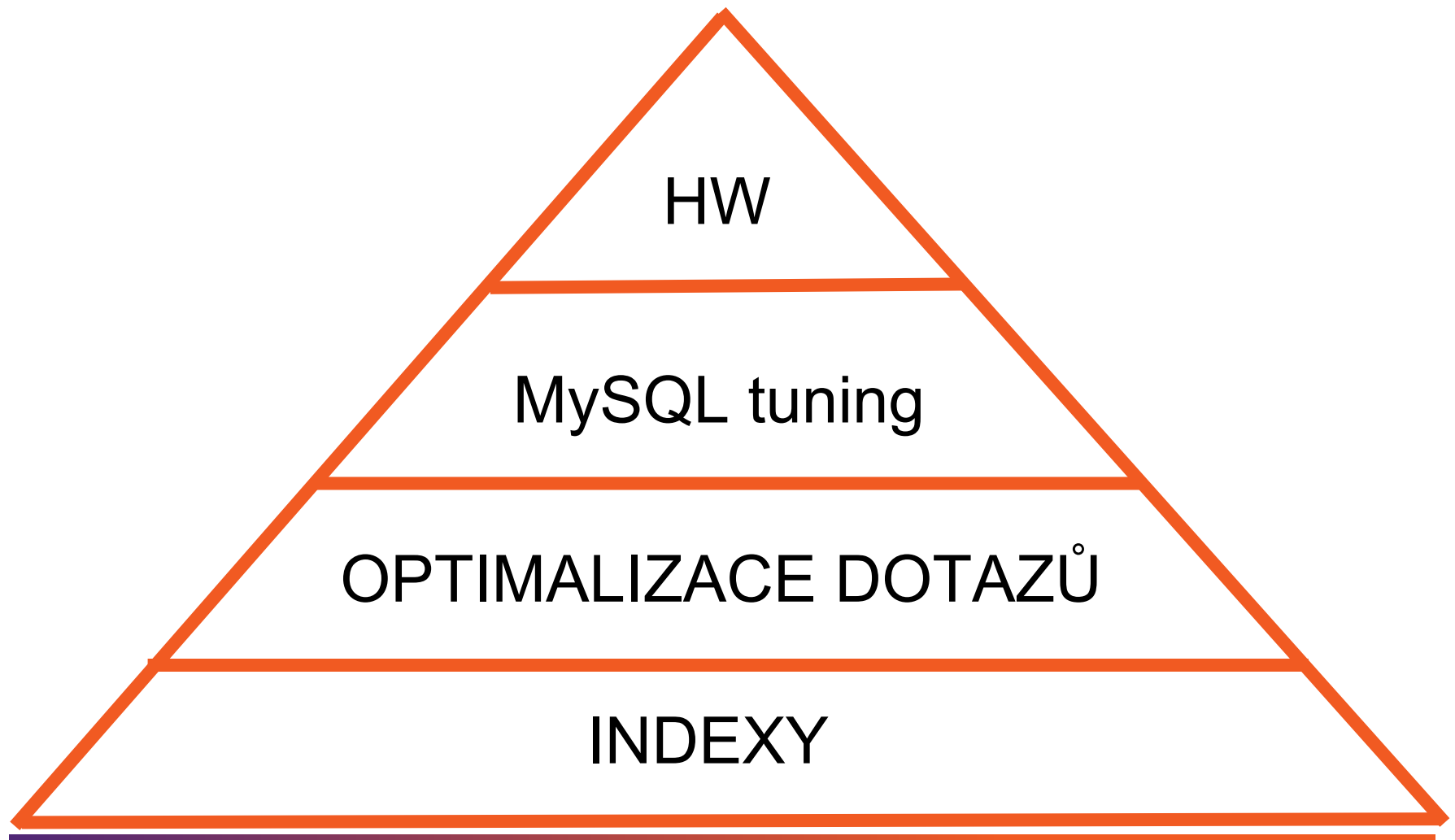
O čem to bude?

O čem to bude?

- výběr HW

- tuning GNU/Linux

- optimalizace paramerů MySQL

- analýza provozu a řešení problémů

etnetera

MySQL v ETN

- 4.1, 5.0, 5.1, 5.5

- datasety ~100MB - ~10GB

- 100% Linux

- x86_64, Intel

Efektivita ladění

HW

MySQL tuning

OPTIMALIZACE DOTAZŮ

INDEXY

etnetera

Paretovo pravidlo

80% - 20%

etnetera

Čísla, které by měl každý znát ...

http://odbms.org/download/dean-keynote-ladis2009.pdf

L1 cache reference	**0.5 ns**
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	25 ns
Main memory reference	**100 ns**
Compress 1K bytes with Zippy	3,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	**250,000 ns**
Round trip within same datacenter	500,000 ns
Disk seek	**10,000,000 ns**
Read 1 MB sequentially from disk	**20,000,000 ns**
Send packet CA->Netherlands->CA	150,000,000 ns

etnetera

Hardware

Hardware

- **CPU**
  - MySQL neumí rozkládat dotazy na víc jader
    - => výkon CPU je důležitý
  - každé spojení má vlastní thread
    - => dostatečný počet jader

- **RAM**
  - OLTP - aktivně používaná data se musí vejít do RAM

- **SÍŤ**
  - komunikace s aplikací
  - replikace
   - dedikovaná síť, bonding 2x 1Gbit

etnetera

Hardware - I/O

- potřebujeme maximální rychlost a **minimální latenci**
  - **raid10**
  - hardwarový řadič
    - "zadrátovaná" logika je rychlejší
    - baterie pro **writeback cache**
    - vypnout write barriers

      ```
      mount -o nobarrier
      ```

etnetera

GNU/Linux

- read-ahead
  - `blockdev --setra <#> /dev/<...>`

- vhodný filesystém
  - XFS
  - ext4

- I/O scheduler - deadline

- snížit swappiness
  - `echo "20" > /proc/sys/vm/swappiness`

etnetera

Optimalizace parametrů MySQL serveru
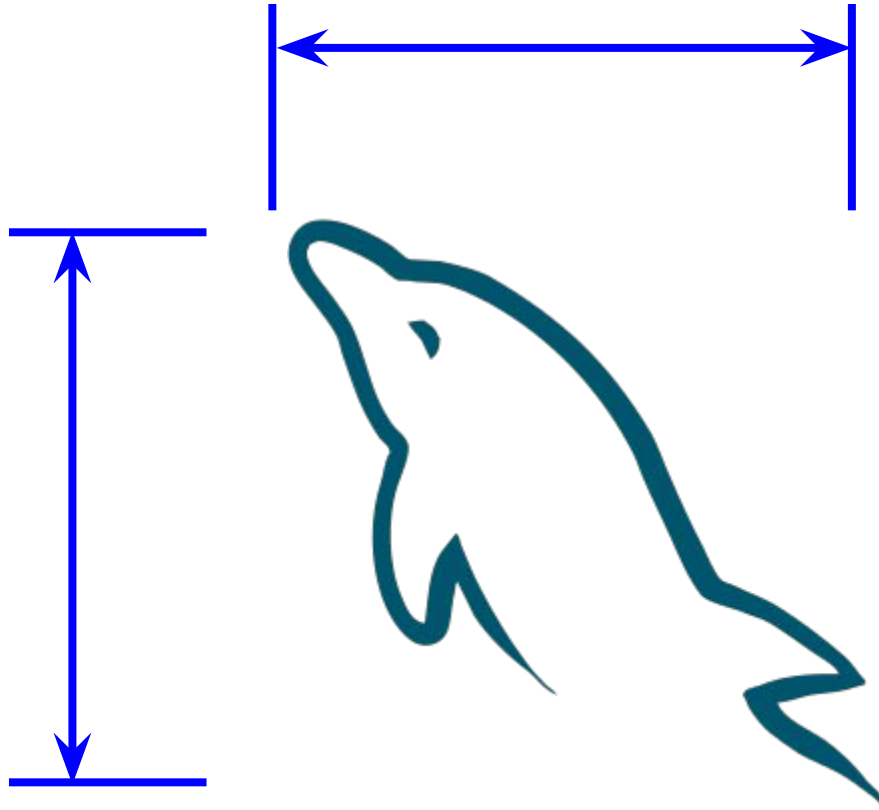
Alokace paměti

```
set @giga=1024*1024*1024;

SELECT ( @@key_buffer_size +
  @@query_cache_size +
  @@innodb_buffer_pool_size +
  @@innodb_additional_mem_pool_size +
  @@innodb_log_buffer_size + @@max_connections
  * (
  least(@@tmp_table_size,@@max_heap_table_size
  ) + @@read_buffer_size +
  @@read_rnd_buffer_size + @@sort_buffer_size
  + @@join_buffer_size + @@binlog_cache_size +
  @@thread_stack) ) / @giga AS MAX_MEM;
```

Fiktivní MySQL instance

- 24 core
- 64GB RAM
- 6x SAS 15k, raid 10 512MB cache, write-back

- ~ 10 schémat
- ~ 1000 tabulek
- InnoDB only
- dataset 25GB
- 80% čtení - 20% zápis

etnetera

Parametry MySQL serveru

```
max_connections = 400
back_log = 1024
thread_cache_size = 300
skip_name_resolve = ON
tmp_table_size = 128MB
max_heap_table_size = 128MB
sort_buffer_size = 8MB
join_buffer_size = 8MB
```
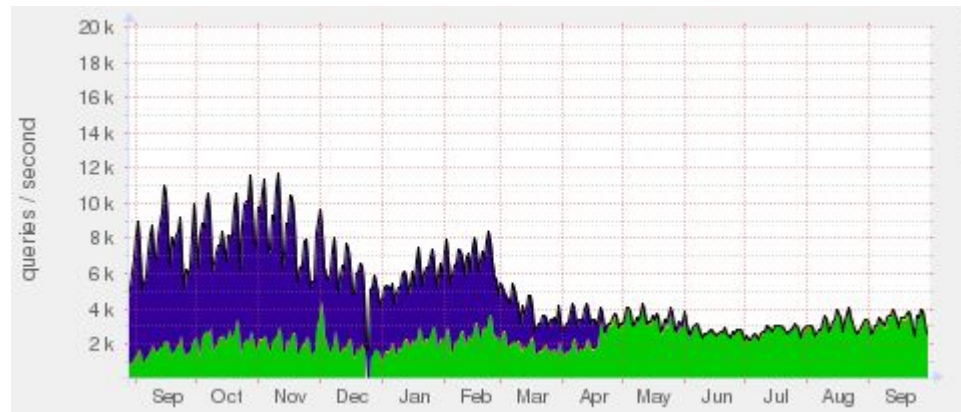
Query cache

- co nelze cachovat
  - `now(), rand(), uuid()`
  - tabulka s právy na úrovni sloupců
  - …

```
query_cache_size = 64-256MB
query_cache_limit = 16MB
query_cache_min_res_unit =
```

Query cache

$$QC\_min\_res = \frac{query\_cache\_size - Qcache\_free\_memory}{Qcache\_queries\_in\_cache}$$

```
show global variables like 'query_cache_size';
show global status like 'Qcache_free_memory';
show global status like 'Qcache_queries_in_cache';
```

etnetera

Parametry InnoDB enginu

```
innodb_buffer_pool_size        = 32G
innodb_buffer_pool_instances  = 32
innodb_purge_threads           = 1
innodb_log_file_size            = 2G
innodb_flush_method         = O_DIRECT
innodb_io_capacity            = 10000
innodb_io_capacity_max        = 15000
innodb_lru_scan_depth         = 10000
```

Analýza provozu a řešení problémů

Náročné operace pro MySQL

- query cache miss

- table scan, full join

- tmp tabulka

- tmp tabulka vytvořená na disku, filesort na disku

etnetera

Slow query

```
slow-query-log-file      =
 /var/log/mysql/mysql-slow.log
slow-query-log               = 1
```

slow_launch_time   = 1000

log-queries-not-using-indexes

```
long_slow_filter =
"tmp_table_on_disk,filesort_on_disk,full_joi
n"
```

Slow query - pt-query-digest

ukázka

Detekce zámků (na úrovni InnoDB)

```
SELECT r.trx_id waiting_trx_id, r.trx_mysql_thread_id
  waiting_thread, p1.USER waiting_user, p1.HOST
  waiting_host, r.trx_query waiting_query, b.trx_id
  blocking_trx_id, b.trx_mysql_thread_id
  blocking_thread, p2.USER blocking_user, p2.HOST
  blocking_host, b.trx_query blocking_query FROM
  information_schema.innodb_lock_waits w INNER JOIN
  information_schema.innodb_trx b ON b.trx_id =
  w.blocking_trx_id INNER JOIN
  information_schema.innodb_trx r ON r.trx_id =
  w.requesting_trx_id INNER JOIN
  information_schema.PROCESSLIST p1 ON p1.id =
  r.trx_mysql_thread_id INNER JOIN
  information_schema.PROCESSLIST p2 ON p2.id =
  b.trx_mysql_thread_id;
```

Nástroje optimalizace

- explain

- profiling

- performance schema

- (show status, show engine innodb status)

etnetera

Explain

- jak je dotaz vyhodnocen?
- jaké indexy se používají?

```
explain select SQL_NO_CACHE item.id
 from item JOIN item_review on
 item.id=item_review.item_id group
 by item.id order by item.id\G;
```

Explain

```
           id: 1
  select_type: SIMPLE
        table: item_review
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 41310
        Extra: Using temporary; Using
  filesort; Start temporary
```

etnetera

Explain

```
          id: 1
 select_type: SIMPLE
       table: item_review
        type: index
possible_keys: item_order
         key: item_order
     key_len: 8
         ref: NULL
        rows: 41310
       Extra: Using index; Using temporary;
  Using filesort
```

Profilování

```
select SQL_NO_CACHE distinct id from item
  where id IN (select item_id from
  item_review) order by id;
```

9038 rows in set (1.31 sec)

```
select SQL_NO_CACHE item.id from item JOIN
  item_review on item.id=item_review.item_id
  group by item.id order by item.id;
```

9038 rows in set (0.07 sec)

Profilování

```
set profiling = 1;
select Q1 …
select Q2 …
set profiling = 0;

show profiles;
show profile for query 1;
show profile for query 2;
```

Profilování

| Status | Duration 1 | Duration 2 |
+---------------------+------------+------------+
optimizing	0.000007	0.000006
Creating tmp table	0.000000	0.000027
executing	0.000006	0.000003
Copying to tmp table	**0.000000**	**0.065036**
Sorting result	0.000004	0.001362
Sending data	0.000038	0.002149
optimizing	0.000007	0.001362
statistics	0.000013	0.002149
preparing	**1.306187**	**0.000005**

Performance schéma

- MySQL 5.5+
- systémová databáze
- SQL rozhraní

```
performance_schema = ON
show variables like 'perf%';
show engine performance_schema
status;
```

Performance schéma

- zakladni nastavení pomocí setup_ tabulek
  - `setup_instruments`
  - `setup_consumers`
  - `setup_timers`
  - `setup_actors`
  - `setup_objects`
- ps_helper (Mark Leith)

etnetera

Performance schéma - instruments

```
wait/io/file/sql/binlog
wait/io/file/sql/binlog_index
...
statement/sql/select
statement/sql/update
statement/sql/insert
...
stage/sql/Opening tables
stage/sql/optimizing
stage/sql/preparing
```

Performance schéma (ps_helper)

```
mysql> select * from schema_unused_indexes where
  object_schema != 'mysql' and index_name !=
  'PRIMARY';

+-------------------+-------------+-------------------+
| object_schema     | object_name | index_name        |
+-------------------+-------------+-------------------+
| testcz            | item        | title_cz          |
| testcz            | item        | url_id_cz         |
| testcz            | item        | item_segment_id   |
| testcz            | item_review | PRIMARY           |
| testcz            | item_review | eid               |
+-------------------+-------------+-------------------+
```

etnetera

Optimalizační hinty

- SQL_NO_CACHE/SQL_CACHE

- USE/FORCE/IGNORE INDEX

- STRAIGHT_JOIN

- SQL_BIG_RESULT

- ….

etnetera

Optimalizační hinty

```
SELECT SQL_CALC_FOUND_ROWS STRAIGHT_JOIN
  DISTINCT item.id FROM  item JOIN
  item_available ON


SELECT * FROM table1 USE INDEX
  (col1_index,col2_index)
  WHERE col1=1 AND col2=2 AND col3=3;
```

etnetera
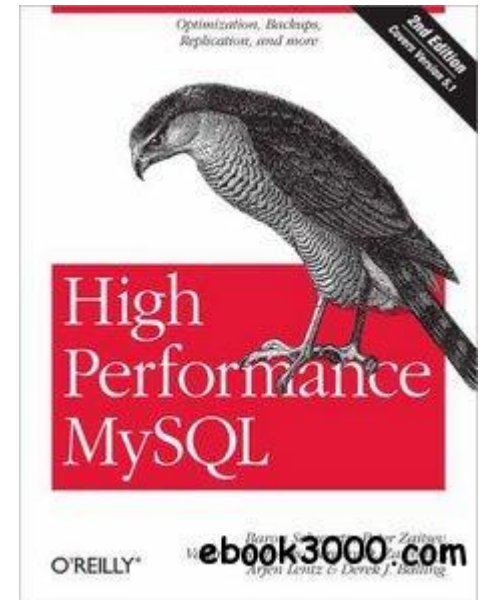
Co když to nestačí ...

- replikace
  - master - slave, master - master

- Cluster
  - Galera (Percona XtraDB Cluster), MySQL cluster

- sharding

etnetera

Užitečné tooly

- percona toolkit
  - pt-query-digest
  - pt-mext
- ps_helper
- mysqltunner.pl
- mysql_primer
- munin mysql (https://github.com/kjellm)
- percona nagios plugins

Zdroje

- High Performance MySQL (3. edice)
- http://www.mysqlperformanceblog.com/
- http://blog.jcole.us/
- http://dev.mysql.com/doc/refman/5.6/en/

COME TO THE DARK SIDE
WE HAVE COOKIES

etnetera

Q&A

petr.medonos@etnetera.cz

anna.janackova@etnetera.cz