*Coda is derived from AFS. What aspects of AFS simplify the design of Coda?*

Answer:

*1) trades off consistency for achieving high availability. Coda borrows design principles from AFS and trades off consistency for achieving high availability. The two techniques of server replication and disconnected operation are used to achieve high availability.*

*2) Whole file caching. Bring over copy of file on open and set up a callback on the server (keep state on server of which clients have file cached). When close, flush changes to server. When server sees changes to file, it breaks any outstanding callbacks so that client will re-fetch the file the next time it opens it. Otherwise, if client still has good callback, it doesn’t have to re-fetch file on open.*

*3) Close-to-open consistency: client when it calls open will get version of file that was last closed.*

Paper Review (Coda):

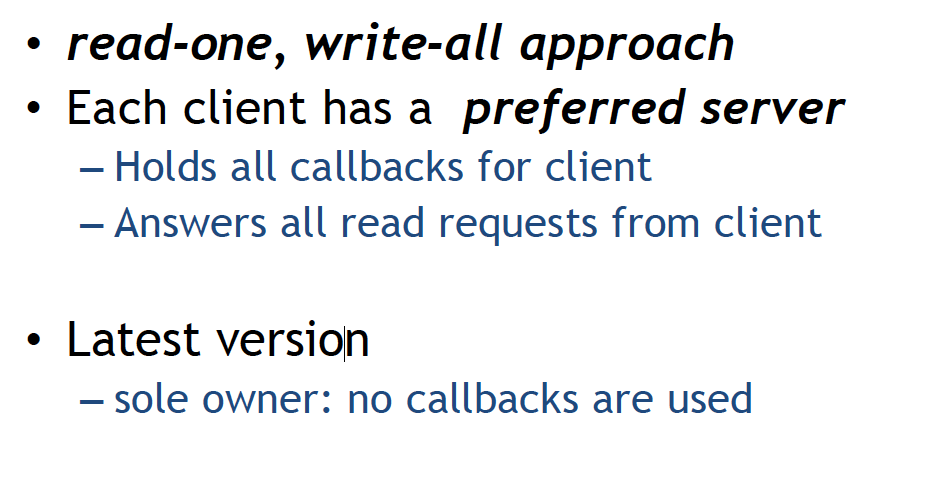
The most important designs of Coda are Server replication and Disconnected operation, the goal of both of them is availability.

*Coda uses a local cache to provide access to server data when the network connection is lost. During normal operation, a user reads and writes to the file system normally, while the client fetches, or "hoards", all of the data the user has listed as important in the event of network disconnection. If the network connection is lost, the Coda client's local cache serves data from this cache and logs all updates. This operating state is called disconnected operation. Upon network reconnection, the client moves to reintegration state; it sends logged updates to the servers. Then it transitions back to normal connected-mode operation.*

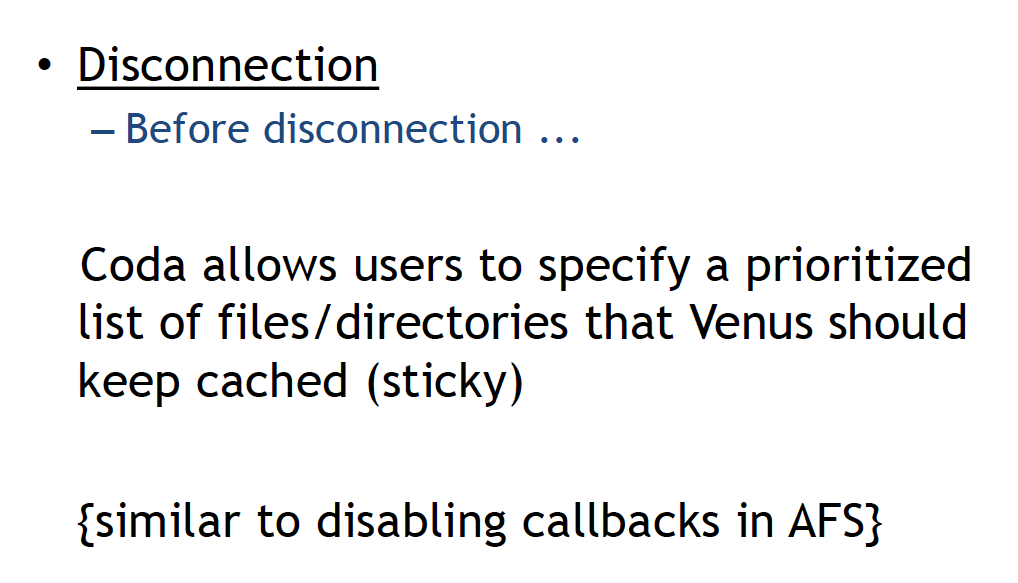
*Also different from AFS is Coda's data replication method. AFS uses a pessimistic replication strategy with its files, only allowing one read/write server to receive updates and all other servers acting as read-only replicas. Coda allows all servers to receive updates, allowing for a greater availability of server data in the event of network partitions, a case which AFS cannot handle.*

*These unique features introduce the possibility of semantically diverging copies of the same files or directories, known as "conflicts". Disconnected operation's local updates can potentially clash with other connected users' updates on the same objects, preventing reintegration. Optimistic replication can potentially cause concurrent updates to different servers on the same object, preventing replication. The former case is called a "local/global" conflict, and the latter case a "server/server" conflict. Coda has extensive repair tools, both manual and automated, to handle and repair both types of conflicts.*

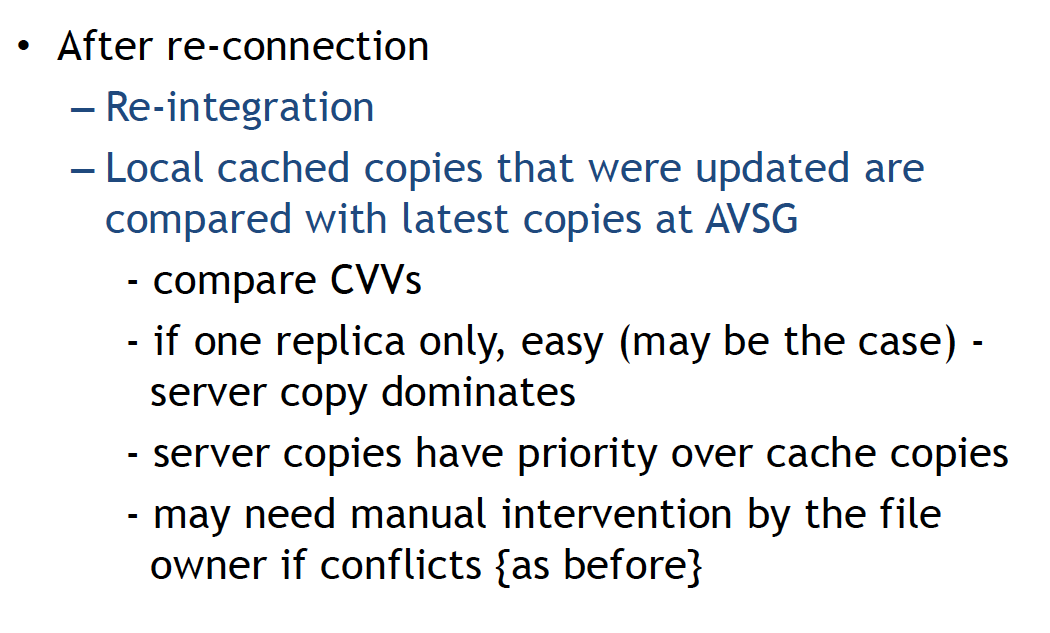
Server replication:



Disconnected operation:



Disconnect re-connect



pro & con

