Disconnected Operation in the Coda File System (SOSP'91)

James J. Kistler and M. Satyanarayanan.

ACM Transactions on Computer Systems (TOCS) 10(1), February 1992, 3-25.

Motivation

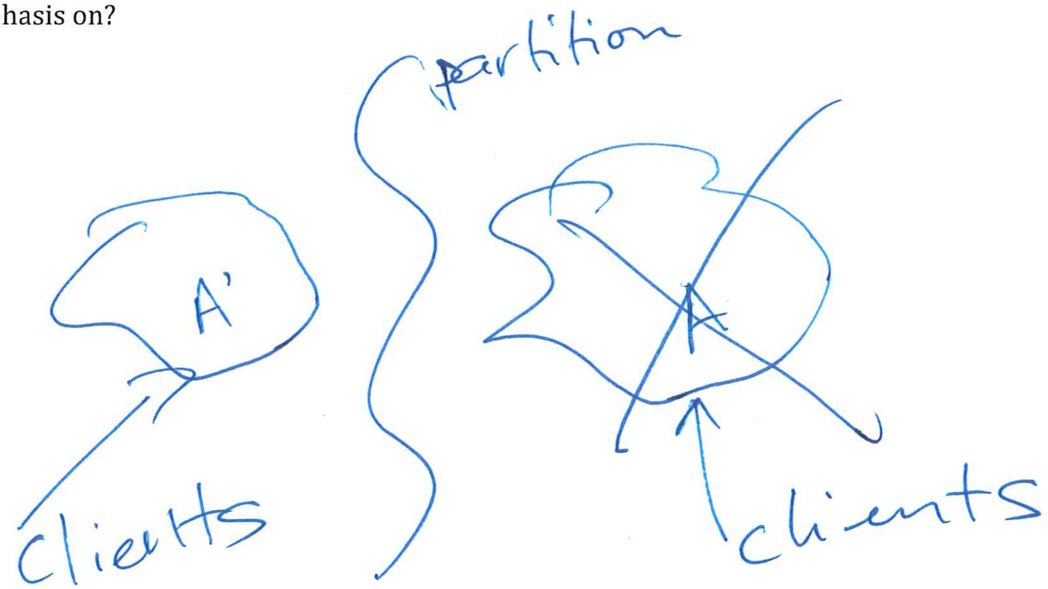
1. What were the goals of Coda? What is the difference between voluntary and involuntary disconnection? What assumptions did Coda make? How good of a job did they do predicting technology trends?

-Disconnected operations

-Voluntary: planned - laptop

-Involuntary: unplanned

2. Replication is often used to increase availability, but there are trade-offs that must be considered. Is it possible to simultaneously achieve perfect consistency and availability when suffering from network partitions? Why or why not? Which does Coda place more emphasis on?

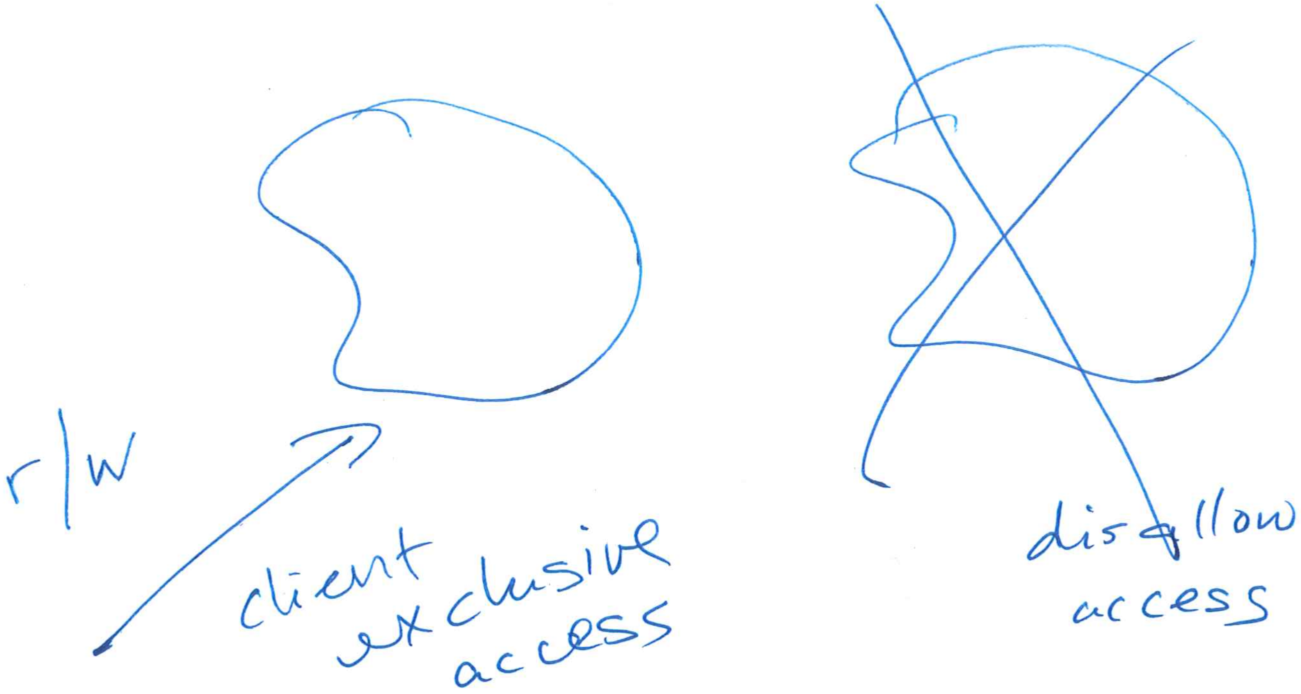


not consistent
or
not available

No!

Availability

3. When a network is partitioned, replicas can be controlled with either pessimistic or optimistic replica control. What is pessimistic replica control? What are the pros and cons of it? Why don't leases solve the problem?



- know to grab lock (voluntary)
- one malicious client can stop all other

Leases:

4. What is optimistic replica control? What are its pros and cons? Why was optimistic replica control chosen in Coda? Can you think of an environment where pessimistic replica control would be more appropriate?

- No conflicts -assumption; Detect
- Conflict: (manual) merge

  ↓

  automatic?
- Little sharing
- Transactional/financial are not good match

5. Coda performs replication on both the servers (VSG, volume storage group) and clients. What are the differences between these two types of replicas? What does Coda do if some, but not all, servers are available?

servers: first-class

superior quality

clients: second-class

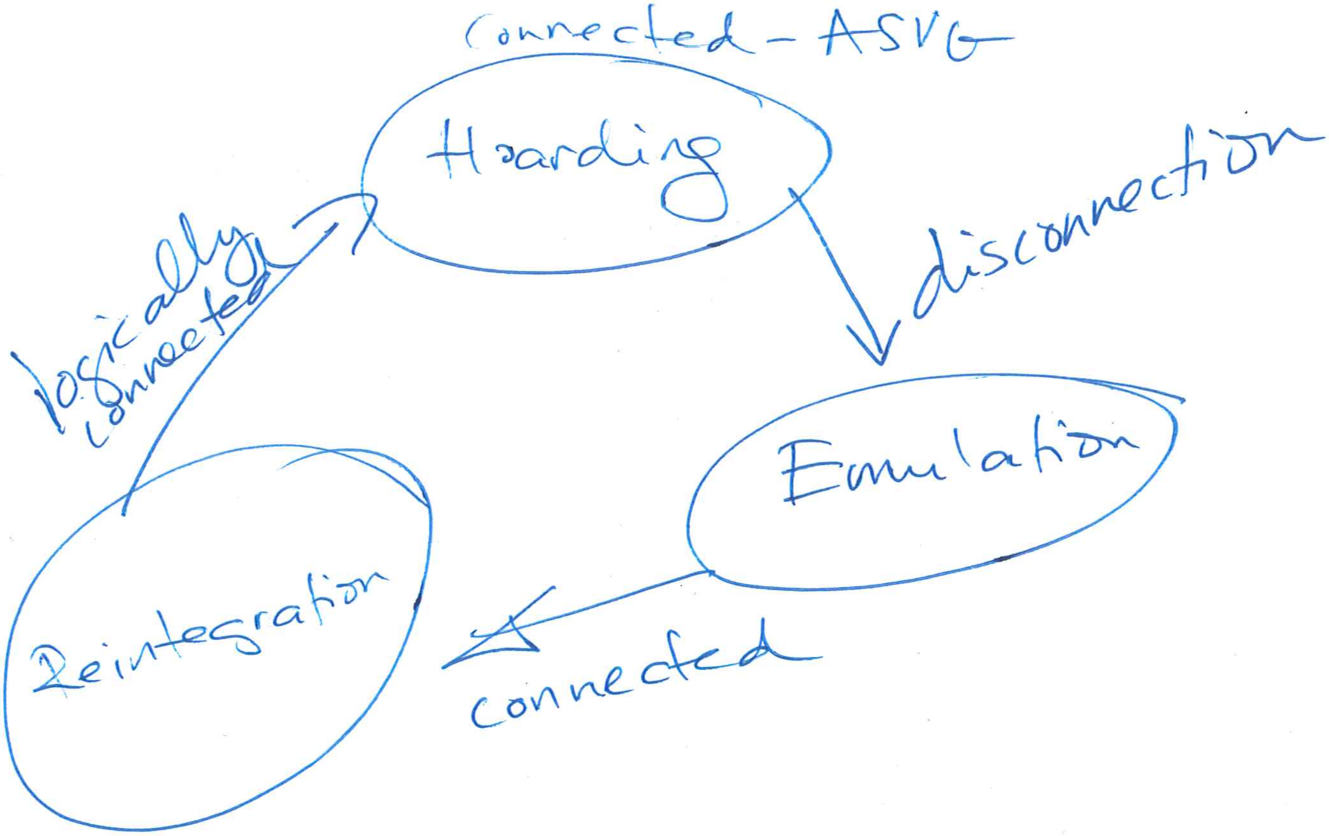
~~up of~~

out of date

AVSG - as long as 1 server

Detailed Design and Implementation

6. Clients are managed by a software layer called Venus. How does the state and behavior of Venus change as the client becomes disconnected or connected?



7. Consider the hoarding state, in which Venus attempts to hoard useful data in anticipation of disconnection. The challenge for hoarding is that the amount of cache space on the clients is limited. During hoarding, what tensions must Venus balance in how it manages the client cache? How does Venus decide what is cached? What information is given infinite priority in the local cache? Why?

- Working set - dynamically - performance
- Essential files - when disconnected - files.

Balance priority of both

Parent directories - contents

8. Is Venus during the hoarding stage identical to AFS? Why might the performance of Coda with Hoarding be worse than AFS?

Essential files: ~~take~~ up disk
less room for working set

Hoard walk: Periodically
File A accessing > Essential File B
equilibrium
prio drops < B

Refetch B - extra work

AFS: callback break?
- next open, refetch file (on-demand)

Coda: callback break
- hoard walk: refetch files that had broken callbacks

9. During emulation, Venus on the client performs many of the actions normally handled by the servers. What types of tasks does this include? How does Venus record enough information to update the servers during reintegration? How does Venus save space? What happens when all space is consumed?

- Disconnected
  - fids
  - record all close to log
  - space savings:
    - don't record every write (close)
    - remove
- No more mods!

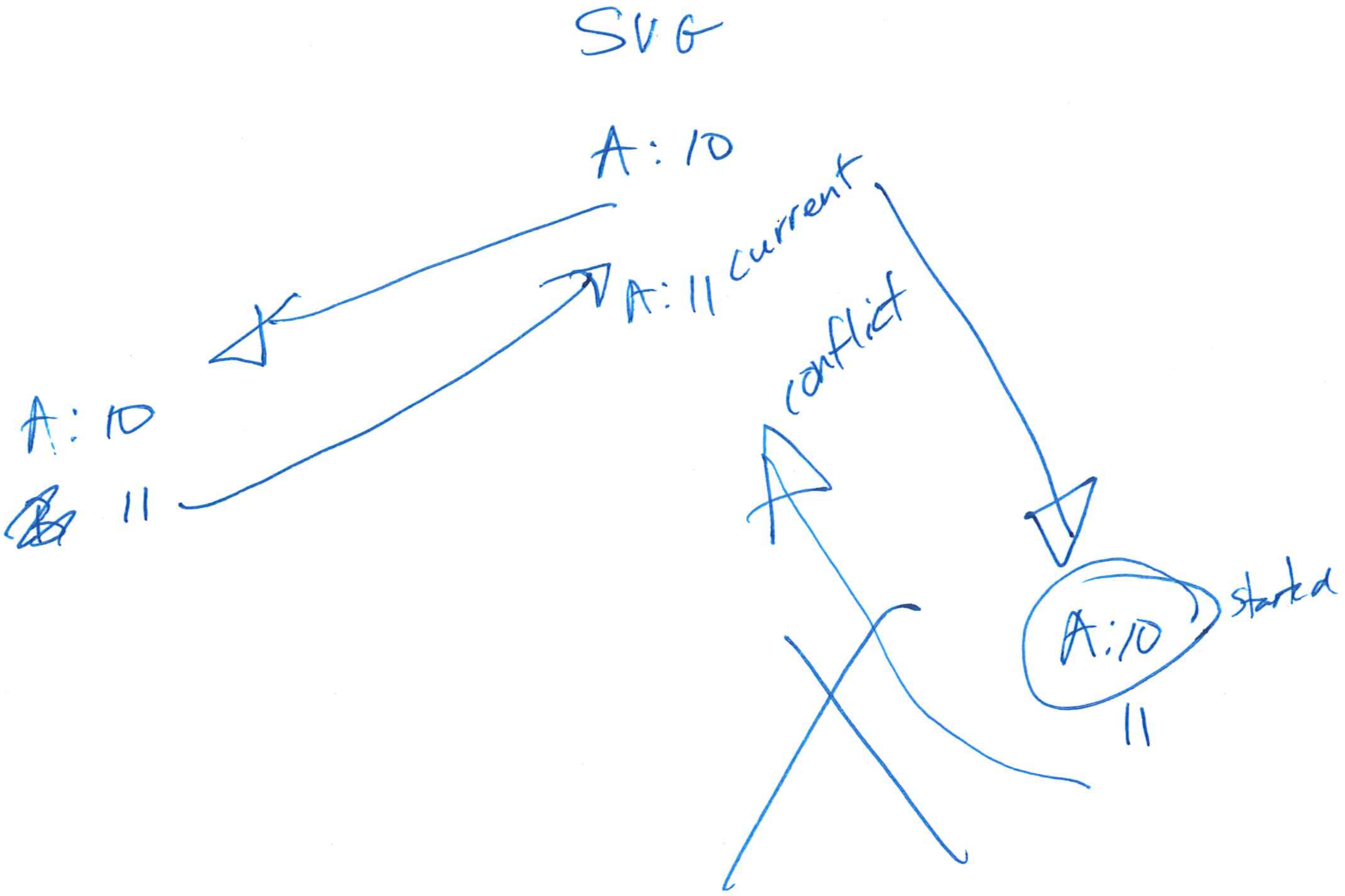
10. During reintegration, Venus propagates changes made during emulation to the servers and updates its cache to reflect current server state. Under what circumstances will the replay fail and how is failure detected? What happens when the replay fails?

-Write | write conflicts

storeid - every file

-inc every mods

-Whole replay fails on conflict



Status and Evaluation

11. They look at 3 questions. How long does reintegration take? How large a local disk does one need? And, how likely are conflicts? Which question do you think has the most impact on whether or not Coda could be successful?

12. About how long is reintegration expected to take? Why is the time for this step crucial? How are technology or workload trends likely to impact this time? Is a design change needed?

1 minute → hours offline

→ bg

→ concurrent

13. How did they determine the size of a needed local disk? How are technology or workload trends likely to impact this? Is a design change needed?

10 hours → 30MB in worst case

no worry

local disks large enough

14. How likely is a conflict during reintegration? Will technology or workload trends impact this? Is a design change needed?

99% mods were by same user

Conclusions

15. Coda handles both voluntary and involuntary disconnection of a client from the network. Where could Coda have made different (or simpler) decisions if they had handled only voluntary disconnection?