

OPTIMALIZACE dotazů a výkonu v MySQL



DOZVÍTE SE:

- ▶ Klíčové konfigurační volby a jejich nastavení
- ▶ Příčiny zamykání záznamů a deadlocků a jejich minimalizace
 - ▶ Efektivní využití indexů pro rychlejší dotazy
- ▶ Další optimalizační možnosti, včetně využití AI

 **26. 4. 2023**

 **od 18 hod**

 **Restaurace U Salzmannů**
salónek v 1. patře, občerstvení zdarma



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

RTsoft s.r.o.

- Plzeňská softwarová společnost působící na trhu již přes 20 let
- Vývoj webových a mobilních aplikací na míru
- Více než 300 dokončených projektů
- Mezi naše partnery patří MND, Knihy Dobrovský (3× eshop roku), Sazka, Sportisimo, Darujme.cz a mnoho dalších
- Více než 40 vývojářů



Informační systémy



Business



Mobilní



Weby



MND





Pavel Polívka

Software Developer

polivka@rtsoft.cz

www.rtsoft.cz

- Konfigurační volby MySQL
- Cizí klíče, zamykání a deadlocky
- Indexy
- Další možnosti optimalizací

+ hodnocení znalostí AI ChatGPT 3.5/4



Konfigurační volby MySQL

Konfigurace



- Stovky konfiguračních voleb
- Zásadních pro výkon je pouze několik
- Testovat a monitorovat vliv na výkon
- Pozor na rozdíly PROD/TEST/UAT prostředí

innodb_buffer_pool_size



- **Nejdůležitější konfigurační parametr**
- Obsahuje cache dat a indexů
- Nastavit v závislosti velikosti DB a RAM serveru

```
SELECT SUM(data_length + index_length) AS total_size  
FROM information_schema.tables;
```


innodb_log_file_size



- **Druhý nejdůležitější konfigurační parametr**
- Udržuje data transakcí pro případ ROLLBACKu
- Nenastavovat zbytečně veliký
- Velikost nastavit tak, aby stačila na cca hodinu provozu

```
SHOW STATUS LIKE 'Innodb_os_log_written';
```



tmp_table_size max_heap_table_size

- Určuje max. velikost temp tabulek – interní a uživatelsky definované (`CREATE TEMP TABLE xxx`)
- Pozor: při překročení velikosti jsou tabulky ukládané na disk

sort_buffer_size



- Cache pro řazení, ovlivňuje rychlost použití ORDER BY
- Každé DB spojení využívá vlastní cache
- **Pokud je potřeba, tak se vždy alokuje celá velikost**
- Navyšování provádět postupně a monitorovat
- Případně navýšit selektivně pouze pro vybrané dotazy




query_cache_type

query_cache_size

- Cachování výstupu pro SELECT dotazy
- Nevyzpytatelné chování
- Deprecated (MySQL 5.7), removed MySQL 8.0
- V MariaDB stále podporováno

Nástroje pro validaci a kontrolu nastavení

- Existuje několik SW nebo i webových nástrojů
- S přístupem k db více informací a lepší doporučení
- Doporučujeme  MySQLTuner

<https://github.com/major/MySQLTuner-perl>

Cizí klíče, zamýkání a deadlocky

Setkáváte se s?



- **SQLSTATE[HY000]: General error: 1205 Lock wait timeout exceeded; try restarting transaction**
- **SQLSTATE[40001]: Serialization failure: 1213 Deadlock found when trying to get lock; try restarting transaction**

Zobrazení blokujících/čekajících dotazů



```
SELECT r.trx_id      waiting_trx_id,  
       r.trx_query  waiting_query,  
       b.trx_id     blocking_trx_id,  
       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;
```


Jaké jsou typy zámků?



- Shared (čtení) and Exclusive (zápis) locks
- Intention Locks - `LOCK TABLES; SELECT FOR UPDATE;`
- Gap Locks - `WHERE id BETWEEN 1 AND 10;`
- Record Locks - `WHERE id = 10;`
- AUTO-INC Locks – Pro sloupce s `AUTO_INCREMENT`

Co je deadlock?



- Je situace, kdy dvě nebo více transakcí navzájem čekají na uvolnění zdrojů, které drží ostatní transakce
- Jsou detekovány okamžitě při vzniku
- DB automaticky KILLne tu „jednodušší“ transakci

Příklad deadlocku



- Transakce A provádí úpravu objednávky O.
- Transakce B provádí úpravu řádky Ř u objednávky O.
- Transakce A chce aktualizovat řádky objednávky O – musí ale čekat na uvolnění zámku na řádku Ř.
- Transakce B chce aktualizovat i příslušnou objednávku O – ta je již zamčena transakcí A – **DEADLOCK!**

Jak lze tyto problémy minimalizovat?



- Zkrátit dobu trvání dotazů a tedy držení zámků
- Odebrat nepotřebné dotazy z transakcí, rozdělit je do více jednotlivých nebo využít SQL funkce/procedury
- Vyhýbat se paralelním zpracování (joby/batche/MQ)
- Udržovat stejnou logiku provádění úprav tabulek
- **Nepoužívat cizí klíče** 😊

Jak cizí klíče způsobují problémy?



- Cizí klíče vytváří zámky na odkazovaných tabulkách
- Pokud se vrátíme k příkladu z deadlocku, kdy:
 - Transakce A provádí úpravu objednávky O.
- A spustíme akci:
 - Transakce B chce přidat řádek k objednávce O -
WAIT

Indexy

Standardní indexy - vlastnosti



- V jednom dotazu je použit max. jeden index
- Non-cluster, B-tree index
- Vždy v sobě zahrnují PRIMARY KEY
- Lze využít pro „Query coverage“

Obecná doporučení



- Vytvářet nad sloupci s vyšší kardinalitou
- Vícesloupcové indexy, pozor na správné pořadí
- Odstraňovat duplicitní indexy
- Odstraňovat nepoužívané indexy

Pořadí ve vykonání SELECT dotazu



- Optimizer si může pořadí upravit, ale zpravidla je:
 - FROM + JOIN
 - WHERE
 - GROUP BY
 - SELECT
 - ORDER BY

Primární klíče



- **Každá tabulka by měla mít primární klíč!**
- Je součástí všech ostatních indexů
- Volit co nejmenší datový typ
- Při skládání více sloupců použít raději unikátní index
- Cluster index – využívat sekvence

Modifikace hodnoty sloupce

- Na modifikované sloupce nelze použít index
- V následujícím příkladu je ID PRIMARY KEY:
- `SELECT id FROM contracts WHERE id + 1 = 2;`
- Nad tabulkou s 2M záznamy trvá cca 250 ms.
- Jiný, reálnější příklad: `MONTH(event_date) = 12`

Function-based indexy



- Pomocí virtuálních (generovaných) sloupců
- Od MySQL 8.0.13 standardní podpora
- MariaDB pouze pomocí virtuálních sloupců

Virtuální sloupec s indexem



```
CREATE TABLE events (  
    event_date DATE,  
    event_month INT AS (MONTH(event_date))  
    PERSISTENT  
);
```

```
CREATE INDEX idx_event_month ON events  
(event_month);
```

Funkční index



```
CREATE TABLE events (  
    event_date DATE  
);
```

```
CREATE INDEX idx_event_month ON events  
(MONTH(event_date));
```

Analýza dotazů

- `SELECT id FROM contracts WHERE id + 1 = 2;`
- EXPLAIN – zobrazuje exekuční plán
- (EXPLAIN) ANALYZE – profilování dotazu
- Co na to ChatGPT?
- 1,SIMPLE,contracts,index,,type,1,,1881416,1871677,100,0,Using where; Using index

- Zjednodušení podmínky WHERE: V současné podmínce WHERE je uvedeno $id + 1 = 2$, což znamená, že hledáme řádek, kde je hodnota id rovna 1. Místo použití operace sčítání v podmínce můžeme podmínku zjednodušit na $id = 1$.
- Využití primárního klíče nebo jedinečného indexu: Pokud je sloupec id primární klíč nebo jedinečný index, databázový systém by měl být schopen najít požadovaný řádek velmi rychle. Ujistěte se, že máte na sloupci id primární klíč nebo jedinečný index, který zlepší výkon dotazu.

Další možnosti optimalizací

DB návrh tabulek a struktura



- Využití normalizace a denormalizace tabulek
- Používání odpovídajících datových typů
- Zvolit vhodný formát řádků pro optimalizaci ukládání a načítání dat (Compact/Dynamic/Compressed)

Datové typy sloupců



- Správně určit velikost (TINYINT/INT atp.)
- Definovat sloupce jako NOT NULL, pokud je to možné
- Co nejjednodušší – např. nepoužívat více COLLATIONS
- Používat CHAR pro pevně dlouhé řetězce a u VARCHAR definovat co nejpřesněji max. délku
- Používat ENUMy místo CHAR/VARCHAR

Table partitioning



- Rozdělení tabulky do více fyzických částí
- Možné zvýšení rychlosti čtení
- Jednodušší správa jednotlivých částí (např. pomocí ALTER TABLE)
- **Nelze použít pokud jsou v tabulce cizí klíče**

Přepis dotazů - subdotazy



- Tabulkové subdotazy nejsou dobře optimalizované
- Možná řešení jsou:
 - Rozdělení do více dotazů
 - Zopakovat podmínky z hlavního dotazu
 - CTE / UNION / (NOT) EXISTS / JOIN
 - Nahradit skalárním subdotazem

Přepis dotazů – GROUP BY



- GROUP BY může být výkonnostně velmi náročný
- Používat pouze pro agregační funkce
- Nevyužívat funkčnosti MySQL ve vypisování sloupců, které nejsou vyjmenované v GROUP BY

Závěr k využití AI ChatGPT

- Verze 4 poskytuje kvalitnější a přesnější výsledky
- Velmi dobrá znalost tematiky
- Velmi dobrá schopnost generování SQL skriptů
- Problematické jsou informace ze starších verzí MySQL a mix informací z MSSQL/Oracle



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

Děkuji za pozornost



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

Stále nabíráme!

www.rtsoft.cz/kariera



AI manéž

31.5. v 18:00 U Salzmanů

- ★ LLM a jak na ně!
- ★ Praktické využití!
- ★ Jak fungují pod kapotou?
- ★ Jaký budou mít dopad?
- ★ ...a také: jak si rozběhat vlastní ChatGPT!