



INSTITUTO SUPERIOR TÉCNICO

RELATÓRIO DO PROJECTO 2017

Programação de Sistemas

Grupo 19

João Loureiro
Mariana Martins

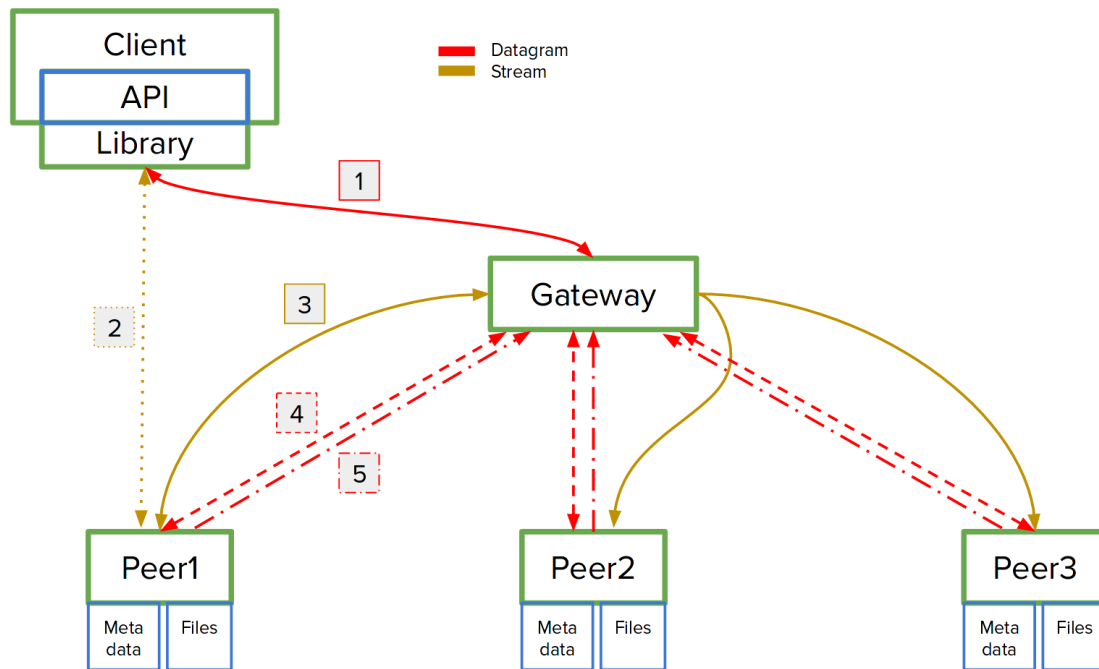
81164
80856

Conteúdo

1	Arquitectura e Componentes	2
2	Gestão de threads	3
3	Estruturas de dados e sincronização	4
4	Replicação	6
5	Desactivação de peers	6
6	Download da informação para novos peers	7
7	Critérios de avaliação	7

1 Arquitectura e Componentes

O projecto segue a seguinte arquitectura:



- **Ligação 1:** Permite o cliente fazer um pedido de endereço à gateway e a gateway enviar um endereço da lista de peers ao cliente.
- **Ligação 2:** Permite o cliente enviar informação ao peer para executar um qualquer pedido no sistema (add photo, add keyword, ...) e o peer enviar ao cliente o retorno desse pedido. Há tantos sockets quanto os clientes a utilizar esse peer.
- **Ligação 3:** Permite o peer enviar a informação à gateway e a gateway enviar a mesma a todos os peers (replicação). Há tantos sockets quanto os peers existentes.
- **Ligação 4:** Permite a gateway enviar um ping ao peer e o peer de volta para a gateway.
- **Ligação 5:** Permite o peer enviar o seu endereço à gateway.

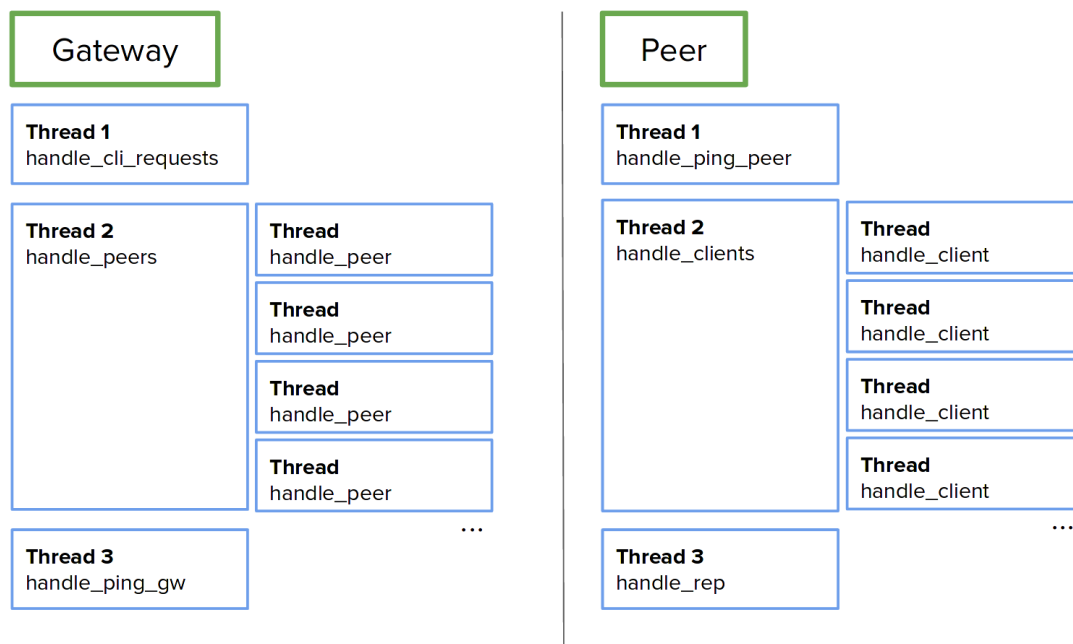
Quando um novo cliente se liga ao sistema, contacta a gateway (ligação 1), a gateway retorna um endereço de um peer da lista de peers já existentes. O cliente liga-se a esse peer (ligação 2), por onde são analisados os seus pedidos.

Quando é recebido um pedido de um cliente num peer, este é realizado nesse peer caso seja um pedido de leitura do sistema ou é mandado para a gateway (ligação 3) caso seja um pedido de escrita no sistema. A gateway redirecciona esse pedido para todos os peers na lista de peers (ligação 3) e cada peer executa o pedido. O peer que mantém o contacto com o cliente manda-lhe o retorno do pedido.

A gateway corre infinitamente a lista de peers, e manda a cada um ping e o peer se estiver vivo responde de volta (ligação 4).

Quando aparece um novo peer, este envia o seu endereço à gateway (ligação 5) e recebe a informação já existente no sistema pela ligação 3 (a gateway manda um pedido ao primeiro peer da lista, este envia a informação existente para a gateway e a gateway envia-a para o novo peer).

2 Gestão de threads



Na **gateway** existem 3 threads principais como representado, todas criadas no main do gateway.c, a seguir indicam-se as suas funções:

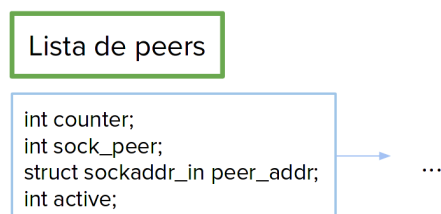
- Thread 1 é responsável por receber do cliente pedidos de endereços, e enviar o endereço de volta.
- Thread 2 é responsável por receber novos peers. Sempre que é recebido um peer, é criada uma subthread para ele. Nessa subthread são recebidos e redireccionados todos os pedidos respeitante a esse peer.
- Thread 3 é responsável por correr a lista de peers, mandando e recebendo pings de cada um, para verificar se estão vivos.

No **peer** existem 3 threads principais como representado, todas criadas no main do peer.c, assequir indicam-se as suas funções:

- Thread 1 é responsável por receber e enviar um ping a gateway a indicar que o peer está vivo.
- Thread 2 é responsável por receber novos clientes. Sempre que é recebido um cliente, é criada uma subthread para ele. Nessa subthread são recebidos e analisados todos os pedidos respeitantes a esse cliente, caso sejam pedidos de leitura é também aqui que são executados.
- Thread 3 é responsável por receber pedidos provenientes da gateway e executá-los (replicação), enviado o retorno de volta para o cliente.

3 Estruturas de dados e sincronização

- counter - conta o nº de vezes que o servidor não respondeu ao ping
- sock_peer - guarda o socket stream que permite passar informação entre a gateway e o peer (ligação 3)
- peer_addr - contém o endereço do peer
- active - define se um novo peer já recebeu a informação existente ou não (activo ou não respectivamente)



Nas **threads da gateway é partilhada a lista de servidores.**

Na thread 1, é utilizada para ler um peer da lista e enviar o seu endereço ao cliente (leitura).

Na thread 2, é utilizada para inserir peers na lista (escrita). Em cada subthread da thread 2, é utilizada para redireccionar os pedidos ao sistema a cada peer - replicação

- e para enviar a informação existente a um novo peer (leitura).

Na thread 3, é utilizada para ler peers da lista e mandar-lhes pings.

- file_name - contém o nome da foto que o cliente enviou
- id_photo - contém o id da foto na lista
- keyword - contém a possível keyword dada pelo cliente
- photo_size - contém o tamanho da foto
- cli_sock - contém o socket do cliente que fez o pedido
- peer_pid - contém o pid do peer que está a tratar do cliente que fez o pedido



Nas **threads 2, 3 e subthreads do peer é partilhada a lista de fotos e o socket stream que comunica com a gateway (ligação 3).**

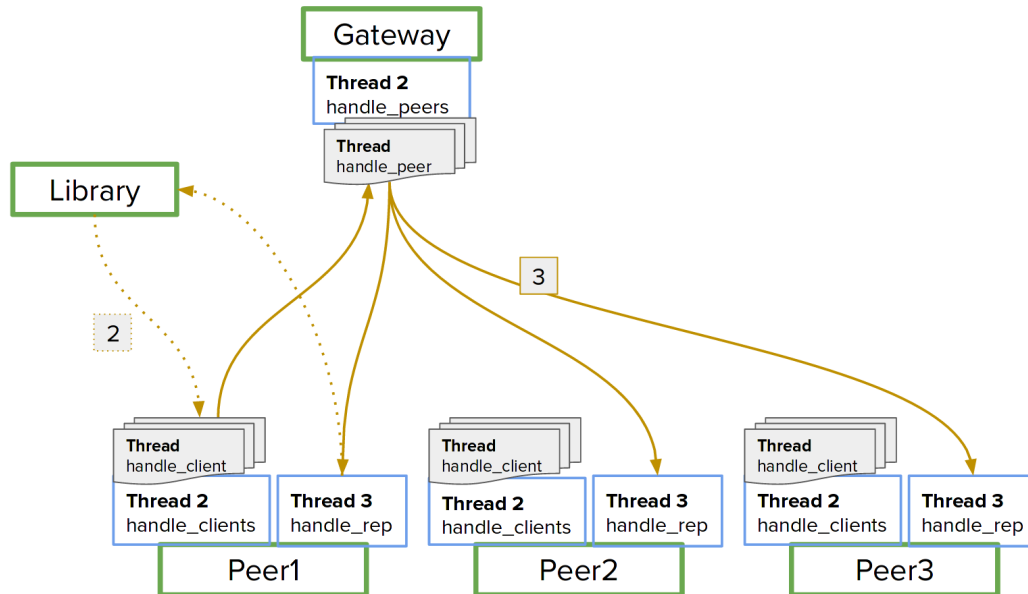
Em cada subthread da thread 2, a lista de fotos é utilizada para analisar os pedidos do cliente (leitura) e executá-los - caso não sejam de escrita. Neste último caso, o socket stream é utilizado para enviar o retorno ao cliente.

Na thread 3, a lista de fotos é utilizada para inserir novos elementos - add photo -, para escrever nova informação em já existentes - add keyword - ou para os remover - delete photo - (leitura e escrita). O socket stream é usado para mandar o retorno dos pedidos do cliente que foram executados nesta thread ao mesmo.

A sincronização é feita usando 3 mutexes que bloqueiam leituras ou alterações de estruturas de dados partilhadas entre as threads da gateway e entre as threads de cada peer.

Na gateway actua o mutex que bloqueia as regiões onde a lista de peers é acedida ou modificada e outro mutex na região onde é gerado o id das fotos entre as várias threads.

Em cada peer há apenas um mutex entre as várias threads que bloqueia as regiões onde se acede e modifica a lista de fotos.



4 Replicação

Quando o cliente pede add photo, add keyword ou delete photo, estas operações são replicadas para que todos os peers as executem.

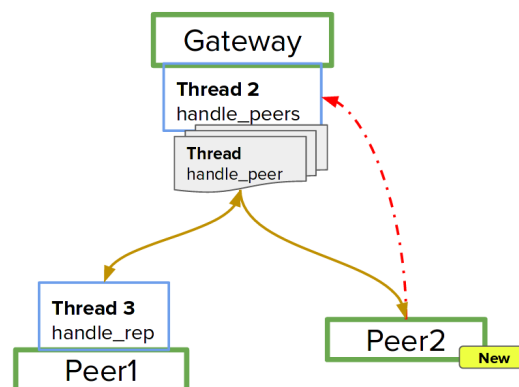
Exemplificando, quando um cliente faz um desses pedidos, este é recebido pela subthread da thread 2 que está a tratar desse cliente (ligação 2). A subthread envia o pedido à gateway que o redirecciona para a thread da replicação (thread 3) de cada peer da lista de peers (ligação 3). Nessa thread, é então executado o pedido, e enviado o retorno do mesmo ao cliente (ligação 2). Este retorno é apenas enviado pelo peer que está a tratar do cliente, daí haver a variável `peer_pid` nas estruturas da lista de fotos.

5 Desactivação de peers

A gateway está constantemente a enviar e a receber pings dos peers da lista de peers existentes. Quando um desses peers não responde, é aumentado o seu counter. Se ele voltar a responder, o counter volta a zero. Se o counter chegar a 3, o peer é considerado morto e é retirado da lista de peers. O ping corresponde a um inteiro com valor 1, dado o seu conteúdo ser irrelevante para aqui.

6 Download da informação para novos peers

Quando aparece um novo peer, este envia o seu endereço à gateway. Ao receber, a gateway faz um pedido ao primeiro peer da lista de peers já existentes. A thread 3 (da replicação) desse peer recebe o pedido, e envia a informação existente no sistema de volta para a gateway, que a recebe na thread desse mesmo peer. Este por sua vez, envia a informação para o novo peer.



7 Critérios de avaliação

Todos os critérios são atendidos visto que:

- foi implementada uma gateway
- o cliente contacta a gateway que lhe envia a informação de um dos peers
- os peers podem ser inseridos depois de já ter sido inserida informação no sistema e fazem download da informação existente
- existe a verificação da morte de peers se estes deixarem de responder a pings que são enviados pela gateway durante todo o funcionamento do programa
- são gerados ids para as fotos sendo que é garantido que nenhuma das fotos têm o mesmo id
- é realizada a replicação para todos os peers cada vez que um peer recebe uma operação e todos os peers têm em qualquer instante a mesma informação
- as operações que manipulam estruturas de dados partilhadas por threads estão sincronizadas a partir da utilização de mutexes
- ficheiros das imagens são corretamente enviados e guardados no cliente e nos peers