CMPSC 441
Distributed Systems
Spring 2016

Laboratory Assignment Four: Client-Server Communication with Compression

Introduction

In many distributed systems, a client is dynamically configured to communicate with a server by first downloading the necessary code from a repository. In this laboratory assignment, you will investigate how the use of compression influences the response time and throughput associated with these transfers of code and data from a repository to the client. In particular, you will learn how to perform compression using at least three different algorithms (e.g., Zip, Pack200, and Pack200+Gzip) and you will also discover how to start and monitor an HTTP server using Python.

Next, you will pick several real-world code and data repositories of compiled Java programs (i.e., they should not contain source code) and conduct experiments to measure how compression changes the performance characteristics of a distributed system. 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.

Prepare for the Assignment

Before starting this laboratory assignment, you and your partner should read and discuss the content in Chapter 3 of your textbook. As you review this material, please make sure that you focus on the discussion of code migration in Section 3.5, paying particularly close attention to Figure 3-17 and the content that explains this technical diagram. To better understand the Pack200 compression technique, you and your partner should download from the ACM Digital Library and read "Compressing Java Class Files", a paper published by William Pugh in a past edition of the Proceedings of the International Conference on Programming Language Design and Implementation. Finally, you should study the manual pages for the `jar` and `pack200` programs.

Exploring Client-Server Communication with Compression

To start this assignment, you and you partner should work together to find a "Java Archive", or Jar, file that contains only Java bytecodes and the other resources (e.g., images or data files) that are needed to run a Java program. Please make sure that you are not picking Jar files that contain Java source code; to guarantee that your experiments directly connect to the scenario depicted in Figure 3-17 the chosen archives should only contain Java bytecodes and other types of files that would support the execution of a program. In addition to running experiments with the Jar files that you and you partner find, I have provided two files to you that you can obtain through the "share" repository for this course; please type the "`git pull`" to download these files.

What is the size of the two Jar files that I have provided to you? Can you report the size of these files in bytes? Once you know their correct size, go ahead and learn more about how to run the

`pack200` command in your terminal window. For example, if I wanted to perform both Pack200 and Gzip compression on one of the provided Jar files, I could type the following command "`pack200 JavaRunner.pack.gz JavaRunner.jar`" in the terminal. Now, which of the two archives for the JavaRunner program are bigger? Can you explain why this is the case? After studying the manual page for `pack200`, can you learn how to perform Pack200 compression in a standalone fashion?

Note that you already have these files, such as `JavaRunner.jar`, in a Jar file format. If you want to discover the performance of the `jar` compression technique, then you will have to first decompress a Jar file and then run the command again in compression mode. If you want to learn more about how to run `jar`, please read its manual page or talk with the instructor. Since it is important to quantify the time required for compression with either the `jar` or the `pack200` commands, you should also ensure that you know how to preface one of your previously executed command-lines with either `time` or `/usr/bin/time`. Using one of these tools, can you determine how long it takes to compress the code and data associated with your chosen Java programs? Finally, you and your partner should discuss an equation to calculate a "percentage reduction" and then use this equation for these files. Which method is the best at compressing the code and data?

So far, you have not actually transferred these archives across the network, as is depicted in Figure 3-17 of your textbook. To accomplish this task, please change into the directory that contains all of your archives and type the command "`python -m SimpleHTTPServer 4200`" to start an HTTP server using the Python programming language. Now, you can use the `wget` command in another terminal window to download a compressed archive. You and your partner should now discuss equations and means for calculating both response time and throughput. Once you have agreed on the correct equations, you should use them to compute these metrics for some of the archives. Which compression technique tends to yield the best values? Can you explain why? Please remember that if you want to transfer files with HTTP from one computer in the Alden Hall network to another, then you need to use certain ports that are not blocked by the firewall.

Empirically Studying Compression Methods

You and your partner should create a new repository called `cs441S2016-lab04-<first user name>-<second use name>`. Next, make sure that the 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 leaving the laboratory session today. Finally, please make a Slack channel, for your team and the instructor, so that you can easily communicate while completing this assignment.

If you and your team member did not already do so, please select eight Java programs, in addition to the two already provided to you by the instructor, to support your empirical study of compression methods. You should try to select Java programs that have archives that vary in their size and contents. For instance, please avoid selecting programs that all result in small archives or that only contain Java bytecodes. Next, create a directory structure in your repository that will make it easy to store all of these archives and then later transfer them using an HTTP server.

For this experiment, you will investigate the use of the Zip, Pack200, and Pack200+Gzip compression techniques. First, you will measure the time overhead associated with running each of these methods for the ten chosen Java programs. Next, you should calculate the percentage change in size when compression with Zip is considered the "baseline" technique and either Pack200 or

Pack200+Gzip is the "improved" approach. Additionally, you should time how long it takes to download these archives using the `wget` command as it connects to the `SimpleHTTPServer`. Finally, please calculate the response time and throughput associated with transferring the various Java archives. Whenever your experimental design requires you to collect a data value that is subject to variation, you should conduct multiple trials and report averages and standard deviations. Finally, you should consider all of your data points (e.g., compression times, response times, and throughputs) to determine when to use compression during client-server communication.

At minimum, 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 an attempt to find and explain patterns in the data. Overall, what do your results show you about the costs and benefits of performing compression during client-server communication? Once you are finished running these experiments, can you identify the best ways to use compression to improve the performance of this communication? In addition to merely noting the trends in your data sets (i.e., "compression technique A is the best for Java program P"), you and your team member should always attempt to use background resources to explain why these trends are evident in your data sets. 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. A detailed listing of the commands that you typed to perform compression and file transfer.
2. Using both text and diagrams, a description of client-server communication with compression.
3. A full-featured description of all of the equations for this assignment's evaluation metrics.
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.
6. A detailed listing of the tasks completed by each of the member of your partnership.

In adherence to the Honor Code, students should complete this assignment only with their team member. 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 commands for the terminal, 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 member 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.