CMPSC 441
Distributed Systems
Spring 2016

Laboratory Assignment One: Using Sockets to Explore Client-Server Communication

Introduction

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 several programs that perform client-server communication through the use of a "socket" in the Java programming language. Additionally, you will implement four programs that will support the experimental evaluation of the performance of using sockets for client-server communication on a single host. Finally, using either Markdown or the LaTeX text formatting language, you will write a detailed scientific report explaining the performance results that you identified while finishing this assignment. You will complete this project by working with a partner; you and your partner will collaborate through a Slack channel and a Git version control repository throughout the week during which you finish this assignment.

Review Your Textbook

Before starting this laboratory assignment, you and your partner should read and discuss the content in Chapters 1 and 2 of your textbook. As you review this material, please make sure that you focus on the discussion of different types of communication middleware on page 24. Also, you and your partner should study 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.

Access the Course Repository

Throughout this semester, the course instructor will deliver code to the students in this class through a Git repository, hosted by BitBucket. During this assignment and subsequent assignments, we will securely communicate with the Bitbucket.org servers that will host all of our projects. In this assignment, we will perform all of the steps to configure the accounts on the departmental servers and the Bitbucket service. Throughout the assignment, you should refer to the following Web site for additional information: `https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101`. As you will be required to use Git in the remaining laboratory assignments and during the class sessions, please be sure to keep a record of all of the steps that you complete and the challenges that you face. You are also responsible for communicating with a partner to ensure that each of you is able to successfully complete each of the steps outlined in this assignment.

1. If you do not already have a Bitbucket account, then please go to the Bitbucket Web site and create one — make sure that you use your `allegheny.edu` email address so that you can create an unlimited number of free Bitbucket repositories while you are a student.

2. If you have never done so before, you must use the `ssh-keygen` program to create secure-shell keys that you can use to support your communication with the Bitbucket servers. Follow

the prompts to create your keys and save them in the default directory (press "Enter" after you are prompted: "`Enter file in which to save the key ... :`", then press "Enter" twice if you do not wish to create a passphrase at this time or type your selected passphrase if you do). Type `man ssh-keygen` and talk with your partner to learn more about this program. What files does `ssh-keygen` produce? Where does this program store these files by default?

Once you and your partner have created your ssh keys, you should, if necessary, raise your hand to invite the course instructor to help you with the next steps. First, you must log into Bitbucket and look in the right corner for an account avatar with a down arrow. Click on this blue link and then select the "Manage account" option. Now, scroll down until you find the "SSH keys" option and upload your ssh key to Bitbucket. You can copy your SSH key by going to the terminal and typing "`cat ~/.ssh/id_rsa.pub`" command.

3. Now, you need to test to see if you can authenticate with the Bitbucket servers. First, show the course instructor that you have correctly configured your Bitbucket account. Now, ask your instructor to share the course's Git repository with you. Open a terminal window on your workstation and change into the directory where you will store your files for this course. If needed, you can open a terminal window and type the command "`mkdir cs441S2016`", thus making a `cs441S2016/` directory that will contain the Git repository that the instructor will always use to share files with you. Once you have done so and, additionally, asked the course instructor to share this repository with you, please type the following command "`git clone git@bitbucket.org:gkapfham/cs441s2016-share.git`".

   If everything worked correctly, you should be able to download all of the files that you will need for this laboratory assignment. Please resolve any problems that you encountered by first reviewing the Bitbucket documentation and then discussing the matter with a teaching assistant. If you are still not able to run "`git clone`", then please see the course instructor and work with your partner to resolve this issue. One problem that students commonly confront is the incorrect addition of their SSH key to the Bitbucket system.

4. Using your terminal window, you should browse the files that are in this Git repository. In particular, please look in the `cs441s2016-share/labs/lab1/` directory and use `gvim` to study the Java classes that you find. Remember, the "`cd`" command allows you to change into a directory. Make sure that both you and your partner can access these files and then make sure to copy them to the shared repository, which you will setup in the following step.

It is worth noting that many of the students in this class may have already configured `git` and Bitbucket as a means for sharing source code. If this is the case for both you and your partner, then you may skip all of the previous steps and create your own repository that you will use to complete this laboratory assignment. Students who have not used Bitbucket in the past should work with the course instructor and their partner to ensure that they can use this important tool throughout the remainder of the semester. Again, once you have gained access to the `cs441S2016-share` repository, you and your partner should create a new repository called `cs441S2016-lab01-<first user name>-<second use name>`. Finally, make sure that both individuals in your partnership, and the course instructor, have access to the repository that you will use for this assignment. Students who have questions about the use of the Git version control system should ensure that they have resolved them before the completion of this first laboratory assignment.

Exploring Client-Server Communication

Please find the `labs/lab1/` directory in our course's repository and work with your partner to study and understand the source code for the files called `FactorizationSocketClient` and `FactorizationSocketServer`. What are the key aspects of how these two Java classes support client-server communication? For instance, you will notice that these Java classes specify items like a "host" and a "port" — what are 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 and your partner should work together to 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 `FactorizationSocketServer` in one terminal window and then attempt to have the `FactorizationSocketClient` connect to it. For now, you should run both of these programs on the same computational node. Once you have gotten the client-server communication to work correctly, record the output of the client and the server to demonstrate that they are working. If you are interested in exploring this further, please try to run the client on the computer of one partner and the server on the computer of another. Students should be aware that, depending on their setup, they may need to enhance these two Java classes to fully support remote communication.

Finally, you will notice that the `FactorizationSocketClient` and `FactorizationSocketServer` 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 recompiles the Java class. Since this is not ideal from either a usability or an experimentation perspective, you and your partner should revise the 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/`. Please see the instructor with questions about this task.

Implementing and Using a Benchmarking Framework

Now, you and your partner should study the code for the `LSSocketClient` and the `LSSocketServer` — these Java classes will serve as the start of a simple benchmarking framework that you and your partner will implement. The "`LS`" abbreviated in the names of these classes is for the words "Large" and "Small" — what about the client-server interaction in this system can be classified in this fashion? After discussing this question with your partner, you should implement six additional Java classes respectively called, `SLSocketClient`, `SLSocketServer`, `SSSocketClient`, `SSSocketServer`, `LLSocketClient`, and `LLSocketServer`. As you are implementing all of these Java classes make sure that the ways in which these clients and servers interact adhere to the names that you have assigned to the classes. Overall, these programs will enable you to establish some "baseline" measurements for the cost of performing client-server communication on a single computer. Next, you and your partner should implement one final program that is more like the `FactorizationSocketClient` and `FactorizationSocketServer` — that is, it should perform some type of "useful" computation on the server whenever the client makes the appropriate request. Teams that are unsure of what type of additional computation to implement should take time to brainstorm ideas with the instructor.

As you are creating your programs, make sure that you give them command-line arguments that will effectively support running experiments and collecting data about the response time of the client-server interaction. Once you have implemented all of the Java classes that you will use in your benchmarking framework, you and your partner should divide up the work associated with running your experiments. For each of the four "baseline" client-server configurations, you should run the client multiple times to collect many timings and then calculate, at minimum, arithmetic means and standard deviations. Then, you should run both of the "useful" computations (i.e., the one for factorization and the one that you have implemented with your partner), again collecting multiple timings and calculating arithmetic means and standard deviations. For this laboratory assignment, you and your partner should focus on benchmarking local client-server communication on the same computational node. However, teams that wish to potentially earn extra credit should consider also running their experiments to measure the costs of remote communication.

You and your partner 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 communication? 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 and your partner 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 then presents and analyzes the results.

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 the repository that you created for this assignment.

1. The well-commented source code of the Java classes that form the two "useful" benchmarks,
2. The well-commented source code of the Java classes for the four "baseline" benchmarks.
3. Using both text and diagrams, a description of client-server communication with sockets.
4. A detailed paper that reports on the empirical results arising from the use of the benchmarks.
5. 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 two members of the partnership. Students who have questions about the Honor Code and how it applies to this assignment should schedule a meeting with the course instructor before this assignment's due date.