Distributed System

Fault Tolerance

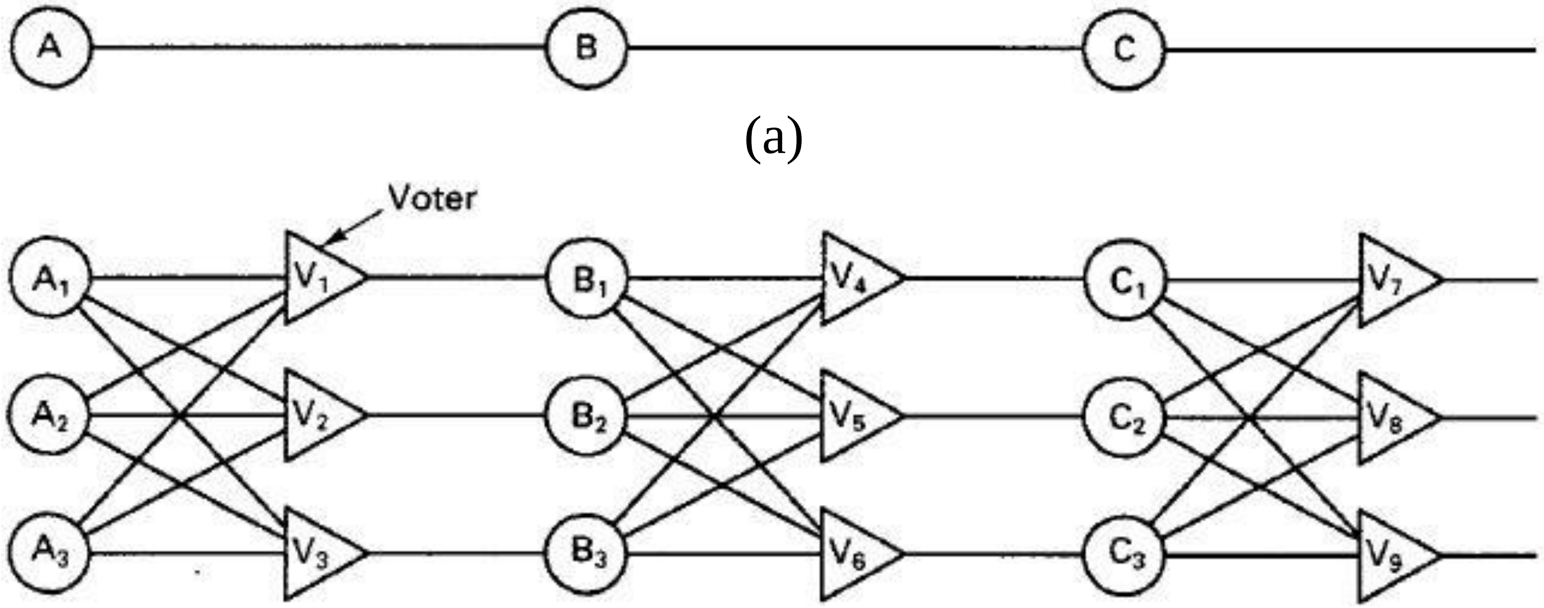
- A system is said to fail when it does not meet its specification
- Banking system
- Component faults
  - Failing due to fault in some component (processor, memory, cable)
  - Fault: Malfunction caused by design error, manufacturing error, programming error, physical damage, deterioration in the course of time, unexpected inputs,…..
  - Transient faults
    - Occur once and then disappear
    - Fault goes away if operation is repeated
  - Intermittent fault
    - Occurs, vanishes, reappears,…..
    - Loose contact on a connector
    - Difficult to diagnose
  - Permanent fault
    - Continues to exist until the faulty component is repaired
    - Burnt-out chips, software bugs
  - Fault-tolerant system: Ensuring that a system as a whole continues to function correctly, even in the presence of faults

Fault Tolerance

- System failures
  - Surviving component faults rather than making these unlikely
  - Fail-silent faults/Fail-stop faults
    - A faulty processor just stops and does not respond to subsequent input or produce further output
    - Easier to identify and resolve
  - Byzantine faults
    - A faulty processor continues to run issuing wrong answers to questions
    - Works together maliciously with other faulty processors to give the impression that they are all working correctly when they are not
    - Difficult to deal with
- Use of redundancy
  - General approach to fault tolerance
  - Information redundancy
    - Extra bits are added to allow recovery from garbled bits
    - Hamming code
  - Time redundancy
    - An action is performed and then if needed it is performed again
    - Redoing atomic transaction if it gets aborted
    - Transient, intermittent

Fault Tolerance

- Use of redundancy
  - Physical redundancy
    - Adding extra equipment to tolerate loss or malfunctioning
    - Adding extra processors
    - Active replication
      - TMR
      - Why 3?
      - A_1 fails
      - V_1 malfunctions



(a)

(b)

Figure 1: Triple modular redundancy

Fault Tolerance

- Use of redundancy
  - Physical redundancy
    - › Active replication
      - – K fault tolerant: Survive faults in k components and still meet its specifications
      - – Fail silent: K+1 for achieving K fault tolerance
      - – Byzantine failure: 2K+1 for achieving K fault tolerance
      - – Replicas can also fail
    - › Primary backup
      - – One server is primary
      - – If primary fails backup takes over
      - – Requires fewer machines
      - – Works poorly in presence of byzantine faults
      - – Primary crash
        1. After doing the work, before sending the update
           - → Backup takes over
           - → Request comes again
           - → Work will be done 2nd time
           - → Problematic in some cases

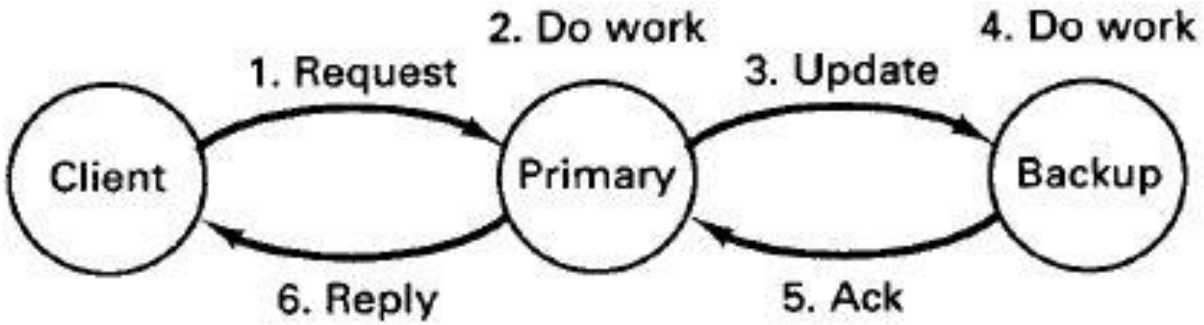


Figure 2: A simple primary backup protocol

Fault Tolerance

- Use of redundancy
  - Physical redundancy
    - Primary backup
      - Primary crash
        2. After doing the work in backup, before sending reply to client
           → Doing the work three times
           → By primary
           → In backup
           → When backup becomes primary
           → When to takeover the primary
           → Disk between primary and secondary

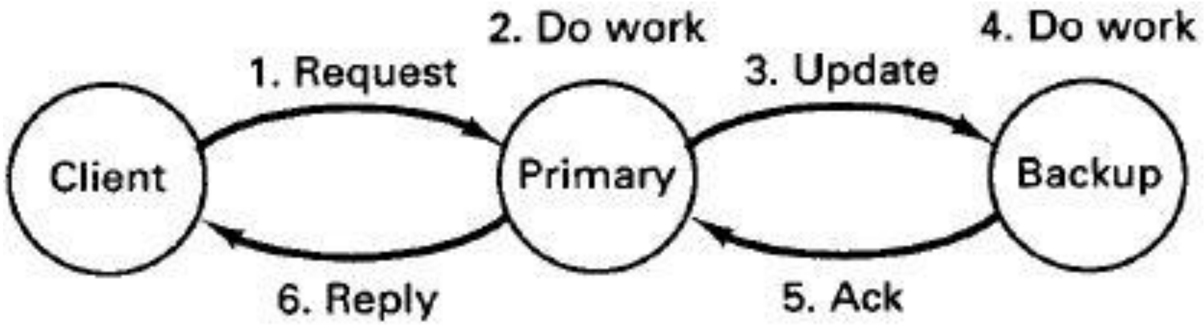


Figure 2: A simple primary backup protocol