




EP1 - MAC0352

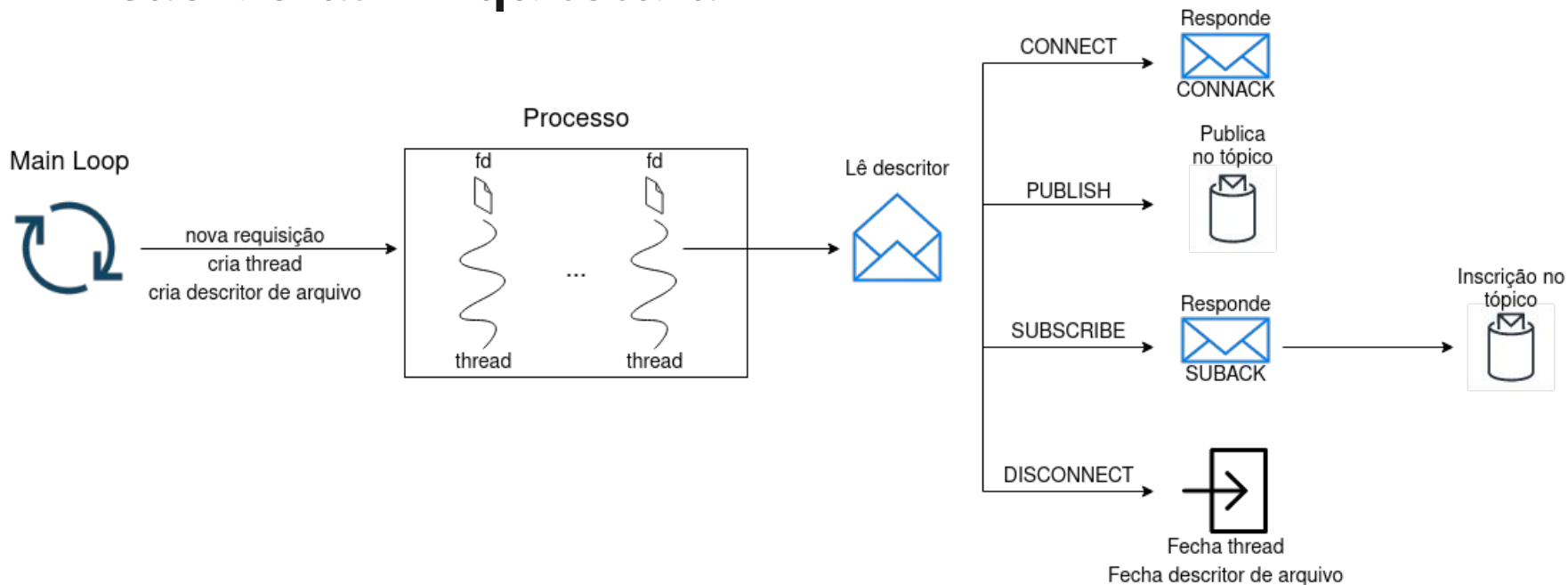
Servidor MQTT

Ygor Tavela Alves

nUSP: 10687642

- 
- **Visão Geral - Arquitetura**
 - **Detalhes de Implementação**
 - **Desempenho**

Visão Geral - Arquitetura





Detalhes de implementação

Criação de requisições:

- Aceita nova conexão no *file descriptor* de escuta;
- Cria nova *struct* de requisição que carrega informações como o *file descriptor* da conexão e um ponteiro pra uma *thread*;
- Inserir *struct* de requisição em uma lista ligada dupla que é utilizada para armazenar os endereços de memória das requisições e facilitar a limpeza na memória ao final;
- Chama função para ler as informações do socket.



Detalhes de implementação

Leitura pacote socket:

- Desempacota *header* fixo da mensagem MQTT, armazenando informações como o tipo da mensagem e o tamanho da mensagem;
- A depender do tipo da mensagem uma ação é tomada;
 - CONNECT: responde com mensagem CONNACK;
 - PUBLISH: Verifica se tópico existe e, caso sim, responde todos os inscritos no tópico com a mensagem publicada;
 - SUBSCRIBE: responde com mensagem SUBACK e se inscreve no tópico desejado;
 - DISCONNECT: fecha *thread* e *file descriptor*.



Detalhes de implementação

Publicação de mensagens:

- Busca em uma *Hash Table* pelo tópico desejado, uma *struct* que contém informações como nome do tópico e o seu tamanho, uma lista dinâmica de *file descriptors* dos *subscribers* inscritos nesse tópico e um *lock* de *thread* para leitura/escrita do tópico de forma segura entre as threads;
- Caso tópico exista, a lista de *file descriptors* dos *subscribers* é percorrida para enviar a mensagem publicada para cada um destes *subscribers*;
- Caso tópico não exista, nada é feito.



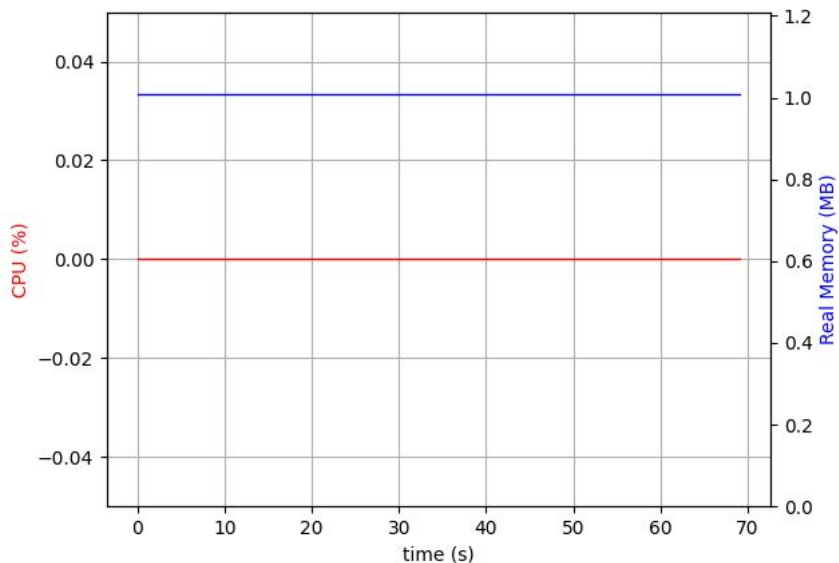
Detalhes de implementação

Inscrição em tópicos:

- Busca em uma *Hash Table* pelo tópico desejado;
- Caso tópico não exista, uma nova *struct* de tópico é criada e inserida na *Hash Table* utilizando como chave o nome do tópico;
- Insere-se o *file descriptor* correspondente ao *subscriber* atual na lista dinâmica de *subscribers* do tópico;
- Inicia-se um *loop* de inscrição para escutar mensagens com os tipos PINGREQ e DISCONNECT;
 - PINGREQ: responde com mensagem PINGRESP;
 - DISCONNECT: deleta *file descriptor* da lista dinâmica de *subscribers* do tópico, fecha *thread* e *file descriptor*.

Desempenho

0 Cliente:

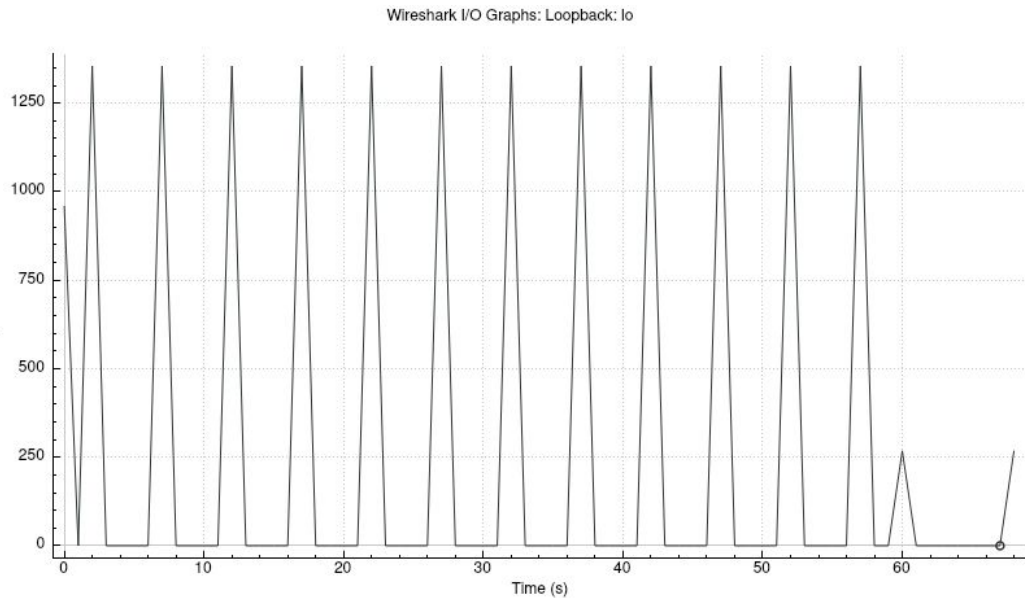
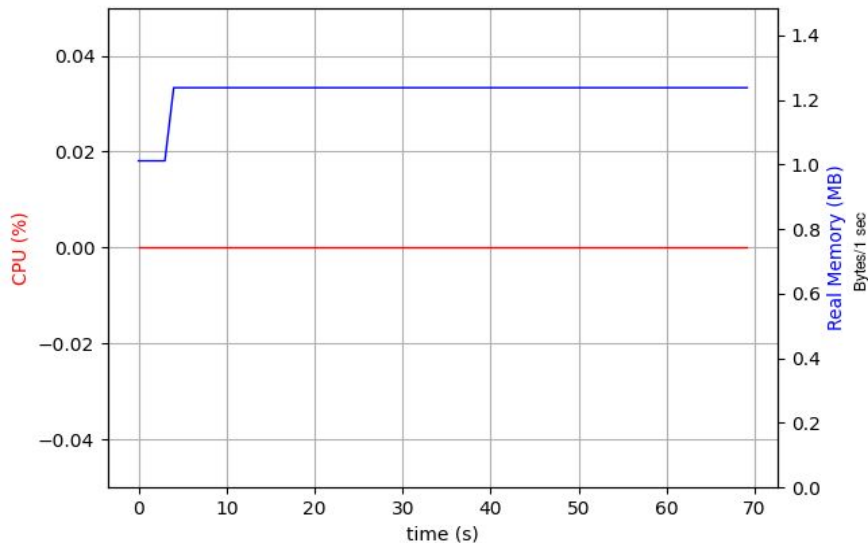


- Não houve consumo de CPU/rede para este caso
- O consumo de memória foi constante, pois não houve alocação de recursos para atender nenhuma requisição

Desempenho

1 Cliente (1 Pub/1 Sub):

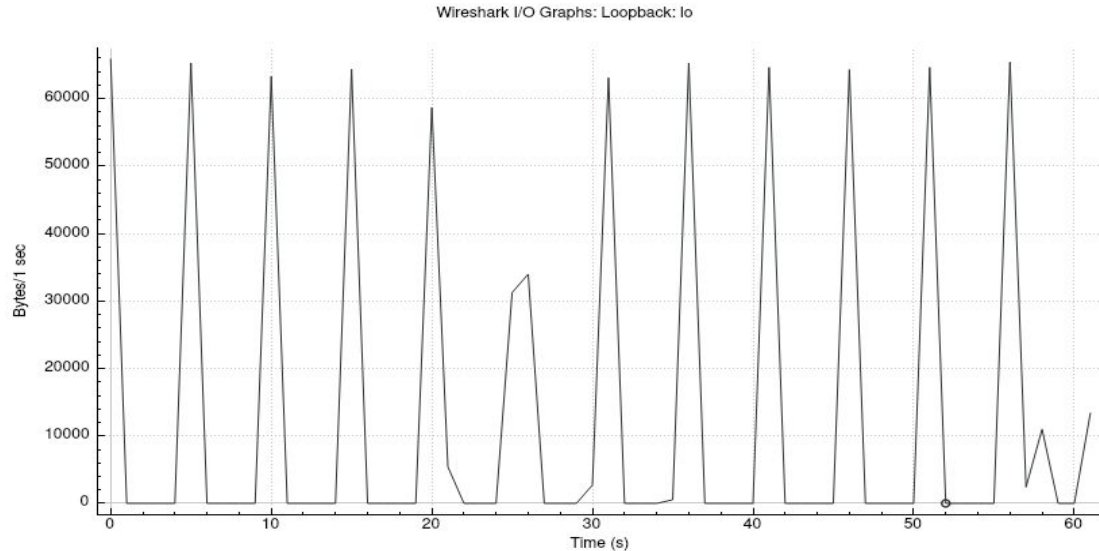
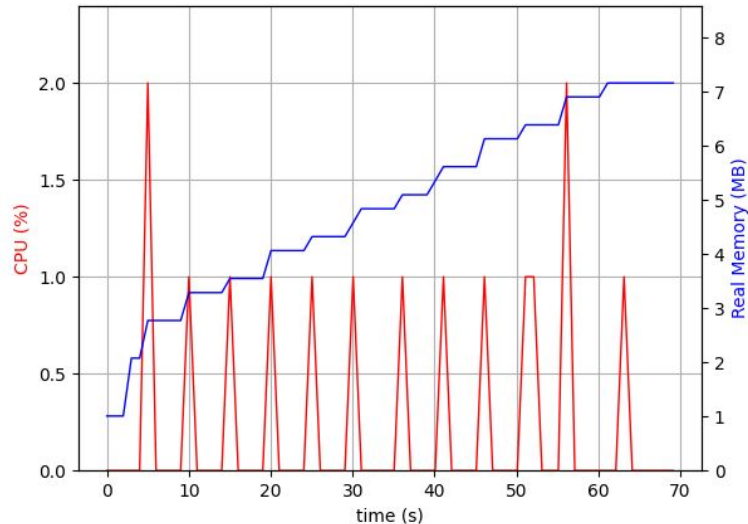
- Consumo de CPU próximo de 0 (psrecord não possui tanta precisão).
- Consumo de Memória praticamente constante na média de 1.2Mb.
- Consumo de Rede em picos, atingindo máximas de 1250 bytes consumidos por segundo.



Desempenho

50 Clientes (50 Pubs/50 Subs):

- Consumo de CPU em picos, em torno da média de 1% da CPU consumido, com no máximo 2% de consumo.
- Consumo de Memória crescente, podendo indicar vazamento de memória, atingindo consumo máximo de 7Mb.
- Consumo de Rede em picos, atingindo máximas de 65000 bytes consumidos por segundo.





Desempenho

Ambiente de Testes:

- Arquitetura x86_64
- Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
- 8GB RAM