



CCMP0077 – Avaliação de Desempenho de Sistemas

Relatório Técnico

Medição do tempo de resposta do Banco de Dados MariaDB com Python 3

Jose Adolfo de Castro Neto
Juazeiro-BA, 18/08/2021

Resumo: Foi implementado um programa sintético em Python 3 para medição do tempo de resposta do Banco de Dados MariaDB. O *overhead* deste programa também foi medido. O *overhead* e o tempo de resposta foram estimados executando repetições de blocos de instruções *insert*, *select* e *update*, utilizando o banco de dados MariaDB versão 10.6.3. O tempo de resposta das operações *insert*, *select* e *update* foi igual a 0.707748; 0.053272 e 0.903450 s, respectivamente. Os valores dos *overheads* referentes a estas operações podem ser desprezados.

Introdução

MariaDB é um SGBD (Sistema de Gerenciamento de Banco de Dados), que surgiu como uma ramificação do MySQL com a intenção de manter uma alta fidelidade com o MySQL só que sendo um banco de dados relacional totalmente free (MariaDB, 2021).

Na pagina oficial do banco MariaDB é possível obter o download do servidor do banco através do site: <https://downloads.mariadb.org/> . Disponível para vários sistemas operacionais. A versão utilizada para este relatório do banco foi 10.6.3 para Windows x86_64 com o tipo de pacote **MSI Package** que é um instalador para o Windows tendo 54,6MB de tamanho. Para mais detalhes sobre a instalação consulte o anexo.

O presente trabalho tem como objetivo medir o tempo de resposta do Banco de Dados MariaDB. As cargas de trabalho são operações de inserção, seleção e atualização. Para isso, foi elaborado um programa sintético na linguagem Python; também foi medido o *overhead* deste programa.

Materiais e Métodos

Foi implementado o programa sintético **Desempenho.py**, na linguagem Python, que inclui no cabeçalho quatro imports de modo que possa fazer uso da biblioteca mariadb para a conexão com o banco de dados, a biblioteca time para utilizar as

funções de tempo e a biblioteca statistics para utilizar as funções de média e desvio padrão.

O programa Desempenho.py, anexo, utiliza a tabela descrita na Tabela 1 e foi configurado para executar cinco mil de operações insert, select e update, divididas em 10 blocos de 500 operações. A duração média de cada bloco destas operações, medida em segundos, foi gravada em um arquivo-texto resumo.txt localizado na pasta DADOS.

Tabela 1 – Estrutura da tabela do banco de dados MariaDB com campo, tipo, valor padrão e chave primaria.

Campo	Tipo	Valor Padrão	
A0	INT	NULL	AUTO_INCREMENT
A1	INT	NULL	
A2	FLOAT	NULL	
A3	NVARCHAR(26)	NULL	
A4	VARCHAR(26)	NULL	
A5	TIME	CURRENT_TIME NULL	
A6	DATE	CURRENT_DATE NULL	
PRIMARY KEY (A0)			

A estimativa do tempo de resposta das operações de inserção, seleção e atualização, foi feita por meio da media aritmética da duração dos blocos de cada uma destas operações. Também foi obtido o desvio padrão do tempo de resposta de cada uma destas operações.

A estimativa do *overhead* do programa Desempenho.py foi feita comentando as instruções **cur.execute()** dentro das 3 funções def Insert(), def Select() e def Update(). A configuração adotada foi a mesma para a obtenção do tempo de resposta. Os valores médios do *overhead* das operações de inserção, seleção e atualização, foi feita por meio da media aritmética da duração dos blocos de cada uma destas operações.

O software Desempenho.py possui funções para fazer a conexão com banco de dados MariaDB. Ele foi executado em um notebook, usando o sistema operacional Windows 10, CPU Intel[®] Core™ i5-8265U CPU@ 1,60 GHz e 16GB de RAM DDR4 2400 MHz, usando o terminal do Windows no diretório do programa, com o comando python3 Desempenho.py.

Resultados e Discussão

O software Desempenho.py possui as funções def Insert(), def Select() e def Update() para calcular a duração das operações *insert*, *select* e *update*, respectivamente.

Estas funções também permitem calcular a duração do *overhead* do programa bastando apenas comentar as funções **cur.execute()** dentro das 3 funções def Insert(), def Select() e def Update().

O Desempenho.py foi configurado para fazer 10 repetições (RE=10) de 500 blocos de *insert* (BI=500), 500 blocos de *select* (BS=500) e 500 blocos de *update* (BU=500). Para calcular a duração dos blocos de cada operações, este programa utiliza a biblioteca time, disponível no Python 3.

A Tabela 2 apresenta a duração média dos blocos das operações *insert*, *select* e *update*, em segundos, organizadas por bloco de 500 operações; e seus respectivos valores médios e desvio-padrão.

A duração média dos blocos referentes à operação *insert* variou de 0.712060690 a 0.774371386 s, com média aritmética igual a 0.748676157s e desvio-padrão de 0.020862534 s.

A duração média dos blocos referentes à operação *select* variou de 0.050863028 a 0.055257797 s, com média aritmética igual a 0.052867126 s e desvio-padrão de 0.001497508 s.

A duração média dos blocos referentes à operação *update* variou de 0.711958647 a 0.816478491 s, com média aritmética igual a 0.749223185 s e desvio-padrão de 0.029932726 s.

Tabela 2 – Duração média dos blocos das operações *insert*, *select* e *update*, em segundos, organizadas por bloco de 500 operações; e seus respectivos valores médios e desvio-padrão

Bloco (x500)	Insert (s)	Select (s)	Update (s)
1	0.712060690	0.053856134	0.762755394
2	0.774371386	0.050863028	0.736161232
3	0.750755072	0.055257797	0.729774952
4	0.720533848	0.051848888	0.753860474
5	0.758156300	0.054907560	0.816478491
6	0.730562925	0.051888227	0.747424364
7	0.760778427	0.051860571	0.745075226
8	0.749106169	0.053829432	0.718576193
9	0.763211250	0.052643776	0.770166874
10	0.767225504	0.051715851	0.711958647
m	0.748676157	0.052867126	0.749223185
s	0.020862534	0.001497508	0.029932726

A Tabela 3 apresenta a duração média dos *overhead* dos blocos referentes às operações *insert*, *select* e *update*, em segundos, organizadas por bloco de 500 operações; e seus respectivos valores médios

A duração média dos *overhead* dos blocos referentes à operação *insert* variou de 0.000000000 a 0.001000166 s, com média aritmética igual a 0.000100017 s e desvio-padrão de 0.000316280 s.

A duração média dos *overhead* dos blocos referentes à operação *select* variou de 0.000000000 a 0.000975370 s, com média aritmética igual a 0.000097537s e desvio-padrão de 0.000308439 s.

A duração média dos *overhead* dos blocos referentes à operação *update* variou de 0.000000000 a 0.000996590 s, com média aritmética igual a 0.000099659 s e desvio-padrão de 0.000315149 s.

Tabela 3 – Duração média dos *overhead* dos blocos referentes às operações *insert*, *select* e *update*, em segundos, organizadas por bloco de 500 operações; e seus respectivos valores médios

Bloco (x500)	Insert (s)	Select (s)	Update (s)
1	0.000000000	0.000000000	0.000000000
2	0.000000000	0.000000000	0.000000000
3	0.000000000	0.000000000	0.000996590
4	0.000000000	0.000000000	0.000000000
5	0.000000000	0.000000000	0.000000000
6	0.000000000	0.000975370	0.000000000
7	0.000000000	0.000000000	0.000000000
8	0.001000166	0.000000000	0.000000000
9	0.000000000	0.000000000	0.000000000
10	0.000000000	0.000000000	0.000000000
m	0.000100017	0.000097537	0.000099659
s	0.000316280	0.000308439	0.000315149

Os valores dos *overheads* referentes às operações *insert*, *select* e *update* são muito menores que seus respectivos tempos de reposta, e podem ser desprezados.

Conclusão

Foi implementado um programa sintético em Python 3 para medição do tempo de resposta do Banco de Dados MariaDB para as operações de inserção, seleção e atualização.

O *overhead* deste programa também foi medido para estas operações. O tempo de resposta das operações *insert* e *update* foi comparável, com valores da ordem de décimos de segundo. Já o tempo de resposta da operação *select* foi menor do que os demais, com valores da ordem de centésimos de segundo, dez vezes menor.

O tempo de resposta das operações *insert*, *select* e *update* foi igual a 0.748676157; 0.052867126 e 0.749223185 s, respectivamente. Os valores dos *overheads* referentes às operações *insert*, *select* e *update* podem ser desprezados já que estão na ordem de microssegundos.

Referências

How to connect Python programs to MariaDB. Disponível em:

<https://mariadb.com/resources/blog/how-to-connect-python-programs-to-mariadb/>

Acesso:18/08/2021

Funções estatísticas. Disponível em:

<https://docs.python.org/pt-br/3/library/statistics.html#statistics.fmean>

Acesso:18/08/2021

Download do banco MariaDB. Disponível em :

<https://downloads.mariadb.org/>

Acesso:18/08/2021

Anexo

```
# module Imports
import mariadb
import sys
import time
import statistics

# connect to MariaDB Platform
try:
    conn = mariadb.connect(
        user = "root",
        password = "123456",
        host = "127.0.0.1",
        port = 3306,
        database = "bdteste"
    )
except mariadb.Error as e:
    print(f"Error connection to MariaDB Platform:{e}")
    sys.exit(1)

# Get Cursor

RE = 10 # repeticao
BI = 500 # blocos de insert
BS = 500 # blocos de select
BU = 500 # blocos de update

def Insert():
    arquivo.write("#####")
    arquivo.write("\nTempo Blocos Insert em segundos:\n")
    print("\n#####")
    print("Tempo Blocos Insert em segundos:\n")
    vetor = []
    for r in range(RE):

        inicio = time.time()
        for b in range(BI):
            cur = conn.cursor()
            cur.execute(''' INSERT INTO A(A1,A2,A3,A4,A5,A6)
VALUES(1,1.2,'abcdefghijklmnopqrstuvxz','abcdefghijklmnopqrstuvxz','15:25:22','2021-05-11');''')
            conn.commit()
        fim = time.time()
        tempo = fim - inicio
        vetor.append(tempo) # Adiciona no final do vetor o valor tempo

    arquivo.write("\n%.1.9f" % tempo) # Escreve os valores de tempo no arquivo
    print("Media Bloco Insert", r, "%.1.9f" % tempo, "s")

    Media = statistics.fmean(vetor)
    DesvioPadrao = statistics.stdev(vetor)
    arquivo.write("\n\nMedia : %.1.9f" % Media) # Escreve a media no arquivo
    arquivo.write("\nDesvio Padrao: %.1.9f" % DesvioPadrao) # Escreve o desvio padrão no arquivo
```

```

print("\nMedia          : %1.9f" % Media,"s")          # Escreve a media no Terminal
print("Desvio Padrao: %1.9f" % DesvioPadrao,"s")      # Escreve o desvio padrão no Terminal

def Select():
    arquivo.write("\n\n#####")
    arquivo.write("\nTempo Blocos Select em segundos:\n")
    print("\n\n#####")
    print("Tempo Blocos Select em segundos:\n")
    vetor = []
    c = 1
    for r in range(RE):

        inicio = time.time()
        for b in range(BS):
            cur = conn.cursor()
            cur.execute( "SELECT * from A where A0=%d" % c )
            c += 1
        fim = time.time()
        tempo = fim - inicio
        vetor.append(tempo)                                # Adiciona no final do vetor o valor tempo
        arquivo.write("\n%1.9f" % tempo)                  # Escreve os valores de tempo no arquivo
        print( "Media Bloco Select", r,"%1.9f" % tempo,"s")
    Media = statistics.fmean(vetor)
    DesvioPadrao = statistics.stdev(vetor)
    arquivo.write("\n\nMedia          : %1.9f" % Media)    # Escreve a media no arquivo
    arquivo.write("\nDesvio Padrao: %1.9f" % DesvioPadrao) # Escreve o desvio padrão no arquivo

    print("\nMedia          : %1.9f" % Media,"s")          # Escreve a media no Terminal
    print("Desvio Padrao: %1.9f" % DesvioPadrao,"s")      # Escreve o desvio padrão no Terminal

def Update():
    arquivo.write("\n\n#####")
    arquivo.write("\nTempo Blocos Update em segundos:\n")
    print("\n\n#####")
    print("Tempo Blocos Update em segundos:\n")
    vetor = []
    c = 1
    for r in range(RE):

        inicio = time.time()
        for b in range(BU):
            cur = conn.cursor()
            cur.execute( "UPDATE A SET A1=9, A2=2.222, A3='ABCDEFGHIJKLMN0PQRSTUVWXYZ', A4='ABCDEFGHIJKLMNO
PQRSTUVWXYZ', A5= '16:25:22', A6='2031-05-11' where A0=%d" % c )
            conn.commit()
            c += 1
        fim = time.time()
        tempo = fim - inicio
        vetor.append(tempo)                                # Adiciona no final do vetor o valor temp
o
        arquivo.write("\n%1.9f" % tempo)                  # Escreve os valores de tempo no arquivo
        print( "Media Bloco Update", r,"%1.9f" % tempo,"s")
    Media = statistics.fmean(vetor)
    DesvioPadrao = statistics.stdev(vetor)
    arquivo.write("\n\nMedia          : %1.9f" % Media)    # Escreve a media no arquivo
    arquivo.write("\nDesvio Padrao: %1.9f" % DesvioPadrao) # Escreve o desvio padrão no arquivo

```

```

        print("\nMedia          : %1.9f" % Media,"s")          # Escreve a media no Terminal
        print("Desvio Padrao: %1.9f" % DesvioPadrao,"s")      # Escreve o desvio padrão no Terminal

# INICIO

cur = conn.cursor()
cur.execute("DROP TABLE IF EXISTS A;") # Apaga a tabela A

cur.execute(
    '''CREATE TABLE A(
        A0 INT          NULL      AUTO_INCREMENT ,
        A1 INT          NULL      ,
        A2 FLOAT        NULL      ,
        A3 NVARCHAR(26) NULL      ,
        A4 VARCHAR(26)   NULL      ,
        A5 TIME DEFAULT CURRENT_TIME NULL ,
        A6 DATE DEFAULT CURRENT_DATE NULL  ,
        PRIMARY KEY (A0));
    ''')
)

arquivo = open('DADOS/resumo.txt','w') # Cria um arquivo chamado resumo.txt na pasta DADOS
arquivo.write("#####\n#####")
arquivo.write("\n***  RELATORIO DO PROGRAMA PARA MEDIR O TEMPO DE RESPOSTA DO BANCO DE DADOS MARIADB  ***\n")
arquivo.write("\n Autor: Jose Adolfo de Castro Neto")
arquivo.write("\n Data : 18/08/2021")
arquivo.write("\n\n QUANTIDADE DE REPETIÇÕES PARA CADA BLOCO: %d" % RE)
arquivo.write("\n QUANTIDADE DE BLOCOS DE INSERT          : %d" % BI)
arquivo.write("\n QUANTIDADE DE BLOCOS DE SELECT          : %d" % BS)
arquivo.write("\n QUANTIDADE DE BLOCOS DE UPDATE          : %d" % BU)
arquivo.write("\n#####\n#####\n\n")

print("\n#####\n###")
print("\n***  RELATORIO DO PROGRAMA PARA MEDIR O TEMPO DE RESPOSTA DO BANCO DE DADOS MARIADB  ***\n")
print(" Autor: Jose Adolfo de Castro Neto")
print(" Data : 18/08/2021")
print("\n\n QUANTIDADE DE REPETIÇÕES PARA CADA BLOCO: %d" % RE)
print(" QUANTIDADE DE BLOCOS DE INSERT          : %d" % BI)
print(" QUANTIDADE DE BLOCOS DE SELECT          : %d" % BS)
print(" QUANTIDADE DE BLOCOS DE UPDATE          : %d" % BU)
print("#####\n#\n\n")

Insert()
Select()
Update()
arquivo.close() # Fecha o arquivo
conn.close()    # Fecha a conexão com o banco de dados MariaDB

```


Biblioteca: mariadb, time e statistics.

Instalação:

Comando para instalação da biblioteca mariadb no terminal do Windows:

```
pip3 install mariadb
```

Comando para instalação da biblioteca time no terminal do Windows:

```
pip3 install time
```

Comando para instalação da biblioteca statistics no terminal do Windows:

```
pip3 install statistics
```

Video Aula - Baixando e Instalando o MariaDB Server:

https://www.youtube.com/watch?v=oBjs4Odl-BE&ab_channel=FernandoRobertoProen%C3%A7a

Video Aula - Criando um Banco de Dados e uma Tabela

https://www.youtube.com/watch?v=zsa1K4nDxBw&t=1s&ab_channel=FernandoRobertoProen%C3%A7a

A senha do root utilizada foi 123456 com o nome do banco sendo bdteste. Para que o programa Desempenho.py funcione é necessário antes criar esse banco de dados na ferramenta fornecida já na instalação do servidor mariaDB com o nome bdteste. Abaixo estão as especificações para a conexão com o banco.

```
user = "root",  
password = "123456",  
host = "127.0.0.1",  
port = 3306,  
database = "bdteste"
```