

Sistemas Distribuídos

*Relatório do Primeiro Projeto*

*Mestrado Integrado em Engenharia Informática e Computação*

*(2 de abril de 2018)*

**Grupo T3G12**

Bárbara Sofia Silva **up201505628**@fe.up.pt

Julieta Frade **up201506530**@fe.up.pt

*Introdução*

Este relatório tem como objetivo explicar detalhadamente a melhoria implementada no protocolo base do projeto: ***backup***. Assim como também, descrever o design escolhido que permite a execução simultânea de protocolos e explicar a sua implementação.

Por fim, o projeto foi desenvolvido no âmbito da unidade curricular de Sistemas Distribuídos.

*Protocolo de Backup*

Relativamente à melhoria do protocolo de Backup, esta tem como objetivo garantir o grau de replicação desejado e consequentemente, poupar memória.

A forma mais eficiente, que o grupo encontrou, de implementar este melhoramento foi instanciar em cada *peer* a classe **Storage**, que por sua vez contém estruturas de dados que auxiliam na análise e gestão da informação. Neste caso, é de salientar a tabela **storedOccurrences**, cuja chave é uma *string* da combinação do ID do ficheiro com o número do *chunk* e o valor é o número de ocorrências da mensagem *STORED*, ou seja, quantas vezes é que esta mensagem foi recebida em relação a um *chunk* específico.

Assim, a melhoria foi implementada invertendo a ordem de operações do protocolo, isto é, sempre que é recebida uma mensagem *PUTCHUNK*, o *peer* espera um tempo aleatório entre 0 a 400ms até começar a escrever o ficheiro. No entanto, antes de o fazer, consulta a tabela **storedOccurrences**, onde tem acesso ao grau de replicação atual do *chunk* em questão e verifica se este é maior ou igual ao desejado. Caso a condição seja verdadeira, o *peer* descarta o *chunk* e aborta a sua escrita, no caso de ser falsa, o *peer* atualiza a estrutura de dados, escreve o ficheiro e envia uma mensagem *STORED*, que por sua vez vai fazer com que todos os *peers* atualizem a sua tabela.

Em suma, esta solução revelou-se ser bastante eficiente, visto a probabilidade de o grau de replicação de um *chunk* ser superior ao desejado ser muito baixa.

*Execução simultânea de protocolos*

Descrever o design que escolhemos, mostrar prints do código e explicar como implementamos. E s t o u f a r t a.

usei synchronized em varias funçoes para as threads nao acederem todas ao mesmo tempo

podes falar tmb no serializable que guarda e faz load do estado do programas

**Describe your concurrency design in the report and refer to your code to explain how you implemented that design. (This description alone is worth up to 5%, and may take two or three pages, as necessary)**