

**CMPSC 441
Distributed Systems
Spring 2016**

Laboratory Assignment Eight: Benchmarking Distributed and Local File Systems

Introduction

As noted by Tanenbaum and Van Steen, “distributed file systems form the basis for many distributed applications.” In this laboratory assignment, you will create and run your own file system benchmark on both a local and a distributed file system. Then, you will analyze the results arising from multiple runs of your benchmark to assess the relative performance trade-offs associated with the use of local and distributed file systems. Next, you will use an established benchmarking program to conduct further experiments to evaluate the performance of the same file systems. Before conducting your benchmarks, you will carefully read a journal paper, published in *ACM Transactions on Storage*, that reports on the challenges and best practices in the field of file system and storage benchmarking. Additionally, you will learn more about how leading experts in the field suggest that you evaluate the performance of a file system. Ultimately, you will develop a full-featured understanding of the trade-offs related to both file systems and their benchmarking.

Review Your Textbook

Before starting this assignment, you should review Chapters 1 through 4 of your textbook. Since this laboratory assignment builds on the fundamentals developed in these chapters, you should resolve question about this content before starting this laboratory assignment. Next, you should examine the material in Sections 11.1 through 11.4 that discusses the issues related to the design, implementation, and use of a distributed file system like the Network File System (NFS). Students should pay close attention to the content in Section 10.1 that introduces the basic NFS architecture for a Unix-based system. Although it is not this assignment’s emphasis, students who want to learn more about topics such as consistency, replication, fault tolerance, and security in distributed file systems should review Sections 11.5 through 11.8. Please see the instructor if you have questions.

Leverage Past Experience in File System Benchmarking

The authors of the paper “A Nine Year Study of File System and Storage Benchmarking”, published in *ACM Transactions on Storage*, make the following statement in the abstract of their paper: “We found that most popular benchmarks are flawed and many research papers do not provide a clear indication of true performance.” To learn more about the claims made in this paper written by Traeger et al. please go to the ACM Digital Library and download its PDF and, to ensure that you can cite this paper in your final report, its BibTeX reference. Students should carefully read this entire paper before implementing and running all of their file system benchmarks; please see the course instructor if you have questions about this paper. To start, it is worth noting that this paper develops two underlying themes, which the authors summarize in the following fashion:

1. “Explain what was done in as much detail as possible.”
2. “In addition to saying what was done, say why it was done that way.”

Additionally, the authors suggest that there are three main types of benchmarks. As you are completing this laboratory assignment, please make sure that you are always following the two main suggestions of this paper and that you are aware of the type of benchmark that you are either creating or running. When you are creating your own benchmark or running an existing tool to perform benchmarking, you should first think about the research question that you are attempting to answer. Then, you need to clearly state a hypothesis for this research question and, finally, define the metrics and experimentation protocol that will enable you to evaluate the hypothesis and respond to the question. Please see the course instructor if you have questions about this step.

In addition to reading the paper published in *ACM Transactions on Storage*, you should study some of the many useful online resources that explain further how to rigorously conduct performance benchmarking. For instance, Brendan Gregg works at Netflix and he has written a series of informative blog posts about benchmarking. Please read the articles that Gregg has written to define and compare the concepts of “active” and “passive” benchmarking. What type of performance benchmarking does Gregg advocate? Why? As you investigate these concepts and respond to these questions, please make sure that you read the article that Gregg published about active benchmarking of file systems with the `bonnie` and `bonnie++` tools. Finally, as you work on this assignment you should take to heart the following assertion that Brendan Gregg made in one of his recent presentations: it can “take one to two weeks to debug a single benchmark”!

Create and Use Your Own File System Benchmark

Now that you have investigated and better understood some of the benefits and challenges associated with benchmarking local and distributed file systems, you are ready to create and use your own file system benchmark. Using the programming language of your choice, you should implement your own benchmark after following the preparatory steps outlined in the previous sections of this assignment sheet. For instance, before you start programming your benchmark, you should formulate a research question, state a hypothesis, define evaluation metric(s), and prepare a experimentation protocol. Then, you should create your benchmark and run it so that you can evaluate the performance characteristics of storing and retrieving data through the use of distributed and local file systems. Students who want to run their benchmark on a local disk should point it to the `/tmp/` directory. Otherwise, if you want to store files in a NFS mount, you can store the files in your home directory. Finally, students should classify the benchmark that they created according to the scheme outlined by Traeger et al. in “A Nine Year Study of File System and Storage Benchmarking”. Please see the instructor with any questions about this step of the assignment.

Judiciously Use a File System Benchmarking Tool

Of course, there are a wide variety of file system benchmarking tools that have already been implemented and are suitable for comparing the performance of distributed and local file systems. Two noteworthy examples of tools are `bonnie` and `bonnie++`, the second of which is installed on our laboratory computers. This tool provides many parameters that make it possible to collect performance metrics about the performance of a file system. For instance, `bonnie++` can help you to estimate the throughput, in KB/sec, of operations such as sequential input and output, random seeks, and sequential and random file creation. This tool also comes with a support program that allows you to format its output — which is standardly presented as a comma-separated value (CSV)

file — in an easy-to-read table. For instance, here is one command that specifies a computer has 2 GB of total memory and that the program should use the `/tmp/` file system to run a series of experiments with output eventually being formatted in an HTML-based table.

```
bonnie++ /tmp/test -r 2048 -u gkapfham | bon_csv2html > /bonnietest.html
```

In this final phase of the laboratory assignment, you should follow the procedure outlined in the previous sections of this document to learn more about the performance of distributed and local file systems. As you work on this part of the assignment, it is critically important to remember that the network file system in Alden Hall is a shared resource used by all of the students and faculty of the Department of Computer Science. While it may be tempting to run `bonnie++` in many different configurations with the hope that some interesting results might ultimately emerge, students should avoid taking this approach when running on a distributed file system. This method will place too much load on our NFS servers and potentially prevent others from completing assignments in this and other classes — in addition to yielding results that are likely incorrect and misleading.

Instead, students should first learn how to use `bonnie++` by running experiments on the `/tmp/` directory that is stored on the local file system. Then, you should conduct a series of preliminary experiments — after, of course, defining research questions and all of the other machinery needed to conduct a meaningful experiment — while exclusively using the local disk. Before running experiments on the network file system, a student should use Slack to ensure that no other students are also running experiments for this assignment. Once it is clear that no other experiments are currently being performed, the student should use Slack to mark the start of a new campaign of experiments, conduct the necessary experiments, and then again use Slack to declare that all experiments are finished. Please see the instructor if you have questions about this procedure.

Summary of the Required Deliverables

This assignment invites you to submit printed and signed versions of these deliverables. All of these deliverables must also be in a `cs441S2016-<your user name>` repository created for this course.

1. The well-commented source code and output for your own file system benchmark.
2. Using both text and diagrams, a description of all benchmarks used in your experiments.
3. A document that summarizes the published paper that explains storage system benchmarking.
4. A summary paper that explains the key features and parameters of the `bonnie++` tool.
5. A detailed report presenting your experimentation framework and analyzing your results.
6. A reflection on 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.