

Relatório de Projeto

Sistemas Distribuídos



Turma 7 Grupo 06:

Carlos Daniel Gomes
Abel Tiago

Aspetos Gerais

Apenas foram implementados 3 protocolos: Back, Restore e Delete. O reclaim permaneceu na sua estrutura inicialmente pensada mas nem perto de ser concluída.

Enhancements

Tentamos fazer a enhancement do Backup porém não tivemos tempo para testar.

Nesta enhancement todos os peers ligados recebem os Stored messages e guardam num ConcurrentHashMap com uma key diferente para cada Chunk Guardado no sistema. Assim, quando recebem a mensagem PUTCHUNK, cada peer vê se possui na estrutura de dados referida já o chunk em questão e se este já atingiu o grau de replicação desejado. Caso já tenha atingido, esta mensagem é ignorada.

Mais nenhum enhancement é posto em prática.

Para além disto é usado RMI para a comunicação entre o Cliente e o Sistema de Peers.

Concorrência

Utilizamos threads para quase tudo, para cada canal de comunicação - em que temos os 3 enunciado especifica - para este ficar à escuta. É criada uma thread por cada mensagem enviada e por cada mensagem recebida. Para lidar com esta quantidade de threads é uma ThreadPoolExecutor na qual são todas as threads inseridas e postas ao cargo desta estrutura. Devido à enorme quantidade de threads, a pool em questão possui como parâmetro 200. Esta estrutura também foi útil para quando era necessário adicionar um tempo de espera aleatório para que alguma função acontecesse. Por exemplo, era necessário esperar um tempo random entre 0 e 400 ms para enviar uma chunk message. Utilizando, o método schedule da estrutura isto era fácil de fazer visto que era só dar o tempo necessário de delay para esta funcionalidade começar a correr. Certas informações são guardadas, como por exemplo: os id's dos ficheiros que um peer faz backup, a quantidade de mensagens STORED recebidas por um peer (cada um guarda a sua informação). As informações que eram mantidas pelo programa situam-se em ConcurrentHashMaps ou ConcurrentSkipListSet que permitem uma maior concorrência no programa. Apesar de não ter sido testado, a cada 2 segundos tentou se guardar as informações suscetíveis do programa através da interface Serializable, uma vês mais, graças à utilidade da estrutura que lida com as threads.