

Enhancement: Backup

O subprotocolo de backup sem alterações guarda sempre um chunk desde que tenha espaço, mesmo que outros peers já o tenham guardado em numero suficiente para satisfazer o replication degree do chunk. Isto leva a que o espaço alocado para guardar chunks seja usado rapidamente, o que posteriormente leva a que muitos peers tenham que remover chunks da memória, levando a uma quantidade aumentada de tráfego.

Para resolver este problema decidiu-se que apenas alguns peers deveriam guardar os chunks imediatamente. Os peers têm uma probabilidade (um terço no nosso caso) de guardarem o chunk imediatamente, enviando o packet 'STORED'. Os peers que não guardaram o chunk, aguardam um quantidade de tempo aleatória entre 0 e 400ms. Se ao final desse período tiverem recebido respostas 'STORED' suficientes para satisfazer o replication degree do chunk indicado, descartam o chunk, caso contrário aumenta-se a probabilidade de eles guardarem o chunk e repete-se o processo até a probabilidade ser de 100%.

O protocolo continua a garantir que o chunk é gravado, pois os peers chegam sempre a 100% de probabilidade e acabam por guardar o chunk antes do peer iniciador realizar os 5 timeouts.

Enhancement: Deleção

Assume-se a situação em que um pacote Delete é enviado para uma rede. Um *peer* que tem uma cópia do ficheiro, porém, não está em rede, não recebendo portanto esse pacote.

O *enhancement* tem como objectivo permitir que o utilizador possa verificar se, enquanto esteve fora da rede, foram ordenadas as deleções de algum ficheiro que tenha em backup.

Para permitir interoperabilidade entre outras implementações do protocolo, o *enhancement* faz uso dos pacotes pré-existentes *GETCHUNK*, *CHUNK* e *DELETE*. Espera-se portanto que outros protocolos que interajam com este tenham implementado a compatibilidade básica requerida no enunciado com esses pacotes.

Quando se liga novamente à rede, o protocolo encarregue do programa envia um pacote *GETCHUNK* para a rede por cada ficheiro que tiver em backup, contendo o *fileID*, o nome do ficheiro, e o número de *chunk* zero. O número do chunk é irrelevante, mas o uso do zero garante que, se existir, então tem pelo menos um, garantindo assim uma resposta.

O protocolo espera então por receber um pacote *CHUNK*, contendo o *chunk* pedido atrás. Se, de facto, receber o pacote *CHUNK*, então o ficheiro existe algures, e o protocolo passa a descartar este pacote e os seus conteúdos e passa simplesmente ao próximo ficheiro em lista.

Caso não receba de imediato o pacote *CHUNK*, o protocolo permite mais duas tentativas, com tempos de espera crescentes. Se mesmo assim não receber o pacote *CHUNK*, então passa a emitir um pacote *DELETE*. Dado que os outros *peers*, já não têm nenhum *backup* do ficheiro, simplesmente descartam este pacote, e o protocolo apaga o seus *chunks* guardados.

