

PERFORMANCE E TUNING COM MySQL

www.mysqlbrasil.com.br

Wagner Bianchi é:

Certified MySQL Associate

Certified MySQL Developer

Certified MySQL Database Administrator



Teste de Performance



“MySQL aguenta mesmo o tranco?”

- 🐟 Que tal inserirmos 1 milhão de registros em uma tabela sem *tuning* durante esta palestra?
- 🐟 Versão utilizada nesta palestra: 5.1.22rc-2

www.mysqlbrasil.com.br

Procedimentos



➤ Criar o banco de dados test:

```
mysql> create database test;  
Query OK, 1 row affected (0.00 sec)
```

Procedimentos



🐟 Crie uma tabela no banco de dados test chamada t1:

```
mysql> CREATE TABLE t1 (
```

```
-> id int not null auto_increment primary key,
```

```
-> nome char(40) not null,
```

```
-> dt_cadastro timestamp default
```

```
-> current_timestamp()
```

```
-> ) ENGINE=InnoDB;
```

Query OK, 0 rows affected (0.00 sec)

Procedimentos

✎ Criar um *Stored Procedure* para popular a tabela t1;

```
mysql> delimiter //
```

```
mysql> create procedure sp_test(v_num int, v_nome char(40))
```

```
-> begin
```

```
-> set @ var = 0;
```

```
-> while (@ var <= v_num) do
```

```
-> insert into t1 set
```

```
→ nome =concat(v_nome,'_',@ var);
```

```
-> set @ var = @ var + 1;
```

```
-> end while;
```

```
-> end;
```

```
-> //
```

Query OK, 0 rows affected (0.00 sec)

Procedimentos



↳ Chama a *procedure*, passando o número de registros e um nome como parâmetros:

```
mysql> call sp_test(1000000,'M ySQL')//  
Query OK , 0 rows affected (6 min 9.10 sec)
```



By Wagner Bianchi

Tópicos



Otimização de Servidor;

- *Storage Engines*;
- Verificação de variáveis de ambiente;
- Configurações *Server-Wide* e *Per-Client*;

Otimização de Consultas;

- “O que otimizar” e “Como otimizar”?
- Mecanismos disponíveis no servidor;
- MySQL Query Cache;

Tópicos



↳ O que mudou nas versões mais recentes do servidor de bancos de dados MySQL;

↳ Perguntas;

Otimização de Servidor



- Antes de partirmos para a otimização, saber sobre os *Storage Engines* que controlam as tabelas do banco de dados é muito importante;
- Storage Engines* são os motores de armazenamento que trás ao ambiente determinadas características importantes para o negócio;

Otimização de Servidor



↳ São exemplos de Storage Engines:

- InnoDB;
- MyISAM;
- Archive;
- Maria;
- Falcon;
- Federated;

Otimização de Servidor



InnoDB

- Storage Engine transacional;
- Suporte a integridade referencial;
- Tablespace compartilhado ou per-table;
- Logs consistentes (undo, redo e segmento de rollback);
- Suporte à transações com bloqueio em nível de linha;

Otimização de Servidor



InnoDB

```
mysql> UPDATE TBL_CLIENTE SET DT_NASC ='20/12/1900'  
-> WHERE ID =2;
```

TBL_CLIENTE



1	Wagner Bianchi	20/03/1990
2	Nicolas Cage	17/08/1923
3	John Travolta	03/09/1967

Se @@autocommit=1, após o término do comando UPDATE e se este executou com sucesso, ocorrerá um COMMIT interno e o bloqueio será liberado;

Otimização de Servidor



MyISAM

- *Storage Engine* nativo e muito rápido;
- Não poderá ser desabilitado;
- Muito sensível a quedas de energia;
- Não dá suporte a transações;
- Não dá suporte a integridade referencial;
- Pode ser reparado com `myisamchk`;
- Bloqueios (LOCK) em nível de tabela;

Otimização de Servidor



MyISAM

```
mysql> UPDATE TBL_CLIENTE SET DT_NASC ='21/12/1900'  
-> WHERE ID =2;
```

TBL_CLIENTE



1	Wagner Bianchi	20/03/1990
2	Nicolas Cage	20/12/1900
3	John Travolta	03/09/1967

- Ambientes com muitas **leituras e escritas** são mais lentos utilizando tabelas controladas pelo *Storage Engine* MyISAM;

Otimização de Servidor



- ↳ Cada *Storage Engine* possui as suas principais características e também os seus próprios parâmetros de ajuste de performance;
- ↳ Tais parâmetros são configurados com os seu respectivos valores que impactam no funcionamento do servidor de bancos de dados;

Otimização de Servidor



O comando `SHOW VARIABLES` poderá mostrar ao administrador do servidor de bancos de dados os valores das variáveis de ambiente do MySQL;

```
mysql> SHOW VARIABLES;
```

```
+-----+-----+
| Variable_name          | Value          |
+-----+-----+
| auto_increment_increment | 1              |
```

...

Otimização de Servidor



As variáveis são subdivididas:

- *Per-Client*;
- *Server-Wide*;

Per-Client se aplicam à sessão do usuário;

Server-Wide podem ser configuradas de modo global no runtime do servidor de bancos de dados ou no arquivo de configurações **my.cnf**;

Otimização de Servidor



- Os parâmetros chamados de *Per-Client* atuam nas sessões dos usuários com o intuito de melhor trafegar os dados resultantes de uma consulta;
- Necessariamente, para que estes funcionem de maneira eficiente, devemos trabalhar juntamente com os parâmetros *Server-Wide*;

Otimização de Servidor



Per-Client parameters

- read_buffer_size;
- read_rnd_buffer_size;
- sort_buffer_size;
- join_buffer_size;
- max_allowed_packet;
- net_buffer_length;

Otimização de Servidor



Per-Client parameters

Para melhorar a performance por sessão:

```
mysql> SET read_buffer_size=64*1024*1024;
```

```
Query OK, 0 rows affected (0.00 sec)
```

Valores são informados em bytes. Para checar o novo valor da variável de ambiente, utilize SHOW VARIABLES + LIKE;

Otimização de Servidor



Per-Client parameters

- Podem ser configurados no arquivo de configuração, no agrupamento [mysqld]:

```
[mysqld]
read_buffer_size=128M
read_rnd_buffer_size=256M
sort_buffer_size=128M
max_allowed_packet=128M
net_buffer_length=256M
```

Otimização de Servidor



InnoDB (Server-Wide)

- innodb_buffer_pool_size;
- innodb_additional_mem_pool_size;
- innodb_open_files;
- innodb_data_file_path;
- innodb_data_home_dir;
- innodb_log_group_home_dir;
- innodb_log_buffer_size;
- innodb_log_file_size;

Otimização de Servidor



MyISAM (Server-Wide)

- `key_buffer_size`;
- `myisam_block_size`;
- `myisam_recover`;
- `myisam_data_pointer_size`;
- `myisam_sort_buffer_size`;
- `myisam_stats_method`;
- `skip-concurrent-insert`;

Otimização de Servidor



- ❏ Variáveis *Server-Wide* são configuradas no arquivo de opções;
- ❏ Necessita reinicialização para que as novas configurações comecem a prevalecer;
- ❏ No Linux, o arquivo de opções fica em:
– `/var/lib/mysql` .

Tópicos



🐟 Otimização de Servidor;

- *Storage Engines;*
- Verificação de variáveis de ambiente;
- Configurações *Server-Wide* e *Per-Client*;

🐟 Otimização de Consultas;

- “O que otimizar” e “Como otimizar”?
- Mecanismos disponíveis no servidor;
- MySQL Query Cache;

Otimização de Consultas



- ↘ O servidor de bancos de dados MySQL nos provê uma série de atributos para análise de performance relacionado com consultas;
- ↘ Sabe eleger quais consultas realmente precisam de tuning é o “x” da questão;

Otimização de Consultas



- O comando `SHOW [FULL] PROCESSLIST` poderá lhe fornecer uma visão de quais consultas levam maior tempo para executar;
- A coluna `TIME` mede o tempo de execução em segundos;

Otimização de Consultas



- **SHOW PROCESSLIST** exibe os processos ativos, rodando no servidor, desde seu usuário tenha o privilégio SUPER;
- **SHOW FULL PROCESSLIST** exibirá a consulta em execução inteira para o caso de você já querer elegê-la para tuning;

Otimização de Consultas



↘ Análise do log de consultas lentas:

- Log Slow Queries: este é o log de queries lentas que pode registrar as consultas que levam mais tempo que o valor configurado em `long_query_time` e também consultas que não utilizam índices;

Otimização de Consultas



↳ Habilitando o log de consultas lentas:

```
[mysqld]
```

```
# Here you can see queries with especially
```

```
# long duration
```

```
log_slow_queries= /var/log/mysql/mysql-slow.log
```

```
long_query_time = 2
```

```
log-queries-not-using-indexes
```

Otimização de Consultas



- Para extrair as informações de puro texto de forma ordenada do arquivo de log de consultas lentas, utilize o aplicativo mysqldumpslow;
- Após verificar as informações extraídas do log de consultas lentas, podemos eleger a que mais nos interessa e rodá-la com o EXPLAIN;
- EXPLAIN nos auxilia na observação do plano de execução das consultas;

Otimização de Consultas



↳ Estratégias de criação de índices:

- Procure posicionar colunas indexadas nas cláusulas WHERE, GROUP BY, HAVING e ORDER BY;
- Utilize USING, FORCE e IGNORE INDEX para forçar a utilização e a não utilização de índices;

Perguntas



Wagner Bianchi é Tecnólogo em Gerenciamento de Banco de Dados pela Faculdade Infórium de Tecnologia, Diretor de Tecnologia em Bancos de Dados da MySQL Brasil com as seguintes certificações:

- Certified MySQL Associate;
- Certified MySQL Developer I e II;
- Certified MySQL Database Administrator I e II;



TEM VONTADE DE TRABALHAR COMO DBA MySQL?
Envie-nos o seu currículo: www.mysqlbrasil.com.br



By Wagner Bianchi