A. Arpaci-Dusseau
CS739: Distributed Systems

Department of Computer Science
University of Wisconsin, Madison

A Low-Bandwdith Network File System – SOSP'01

1 Introduction

- What is the motivation for LBFS? Why do they feel a network file system is needed?
- What workloads/usage patterns does LBFS target?

2 Related Work

- The authors state that a number of file systems have properties that help them tolerate high network latency and go on to describe AFS, leases, NFS4, and Coda, all of which we have read in class. Do you agree with their characterization of each of these systems?

3 Design

Basics

- The main idea behind LBFS is to not send chunks of files that already exist on the other side of the slow network (whether that is the client or the server). Chunks are identified by a SHA-1 hash over their content. If both the client and server have a chunk with the same SHA-1 hash, they then assume they are the same chunk.

  One challenge is to figure out how a file should be divided into chunks. What are the two "strawman" proposals described in the paper? What are the pros and cons of each?
- What is the related work of Rsync and how does it work? What are its limitations?
- How does LBFS define a chunk? What is the expected size of a chunk? Why is this method attractive for this domain (that is, can you explain the cases shown in Figure 1)?
- What pathological cases can occur and how does LBS handle them?
- The LBFS protocol performs whole file caching and attempts to minimize when a file is fetched from the server using close-to-open consistency with leases. What happens when a client opens a file? What happens when a client closes a file? How does this compare to AFS?
- Figure 2 shows the protocol for how an LBFS client fetches a file from the server. What do each of the steps mean? What work does the server need to do and could it be avoided with different assumptions? Under what conditions might LBFS perform worse than AFS?

- Figure 3 shows the protocol for how an LBFS client writes a file to the server. What do each of the steps mean? Why does the server write to a temporary file? Why does it help performance that the client gets to pick the file descriptor for the temporary file?
- Does the LBFS client need to try sending the whole file as in Figure 3? Could it track which chunks are dirty locally and only worry about sending those back?

Implications and Extensions

- LBFS keeps the mapping of each SHA-1 hash to its chunk (file and offset) in a chunk database. However, this database is treated only as a hint (i.e., it can be wrong). What is attractive about using the contents of the SHA-1 hash database as simply a hint? What extra work do the client and server need to do because of this? Do you think this is a good trade-off?
- Will the LBFS protocol work for finding identical chunks *within* a single file?
- LBFS assumes that there are never collisions of SHA-1 hashes. What would happen if there was a collision? Could a malicious user exploit this in any way?
- The paper explains that a covert channel exists that can leak information; specifically, a malicious client could learn that a server has specific contents in some file by measuring timing differences for the COND_WRITE operation (even though the operation still fails if the user does not have permission to access that file).

  What exactly can the malicious client learn? Does this information leak seem worrisome to you? Should they instead allow sharing across these chunks?

4 Evaluation

- What questions do you think they should answer to evaluate LBFS?
- What is the point of the experiments in Table 1? Is anything discouraging? What do the results in Figure 5 mean?
- What are the workloads used for Figure 6? What is similar about all of their workloads? What is bandwidth normalized to? Are there significant bandwidth savings? Are the results in Figure 7 any different? Any other comparisons you would have liked to have seen?
- Conclusion from Figures 8-10?

5 Conclusions

- What are the strengths of LBFS and the paper?
- What are the weaknesses of LBFS and the paper?