

ASD: Trabalho 2

v1-1617

O objetivo deste trabalho consiste em implementar duas versões dum repositório chave/valor em que os valores guardados são do tipo conjunto.

Assim, a interface do repositório consistirá nas seguintes operações de escrita:

- *void insert(String key, String elem)* – insere o elemento *elem* no conjunto associado à chave *key*.
- *void remove(String key, String elem)* – remove o elemento *elem* do conjunto associado à chave *key*.
- NOTA: por simplicidade não se pede que implementem operações para criar ou remover um conjunto associado a uma chave – assume-se que ao inserir o primeiro elemento numa chave, o conjunto será criado caso não exista; ao remover o último elemento do conjunto, o conjunto continuará a existir, mas não terá nenhum elemento.

Adicionalmente, estarão disponíveis as seguintes operações de leitura:

- *boolean isElement (String key, String elem)* – retorna *true* se o elemento *elem* estiver presente no conjunto associado à chave *key*.
- *List<String> list (String key)* – devolve uma lista com os elementos presentes no conjunto associado à chave *key*.

O(s) repositório(s) chave/valor a implementar deve(m) executar num conjunto de N servidores, em que cada chave é replicada em K servidores (com $K < N$, por exemplo, $N = 7$, $K = 3$).

Versão 1

O repositório deve ser implementado recorrendo ao algoritmo *Paxos* para ordenar as operações executadas pelos clientes, de forma a que todas as réplicas dum conjunto tenham o mesmo valor.

As seguintes funcionalidades são opcionais:

- implementação da variante *multi-Paxos*, para diminuir o número de mensagens trocadas na execução do protocolo;
- implementação dum mecanismo para adicionar e remover réplicas durante a execução do sistema (fazendo variar N) e ao conjunto de servidores que replica cada chave (fazendo variar K).

Sugestões: Pode implementar esta versão a partir do primeiro trabalho prático, fazendo as seguintes extensões/adaptações:

1. estender o algoritmo para permitir uma sequência de decisões relativas a uma mesma chave (isto permitirá que para uma chave possa fazer uma sequência de escritas);
2. mapear as operações de *insert(k,v)* e *remove(k,v)* num *write(k,op)*, em que *op=insert(v)* ou *op=remove(v)* respetivamente;
3. (opcionalmente) estender o algoritmo para suportar a variante Multi-

Paxos e a adição/remoção de elementos ao conjunto de réplicas associadas a cada chave.

Versão 2

O repositório deve ser implementado recorrendo a um sistema inspirado no Dynamo (com algumas diferenças), com as seguintes características:

- o conjunto associado a cada chave deve ser guardada num conjunto de servidores.
- Uma operação de leitura é efetuada acedendo ao valor de $R \leq K$ servidores, caso existam servidores com diferentes valores deve combinar o resultado dos vários servidores – sugestão: usando um CRDT conjunto.
- Uma operação de escrita é efetuada em dois passos: (i) lendo o valor do conjunto a partir dum conjunto de $W \leq K$ servidores, combinando os vários valores caso necessário; (ii) aplicando a operação localmente ao estado do conjunto; (iii) escrevendo o novo estado do conjunto nos mesmos W servidores.

Nesta versão sugere-se que o cliente envia as operações para um servidor do sistema, o qual se encarregará de fazer os passos necessários à execução da operação.

Avaliação

Para avaliar o seu trabalho deve executar as suas versões do repositório chave-valor com 7 servidores e 3 clientes, em que cada cliente executa uma sequência de 10000 operações sobre 100 chaves diferentes, com a seguinte proporção: insert: 40%; remove: 10%; isElement: 40%, list: 10%. As chaves são acedidas usando uma distribuição uniforme.

Os resultados a obter devem permitir uma avaliação comparativa entre:

- Versão usando PAXOS vs. Versão usando Dynamo ou opcionalmente
- Versão usando MultiPaxos vs. Usando Dynamo

Notas complementares:

- O projeto pode ser implementado em qualquer linguagem
- Sugere-se a utilização do sistema Akka (valorizável). Neste sistema, um servidor pode ser implementado como um Actor
- Detalhes e instruções para entrega do trabalho e valorização do trabalho podem ser consultadas no sistema CLIP, incluindo o *template*/formato do relatório a entregar.