CS 550: Advanced Operating Systems

Work realized by

Florentin Bekier
Rémi Blaise

P2P File Sharing System: Performance Results

Taught by
Dr. Zhiling Lan

Master of Computer Science, Spring 2020

Contents

1 Introduction **3**

2 Single peer **3**

3 Multiple peers **3**

4 Conclusion **4**

1 Introduction

To verify that our system is capable of withstanding different load intensities, we have carried out performance tests.

We set up 3 peers on the same machine, each sharing at least 10 files ranging in size from 1 to 20 kB to verify that the file transfers ran smoothly.

We have also implemented bash scripts to simulate a large number of sequential and concurrent queries. The results are detailed in the following sections.

2 Single peer

We created a script to automate 500 sequential file searches and downloads with the same peer. This script can be found in our source code at this path: `peer/test`. The requested file is 10 B size. Performing this test allows us to measure the time for those 500 operations and calculate the average response time.

We obtained a total time of 13.651 s, thus the average time for each search and download is **27.3 ms**. This result is really low since our Index server and Peer are located on the same computer but it shows us that the Index server performs quite well to process client search requests.

3 Multiple peers

We also created another more complex script that allows us to simulate multiple clients and making them simultaneously search and download a 1 MB file from the same peer. The script then measures the response time so we can compare the results for different values.
You can run the script by executing this command: `test/concurrential 500`.

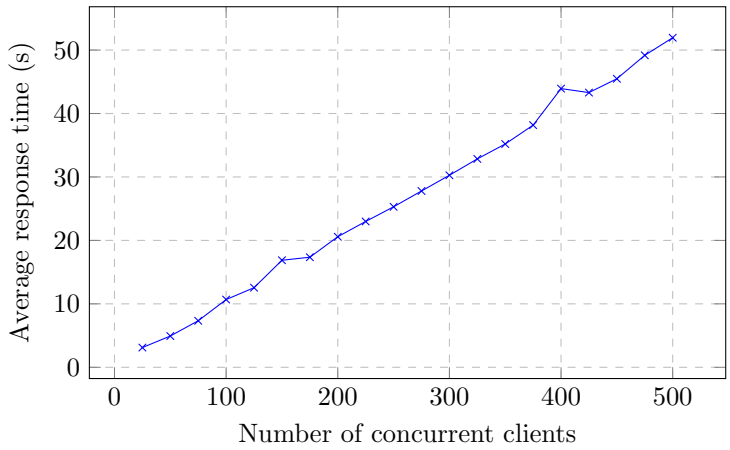

Figure 1: Measured average response time for different concurrent clients

Our results are summarized in the plot graph from Figure 1. We can see that over 500 simultaneous requests, the time evolution is linear. It shows that the server-Peer can scale almost perfectly until at least 500 requests.

4 Conclusion

Based on those performance results, we can conclude that our system is capable of handling many requests at the same time and therefore works as expected and is ready to be used by several users.

However, complementary tests need to be performed with peers on different computers to make sure that the behavior is not different than this.