Assignment 3
P2P
Datanet 2015

Christian Enevoldsen

Department Of Science, Copenhagen University

This document covers the design of a simple Peer program for the Kascade p2p protocol. It starts with a brief introduction to p2p followed by the design face. Next it discusses some of the bottlenecks of the program and finally answers some theoretical questions.

Categories and Subject Descriptors:

General Terms: Programming, Experimentation

Additional Key Words and Phrases: HTTP, SERVER, P2P, PEER, TRACKER

1. INTRODUCTION

Peer-to-peer (short, p2p) is a networking protocol for partitioning tasks among multiple peers. This is usually used for sharing data, without the need to have a central server. Data can be everything from files to game stats. In this experiment we are going to implement a simple peer that will be able to download and distribute a set of files.

2. DESIGN

The peer is divided into 4 submodules: The Client, Kascade, Peers and Server.

2.1 Client

The client's responsibility is to keep track of Kascade files, update them and download the contents from other Peers.

2.2 Kascade

Kascade is mainly an abstraction of the cascade files. There's a kascade loader which helps loading in the kascade files and creates Kascade objects from them. Kascade objects contains all information stored in the file and also additional information such as peers.

2.3 Peer

The peer module is also an abstraction. A Peer object contains information about the peers that you get from the tracker.

2.4 Server

The server module instantiates an Express node server which will serve other Peers, such that they can download data from the client.

2.5 Libaries and Frameworks

The Peer uses 3rd party libraries and a few build in ones.

—express - for networking

—fs - for reading and writing files

—crypto - for hashing

3. BOTTLENECKS

The peer has troubles loading the kascade files sometimes. This might be due to the files not being closed or opened correctly. The workaround is to restart the computer or wait some minutes and try again. Files cannot be written or read by multiple threads so this has an impact on the download time. The Kascade file system is not very well compressed so this has a significant impact in the downloading time, since it takes some seconds to load a big file.

4. THEORY QUESTIONS

Is TCP the best choice for the tracker communication?

UDP is better because it's stateless and doesn't cause overhead on packets. Using UDP can reduce this overhead by approximately 50

Is TCP the best choice for the peer communication?

When downloading data any package loss can cause significant errors, so since TCP is resending dropped packages this might be a better solution.

Is HTTP the best choice for the tracker communication?

For the purpose of this assignment HTTP works great, but in the real world it is too slow.

Is HTTP the best choice for the peer communication?

Direct socket stream with custom events would be better.