Quiz on RPC

- In RPC implementation, server crashes are difficult to handle. Is there any way to differentiate between server crashing before executing the RPC and server crashing after executing the RPC? Why is this important?
- In general, what kind of RPC semantics can be provided regarding server failure?
- Describe how RPC implementations can pass an integer array int a[1024] as a parameter in a RPC call.

Disconnected Operation in the Coda File System

- Motivation
- Design Overview
- Implementation
- Performance

Motivation

- the "AFS server not responding" problem
- Why do we use distributed file system in the first place?
- What would be the ideal situation?
  - providing the ideal sharing for read-only files
  - providing the ideal sharing for user's home directory
  - providing the ideal sharing for truly shared files

Coda file system

- server replication
  - volume storage group
  - accessible volume storage group
- caching for availability
  - when disconnected, rely on cache for file contents
  - problems:
    - what do the servers see
    - what do the clients see

Design Considerations

- placing of functionality on clients
- avoid system-wide consensus
- predictable and un-predictable disconnected operations
- server replication versus client replication

Handling Server and Client Failure in Callbacks

- Client power up/down
  - lossing invalidation message
- Server failure
  - consistency semantics
- Network failure
  - temporary partition
  - long-term partition

Replica Control

- The big problem: how do we handle consistency???
  - strict consistency: acquiring locks
    - variation: acquiring locks with leases
  - optimistic synchronization: let writes go, and resolve conflicts later
  - what needs to be provided at the file system level?

Venus (client) Revisited

- Why did they make venus a user process
  - easy debugging + better portability versus performance loss
  - Minicache: reducing the performance losses
    - for example, caching recent stat calls
- States of the AFS client
  - connected (*hoarding*)
  - disconnected (*emulation*)
  - reconnected (*reintegration*)

In the Connected State

- user specification: hoarding profile
- balancing the user of the cache
  - hoard walk

In the Disconnected State

Upon Re-Connection

- Reflecting file modifications
  - replay the log
- Handling conflicts
  - handling directory conflicts
  - handling file conflicts

Evaluation

- Re-Integration time
- Type of modification conflicts
- Cache size consideration

General Approaches to Handle Network Unavailability

- Fault tolerance or high availability
  - geographically fault tolerance
  - double the resources
- Server replication
- Disconnected operations:
  - laptops
  - pilots
  - application-specific conflict resolution