



Computação Paralela e Distribuída 2021 – 2022

# **Distributed and Partitioned Key-Value Store**

João Silva 201906478

Nuno Castro 202003324

Pedro Vale 201806083

## Conteúdo

Membership Service .....	3
Message Format.....	3
Details .....	3
Storage Service.....	4
Arquitetura.....	4
Key Value Transfer .....	5
Join Event / Leave Event .....	5
Message Format.....	5
Details .....	6

## Membership Service

### Message Format

Sempre que se dá um **JOIN** event ou um **LEAVE** event o nó que se juntou ou deixou o cluster envia por IP multicast uma mensagem a sinalizar esta operação.

Ambas operações **JOIN** e **LEAVE** são responsabilidade da classe **TCPReceiver** que recebe informação vinda do cliente em relação ao tipo de função a executar.

Perante um **Join** a função **joinOp()** é invocada e num **Leave** a função a executar é **leaveOp()**.

Nestas duas funções é inicializado um objeto do tipo **MulticastSender** que envia uma mensagem via IP multicast e é inicializado um objeto do tipo **MulticastMessage** que trata de formular a mensagem a enviar por multicast consoante a operação realizada.

#### Join Message:

No objeto **MulticastMessage** é passado como parâmetros o **membership counter** do cluster, o **hash id** (hashed IP address) do nó que pretende juntar-se, a **porta (random)** correspondente a uma nova Socket criada no momento do join, o valor em termos de verdadeiro ou falso do uso de mecanismo **antiFail** (envio de mensagem UDP de 1 em 1 segundo para a rede com o conteúdo do seu membership log) e finalmente o **IPaddress** correspondente.

#### Leave Message:

A formatação da mensagem de operação Leave é semelhante ao formato da operação Join:

No objeto **MulticastMessage** é passado como parâmetros o **membership counter** incrementado do cluster, o **hash id** (hashed IP address) do nó que pretende juntar-se, a **porta** correspondente ao nó, o valor em termos de verdadeiro ou falso do uso de mecanismo **antiFail** (envio de mensagem UDP de 1 em 1 segundo para a rede com o conteúdo do seu membership log) e finalmente o **IPaddress** correspondente.

### Details

Em relação à **anulação da propagação de envio de informação obsoleta**, a escolha dos nós que enviam a um novo membro a sua visualização da **Membership** não é feita com qualquer critério, apenas são selecionados os três primeiros nós e assim feita a seleção estes enviam as **Membership messages**.

Em relação às **falhas**, foi implementado como já fora referido anteriormente um mecanismo de **anti fail** que quando acionado consiste no envio de uma mensagem UDP de 1 em 1 segundo contendo o **Membership log** do nó que ativa este procedimento.

## Storage Service

### Arquitetura

A classe **TCPReceiver** é novamente responsável por tratar de grande parte do serviço de **store** dos pares **Key-Value**.

Efetuada um **Join** por parte do cliente, é feito um **Store** do nó caracterizado pelos argumentos do programa e atribuído a este uma pasta (se não estiver já criada) no sistema de ficheiros e um hash ID calculado através do seu IP address.

Existem 3 operações invocadas pelo cliente que manipulam este sistema de storage. Sendo elas a operação de **Put** parametrizada com a **key** e com o **value** correspondente a esta key, a operação de **Delete** de um par **Key-Value** e a operação **Get** que recolhe um valor para o utilizador dada a Key a pesquisar.

Na operação de **Put** é criado um ficheiro dentro da pasta do nó correspondente nomeado pela **Key** passada como argumento e preenchido com o **value** equivalente.

Na operação de **Delete** é formulado o caminho no sistema de ficheiros criado compatível com a **Key** dada e apagado a o ficheiro contendo o **value**.

Na operação de **Get** entrega-se ao cliente o valor equivalente à **Key** dada.

## Key Value Transfer

O protocolo de transferência de pares Key-Value é ativado por mudanças no membership, seja por eventos de Join ou por eventos de Leave.

Tal como referido, é nas operações de Join ou de Leave que o procedimento adotado para a transferência destes itens é efetuado. Sendo assim, nas funções joinOp e LeaveOp definidas na classe TCPReceiver são criadas instâncias da classe KeyValueTransfer que tratam destes eventos de acordo com as regras estabelecidas para esta transferência.

### Join Event / Leave Event

- É feito um processamento inicial para recolher todos os hashID dos nós que se apresentam online no cluster membership.
- Uma pesquisa do sucessor do nó com o ID enviado por parâmetro.
- Para um Join event, é enviado uma mensagem TCP para o sucessor a pedir os pares Key-Value adequados.
- Para um Leave event, é enviado uma mensagem TCP para o sucessor com os pares Key-Value do nó passado como argumento.

## Message Format

No serviço de store é utilizado três tipos de mensagem:

Store:

- `store;nodeKey;counter;log1---log2---log3`

Join Event - Sucessor:

- `sucessor;id;IPaddress;port`

Leave Event – Resposta – Entrega de pares Key-Value

- `retsuccesor;key1;value---value---value;key2;value---value`

## Details

O conceito de replicação não é implementado neste serviço. O aumento da disponibilidade de um Key-Value pair não é tido em conta na nossa implementação.