INF1406 Programação Distribuída e Concorrente
2021.2

PUC
RIO

Relatório de testes - Trabalho 2 - Parte 2

Professora: Noemi Rodriguez
Grupo: Luiza Del Negro Ciuffo Fernandes - 1721251
Lucas Rebello Damo - 1721275

Teste 1

Parâmetros:

Quantidade de clientes total: 5

Quantidade máxima de conexões com cliente: 10

Tempo de timeout: 5 segundos

Tempo máximo de ausência do cliente: 20 segundos

Tempo de sleep do cliente: 1 segundo

Output:

```
Creating client 0
Creating client 2
Creating client 1
Creating client 4
Client 0 - cliente: conectando a 127.0.0.1
Client 5 : sending random number 91122193
selectserver: new connection from 127.0.0.1 on socket 4
SERVER: Abrindo  arquivo file5.dat para o client 4
Client 2 - cliente: conectando a 127.0.0.1
Client 2 : sending random number 29468330
selectserver: new connection from 127.0.0.1 on socket 6
SERVER: Abrindo  arquivo file2.dat para o client 6
Client 1 - cliente: conectando a 127.0.0.1
Client 1 : sending random number 168188146
selectserver: new connection from 127.0.0.1 on socket 8
Client 2 - Received ready.
Client 0 - Received ready.
SERVER: Abrindo  arquivo file1.dat para o client 8
Creating client 3
Client 1 - Received ready.
Client 4 - cliente: conectando a 127.0.0.1
selectserver: new connection from 127.0.0.1 on socket 10
Client 9 : sending random number 183402820
SERVER: Abrindo  arquivo file9.dat para o client 10
Client 4 - Received ready.
Client 3 - cliente: conectando a 127.0.0.1
Client 2 : sending random number 214167394
selectserver: new connection from 127.0.0.1 on socket 12
SERVER: Abrindo  arquivo file2.dat para o client 12
Client 3 - Received ready.
FINISHED - EXITING CLIENT 4
FINISHED - EXITING CLIENT 6
FINISHED - EXITING CLIENT 10
FINISHED - EXITING CLIENT 12
--- CLIENT 0: Finished reading file contents:
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium
doloremque laudantium, totam rem aperiam, eaque ipsa.
----------------Exited client 0
--- CLIENT 2: Finished reading file contents:
Thee naps all day long, thy cat, dog hate, mouse eat, string barf pillow no
baths hates everything but kitty poochy. Cat sleeps.
----------------Exited client 2
```

```
FINISHED - EXITING CLIENT 8
--- CLIENT 4: Finished reading file contents:
abc def ghi jkl mno pqrs tuv wxyz ABC DEF GHI JKL MNO PQRS TUV WXYZ !xx $%&
/() =?* '<> #|; 23~ @`´ csc 34× {} abc def ghi jk.
----------------Exited client 4
--- CLIENT 3: Finished reading file contents:
Thee naps all day long, thy cat, dog hate, mouse eat, string barf pillow no
baths hates everything but kitty poochy. Cat sleeps.
----------------Exited client 3
--- CLIENT 1: Finished reading file contents:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus urna urna,
luctus eu lectus in, aliquam elementum velit. Nulla rhoncus tellus urna,
eget pretium risus luctus interdum. Pellentesque aliquet vestibulum odio at
luctus. Donec sodales pharetra.
----------------Exited client 1
SERVER: Timeout!
SERVER: Timeout!
```

Observações:
Teste inicial, ocorreu como esperado, todos os clientes foram atendidos, mas o servidor não está sendo muito testado.
(Continua em conclusões finais)

Teste 2

Parâmetros:

- Quantidade de clientes total: 10
- Quantidade máxima de conexões com cliente: 10
- Tempo máximo de ausência do cliente: 20 segundos
- Tempo de timeout: 5
- Tempo de sleep do cliente: 1 segundo

Output:
observação - para facilitar a leitura, diminuímos o conteúdo dos logs, mas eles seguem o mesmo formato demonstrado no teste1.

```
Creating client 0
Creating client 1
Creating client 6
Client 1 - cliente: conectando a 127.0.0.1
selectserver: new connection from 127.0.0.1 on socket 4
Client 8 : sending random number 208217087
selectserver: new connection from 127.0.0.1 on socket 5
SERVER: Abrindo  arquivo file8.dat para o client 5
Creating client 2
Client 1 - Received ready.
Client 6 - cliente: conectando a 127.0.0.1
selectserver: new connection from 127.0.0.1 on socket 7
Client 4 : sending random number 56352295
Client 0 - cliente: conectando a 127.0.0.1
Client 4 : sending random number 24935797
SERVER: Abrindo  arquivo file4.dat para o client 7
```

```
Client 6 - Received ready.
SERVER: Abrindo  arquivo file4.dat para o client 4
Client 0 - Received ready.
Client 2 - cliente: conectando a 127.0.0.1
Client 6 : sending random number 178446441
selectserver: new connection from 127.0.0.1 on socket 10
SERVER: Abrindo  arquivo file6.dat para o client 10
Client 2 - Received ready.
Creating client 3
Client 3 - cliente: conectando a 127.0.0.1
Client 1 : sending random number 41096052
selectserver: new connection from 127.0.0.1 on socket 12
SERVER: Abrindo  arquivo file1.dat para o client 12
Client 3 - Received ready.
Creating client 4
Client 4 - cliente: conectando a 127.0.0.1
Client 6 : sending random number 117504543
selectserver: new connection from 127.0.0.1 on socket 14
SERVER: Abrindo  arquivo file6.dat para o client 14
Client 4 - Received ready.
Creating client 5
Client 5 - cliente: conectando a 127.0.0.1
Client 9 : sending random number 87297413
selectserver: new connection from 127.0.0.1 on socket 16
SERVER: Abrindo  arquivo file9.dat para o client 16
Client 5 - Received ready.
Creating client 7
Client 7 - cliente: conectando a 127.0.0.1
Client 1 : sending random number 26166638
selectserver: new connection from 127.0.0.1 on socket 18
SERVER: Abrindo  arquivo file1.dat para o client 18
Client 7 - Received ready.
Creating client 8
Client 8 - cliente: conectando a 127.0.0.1
Client 7 : sending random number 103961514
selectserver: new connection from 127.0.0.1 on socket 20
SERVER: Abrindo  arquivo file7.dat para o client 20
Client 8 - Received ready.
Creating client 9
Client 9 - cliente: conectando a 127.0.0.1
Client 8 : sending random number 180351500
selectserver: new connection from 127.0.0.1 on socket 22
SERVER: Abrindo  arquivo file8.dat para o client 22
Client 9 - Received ready.
FINISHED - EXITING CLIENT 5
FINISHED - EXITING CLIENT 4
FINISHED - EXITING CLIENT 7
FINISHED - EXITING CLIENT 10
FINISHED - EXITING CLIENT 14
FINISHED - EXITING CLIENT 16
FINISHED - EXITING CLIENT 20
FINISHED - EXITING CLIENT 22
--- CLIENT 1: Finished reading file contents:(...)
----------------Exited client 1
--- CLIENT 6: Finished reading file contents:(...)
```

```
----------------Exited client 6
--- CLIENT 0: Finished reading file contents:(...)
----------------Exited client 0
--- CLIENT 2: Finished reading file contents:(...)
----------------Exited client 2
FINISHED - EXITING CLIENT 12
--- CLIENT 4: Finished reading file contents:(...)
----------------Exited client 4
--- CLIENT 5: Finished reading file contents:(...)
----------------Exited client 5
FINISHED - EXITING CLIENT 18
--- CLIENT 8: Finished reading file contents:(...)
----------------Exited client 8
--- CLIENT 9: Finished reading file contents:(...)
----------------Exited client 9
--- CLIENT 3: Finished reading file contents:(...)
----------------Exited client 3
--- CLIENT 7: Finished reading file contents:(...)
----------------Exited client 7
SERVER: Timeout!
SERVER: Timeout!
SERVER: Timeout!
SERVER: Timeout!
```

Observações:

Agora com a quantidade máxima que o servidor consegue atender. Funciona como o esperado. Pelos tempos baixos de espera e pela quantidade de clientes, nenhum é derrubado por estar antigo demais.

Teste 3

Parâmetros:

Quantidade de clientes total: 15

Quantidade máxima de conexões com cliente: 10

Tempo máximo de ausência do cliente: 20 segundos

Tempo de timeout: 5

Tempo de sleep do cliente: 1

Output:

```
./c
Creating client 0
Creating client 1
Creating client 2
Creating client 3
Client 0 - cliente: conectando a 127.0.0.1
Client 0 : sending random number 6
selectserver: new connection from 127.0.0.1 on socket 4
SERVER: Abrindo  arquivo file6.dat para o client 4
Client 0 - Received ready.
Client 1 - cliente: conectando a 127.0.0.1
```

```
Client 1 : sending random number 8
selectserver: new connection from 127.0.0.1 on socket 6
Client 2 - cliente: conectando a 127.0.0.1
Creating client 8
Client 2 : sending random number 0
selectserver: new connection from 127.0.0.1 on socket 7
SERVER: Abrindo  arquivo file8.dat para o client 6
SERVER: Abrindo  arquivo file0.dat para o client 7
Client 2 - Received ready.
Client 3 - cliente: conectando a 127.0.0.1
Creating client 7
selectserver: new connection from 127.0.0.1 on socket 10
Client 3 : sending random number 2
Client 1 - Received ready.
Creating client 9
SERVER: Abrindo  arquivo file2.dat para o client 10
Client 3 - Received ready.
Creating client 14
selectserver: new connection from 127.0.0.1 on socket 12
Client 8 - cliente: conectando a 127.0.0.1
Client 8 : sending random number 3
SERVER: Abrindo  arquivo file3.dat para o client 12
Client 8 - Received ready.
Client 7 - cliente: conectando a 127.0.0.1
Client 7 : sending random number 3
Creating client 6
selectserver: new connection from 127.0.0.1 on socket 14
SERVER: Abrindo  arquivo file3.dat para o client 14
Client 9 - cliente: conectando a 127.0.0.1
selectserver: new connection from 127.0.0.1 on socket 16
Client 9 : sending random number 5
Client 7 - Received ready.
Creating client 10
SERVER: Abrindo  arquivo file5.dat para o client 16
Client 14 - cliente: conectando a 127.0.0.1
selectserver: new connection from 127.0.0.1 on socket 18
Client 14 : sending random number 7
Client 9 - Received ready.
SERVER: Abrindo  arquivo file7.dat para o client 18
Creating client 11
Client 14 - Received ready.
Creating client 4
Client 6 - cliente: conectando a 127.0.0.1
Client 6 : sending random number 4
selectserver: new connection from 127.0.0.1 on socket 20
SERVER: Abrindo  arquivo file4.dat para o client 20
Client 6 - Received ready.
Creating client 12
Client 10 - cliente: conectando a 127.0.0.1
Client 10 : sending random number 5
Client 11 - cliente: conectando a 127.0.0.1
Client 11 : sending random number 2
Creating client 13
Client 4 - cliente: conectando a 127.0.0.1
Client 4 : sending random number 9
```

```
Creating client 5
Client 12 - cliente: conectando a 127.0.0.1
Client 12 : sending random number 3
selectserver: new connection from 127.0.0.1 on socket 22
SERVER: Reached maximum simultaneous clients connected 10!
SERVER: Abrindo  arquivo file5.dat para o client 22
SERVER: Reached maximum simultaneous clients connected 10!
SERVER: Reached maximum simultaneous clients connected 10!
Client 10 - Received ready.
Client 13 - cliente: conectando a 127.0.0.1
Client 13 : sending random number 1
SERVER: Reached maximum simultaneous clients connected 10!
Client 5 - cliente: conectando a 127.0.0.1
SERVER: Reached maximum simultaneous clients connected 10!
Client 5 : sending random number 9
FINISHED - EXITING CLIENT 4
FINISHED - EXITING CLIENT 7
FINISHED - EXITING CLIENT 6
FINISHED - EXITING CLIENT 10
FINISHED - EXITING CLIENT 16
FINISHED - EXITING CLIENT 18
FINISHED - EXITING CLIENT 20
FINISHED - EXITING CLIENT 22
--- CLIENT 2: Finished reading file contents:(...)
----------------Exited client 2
--- CLIENT 0: Finished reading file contents:(...)
----------------Exited client 0
--- CLIENT 1: Finished reading file contents:(...)
----------------Exited client 1
--- CLIENT 3: Finished reading file contents:(...)
----------------Exited client 3
FINISHED - EXITING CLIENT 12
FINISHED - EXITING CLIENT 14
--- CLIENT 9: Finished reading file contents:(...)
----------------Exited client 9
--- CLIENT 14: Finished reading file contents:(...)
----------------Exited client 14
--- CLIENT 6: Finished reading file contents:(...)
----------------Exited client 6
--- CLIENT 10: Finished reading file contents:(...)
----------------Exited client 10
--- CLIENT 8: Finished reading file contents:(...)
----------------Exited client 8
--- CLIENT 7: Finished reading file contents:(...)
----------------Exited client 7
SERVER: Timeout!
```

Observações:
Nesse momento, usamos mais clientes do que o servidor pode conectar. Como podemos ver, ele não se conecta com os excedentes, mas pela alta tolerância de tempo de conexão com o cliente, nenhum servidor é forçado a fechar.

Teste 4

Parâmetros:

Quantidade de clientes total: 5
Quantidade máxima de conexões com cliente: 10
Tempo de timeout: 3
Tempo máximo de ausência do cliente: 5 segundos
Tempo de sleep do cliente: 10

Output:

```
Creating client 0
Creating client 1
Creating client 2
Creating client 3
Client 0 - cliente: conectando a 127.0.0.1
Client 0 : sending random number 6
Creating client 4
Client 2 - cliente: conectando a 127.0.0.1
Client 2 : sending random number 7
selectserver: new connection from 127.0.0.1 on socket 4
Checking old conn
Time 0  max_age=5
selectserver: new connection from 127.0.0.1 on socket 5
SERVER: Abrindo  arquivo file6.dat para o client 4
Checking old conn
Client 0 - Received ready.
Time 0  max_age=5
Time 0  max_age=5
SERVER: Abrindo  arquivo file7.dat para o client 5
Checking old conn
Client 2 - Received ready.
Time 0  max_age=5
Time 0  max_age=5
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Client 3 - cliente: conectando a 127.0.0.1
Client 3 : sending random number 9
selectserver: new connection from 127.0.0.1 on socket 8
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
SERVER: Abrindo  arquivo file9.dat para o client 8
Checking old conn
Time 0  max_age=5
Client 3 - Received ready.
Time 0  max_age=5
Time 0  max_age=5
```

```
Client 4 - cliente: conectando a 127.0.0.1
Client 4 : sending random number 9
selectserver: new connection from 127.0.0.1 on socket 10
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
SERVER: Abrindo  arquivo file9.dat para o client 10
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Client 4 - Received ready.
Time 0  max_age=5
Client 1 - cliente: conectando a 127.0.0.1
selectserver: new connection from 127.0.0.1 on socket 12
Client 1 : sending random number 0
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
SERVER: Abrindo  arquivo file0.dat para o client 12
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Client 1 - Received ready.
Time 0  max_age=5
Time 0  max_age=5
Checking old conn
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5
Time 0  max_age=5

SERVER: Timeout!
Checking old conn
Time 3  max_age=5
Time 3  max_age=5
Time 3  max_age=5
Time 3  max_age=5
Time 3  max_age=5
SERVER: Timeout!
Checking old conn
Time 6  max_age=5
Dropping old connection 4
```

```
Time 6  max_age=5
Dropping old connection 5
Time 6  max_age=5
Dropping old connection 8
Time 6  max_age=5
Dropping old connection 10
Time 6  max_age=5
Dropping old connection 12
```

Observações:

Para forçar as conexões a serem finalizadas por tempo excedido, tivemos que diminuir os intervalos de tempo de tolerância, timeout e aumentar o tempo de sleep.

Considerações finais:

A motivação para a utilização de concorrência com um único processo é o baixo custo da criação de processos, dividir informações entre todas as conexões. A "concorrência aparente" entre processos que compartilham memória pode ser obtida se o total de requests para o servidor não excede sua capacidade de lidar com eles.

Em uma concorrência real, o servidor tem mais de um processo para lidar com cada nova conexão, então o sistema operacional é o responsável pelo slicing dividindo a CPU entre os processos. No nosso caso, com um único processo, não tem tanta troca de contextos, tendo essa vantagem, mas em contrapartida temos mais lentidão e dificuldade de administrar as requisições.

Ao analisar o funcionamento do atendimento concorrente com requisições de vários clientes, percebemos que os tempos analisados são bastante importantes, pois podem retirar clientes antigos, dando espaço para novos clientes e requisições, mas essa perda também significa perda de informações trocadas.