

Sistemas Distribuídos
Augusto Amorim Cravo Silva
Gabriel Cardoso Candal

30 de Março de 2014
MIEIC – FEUP

Enhancements

Enhancement 1

A partir do momento em que um *peer* recebe pela primeira vez a mensagem *PUTCHUNK* regista os ips dos *peers* que responderam com stored.

Quando chegar a sua vez de guardar e responder (após os ms calculados aleatoriamente entre 0 e 400 ms), se o número de *peers* com o chunk for superior ao necessário, não guarda esse chunk nem responde com Stored.

Assim, assegura-se o número mínimo de *peers* com o chunk evitando assim a sobreocupação de espaço que pode ser escasso e por sua vez a ocupação da rede com mensagens *REMOVED*.

Enhancement 2

Para evitar usar todo o canal multicast para o envio de um chunk como resposta a um pedido *GETCHUNK*, e dado que a priori é sabido o endereço IP de destino desse chunk, cada peer a correr esta versão do protocolo estará também à escuta de uma dada porta (que terá que ser previamente acordada entre peers, já que é passada como argumento na linha de comandos).

Do lado do par que inicia o sub-protocolo de restauro, é verificada a versão do pedido *GETCHUNK*, se esta for superior à versão 1.0 será enviado o pacote com a mensagem *CHUNK* para essa tal porta específica do IP do pacote que enviou a *GETCHUNK*; caso contrário, faz o envio pelo canal multicast.

Enhancement 3

Como forma de contornar o problema pensámos em duas hipóteses: os pares periodicamente perguntarem à rede se o ficheiro ainda existia, ou o *peer* que é dono do ficheiro notificar todos os outros, de X em X tempo, que o ficheiro é para ser apagado (até saber, ou ter uma estimativa razoável, de que tal aconteceu). A abordagem adoptada é a segunda.

Quando o utilizador apaga um dado ficheiro, é adicionada a uma tabela uma entrada com o ID do ficheiro e o número de *peers* que possui esse ficheiro – número obtido pelo *merge* dos IPs de todos os chunks do ficheiro.

Um par que receba uma mensagem *DELETE* responde com uma mensagem de reconhecimento *DELETED* com conteúdo semelhante. O dono do ficheiro irá decrementando a contagem da tabela cada vez que recebe uma mensagem deste tipo. Caso a contagem chegue a zero, o ficheiro considera-se apagado e não são enviadas mais mensagens *DELETE*.

Para esta solução parte-se do princípio que os *peers* que possuem o ficheiro são aqueles cujas mensagens de *STORED* chegaram ao *initiator peer*.

Enhancement 4

No caso de o *replication degree* não ser atingido ou descer a baixo do nível necessário, *peers* que tenham o *chunk* tentam voltar a enviá-lo de forma a assegurar o nível de replicação necessário.

Este envio é feito sempre após 8 segundos acrescidos de entre 0 e 1s calculados aleatoriamente. Esta espera serve para assegurar que o *initiator peer* já desistiu de enviar a mensagem pois já fez 5 tentativas.

Após o 1º envio, os envios seguintes seguem o mesmo esquema originalmente definido para o subprotocolo de backup, isto é, tenta-se 5 vezes e o tempo entre as várias tentativas vai sendo duplicado.