Backup Enhancement

Surge um problema quando o grau de replicação de chunks é muito superior ao grau de replicação requerido por parte do chunk, levando a uma ocupação rápida de memória. Para prevenir isto implementamos uma solução traduzida no pseudo-código abaixo:

Sleep(número aleatório entre 0 e 400)

(ir armazenando as mensagens stored relativamente ao chunk)

-sleep ends

If(númeroMensagensStoredRecebidas < númeroDeReplicaçãoDesejado)

    Guardar o ficheiro e enviar mensagem stored

Else

    Já chegou ao numero de replicação desejado portanto não preciso de fazer nada.

Com esta solução, garantimos quase sempre o grau de replicação desejado.

Restore Enhancement

Quando se envia uma mensagem restore para o canal multicast, só um dos peers está interessado na segunda parte da mensagem (body). Logo estamos a sobrecarregar o canal quando a maior parte dos peers não estão interessados na mensagem total ou parcial.

Portanto para resolver este problema, o peer que vai mandar a mensagem stored, envia a mensagem sem o body para alertar os outros peers que o initiator-peer já recebeu a mensagem e estabelece uma conexão TCP com o initiator-peer e envia-lhe o body.

Delete Enhancement

Quando o initiator-peer manda uma mensagem delete, não tem a certeza se os peers que tinham os chunks relativos à mensagem apagaram, ou porque não recebeream a mensagem ou porque estavam inativos.

Para resolver esta questão implementamos uma nova mensagem “deleted” que cada peer envia por cada chunk que apagar. O peer initiator vai contar as mensagens deleted e se a quantidade for igual à soma dos peers que armazenaram cada chunk do ficheiro referido, significa então que todos os peers apagaram o chunk. Senão, multiplica o tempo de espera anterior por 2 (começando em 1s) e volta a enviar a mensagem. O ciclo termina ao fim de 5 tentativas ou se receber todas as mensagens.

Reclaim Enhancement

Quando o initiator-peer inicia o protocolo backup (motivado por uma mensagem “removed” ou por opção do cliente), pode terminar (“crashar”) antes da conclusão do protocolo. Assim o número de replicação pode ficar abaixo do desejado.

Para resolver este problema e caso o peer implemente uma versão diferente da versão “1.0”, cada vez que o peer inicia, procede-se a um início do protocolo backup de cada chunk que tiver o número de replicação inferior ao desejado. Assim podemos renovar a informação relativa a cada chunk que possa não ter ficado bem concluido quando ocorreu o “crash”.