CS 550: Advanced Operating Systems

Work realized by

Florentin Bekier
Rémi Blaise

Consistent P2P File Sharing System: Performance Results

Taught by
Dr. Zhiling Lan

Master of Computer Science, Spring 2020

Contents

1 Introduction

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

We deployed 10 super-peers, each connected to 3 leaf-nodes. All the super-peers and leaf-nodes were on the same machine. Each leaf-node shared at least 10 files ranging in size from 1 to 20 kB to verify that the file transfers ran smoothly.

We also made an experiment to measure the percentage of invalid query results that come back. The results are detailed in the following section.

2 Measurements

The figure below presents our results in a graph after performing the experiment.

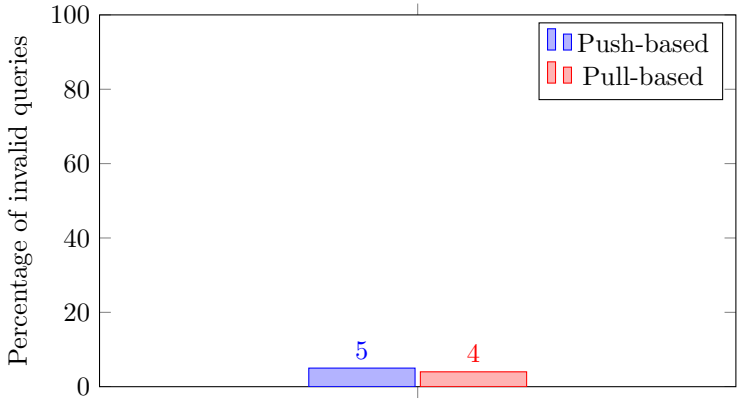

Figure 1: Measured proportion of invalid queries for the different strategies

Based on those results, we cannot see a significant difference between the push-based and pull-based approaches in the experiment. However, in real-life conditions it is likely that there will be more differences.

3 Conclusion

The push-based and pull-based approaches present some differences of conception that we will detail here. In the push-based approach, an invalidate request is broadcasted through the network to every single leaf-node even if this peer doesn't have a version of the file. We can easily see that this is not efficient at all because a lot of unnecessary requests are made. However, this is a simple approach and because the request is sent as soon as a change is made we are sure that the file version stays consistent. In the pull-based approach, it is the peer (leaf-node or super-peer depending on the approach) that polls the origin server for a newer version. In this approach, the origin server needs to handle many requests and the file version is not directly consistent. This will depend on the value of the TTR that must be as small as possible to ensure the consistency but as big as possible not to make too many unnecessary requests to the origin server. The TTR can be chosen based on the average frequency of

modification of a file in the network. If the TTR is wisely chosen, it can be more efficient than the push-based approach and perform fewer requests.
The push-based approach will be preferred when a change must be propagated through the network as soon as it is made and the pull-based approach will be preferred when there can be some (little) delay for the propagation.