

Exercício-Programa 1

MAC0352 - Redes de Computadores e Sistemas
Distribuídos

25/04/2022

Ian Silva Galvão nº usp: 6817482

Implementação

Código

Estrutura

O código utiliza 3 tipos de processo:

- Um processo único que escuta novas conexões.
 - Um processo único que gerencia a troca de mensagens entre os demais.
 - Processos que se comunicam com os clientes. Um por cliente.
-

Processos dos Clientes

Comunicação com o gerenciador de mensagens.

Os processos dos clientes abrem um pipe próprio e enviam o nome dele para o processo gerenciador de mensagens. Junto mandam o tópico e um byte para indicar de são subscribers ou publishers.

Estes dados são enviados por um pipe de controle.

Processos dos Clientes

Comunicação com o gerenciador de mensagens.

Após mandar as informações, o processo do cliente aguarda o gerenciador de mensagens abrir o seu pipe.

Por este pipe ele irá mandar ou receber mensagens do processo gerenciador.

Processo Gerenciador de Mensagens

Distribuição de Mensagens

O processo gerenciador fica em um loop lendo mensagens do pipe de controle. Cada uma possui o tipo de processo que enviou, o tópico e nome do pipe para a comunicação entre eles.

Processo Gerenciador de Mensagens

Distribuição de Mensagens

Se o processo do cliente for subscriber, o gerenciador irá abrir o FIFO como WRONLY enviado e guardá-lo em uma lista encadeada indexada pelo tópico.

Se o processo for publisher ele irá abrir o FIFO como RDONLY, ler a mensagem e a escrever em cada pipe aberto da lista encadeada do tópico correspondente.

MQTT Connect

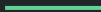
Como é feito

Ao receber um pacote Publish o servidor envia uma resposta **CONNACK**.

MQTT Disconnect

Como é feito

Ao receber um pacote Disconnect o servidor não envia respostas e fecha automaticamente pelo protocolo TCP.



MQTT Publish

Como é feito

Ao receber um pacote Publish o servidor não envia resposta, e aguarda um DISCONNECT.

Para executar, o processo do cliente abre um Pipe, manda o nome para o gerenciador de processos (pelo pipe de controle) e escreve a mensagem no pipe aberto.

O gerenciador, então, escreve no Pipe de cada processo inscrito no tópico, que por sua vez, irão mandar a mensagem para os clientes.

MQTT Subscribe

Como é feito

Ao receber um pacote Subscribe, o servidor responde com um SUBACK.

Para executar o pedido, o processo do cliente abre um pipe, manda o nome para o gerenciador de processos (pelo pipe de controle) e aguarda mensagens neste pipe para mandar para os clientes.

Bibliotecas externas

hashmap

Para indexar os tópicos foi usada uma tabela de hash retirada de: <https://github.com/tidwall/hashmap.c/blob/master/hashmap.c>, desenvolvida por Joshua J Baker

O software em questão possui licença GPL, colocada junto com o projeto.

OBS.: A licença no arquivo LICENSE se refere apenas a essa tabela hash.

Experimentos

Experimentos

Setup

Os experimentos foram feitos utilizando dois computadores ligados na rede wifi.

Foram executados experimentos com 0, 100 e 1000 clientes.

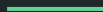
Em cada um, metade dos clientes se inscreveu em um único tópico e a outra metade enviou uma única mensagem para este tópico.

Experimentos

Setup

Dessa forma foram enviadas 0, 2500 e 250000 mensagens, respectivamente.

Algumas mensagens foram enviadas em um mesmo pacote TCP, e portanto o número de pacotes enviados foi menor do que o número de mensagens.



Resultados

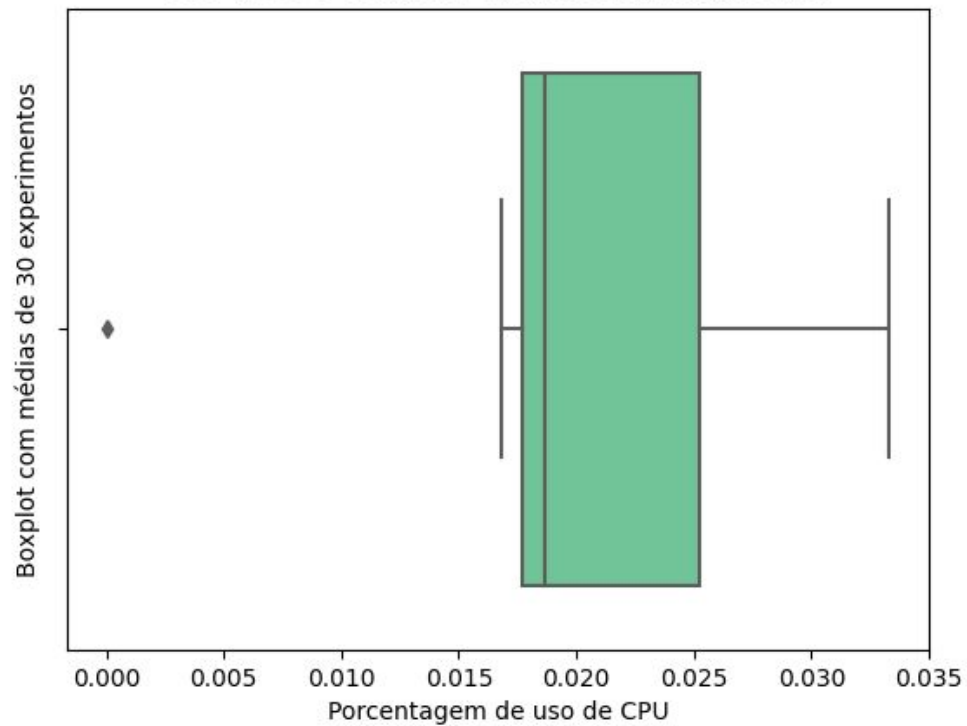
O uso de rede com 0 processos foi nulo. O uso de CPU nesse cenário foi insignificante.

Para os dois outros cenários, o uso de rede e CPU está indicado nos boxplots a seguir.

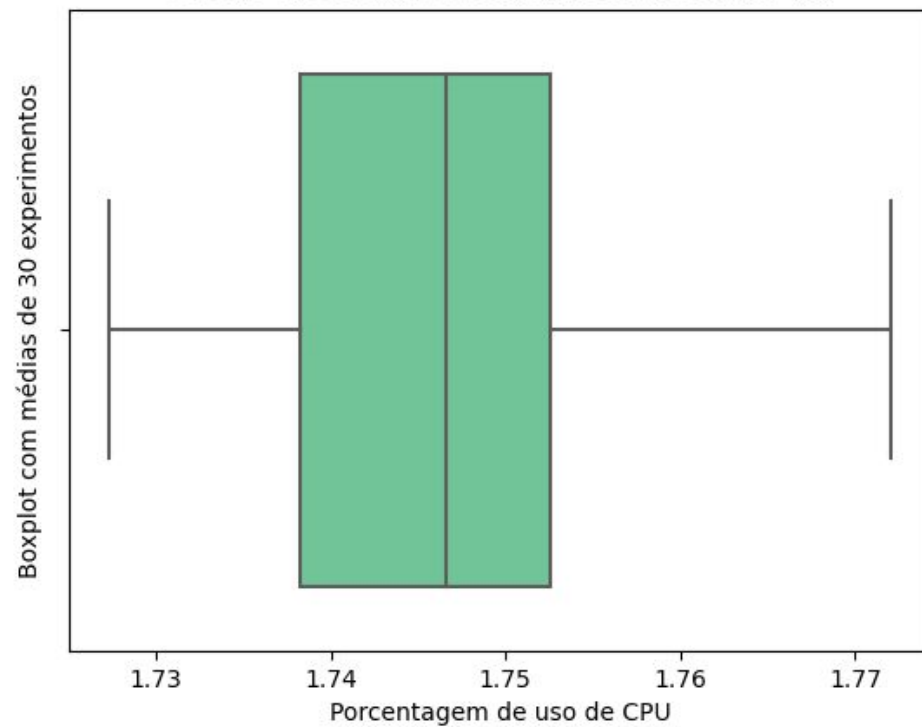
Foram rodados 30 experimentos para cada cenário.

OBS.: O uso da rede foi determinado somando o tamanho de todos os pacotes capturados no Wireshark, com filtro port 1883.

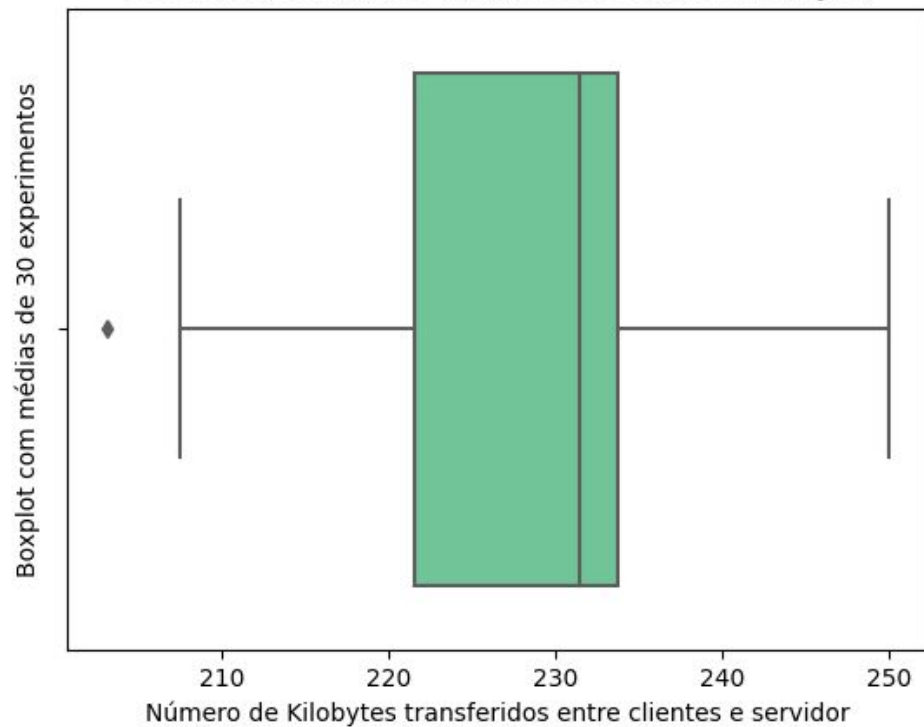
Uso de CPU com 100 clientes conectados (%)



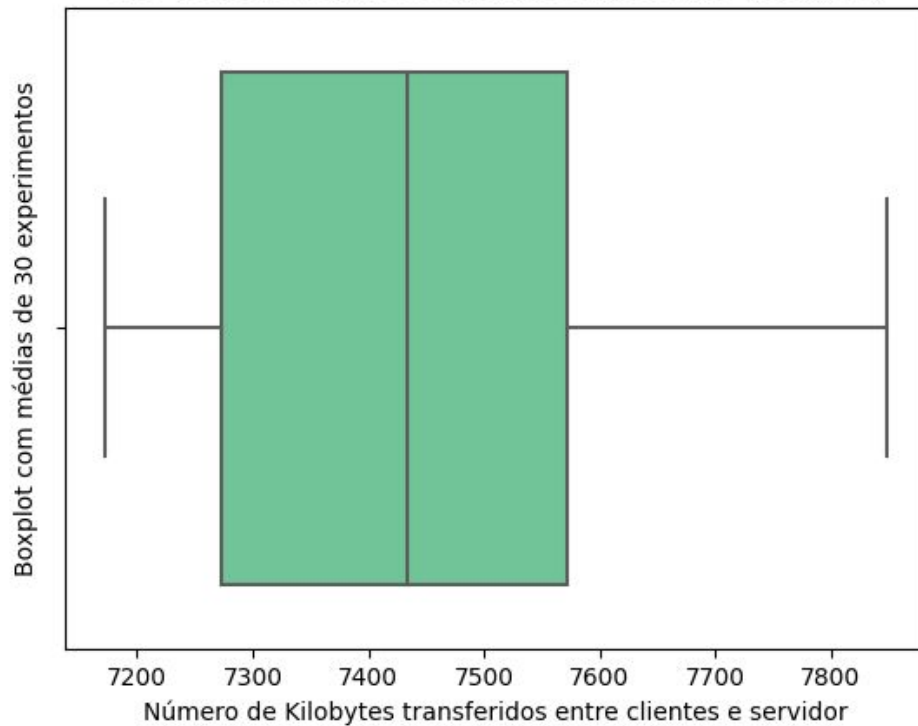
Uso de CPU com 1000 clientes conectados (%)



Uso de rede com 100 clientes conectados (Kilobyte)



Uso de rede com 1000 clientes conectados (Kilobyte)



Fim
