

Greenlee 560 Summary 7

Elliot Greenlee

March 28, 2017

1 Introduction

Disconnected Operation in the Coda File System.

This is source control related.

2 Problems

What problems does this paper try to solve? Why are such problems important?

When working on distributed systems, it is likely that remote failures will occur, causing delays in access to data and ability to work. Workstations at the time were just becoming powerful enough to work standalone, but these were still configured to be dependent on remote resources. There are benefits to the shared repository, but it would be nice to still continue critical work when it is not available.

3 Assumptions

What are assumptions made by this paper? Are they verifiable? Are there logical holes in such assumptions?

The paper assumes that 1-2 days of disconnection are at the normal maximum level for interruption. It seems that this is based on experience. Their observed compensation for this is 100MB of data and about a minute for reconnection.

They specify that the design is optimized for the access and sharing patterns typical of academic and research environments. It is specifically not intended for applications that exhibit highly concurrent, fine granularity data access.

One possibly incorrect assumption is that users consciously choose to disconnect with remote workstations, and this chosen interruption is the same as random interruptions. They assume it is reasonable to get the users help in preparing for random interruptions, but this seems faulty.

4 Solutions

What are the major solutions of this paper? Do you think the solutions in this paper will work for the problem? Do you think the paper evaluates the solutions in a convincing manner? List two limitations of the solutions proposed in this paper, and outline your method to fix them.

Disconnected operation is a mode of operation that enables a client to continue accessing critical data during temporary failures of a shared data repository. This paper describes using data caching to improve availability.

4.1 Limitations

Their design is not for highly concurrent or fine grained data access. On the whole, it seems their system is actually pretty fast and loose, something that was created just to try out rather than as a research experiment. More rigor is needed in a system that handles this sort of consistency work, and the paper suggests many improvements throughout that they did not implement.

The paper seemed to think that conflicts are very rare, and used data from the research group to back it up. However, they noted that everyone in the groups knows they are working on an experimental system, and so were more careful. In my own experience with working together on systems, there are constant conflicts and merges needing to be made. This system does not even provide a merging mechanism, it just fails when concurrent changes are made.

4.2 Discussion

If they did an automatic versioning on disconnections, that is one option. I think it could be modified to have a version number on the server. But Does not address conflicts. Its an optimistic system, which means conflicts can occur, but they just fail if their is a conflict on reintegration.

Changes: Larger files sizes, cache has increased a ton. Most file sharing in this type of environment is assumed to be disconnected, with random connections to merge work. It's a way of bypassing this issue of involuntary disconnects. Usually we just lose access to resources now.

There are also online sharing resources like sharelatex and google docs that allow simultaneous editing. Those resources are external and hardly ever go down though.