CMPSC 441
Distributed Systems
Spring 2016

Laboratory Assignment Three: Exploring and Evaluating Parallel Computation

Introduction

Since processes, or "programs in execution", form the building blocks of distributed systems, it is important to learn how to create and run processes on an operating system. To ensure that distributed systems exhibit the best throughput and response times, it is also critical that these processes are, whenever it is appropriate to do so, executed in a parallel fashion. In this assignment, you will use two techniques for parallelizing computations on a single node. Next, you will conduct an experiment to determine how response time changes as the parameters of the parallel computation are modified. Finally, using either Markdown or the LaTeX text formatting language, you will write a report explaining the performance results that you identified during this assignment.

Review Your Textbook

Before starting this assignment, you should again read the content in Chapters 1 and 2 of your textbook so that you have a strong understanding of the basic concepts in the field of distributed systems. Next, please read the content in Section 3.1, paying particularly close attention to the content about processes and threads. As you review Section 3.1, make sure that you understand the trade-offs associated with using parallel threads and processes in a software system. Students who want to learn more about the `xargs` and `parallel` programs that are part of this laboratory assignment are also encouraged to read their manual pages by using the `man` program.

Performing Parallel Image Conversion

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 many files that are needed to complete this assignment. Now, change into the `labs/lab3/` directory and explore the files that are available in the `scripts` and `images` directories. First, please make sure that you can find the files called `convert_forloop.sh`, `convert_parallel.sh`, and `convert_xargs.sh`. What do these scripts do?

Once you have studied these scripts, you will see that they perform image conversion using the "`convert`" command. This particular set of scripts uses `convert` to produce "thumbnails" of larger images. This is a common task that needs to be completed if you are creating a web site that will showcase a large collection of images with a page that gives small — or "thumbnail" — previews of the images themselves. While you could create these thumbnails by repeatedly running the `convert` program in the terminal window, this would be time consuming and error prone.

As an alternative, these scripts automatically create thumbnails for all of the images in a directory. One of the scripts uses a `for` loop to perform the conversion while the other two respectively use the `parallel` and `xargs` programs in an attempt to speed up the entire process. Now, change into the directory called `images/slides/` and run the `convert_forloop.sh` script (you will prob-

ably need to type `../../scripts/convert/convert_forloop.sh` to get this command to work correctly). How long did it take to perform the conversion? Now, please try out the other two scripts as well. Which approach to image manipulation is the fastest? Can you explain why?

Empirically Evaluating Sequential and Parallel Computation

Using the provided scripts, you are responsible for conducting an experiment to evaluate the performance of sequential and parallel image manipulation. You should explore the performance tradeoffs as you start with directories containing a few images and then gradually increase to ones with many images. To support the population of these directories, please use the `download_dogs.sh` and `download_cats.sh` scripts provided in the `scripts/download/` directory. As part of your experiment, you should identify and explain scenarios in which parallel computation does not provide any appreciable benefits. Using knowledge gathered from Linux manual pages, you should also explore how different parameters to the `parallel` and `xargs` influence the efficiency of the computation.

You must 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 parallel image manipulation? Once you are finished running your baseline experiments, can you identify any ways in which you can improve the performance of the parallel computations? 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 evaluated 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.

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 final version of the scripts that you used to conduct the performance evaluation.
2. Using text and diagrams, a description of sequential and parallel image conversion methods.
3. A description of the challenges associated with performing experiments with image conversion.
4. A detailed paper that reports on the results arising from performing image conversion.
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 student completing this assignment.