

Exercício Programa 1

Broker MQTT

Lucas Paiolla Forastiere, 11221911

IME-USP

17 de maio de 2021

Detalhes de Implementação

Os comandos

- O broker interpreta cada comando de acordo com o primeiro byte do header do pacote. Assim que lemos o pacote, vemos se ele se encontra entre as categorias: `CONNECT_PACKAGE`, `PUBLISH_PACKAGE`, `SUBSCRIBE_PACKAGE`, `PINGREQ_PACKAGE` ou `DISCONNECT_PACKAGE`.
- Qualquer outro pacote é ignorado pelo broker, pois ou não pertence ao escopo do projeto (como QoS maiores que 0) ou é um pacote que deveria chegar apenas ao cliente (como `CONNACK_PACKAGE`).

Detalhes de Implementação

Suposições

- Assumi que todos os inteiros que descrevem tamanhos (descritos como `Variable Byte Integer` na RFC) nunca serão maiores que 128, ou seja, ocupam apenas um byte.
- Da mesma forma, existem inteiros que descrevem tamanho que possuem dois bytes fixos. Eu assumi que esses valores sempre são menores que 128. Ou seja, o primeiro byte do número sempre será 0x00.
- Supõe-se que o número máximo de clientes conectados é algo da ordem de 100.000 (mais exatamente, os PIDs dos processos designados a cada cliente não podem ultrapassar 300.000).

Detalhes de Implementação

Suposições

Outras suposições foram feitas sobre os pacotes recebidos:

- Supõe-se que o `PUBLISH_PACKAGE` e o `SUBSCRIBE_PACKAGE` possuem `properties length` igual a zero.

Detalhes de Implementação

Armazenamento dos Inscritos

- Para armazenar os inscritos e fazer a comunicação entre os processos filhos, utilizou-se memória compartilhada com o uso do `mmap`.
- Existe um array com o nome dos tópicos, um array com o nome dos inscritos em cada tópico, um array com quantos são os inscritos em cada tópico e um array com qual a última mensagem enviada para cada tópico.
- Para armazenar os clientes inscritos em um tópico, utiliza-se o array `topic_subs` de tamanho 300.000 onde cada posição j , i é i se, e somente se, o cliente de PID i se inscreveu no tópico j . Caso contrário, é -1 .

- Os testes foram realizados com o uso de uma máquina virtual com quatro núcleos de CPU dedicados e 8G de memória RAM, e uma máquina host de oito núcleos de CPU e 16G de memória RAM;
- A comunicação foi feita utilizando uma rede local de 100 Mbps;
- Para medir a carga de rede, utilizou-se o Wireshark com um filtro de pacotes MQTT. Como nenhuma máquina na rede local faz uso desse tipo de pacote, temos uma estimativa de quanto de rede se está gastando.

Testes

Apenas o servidor

- Quando apenas o servidor está em operação, sem nenhum cliente querendo conectar-se, não temos absolutamente nenhuma ação acontecendo. O servidor não envia ou recebe quaisquer mensagens.

Testes

Servidor com um inscrito e um publicador

- Se não há clientes inscritos no broker, então qualquer publicação imprimirá a mensagem “Tópico não encontrado.”.
- De forma análoga, se um cliente tenta publicar algo em um tópico que não há clientes inscritos, a mesma mensagem é impressa.
- Quando um cliente pede para se inscrever em um tópico, esse tópico é criado (já que nesses testes sempre existirá apenas um inscrito) e o PID do processo é salvo para depois enviarmos mensagens a ele. O processo que recebeu o `SUBSCRIBE_PACKAGE` cria um `fork` que ficará verificando se novas mensagens chegaram ao tópico utilizando uma espera ocupada.

Testes

Servidor com um inscrito e um publicador

- Quando um cliente publica em um tópico existente, nós alteramos a última mensagem enviada ao tópico e avisamos o inscrito que uma nova mensagem está disponível. Então ele enviará exatamente o mesmo `PUBLISH_PACKAGE` recebido para o cliente inscrito e a mensagem será enviada.
- Quando o cliente inscrito em um tópico se desconecta, nós removemos ele do tópico. Como nesse caso de teste há apenas um inscrito, o tópico agora terá zero inscritos e será também deletado, voltando o servidor para o caso em que não haviam inscritos no broker.

Testes

Servidor com 100 clientes

- De forma geral, o servidor não lida bem com tantos clientes, pois ele fica muito congestionado. Entretanto, caso os clientes publicadores não mandem muitas mensagens por segundo, o servidor consegue lidar melhor do que quando esse fluxo de mensagens é muito grande.
- Tudo o que foi falado para o caso com dois clientes vale. Ou seja, enviar uma mensagem para um tópico inexistente resulta em uma mensagem de erro, mas o servidor continua normalmente.
- Além disso, o limite máximo de tópicos é 8, então se já existem esses oito tópicos e um cliente tenta se inscrever em um novo tópico, esse cliente recebe uma mensagem de erro e é desconectado.

Obrigado!

Lucas Paiolla