Distributed Computing, Peer-to-Peer and GRIDS - ID2210

Amir H. Payberah Fatemeh Rahimian
amir@sics.se fatemeh@sics.se

April 27, 2010

Contents

1 Project Outline

The goal of this project is to implement and evaluate an algorithm for a P2P file sharing system. The project has two main parts:

1. Implementing a file sharing systemm, which is introduced in section 2.
2. Running extensive simulations and writing a report on the findings. For this part you will be given the implemention of the system.

2 The File Sharing System Specification

This section explains the behaviour of a *file sharing (FS)* system. The goal of FS is to disseminiate a file to all the nodes in the system.

Assume there is one single file in the system. This file is split into a number of equal size data chunks, called *peices*, and is initially owned by one node, i.e., *seed*. Other nodes (*leechers*) that come later want to download this file, by exchanging the peices. There exists one *tracker* in FS that knows about all the nodes in the system and the pieces that each one has in its local buffer. A leacher node notifies the tracker everytime it downloads a peice. Therefore, the tracker always knows which peices are downloaded by a node, and which peices are missing. A leecher node becomes a seed node when it downloads all the peices of the file.

When a node (seed or leecher) joins the system, first of all it informs the tracker of its arraival by sending a `RegisterPeer` message to it. Each node, P, periodically checks whether it is a seed or not. If P is not a seed (does not have the whole file), it measures the number of its free download slots:

$$freeDownloadSlots = indegree - numOfUploaders$$

If `freeDownloadSlots` is greater than zero, it sends a separate `GetUploaderRequest` message for each free slot to T, e.g., if P has two free download slots, it sends two messages to T. If T can find a node Q such that Q has a piece that P does not, it sends back a `GetUploaderResponse` to P and informs it about the Q's address and the piece number that P can download from Q. If P is not already connected to Q, sends a `HandshakeRequest` to Q. Whenever node Q receives the handshake message, it checks if it has reached its maximum outdegree.

$$freeUploadSlot = outdegree - numOfDownloaders$$

As a reply, Q sends back a `HandshakeResponse` that contains the status of handshake: *accept* or *reject*. If Q accepts P's request, P adds Q to its uploaders list and starts downloading the selected piece.

To download a piece, node P sends a `DownloadRequest` to Q and Q sends back the requested piece to P, after receiving the message. When downloading a piece is completed, node P receives the `DataMessage` message. Whenever P receives this message, it should close its connection to the uploader, Q, by sending a `CloseConnection` message to it. It also should update its buffer status in the tracker by sending a `UpdateBuffer` message to T.

3 Your Tasks

This is what you should do for this assignment.

3.1 Task One

Your first task is to implement the FS system in Kompics. Attached to this document you are given the skeleton of the code that you can complete, but feel free to implement the system from scratch if you are interested. In the given code, the class `Snapshot` keeps the state of all nodes' buffer (buffer maps) and prints them out periodically. To update the state of buffer in `Snapshot`, you should explicity call `Snapshot.addPiece`, whenever a node completely downloads a piece. The simulation terminates whenever all the nodes become seed. In this assignment we assume there is no failure in the system, so the nodes do not use failure detector. In addition, since the nodes register their address in the tracker, we do not use the bootstrap server as well.

3.2 Task Two

The main idea of task 2 is to analyze the behavior of FS in different network configurations and scenarios. You can change the configuration of the system through `peerConfiguration`, which is defined in `Configuration.java`. The `PeerConfiguration` has six parameteres in the following order: (i) download bandwidth, (ii) upload bandwidth, (iii) indegree, (iv) outdegree, (v) number of pieces and (vi) piece size.

In this task you are supposed to answer to the following questions:

- **Question 1:** How does changing the upload bandwidth influence the download time in all nodes?
  (Hints: Assume there are 200 nodes in the system, the fils is divided into 30 pieces and the size of each piece is $64Kb$. Assume the indegree equalds outdegree equals four. Set the download bandwidth of nodes to $512Kbps$ and change the upload bandwidth of nodes: $16Kbps$, $32Kbps$, $64Kbps$, $128Kbps$ and $256Kbps$. Draw a plot such that the X-axis shows the upload bandwidth and the Y-axis shows the time to download all pieces by all the nodes.)

- **Question 2:** How does changing the piece size influence the download time in all nodes?
  (Hints: Assume there are 200 nodes in the system and the fils is divided into 30 pieces. Assume the indegree equalds outdegree equals four. Set the download bandwidth of nodes to $512Kbps$ and upload bandwidth to $64Kbps$. Change the piece size: $16Kb$, $32Kb$, $64Kb$, $128Kb$ and $256Kb$. Draw a plot such that the X-axis shows the piece size and the Y-axis shows the time to download all pieces by all the nodes.)

- **Question 3:** How does changing the indegree influence the download time in all nodes?
  (Hints: Assume there are 200 nodes in the system, the fils is divided into 30 pieces and

the size of each piece is $64Kb$*. Assume the outdegree equals four, the download bandwidth of all nodes is* $512Kbps$ *and upload bandwidth is* $64Kbps$*. Change the indegree of nodes: 1, 2, 3, 4 and 5. Draw a plot such that the X-axis shows the nodes indegree and the Y-axis shows the time to download all pieces by all the nodes.)*

- **Question 4:** How does changing the outdegree influence the download time in all nodes?

  (Hints: Assume there are 200 nodes in the system, the fils is divided into 30 pieces and the size of each piece is $64Kb$*. Assume the indegree equals four, the download bandwidth of all nodes is* $512Kbps$ *and upload bandwidth is* $64Kbps$*. Change the outdegree of nodes: 1, 2, 3, 4 and 5. Draw a plot such that the X-axis shows the nodes indegree and the Y-axis shows the time to download all pieces by all the nodes.)*

- **Question 5:** How does changing the size of network influence the download time in all nodes?

  (Hints: Assume the fils is divided into 30 pieces and the size of each piece is $64Kb$*. Assume the indegree and outdegree of all nodes equal four. Set the download bandwidth of all nodes to* $512Kbps$ *and upload bandwidth to* $64Kbps$*. Change the size of network: 50, 100, 150, 200, 250. Draw a plot such that the X-axis shows the number of nodes in the system and the Y-axis shows the time to download all pieces by all the nodes.)*