

January 15, 2020

Mestrado em Engenharia Informática e Computadores

**Computação Distribuída**

**Projecto Final**

Grupo 12

Alexandre Santos - 47926

Ana Pereira – 47930

Rúben Santos - 47766

Indice

[1. Introdução 3](#_Toc59563251)

# Introdução

O projecto no âmbito da disciplina de computação distribuída tem como propósito desenvolver um sistema distribuído de armazenamento de dados num cluster de servidores com replicação de dados, com um modelo de consistência garantido por consenso entre todos os servidores usando comunicação por grupos.

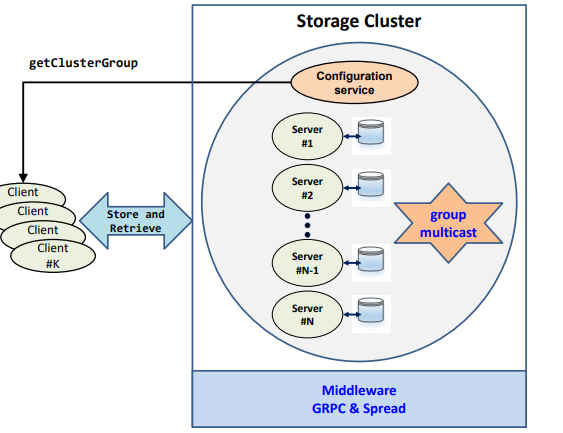
Como representa a figura abaixo pretende desenvolver um sistema distribuído de armazenamento de dados (chave e valor) através da comunicação por grupos. A comunicação entre grupos é feita usando o middleware gRPC (Google Remote Procedure Call) e o Spread Toolkit de comunicação por grupos.

Figura 1- Diagrama das partes envolvidas

No funcionamento normal, cada elemento do cluster corre a sua própria instância do servidor Spread e as aplicações clientes ligam-se localmente ao servidor. Os servidores Spread comunicam então entre si para passar mensagens para as aplicações que as tenham subscrito através do spread Toolkit que fornece um serviço de mensagens resistente a faltas, tanto em redes locais como em redes extensas. Funciona como um transportador de mensagens unificado e fornece multicast, comunicação em grupo e suporte de comunicação ponto a ponto. O toolkit em si consiste num servidor de mensagens e bibliotecas clientes para vários ambientes de desenvolvimento de software.

Por sua vez os clientes irão comunicar com os servidores e os servidores comunicarem entre si também através do Google RPC.

A análise inicial do trabalho consiste em identificar os pontos chaves para o desenvolvimento do trabalho como a escolha do:

* **Leader**:

A existência de um leader de servidores tem como função melhorar a gestão de comunicação entre clientes e servidores, sendo que o leader é o único que responde aos pedidos dos clientes através do grpc e o único que tem a funcionalidade de enviar mensagens multicast para actulizar a base de dados com as operações neste caso através do spread Toolkit.

* **Algoritmo de eleição**:

Através do spread Toolkit sabemos sempre o número de servidores que existem no cluster de servidores com comunicações de grupo, tendo em conta que o spread tem o conhecimento de todos os servidores do grupo temos de definir uma nomenclatura de modo a conseguir ordenar os servidores. E assim que um servidor der lead se esse servidor for o leader os restantes servidores assumem uma nova eleição e escolhem o servidor com o maior ID. Como exemplo, SERVIDORXX onde XX é o número do servidor, (server00, server01, etc.) e assim escolhemos o maior.

Quando acontece uma alteração no grupo de spread os servidores vão assumir o servidor com o hashcode com o código maior, através de spread, é desta forma que é eleito o novo leader.

* **Algoritmo de consenso:**

Os algoritmos de consenso são um processo de tomada de decisão para um grupo, onde cada indivíduo do grupo constrói e apoia a decisão que funcionará para o conjunto do grupo. Em termos simples, é apenas um método capaz de tomar decisões dentro de um grupo garantindo que existe consenso entre todos.

O two-phase commit garante o algoritmo de consenso. Vamos utilizar o algoritmo de consenso de forma a conseguir acordar um valor entre todos os partipantes. Através do spread temos a informação de todos os participantes, uma vez que ele comunica para o grupo tendo o total de participantes ativos e todos têm acesso ao número de participantes activos. Mesmo que um participante esteja em baixo é garantido que se recebem o total de confirmações de consenso.

* **Cordenador:**

Iremos ter também a existência de um coordenador que tenta estabelecer um consenso de todos os participantes em duas fases:

Voting phase - O coordenador sugere o valor, proposto pelo cliente, a todos os participantes e solicita uma resposta (sim ou não). Após receber todas as respostas, toma uma decisão: Se todos os participantes responderam sim (acordo no valor), decide fazer commit ou, caso algum participante não concorde faz abort.

[Commit phase] - O coordenador contata todos os participantes, outra vez, para comunicar a sua decisão de commit ou abort, esta comunicação é feita por spread para o grupo.

Quando é feito o startup é preciso eleger um coordenador para ajudar o leader, a seleção do coordenador, é feita de forma oposta do Leader, uma vez que o coordenador corresponde ao servidor que entra com o menor número de ID, com o menor hashcode, através de spread.

Interfaces:

* Leader – pedir os ips ao configurations.

É por spread mensagem que se chama new leader, vai ter o ip e o port. Quando não existir um leader o servidor com maior hashcode faz broadcast do seu IP, que será o IP do novo leader e o nome do leader. O servidor recebe sempre a informação do novo leader.

Notify no leader - para quando não existe leader.

Election manager – classe que vai gerir a eleição no novo leader

Select new leader – método para escolher o novo leader

Two-face commit – o leader vai enviar uma mensagem (read por spread two-face comit) e depois write two face-comit e depois um grpc se sim ou se não, é uma mensagem síncrona que envia sempre ao leader a dizer sim ou não, que tem um IP (vooting tem que ter uma key e um bolete) por grpc.

Notas:

Situação 1:

1. envia mensagem a dizer que quer escrever, ou seja, envia mensagem a dizer para invalidarem.

2. todos respondem a dizer que invalidaram.

3. Escreve.

4. Responde a dizer que escreveu.

Situação 2:

1. envia mensagem a dizer que quer escrever, ou seja, envia mensagem a dizer para invalidarem.

2. todos respondem a dizer que invalidaram.

3. Durante a escrita, recebe uma mensagem para invalidar.

4. Responde a dizer que tem conflito.

5. ambos vêm a cardinalidade, o que tiver maior escreve

6. O que escreveu responde que escreveu a todos