

6 – Introdução

- Os motores de armazenamento (storage engines) oferecem grande flexibilidade ao MySQL, pois permite que o banco de dados seja customizado para diferentes situações.
- O desenvolvedor deve analisar as características do ambiente e analisar qual engine é a mais adequada para garantir a melhor performance do MySQL. Cada engine é indicada para situações distintas.
- Entretanto, a **InnoDB** é a engine mais utilizada atualmente para quase todos os tipos de ambiente de produção, pois seu desempenho e segurança são muito superiores a maioria dos outros motores de armazenamento.
- A **Oracle** comprou a empresa que desenvolveu o InnoDB e integrou a sua equipe de desenvolvimento com a equipe do MySQL, fazendo com que a plataforma evoluísse consideravelmente nos últimos anos.

6.1 – Características do InnoDB

- O InnoDB é mais do que uma engine rápida baseada em disco, ela oferece todos os recursos exigidos por um banco de dados moderno, são eles:

1. **MVCC**: Várias versões dos dados são mantidas para que usuários possam consultar informações bloqueadas para escrita. Não existe bloqueio de leitura.

2. **ACID**: Atomicidade, consistência, isolamento e durabilidade.

3. **Suporte a transações**: Execução de várias operações como um bloco único.

4. **Bloqueio em nível de linha**: Apenas as linhas que serão modificadas por uma transação são bloqueadas

5. **Integridade referencial**: Relacionamento entre diversas tabelas do banco de dados por meio de chave estrangeira.

6.2 – Verificando InnoDB no MySQL

- O comando abaixo mostra as características das engines instaladas no MySQL:

SHOW engines\G

```
mysql> show engines \G;
***** 1. row *****
      Engine: InnoDB
      Support: DEFAULT
      Comment: Supports transactions, row-level locking, and foreign keys
Transactions: YES
          XA: YES
      Savepoints: YES
***** 2. row *****
      Engine: MRG_MYISAM
      Support: YES
      Comment: Collection of identical MyISAM tables
Transactions: NO
          XA: NO
      Savepoints: NO
***** 3. row *****
      Engine: CSV
      Support: YES
      Comment: CSV storage engine
Transactions: NO
          XA: NO
      Savepoints: NO
```

6.2 – Verificando InnoDB no MySQL

- Também é importante verificar o nível de isolamento que está sendo usado como padrão pelo MySQL. A não ser em casos muito específicos, o nível usado deve ser o **REPEATABLE-READ** (padrão no MySQL), que é totalmente ACID e possui ótimo desempenho.
- O nível **SERIALIZABLE** emula as transações sendo executadas de forma serial (sequencial), garantindo total consistência. O problema deste nível é o desempenho reduzido, logo deve ser evitado a não ser que seja extremamente necessário.

```
SHOW GLOBAL VARIABLES LIKE 'tx_isolation';
```

```
mysql> SHOW GLOBAL VARIABLES LIKE 'tx_isolation';
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| tx_isolation  | REPEATABLE-READ |
+-----+-----+
1 row in set (0,00 sec)
```

6.3 – Configurações do InnoDB

- Existem uma série de parâmetros que podem ser ajustados na engine InnoDB para que o MySQL possa atender suas necessidades do usuário.
- As configurações podem ser feitas de forma temporária, alterando as variáveis de ambiente por meio do terminal, ou de forma definitiva, alterando os parâmetros no arquivo my.cnf.

Alterar variáveis dinamicamente: Funciona apenas na seção do usuário

SET GLOBAL max_connections=130;

Alteração definitiva no my.cnf: É preciso reiniciar o servidor

GEDIT /etc/mysql/my.cnf; (versões mais antigas)

GEDIT /etc/mysql/mysql.conf.d/mysqld.cnf (versão 5.7)

6.4 – Configurações de Memória no InnoDB

- O MySQL gerencia sua memória basicamente de duas formas: Per-thread buffers e por Global buffers.

1. **Per-thread buffers:** Memória (buffer) alocados para cada conexão feita por um usuário no servidor de banco de dados. O total de memória alocado para as conexões é soma dos seguintes parâmetros (número conhecido como **MTB** – MySQL Thread Buffers):

$$\text{MTB} = (\text{read_buffer_size} + \text{read_rnd_buffer_size} + \text{sort_buffer_size} + \text{thread_stack} + \text{join_buffer_size} + \text{binlog_cache_size}) * \text{max_connections}$$

2. **Global buffers:** Memória reservada para o funcionamento geral do banco de dados, independente do número de solicitações e conexões. O **MGB** (MySQL Global Buffers) é dado:

$$\text{MGB} = \text{innodb_buffer_pool_size} + \text{innodb_additional_mem_pool_size} + \text{innodb_log_buffer_size} + \text{key_buffer_size} + \text{query_cache_size}$$

6.4 – Configurações de Memória no InnoDB

- A soma do MTB com o MGB resultam no total de memória usada pelo MySQL. Deve-se cuidar para não reservar memória acima do necessário para um determinado parâmetro de buffer, pois certamente faltará memória para outro buffer.
- É importante equilibrar a quantidade de memória para cada parâmetro, de acordo com a necessidade do usuário.
- É preciso realizar um estudo em cima do banco de dados para que um ajuste fino dos parâmetros de memória possam ser feitos.
- Boas práticas dizem que a soma do MTB com o MGB nunca devem ser maior do que **85%** do total de memória do sistema.

6.5 – Configurações do Datadir

- O diretório de dados, também conhecido como datadir, é o local onde o MySQL armazena seus dados.

Este diretório padrão é /var/lib/mysql. Os passos para trocar o datadir são:

#Cria pasta para novo datadir

```
mkdir /dados/
```

#Copia os arquivos existentes para o novo datadir

```
cp -rpai /var/lib/mysql /dados/
```

#Abre arquivo de configuração

```
GEDIT /etc/mysql/mysql.conf.d/mysqld.cnf
```

#Modifica o datadir com o novo endereço

```
datadir = /dados/mysql
```


6.5 – Configurações do Datadir

#No Ubuntu deve-se alterar o arquivo que determina todos os path's de serviço

gedit /etc/apparmor.d/usr.sbin.mysqld

#Modifica estas duas linhas

/var/lib/mysql/ r,

/var/lib/mysql/** rwk,

#Pelas linhas

/data/mysql/ r,

/data/mysql/** rwk,

#Restarta o serviço

/etc/init.d/mysql restart

6.5 – Parâmetros do MySQL

- Uma **variável estática** (ou parâmetro estático) é definida no arquivo **my.cnf** no momento de inicialização do sistema, e não pode ser mudada durante a operação dos processos do banco de dados. Já as **dinâmicas** pode ser mudadas pelo terminal com o comando **SET**. Os principais parâmetros gerais são:

1. **innodb_buffer_pool_size**: O tamanho do buffer de memória que o InnoDB usa para armazenar dados e índices de suas tabelas. Quanto maior for este valor, menor será a necessidade de E/S de disco para acessar dados na tabela. Em um servidor de banco de dados dedicado você pode definir este parâmetro até 80% do tamanho da memória física da máquina.

2. **transaction-isolation**: Define o nível de isolamento padrão do MySQL. Por padrão é o REPEATABLE-READ. Ex. transaction-isolation = SERIALIZABLE

3. **innodb_table_locks**: Permite que o innodb faça bloqueio em nível de tabela (de forma implícita ou explícita) caso necessário. Por padrão esse parâmetro é TRUE.

6.5.1 – Parâmetros de I/O

1. innodb_flush_log_at_trx_commit: Possui como valor padrão 1, mas possui como valores possíveis 0, 1 e 2. Este parâmetro controla o balanço entre 100% ACID e alto desempenho.

Para valor igual a 1, o MySQL garante consistência, pois o conteúdo do buffer de log são gravados para o arquivo de log em cada commit, para que então o arquivo de log grave os dados em disco. Apesar da segurança na perda de dados, pode gerar gargalos de I/O.

Para valor igual a 0, o MySQL gravará os dados do buffer log no arquivo de log a cada 1 segundo (aproximadamente, depende do SO), para que em seguida os dados sejam efetivados no disco. Nesse caso, é possível perder até 1 segundo de dados em caso de problemas.

Para valor igual a 2, o MySQL gravará os dados do buffer log a cada commit, mas os dados serão gravados do arquivo de log para o disco a cada 1 segundo, havendo risco de perda de dados.

2. innodb_flush_log_at_commit: Valor padrão é 1, com valores possíveis de 1 até 2700. Define o tempo (em segundos) que o arquivo de log será gravado em disco (flush log).

6.5.1 – Parâmetros de I/O

3. **innodb_io_capacity**: Valor padrão é 200. Valores possíveis de 100 até 2×10^6 . Determina o limite superior de operações de I/O feitas pelo background do InnoDB. Esse valor deve ser baseado na demanda do banco de dados e na capacidade de hardware. Caso o valor seja muito demais, comprometerá o sistema de caching. Ex: `innodb_io_capacity=200`

4. **innodb_io_capacity_max**: Em situações de emergência é necessário aumentar temporariamente o número de operações de I/O. Este parâmetro pode ser setado com um valor maior para este tipo de situação. O valor padrão é o dobro do valor de `innodb_io_capacity`, com limite mínimo de 2000. Ex: `innodb_io_capacity_max=4000`

5. **innodb_flush_sync**: Por padrão é TRUE. Este parâmetro permite que o número de operações de I/O (especificado no `innodb_io_capacity_max`) seja aumentado em emergências. Caso seja FALSE, as operações seguirão o valor máximo setado no `innodb_io_capacity`.

6.5.1 – Parâmetros de I/O

6. **innodb_thread_concurrency**: InnoDB usa threads do sistema operacional para processar transações do usuário (tenta manter o mesmo número de thread do SO). Este parâmetro define o número de threads que o InnoDB irá utilizar. Valores possíveis variam de 0 à 1000.

7. **innodb_commit_concurrency**: Define o número de threads que podem ser comitadas simultaneamente. Os valores possíveis varia de 0 à 1000, sendo 0 o padrão. O valor 0 permite que um número ilimitado de threads sejam comitadas simultaneamente.

8. **innodb_fast_shutdown**: Define a forma como o MySQL será desligado. Se o valor for 0, faz o slow shutdown (é demorado, faz flush de todos os buffers e atualiza os logs). Se o valor é 1 (padrão), faz o desligamento rápido (sem flush). Se o valor é 2, o MySQL libera seu logs e faz o chamado cold shutdown, desligando na hora, como se tivesse parado de funcionar (geralmente demora para ligar o banco novamente, pois precisa se recuperar).

6.5.1 – Parâmetros de I/O

9. **innodb_file_per_table**: Valores possíveis são 1 e 0. A partir da versão 5.6.6, este parâmetro está ativado. Com isso, cada tabela possui seu próprio tablespace.
10. **innodb_write_io_threads / innodb_read_io_threads**: Valores possíveis variam de 1 até 64. Define o número de threads que poderão ser usadas para escrita / leitura.
11. **innodb_doublewrite**: Pode estar ativado (padrão) ou desativado. Caso esteja ativado, os dados são gravados duas vezes: no buffer e depois no data dir. Caso o desempenho seja mais importante que a segurança/integridade, pode-se desabilitar este parâmetro.