MAC0352-EP1

Luca Assumpção Dillenburg
NUSP: 11796580

Descrição da Apresentação em Alto Nível

Podemos entender o broker como um servidor que aceita todas as requisições, criando um novo processo filho para lidar com cada requisição.

Tanto os processos que acompanham as requisições de publishers quanto os processos que acompanham as requisições de subscribers são filhos do mesmo processo e se comunicam através de FIFOs: arquivos especiais utilizados para o envio de mensagens entre processos.

Os processos que acompanham requisições de publishers enviam as mensagens para os arquivos FIFOs de um certo tópico e os processos que acompanham requisições de subscribers leem esses arquivos e depois enviam as mensagens aos clientes.

Decisões Relevantes de Projeto: Sistema de Arquivos

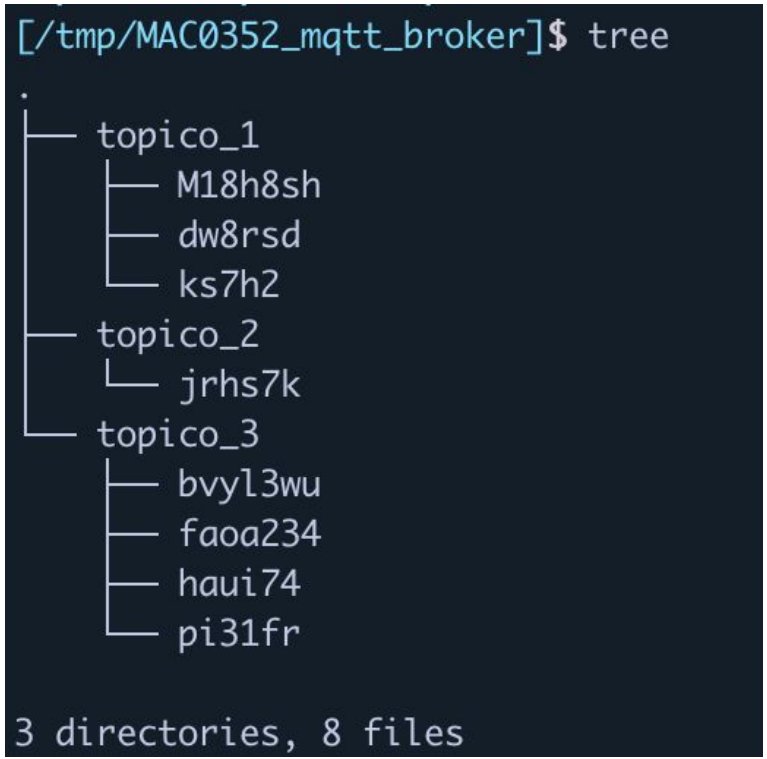
Uma decisão importante na organização do sistema foi onde seriam armazenados os arquivos FIFO de cada tópico. Ao meu ver, haveriam duas formas de fazer: (1.) armazená-las num hashtable que parearia os tópicos (keys) e os arquivos FIFO inscritos em cada tópico ou (2.) criar uma estrutura de arquivos que, por si só, informasse os arquivos FIFO inscritos em cada tópico.

Optei pela segunda opção por remover a necessidade de lidar com o compartilhamento de memória entre diferentes processos. Por isso, tornou-se necessário criar cada arquivo FIFO dentro de uma pasta dedicada ao tópico em que o mesmo está inscrito, possibilitando o envio de mensagens pelos processos do broker responsáveis por encaminhar as mensagens dos publishers.

Decisões Relevantes de Projeto: Sistema de Arquivos

Por conta dessa organização de pastas, tornou-se necessário fazer um encoding dos nomes dos tópicos para garantir que é um nome válido para uma pasta no Linux.

Segue um exemplo (ao lado) da estrutura de arquivos quando há com alguns tópicos e subscribers.



Descrições dos Comandos

- CONNECT: é necessário enviar um packet de resposta CONNACK
- SUBSCRIBE: é necessário enviar um packet de resposta SUBACK, criar um FIFO para aquele tópico e lendo as mensagens dele, enviando para o cliente (foi necessário dividir esse processo em dois para tratar os PINGs)
- PINGREQ: é necessário enviar um packet de resposta PINGRES
- PUBLISH: não é necessário enviar um packet de resposta, mas é necessário enviar os dados para os arquivos FIFOs daqueles tópicos
- DISCONNECT: não precisa fazer nada

Análise de Desempenho

Vimos que o MQQT é muito utilizado em IoT, onde os dispositivos que estão enviando e recebendo mensagens muitas vezes tem uma baixa capacidade de processamento. Por essa razão, procurei uma máquina antiga para rodar os clientes e uma máquina mais nova para rodar o servidor.

A máquina que está rodando o broker/servidor é o Macbook Air M1 com 16gb de RAM, enquanto os clientes estão rodando em um PC Core 2 Duo com 4gb de RAM.

Os dados a serem disponibilizados a seguir são do Macbook Air com o broker recebendo subscribers e publishers, e os dados foram capturados pelo Wireshark (com o filtro da porta e MQTT) e pelo aplicativo Activity Monitor da Apple.

Análise de Desempenho: apenas broker

Bytes Transferidos:

0 bytes

Nenhuma informação foi transferida.

Uso de CPU:

Média: 3.9% com 645.83 processos

Desvio Padrão: 1.138% com 33.58 processos

Gráfico % de CPU:

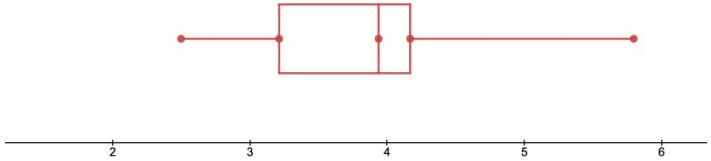
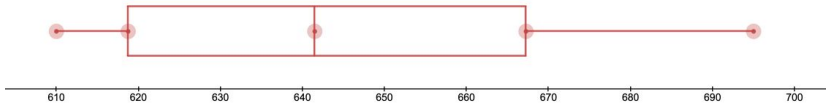


Gráfico Número de Processos:

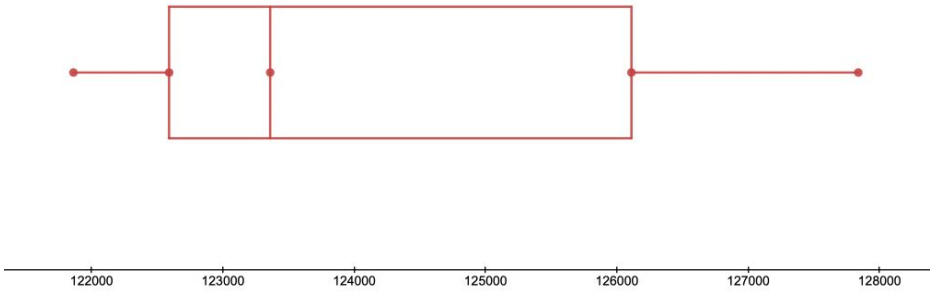


Análise de Desempenho: broker + 100 clientes

Bytes Transferidos:

Média: 124299 bytes

Desvio Padrão: 2471.2 bytes



Uso de CPU:

Média: 7.058% com 795.3 processos

Desvio Padrão: 0.37% com 6.80 processos

Gráfico % de CPU:

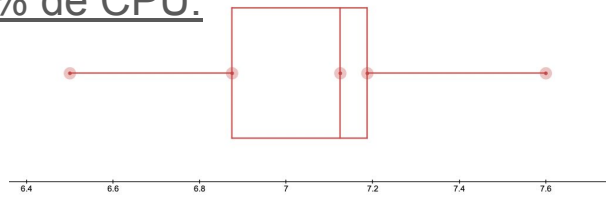
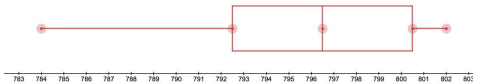


Gráfico Número de Processos:

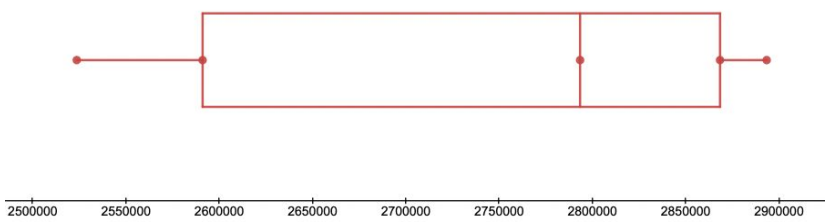


Análise de Desempenho: broker + 1000 clientes

Bytes Transferidos:

Média: 2737019 bytes

Desvio Padrão: 171084 bytes



Uso de CPU:

Média: 12.1% com 1668.8 processos

Desvio Padrão: 1.89% com 10.34 processos

Gráfico % de CPU:

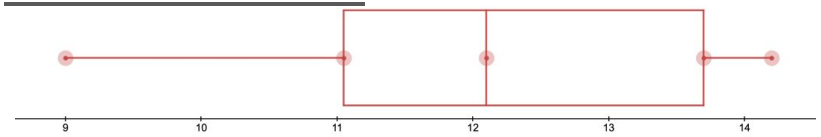


Gráfico Número de Processos: