Versão preliminar do relatório

T08

URL repositório do GitHub

|  |  |
| --- | --- |
| Carolina Carreira | 87641 |
| Miguel Barros | 87691 |

1. Modelo de Faltas

– O sistema é assíncrono e a comunicação pode omitir mensagens

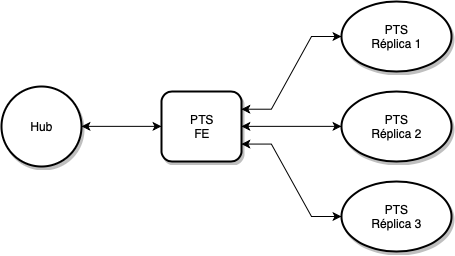
– Existem N gestores de réplica e N é constante e igual a 3

– Os gestores de réplica podem falhar silenciosamente mas não arbitrariamente

– No máximo, existe uma minoria de gestores de réplica em falha em simultâneo

– (TODO)

1. Figura da solução de tolerância a faltas



1. Descrição da figura e breve explicação da solução

O Hub em vez de comunicar diretamente com o PTS comunica com um Front End (FE) e este sim faz a gestão das réplicas do PTS.

Réplicas aceitam apenas operações de leitura ou escrita, por isso PTS FE faz a 'tradução' de operações complexas (exem. incremento) do Hub em operações de leitura e escrita para as réplicas.

Cada réplica, para cada registo guarda o valor e uma tag.

A tag corresponde ao número de sequência de escrita que deu origem ao registo.

FE mantém um contador com número escritas onde mantém o número de sequência para usar aquando a criação de novas tags. Este contado é mantido igual em todas as threads do Hub ao usar *static.*

A cada leitura:

FE

- envia read() para todos

- espera que RT*(read threshold)* respondam

- escolhe resposta com maior tag

A cada escrita:

FE

- constrói newTag com número de sequência de variável interna (referia anteiormente)

- incrementa essa variável interna

- envia write com newTag

- espera WT*(write threshold)* de ACKS

- retorna cliente

O Hub suporta as operações:

* **activateAccount** - ativa uma conta de email

WT = 1

Pois operação não é critica e se não se efetivar pode-se sempre voltar a efetuar (idempotente).

* **loadAccount** - carregar a conta com pontos

WT = 2

Pois operação é critica e não se pode 'repetir' não é idempotente.

* **pointsBalance** - obter pontos de uma conta

RT = 1

Pois operação não é critica e se der resultado desatualizado pode-se sempre voltar a chamar sem consequências negativas (idempotente).

1. Descrição de otimizações/simplificações

No protocolo QC original a tag tinha um Cid.

Na nossa solução a tag não necessita de Client ID pois só há um cliente, o FE.

No protocolo QC original o FE ao fazer uma escrita tinha de fazer primeiro uma leitura para saber o número de sequência para a subsequente escrita.

Na nossa solução este read deixa de precisar de ser feito pois o próprio FE tem um variável interna que usa para incrementar o número de sequência. Isto pode ser feito com segurança pois só há um cliente, o FE.

No protocolo QC todas as escritas são sincronizadas, na nossa solução apenas as escritas na mesma 'conta' são sincronizadas. Sempre que se acede à variável global de sequência de registo há sincronização.

Na nossa solução também alteramos os WT e RT consoante as operações e as suas características específicas (ponto anterior).