

CMPSC 441
Distributed Systems
Spring 2016

Laboratory Assignment Two: Performance Evaluation of File Transfer Using Sockets

Introduction

As mentioned in a previous laboratory assignment, the ability to implement and evaluate software that supports communication between a client and a server is a crucial laboratory skill that will support further work in the field of distributed systems. In this laboratory assignment, you will use two classes that perform client-server communication through the use of a “socket” in the Java programming language. In particular, you will focus on understanding and enhancing a program that uses sockets to perform a file transfer on both a single node and two distinct nodes. You will also perform an experimental evaluation of the performance of using sockets for client-server file transfer. Using either Markdown or the L^AT_EX text formatting language, you will write a scientific report explaining the performance results that you identified during this assignment. Finally, you will compare your finished file transfer system to the peer-to-peer systems described in the textbook.

Review Your Textbook

Before starting this assignment, you should again read the content in Chapters 1 and 2 of your textbook. As you did for a previous assignment, please make sure that you focus on the discussion of different types of communication middleware on page 24. Also, you should review all of the material in Section 2.2, paying particularly close attention to Figure 2-3 and the description of the interactions between a client and a server. Finally, please make sure that you read and understand all of the content about decentralized architectures in Section 2.2.2. Students who would like to learn more about two real-world file transfer systems are encouraged to visit <https://www.getsync.com/> and <https://www.dropbox.com/> to learn about BitTorrent Sync and Dropbox, respectively.

File Transfer through Client-Server Communication

Since you configured your connection to the “share” repository for this course in a previous laboratory assignment, you should be able to change into the `cs441S2016-share` directory and type the “`git pull`” command to gain access to the two Java classes needed to complete this assignment.

Now, please find the `labs/lab2/` directory in our course’s repository so that you can study and understand the source code for the files called `FileSocketClient` and `FileSocketServer`. What are the key aspects of how these two Java classes support file transfer? For instance, you will notice that these Java classes specify items like a “host” and a “port” — as in the previous laboratory assignment, make sure that you understand the meanings of these terms. Can you draw a technical diagram that illustrates the general interaction between a client and a server, customized for these two classes that uses sockets? Please see the instructor if you have questions about these issues.

You should now fully comment all of the source code in these two Java classes, making sure that you correctly explain how all of the methods and lines of code work. If you consult online references to support the development of your comments, then please include a reference to those sources in the

comments. Once you have finished compiling these two classes, please run the `FileSocketServer` in one terminal window and then attempt to have the `FileSocketClient` connect to it and transfer a file. For now, you should run both of these programs on the same computational node. Once you have gotten the client-server-based file transfer to work correctly, record the output of the client and the server to demonstrate that they are working. For instance, you should make sure that the file that the server receives and saves is the same as the one that the client attempted to transfer.

If you are interested in exploring this further, please try to run the client on one computer and the server on another computer. Students should be aware that, depending on their setup, they may need to enhance these two Java classes to fully support remote communication. For instance, right now the programs are configured so that the server can run on “localhost” — if you want to get this to work in a remote fashion then you will need to change this name to the “fully-qualified name” of a computer on the Alden Hall network. Also, as mentioned in the previous laboratory assignment, the network in Alden Hall is configured to only allow connections using certain ports; if your program is currently connecting to a different port then the file transfer will not work correctly.

Also, you will notice that the `FileSocketClient` and `FileSocketServer` have some “hard-coded” values in them. For instance, the port on which socket communication takes place can only be defined if a programmer changes the value in the source code and then recompiles the Java class. Additionally, it is important to note that this client-server program is currently hard-coded to support the transfer of a single PDF file — note that both the name of this file and the actual size of this file are currently included in the system’s source code. Since this is not ideal from either a usability or an experimentation perspective, you should revise the Java source code so that it accepts command-line arguments for any values that you deem to be configurable by the user. While you may use the `args []` array to accept these parameters, students are encouraged to consider the use of the JCommander library available at <http://jcommander.org/>.

Finally, it is worth investigating the ways in which this program will handle various types of user error. For instance, you should run your client program by specifying a file that does not exist on your file system. Next, you should have the client and the server attempt to communicate on different ports. Another idea would be to configure the client to look for the server on the “localhost” node when it is in fact running on a remote computer. After brainstorming ideas with the course instructor and the other members of the class, you should systematically try to “break” the `FileSocketClient` and `FileSocketServer` and record what happens. Please use your experiences in this phase of the laboratory assignment to, whenever possible, derive some principles about the fundamental challenges associated with implementing and testing distributed systems.

Evaluating the Performance of File Transfers

Using the `FileSocketClient` and `FileSocketServer` you should transfer multiple files — of sizes that you deem to be “small”, “medium”, and “large” — as you run the client multiple times to collect many timings and then calculate, at minimum, arithmetic means and standard deviations. For this laboratory assignment, you should focus on benchmarking local client-server communication on the same computational node. However, if you wish to potentially earn extra credit you should consider also running their experiments to measure the costs of remote communication.

You should organize all of your empirical results into tables of data. While not absolutely required, you may consider preparing graphs of your results using the R language for statistical

computation. Next, you should analyze the results in attempt to find and explain patterns in the data. Overall, what do your results show you about the cost of performing client-server-based file transfer? Once you are finished running these experiments, can you identify any ways in which you could have improved the performance of the client or the server? Finally, you should write a detailed report, using either the Markdown or the \LaTeX text formatting language, that introduces the design of your experiment and your research questions, explains how you conducted the experiments and analyzed the results, and then presents and analyzes the results. Please see the course instructor if you have questions about how to write a scientific report using \LaTeX or Markdown.

Comparing to Client-Server and Peer-to-Peer File Transfer

Section 2.2.2 of your textbook mentions “decentralized” architectures for implementing distributed systems. Many of the students in this class may have heard of various decentralized systems for performing file transfer. For instance, the BitTorrent Sync system allows someone to “keep in sync” files on a laptop, desktop, tablet, and mobile phone — all without the need for direct communication through a centralized server. How does a system like BitTorrent Sync work? How is Sync different than both Dropbox and the system that you implemented for this laboratory assignment? What are the benefits and drawbacks associated with using Sync for file transfer? Students who would like to learn more about BitTorrent Sync may ask the course instructor for a demonstration.

Summary of the Required Deliverables

This assignment invites you to submit printed and signed versions of the following deliverables. Additionally, all of these deliverables must be in a `cs441S2016-<your user name>` repository that you created for this class; please make sure that you share this repository with the instructor.

1. The commented source code of the Java classes that perform client-server-based file transfer.
2. Using both text and diagrams, a description of client-server-based file transfer with sockets.
3. A description of the challenges associated with performing client-server communication.
4. A comparison of file transfer methods that use “centralized” and “decentralized” approaches.
5. A detailed paper that reports on the empirical results arising from performing file transfers.
6. A description of the challenges that you encountered when completing this assignment.

In adherence to the Honor Code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else’s work. With the exception of the provided source code, deliverables that are otherwise nearly identical to the work of others will be taken as evidence of violating the Honor Code. This means that, for instance, all of the other comments, source code, data, and written reports should be the original work of the student completing this assignment. Students who have questions about the Honor Code and how it applies to this laboratory assignment should schedule a meeting with the course instructor before this assignment’s due date.