

Preparação de ambiente para exercício

Para a realização do exercício, deverá ser instalado um software chamado Sysbench, que é utilizado na realização de benchmarks de bancos de dados. Além disso, deve-se criar a base de dados e inserir os dados antes do momento da aula.

Sysbench

Instalação

- Ubuntu

```
curl -s https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.sh | sudo bash
sudo apt -y install sysbench
```

- Windows

O suporte para sysbench Windows foi descartado. Para usar no Windows é usando o Windows Subsystem for Linux disponível no Windows 10.

Inicie o PowerShell como Administrador e Instale com o seguinte comando:

```
wsl --install
```

Após reinicie o computador. Inicie novamente o PowerShell como Administrador e ative o wsl com o comando:

```
wsl
```

Depois de instalar o WSL basta seguir as instruções de instalação do Debian/Ubuntu.

- Outros Ver instruções [aqui](#).

Utilização do Sysbench

No momento da aula, será disponibilizado um arquivo *consultas.lua*, com a seguinte estrutura:

```
-- Função chamada pelo Sysbench uma vez ao inicializar o script
function thread_init()
    -- Inicializa o driver MySQL do Sysbench
    drv = sysbench.sql.driver()
    -- Representa a conexão com o MySQL
    con = drv:connect()
end

-- Função chamada pelo Sysbench quando o script termina a execução
function thread_done()
    -- Fecha a conexão com o MySQL
    con:disconnect()
end

-- Função chamada pelo Sysbench para cada execução
function event()
    -- Consulta a ser executada no benchmark
    con:query("SELECT SQL_NO_CACHE COUNT(*) FROM table1;")

    -- Consulta que não será executada no benchmark por estar "comentada"
    -- con:query("SELECT SQL_NO_CACHE COUNT(*) FROM table2;")

    -- Consulta que não será executada no benchmark por estar "comentada"
    -- con:query("SELECT SQL_NO_CACHE COUNT(*) FROM table3;")

end
```

Sendo assim, todas as consultas a serem testadas durante o exercício já virão no script *consultas.lua*, e bastará "descomentar" a consulta que se deseja testar em cada benchmark. Para executar o benchmark utilizando o Sysbench, basta executar via linha de comando (estando no mesmo diretório do *consultas.lua*):

```
sudo sysbench consultas.lua \  
--db-driver=mysql --mysql-user=root \  
--mysql-password=root --mysql-db=marketplace \  
--threads=16 --time=10 run
```

Onde: - *consultas.lua* é o nome do script utilizado; - A flag `--db-driver=mysql` indica que o banco de dados utilizado é o MySQL; - As flags `--mysql-user=root`, `--mysql-password=root` e `--mysql-db=db`, representam, respectivamente, nome de usuário, senha e base de dados para executar o benchmark. - A flag `--threads=16` indica o número de threads utilizadas (quanto maior, mais consultas serão utilizadas ao mesmo tempo); - A flag `--time=10` indica o número de segundos que o benchmark irá durar.

Observação: O número de threads e o tempo de execução do benchmark não precisa necessariamente ser os do exemplo, mas é importante que durante o exercício sejam utilizados os mesmos valores nos testes realizados antes e depois das otimizações.

Importando a base de dados do exercício

Primeiramente, deve-se criar uma base de dados chamada marketplace.

```
mysql> CREATE DATABASE marketplace;  
Query OK, 1 row affected (0.92 sec)
```

Em seguida, basta executar o seguinte comando, estando no mesmo diretório do *marketplace.sql* disponibilizado:

```
mysql> source marketplace.sql
```

Testando o Sysbench com a base de dados

Para testar se a instalação do Sysbench está correta. Execute o seguinte comando estando no mesmo diretório do arquivo *teste.lua* disponibilizado:

```
sudo sysbench teste.lua --db-driver=mysql \  
--mysql-user=root --mysql-password=root \  
--mysql-db=marketplace --threads=16 --time=10 run
```

Observações: Lembre sempre de observar se o usuário root com a senha root são válidos para a sua instalação de MySQL.

Em alguns casos, pode ser necessário executar o comando como administrador (sudo).

Nesse caso, o script teste.lua executará um benchmark com a seguinte consulta:

```
SELECT SQL_NO_CACHE * FROM category;
```

O retorno da execução do Sysbench deve ser algo parecido com isso:

sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:

Number of threads: 16

Initializing random number generator from current time

Initializing worker threads...

Threads started!

SQL statistics:

queries performed:

read: 274555

write: 0

other: 0

total: 274555

transactions: 274555 (27446.62 per sec.)

queries: 274555 (27446.62 per sec.)

ignored errors: 0 (0.00 per sec.)

reconnects: 0 (0.00 per sec.)

General statistics:

total time: 10.0015s

total number of events: 274555

Latency (ms):

min: 0.28

avg: 0.58

max: 95.77

95th percentile: 0.83

sum: 159593.08

Threads fairness:

events (avg/stddev): 17159.6875/58.67

execution time (avg/stddev): 9.9746/0.00

Onde as partes de maior interesse para o contexto do exercícios serão as linhas transactions e avg, que representam: - transactions: Taxa de transferência média, exibida em número de transações por segundo. No exemplo, possui o valor de 27446.62 transações por segundo. - avg (dentro de Latency) Latência média, exibida em número de milissegundos. No exemplo, possui o valor de 0.58 ms.