CMPSC 440
Operating Systems
Spring 2014

Laboratory Assignment Seven: Evaluating the Efficiency of File System Searching

Introduction

Since most software applications ultimately need to store and retrieve information, it is essential that an operating system provide a file system to support these activities. It is also important to ensure that application programs effectively use the primitives provided by the file system. In this laboratory assignment, we will explore and experimentally evaluate two file system searching tools, `grep` and `ag`. One of the tools that we will study, `grep`, was first implemented years ago by Ken Thompson, one of the creators of Unix. The second tool, known as `ag` or the "Silver Searcher", is a new tool for file system searching. As part of this assignment, you will learn the command lines associated with both of these tools and then design a benchmarking framework to determine which one is faster. Finally, you will explore at least one advanced application of these tools.

File System Searching with `grep`, `ag`, and `nautilus`

Before you start using these two tools, you should learn about their history. For instance, you should learn exactly when the tools were implemented and who initially created them. Next, you can run the `man` command in your terminal window and study its output to ensure that you understand the command lines that both tools support. What are the basic commands for searching a specific directory with these two tools? Finally, you should type `nautilus` in your terminal window to run Ubuntu's graphical file system browser. How does `nautilus` support file system searching? Overall, which tool(s) seem to furnish the most robust set of searching features? Why?

Experimentally Evaluating File System Searchers

When you are implementing a large software system, like an operating system or a compiler, it is important for you to have the ability to quickly search through a large collection of files for those that contain a specified word or pattern. For this part of the laboratory assignment, you should find five separate directories of source code that contain a substantial number of files. Next, you should use a Linux command to determine how many files are in each of the chosen directories; please make sure that you clearly explain how you counted the total number of files.

The next step in the experimental evaluation of `grep` and `ag` is to devise at least five different command lines that you can run for each of the tools. Then, for each of the five commands and the five different directories containing code, you should evaluate the efficiency and effectiveness of the tools. In particular, you should evaluate the searchers according to the following metrics: (i) the number of results produced, (ii) the usefulness and accuracy of the output, and (iii) the speed at which the tools produce their output. When conducting the experiments to evaluate efficiency, please run multiple trials and report arithmetic means and standard deviations. After you have completed this basic experiment, you should develop and run one additional study. For instance, you could compare the performance of file system searchers that are run in both your home account (e.g., `/home/g/gkapfham/`) and on a local disk (e.g., the `/tmp` directory on your workstation).

Analysis of the Searching Results

Now that you have defined your benchmarks and run your experiments, you should analyze the searching results. Beyond identifying which tool is faster and the circumstances in which it is better, you should also aim to explain, whenever it is possible to do so, why these results are evident. If you had to pick one of these tools for searching both large and small directories of source code, which one would you select? Why? As you complete a report on the results from your empirical study, make sure that you give all of the relevant details that would be needed to support the replication of your results. Finally, you should comment on whether or not it would be possible to perform the same type of file searching with `nautilus`, Ubuntu's graphical file system browser. Please see the instructor if you have questions about completing these analysis tasks.

Advanced Use of File System Searchers

File system searching tools like `grep` and `ag` have a wide variety of interesting applications. For instance, the Vim text editor can be extended with CTRL-P, a plugin that supports fuzzy file finding. You can learn more about CTRL-P by visiting the following Web site: `http://kien.github.io/ctrlp.vim/`. One critique of CTRL-P is that it is only fast enough for regular use when it leverages caching. However, users will often invalidate the CTRL-P cache when they create new files, forcing the cache to be rebuilt with a time consuming additional scan. One resolution to this problem is to configure CTRL-P to use an external user command such as `ag`. As such, an example of an advanced use of file system searchers that you could report on is the configuration of Vim to use an enhanced version of CTRL-P that leverages `ag`. Alternatively, you could install and explain how to use the `ag.vim` plugin that makes it possible to run `ag` searches directly from Vim. Since Vim also provides integration with the `grep` command, you could explore this advanced feature as well. Students may also look into advanced features that do not involve the use of Vim.

Summary of the Required Deliverables

This assignment invites you to submit printed and signed versions of the following deliverables:

1. A description of the history, features, and drawbacks associated with `grep`, `ag`, and `nautilus`
2. The command lines that you used to run all of your benchmarks with `grep` and `ag`
3. Screenshots demonstrating that you could correctly configure and run `grep`, `ag`, and `nautilus`
4. A comprehensive report that explains the results and responds to all of the stated questions
5. The source code and/or documentation for an advanced use of the file system searching tools
6. Ideas for future experiments and analyses that use and evaluate the file system searchers
7. A reflective discussion 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. As such, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.