

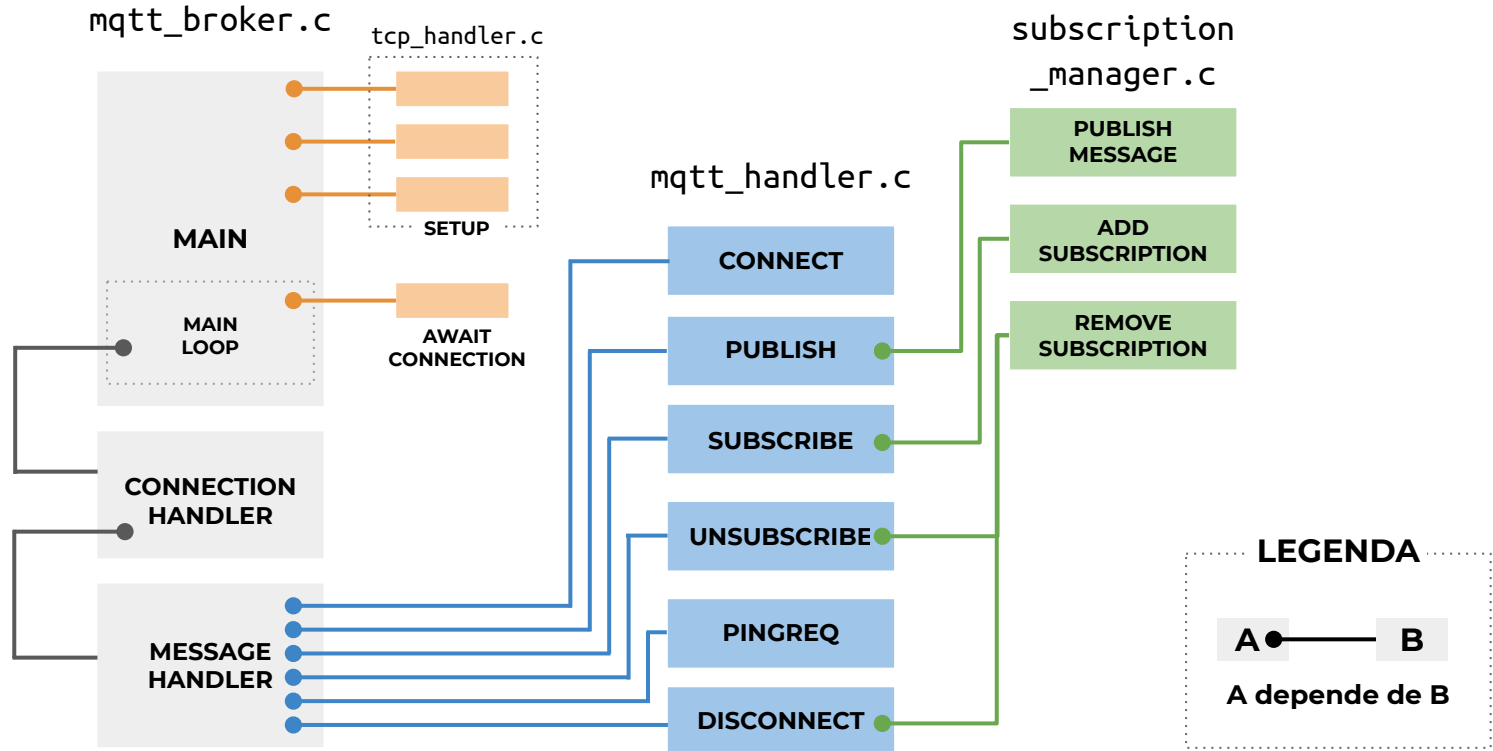
MAC0352 - Redes de Computadores e Sistemas Distribuídos

EP1 - Servidor MQTT

Andrew Ijano Lopes

2021

Estrutura geral



Estrutura de inscrições

A estrutura de inscrições é um array global de tuplas (*tópico, socket*).

Para encontrar uma tupla específica, é necessário percorrer todo o array.

Para remover uma tupla dado seu índice, é feita uma troca com o último elemento e o tamanho do array é decrementado.

subscriptions[]

{ topic, connfd }

...

subscriptions_size



Comandos implementados

CONNECT

- Envia um pacote CONNACK para o cliente.

PUBLISH

- Publica o pacote para todos os clientes inscritos no tópico.

SUBSCRIBE

- Adiciona par (*tópico, socket*) na lista de inscritos e envia pacote SUBACK para o cliente.

PING

- Envia um pacote PINGRESP para o cliente.

UNSUBSCRIBE

- Remove par (*tópico, socket*) da lista de inscritos e envia pacote UNSUBACK para o cliente.

DISCONNECT

- Remove par (*tópico, socket*) da lista de inscritos.

Decisões de projeto

Cada nova conexão é tratada em uma *thread* diferente:

- Biblioteca `pthread.h`
- Permite memória compartilhada
- Uso de mutex para manipulação da estrutura de inscrições

Supõe-se que um cliente só se conecta a um tópico por conexão criada.

Funcionamento

```
client 9 with message type DISCONNECT  
connection 9 closed
```

```
connection 8 open  
client 8 with message type CONNECT  
client 8 with message type PUBLISH  
-- publishing in topic 'test' to subscriber 4  
-- publishing in topic 'test' to subscriber 5  
-- publishing in topic 'test' to subscriber 6  
-- publishing in topic 'test' to subscriber 7  
client 8 with message type DISCONNECT  
connection 8 closed
```

```
❏ > ~ mosquitto_pub -h localhost -m "Hello" -t test -V mqttv5 -p 1883  
❏ > ~ mosquitto_pub -h localhost -m "Hello" -t test -V mqttv5 -p 1883  
❏ > ~ mosquitto_pub -h localhost -m "Hello" -t test -V mqttv5 -p 1883  
❏ > ~ mosquitto_pub -h localhost -m "Hello" -t test -V mqttv5 -p 1883  
❏ > ~ mosquitto_pub -h localhost -m "Hello" -t test -V mqttv5 -p 1883
```

```
❏ > ~ mosquitto_sub -h localhost -t test -V mqttv5 -p 1883 ✓ < 41s
```

```
Hello  
Hello  
Hello  
Hello  
Hello
```

```
❏ > ~ mosquitto_sub -h localhost -t test -V mqttv5 -p 1883 ✓ < 14s
```

```
Hello  
Hello  
Hello  
Hello  
Hello
```

```
❏ > ~ mosquitto_sub -h localhost -t test -V mqttv5 -p 1883 ✓ < 35s
```

```
Hello  
Hello  
Hello  
Hello  
Hello
```

```
❏ > ~ mosquitto_sub -h localhost -t test -V mqttv5 -p 1883 ✓ < 19s
```

```
Hello  
Hello  
Hello  
Hello  
Hello
```

Análise de desempenho

0. AMBIENTE DE TESTES

Para isolar os testes, o servidor foi executado localmente num container docker.

As medidas no container foram feitas usando a ferramenta cAdvisor da Google.

Ambiente computacional:

- CPU Intel Core i7-6500U @ 2.50GHz × 4
- SO Manjaro Linux



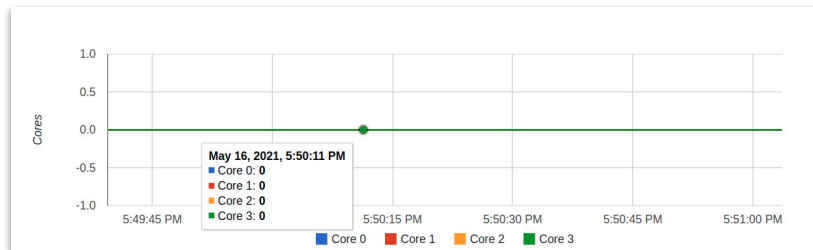
Análise de desempenho

1. APENAS O SERVIDOR

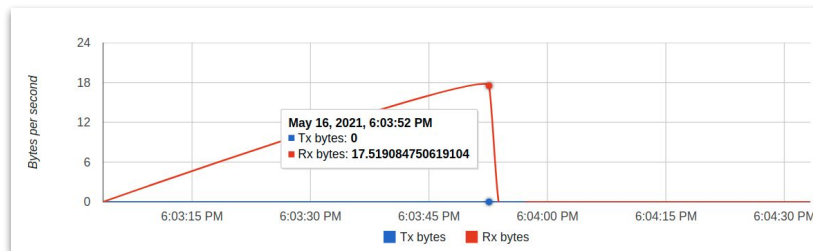
O consumo de CPU foi insignificante, perto de 0.

O uso da rede também foi perto de 0, com alguns poucos bytes sendo recebidos periodicamente.

Consumo de CPU



Uso de rede (*throughput*)



Análise de desempenho

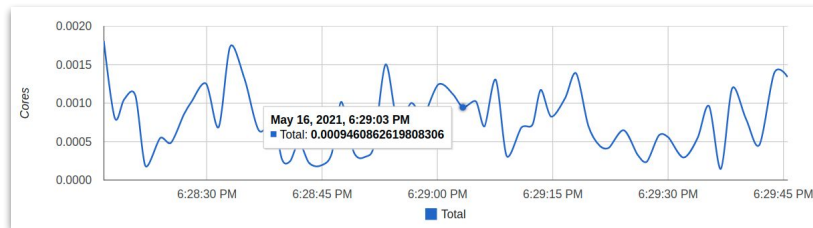
2. SERVIDOR E DOIS CLIENTES

Um cliente é inscrito e outro publica uma mensagem por segundo.

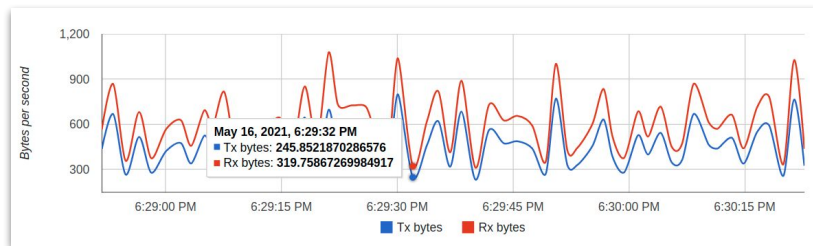
Consumo de até 0,7% de CPU.

Até 1k bytes transferidos por segundo pela rede.

Consumo de CPU



Uso de rede (*throughput*)



Análise de desempenho

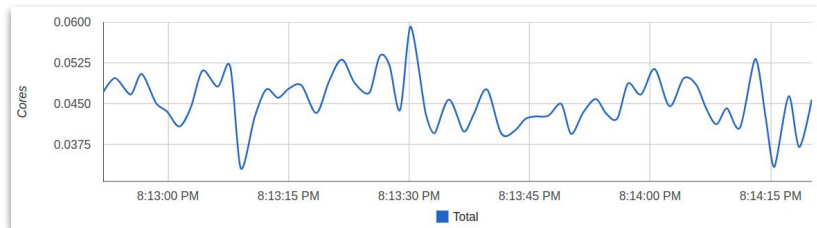
3. SERVIDOR E 100 CLIENTES

50 inscritos e 50 publicadores. Cada inscrição/publicação é feita aleatoriamente dentre 5 possíveis tópicos e por um tempo aleatório.

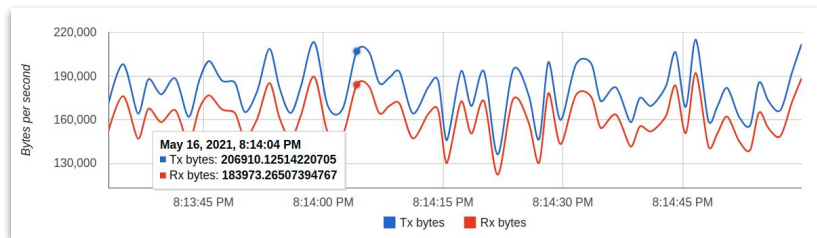
Consumo de até 6% de CPU.

Até 220k bytes transferidos por segundo pela rede.

Consumo de CPU



Uso de rede (*throughput*)



Obrigado