Lan Cloud

Enhancements Report

Mestrado Integrado Em Engenharia Informática e Computação

Sistemas Distribuídos

Turma 1, Grupo 2:

Duarte Manuel Ribeiro Pinto - 201304777

José Miguel Botelho Mendes - 201304828

Faculdade de Engenharia da Universidade do Porto

Rua Roberto Frias, sn, 4200-465 Porto, Portugal

02 de Abril de 2016

Index

Introduction

After implementing all of the protocols, we noticed some could be enhanced, so we implemented a greater version of these protocols. As the original was 1.0, we called the enhancement version 1.3. We’ve implemented the backup, restore and delete enhancements but left out the reclaim.

1. Backup

After implementing the backup protocol in the default version, we came to the conclusion that the way it was implemented could deplete the space allowed for backup rapidly. So, in our enhancement for the backup, we check if the number of peers that are already storing the chunk is greater or equal to the desired replication degree and if it is, the peer that received the PUTCHUNK message does not store it.

2. Restore

Right after we implemented the Restore protocol, we noticed that sending the chunk through the multicast channel was unnecessary because not all of the peers were requesting that chunk and it was overloading the channel. So, what we decided to do consists on still send the GETCHUNK message through the multicast channel but the peer that sent it includes a port (as an extra argument) to connect to via TCP. The peers that have the chunk that was requested will then send the chunk through that TCP connection.

3. Deletion

After implementing the original Deletion protocol, we came across a problem. If a peer sent a DELETE message and a peer that had the chunk wasn’t online at the time, he wouldn’t delete the chunk. In order to solve that, the peer that sent the DELETE message will keep on sending that message until all the peers that are storing the chunk answer with DELETED. The time between messages will keep on rising until it won’t rise any more (but it will keep sending the message). We also implemented a new message that is exactly like the DELETE message but is named CONFIRM_DELETED, for the peers that were offline.