CODA

- Goals
  - Scalability
    - Clients are loaded to avoid FS bottlenecks.
  - Constant Availability
  - Use with Portable computers
    - Clients local disk is a cache
    - Disconnected operation permitted.
- Naming/Location
  - Namespace is hierarchical and divided into volumes. Volume is set of files/directories on a server
  - Each volume is identified by a unique 92 bit FID
    - Vol. No. + Vnode No. + uniquifier
  - Location information is not explicit in name but stored in volume location database mapping file to FID to location

Operational Semantics

- Single Copy Unix
- AFS-1
  - Open succeeds if latest (F,S)
  - Close succeeds if updated(F,S)
- AFS-2
  - Open succeeds if
    latest(F,S,0) OR (latest(F,S,tau) AND lostcallback(S,tau) AND incache(F).
- CODA
  - Open succeeds if
    If some server reachable then AFS2 else if incache
  - Open fails if servers reachable and conflict or not reachable and not in cache
  - Analogously for close

Replication

- When a volume is created, the number of replicas and the servers storing them are recorded in volume replication database. This set of servers is the Volume Storage Group (VSG). The subset of these accessible by a clients cache system (VENUS) is called Accessible VSG(AVSG)
- Read Once Write All type approach
  - Files are cached on client side on demand. They are obtained from a preferred server in the AVSG. Client verifies with others in AVSG to ensure that Pref. Server has latest data. If not, the server with the latest data is made preferred and AVSG notified about stale data.
  - Callback promises are established between client and preferred server. When notified, a client invalidates data and re-fetches.

- Upon close, a file is transferred in parallel to all servers in AVSG using multicast where available.
- Optimistic Replication requires conflict checks on all server operations
  - State is characterized by update history of an object at a server. This consists of storeids of operations at clients.
  - Coda approximates update history by its length and the Latest storeid (LSID). LSID = clientid+monotonically increasing integer. A CVV (coda version vector) is maintained which estimates (conservatively) update length at every other server.
- Replicas may have
  - Strong Equality (LSIDs and CVVs are equal)
  - Weak Equality (LSIDs same but not CVVs)
  - Dominance/Submission (LSIDs are different, but CVV of one subsumed by the other)
  - Inconsistency