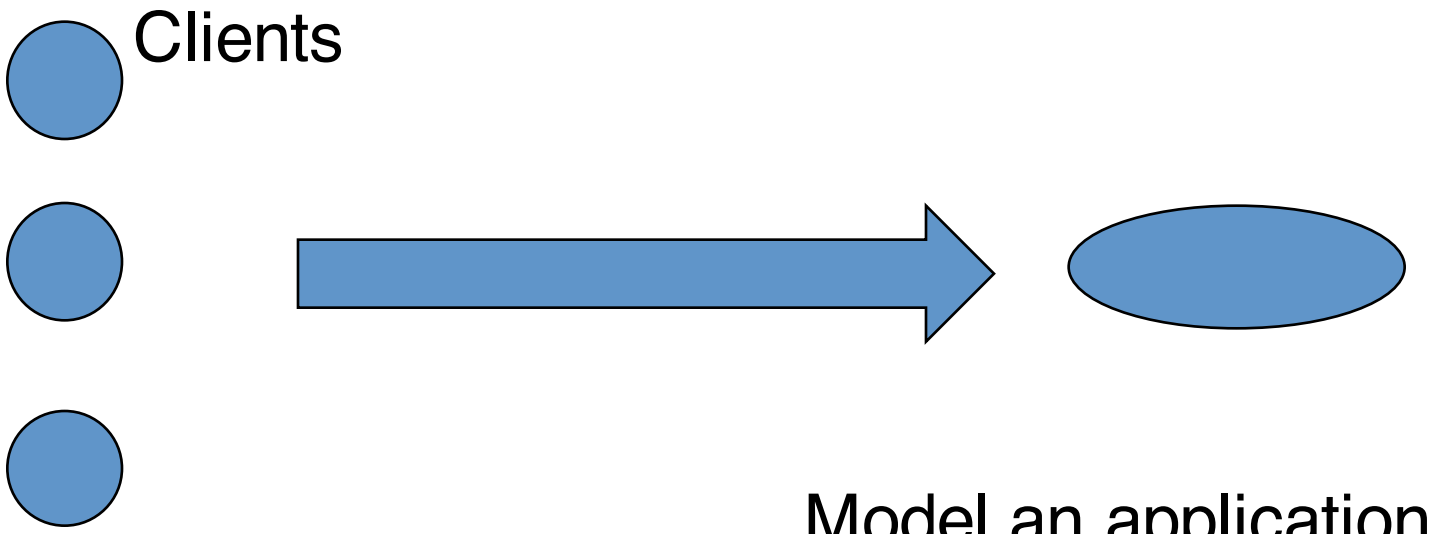
Consensus

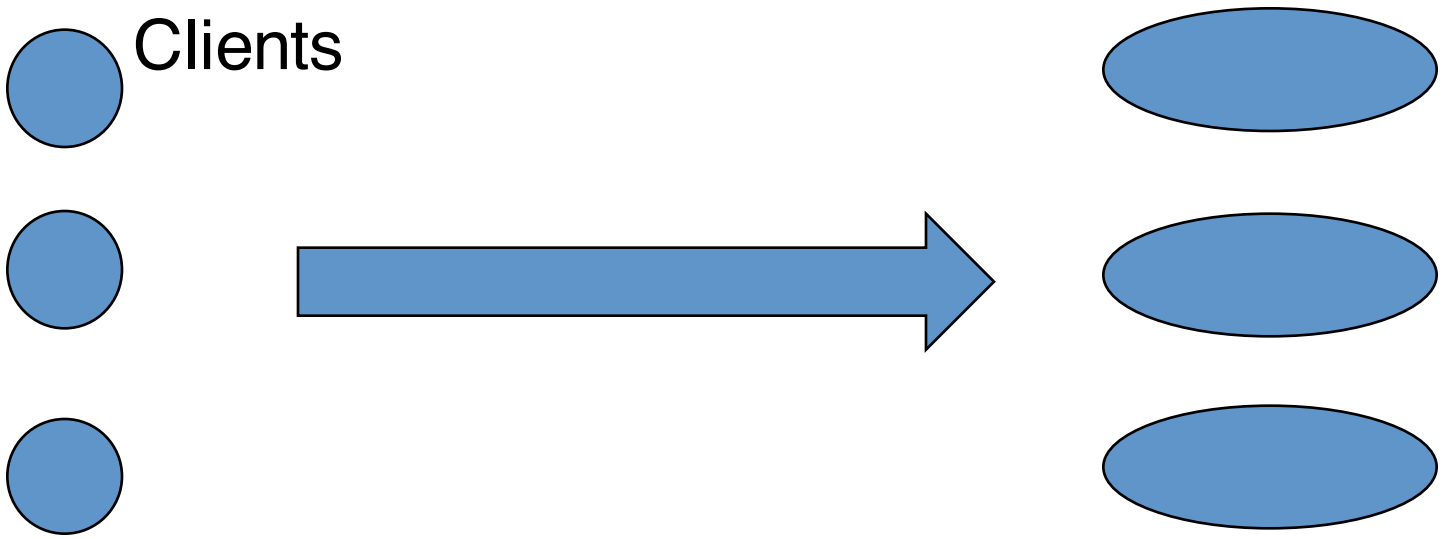
Yang Wang

Review – State machine replication



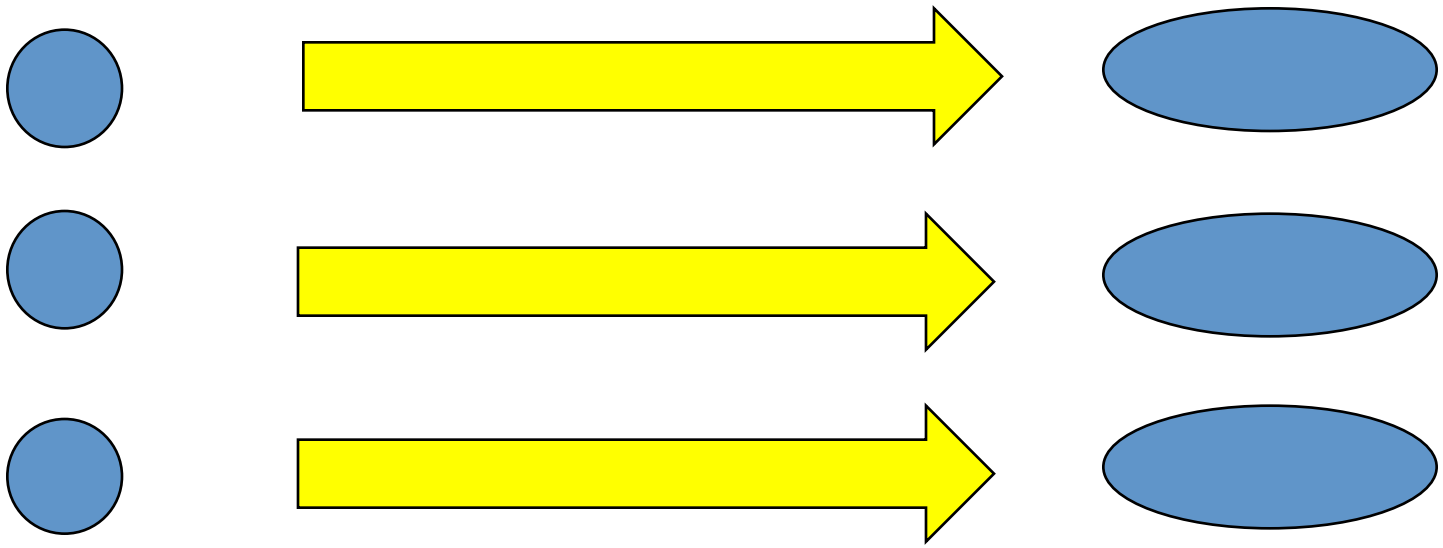
Model an application as a deterministic state machine

Review – State machine replication



For fault tolerance, put multiple copies of state machines on different physical machines

Review – State machine replication



Goal of replication: ensure all correct replicas are identical
In SMR, this is equal to “ensure inputs of all correct replicas are identical”

Review – State machine replication

- How to ensure state machines are deterministic?
  - Hypervisor-based Fault-Tolerance: we need to carefully handle interrupts and environmental instructions
  - There are still open problems: deterministic multithreading
- How to ensure all correct replicas get the same input?
  - Consensus or agreement protocol
  - Examples: Paxos, BFT, (primary backup can also be viewed as one solution)

Review – Primary backup

- Model: fail-stop; synchronous environment
- Basic protocol: clients send to primary; primary forwards to backup; when primary fails, backup becomes primary
  - Obviously when there are no failures, all replicas have the same input
  - When the primary fails, there is only one remaining replica.

- How about asynchronous environment and arbitrary errors?

Consensus

- **Termination**: Every correct process decides some value.
- **Validity**: If all processes propose the same value v, then all correct processes decide v.
- **Integrity**: Every correct process decides at most one value, and if it decides some value v, then v must have been proposed by some process.
- **Agreement**: Every correct process must agree on the same value.

Consensus and replication

- SMR protocols incorporate a consensus protocol to anwser “what is the next request”

Discussion

- We have discussed primary backup, which works in a fail-stop and synchronous model.
- Now let’s move to a fail-stop and asynchronous model.
- First, what can go wrong with primary backup?

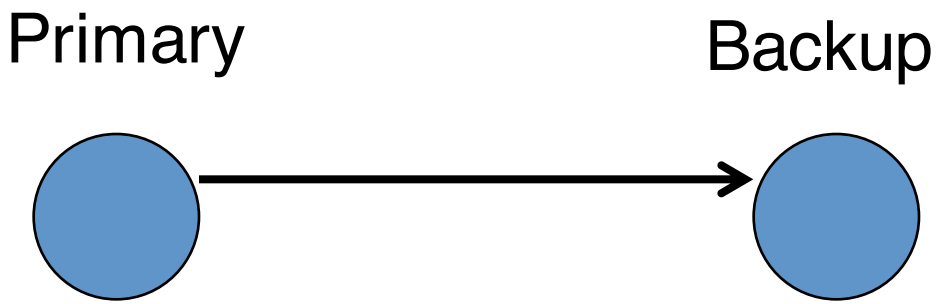
Discussion

- If primary fails, backup becomes primary
  - But how do we know primary fails

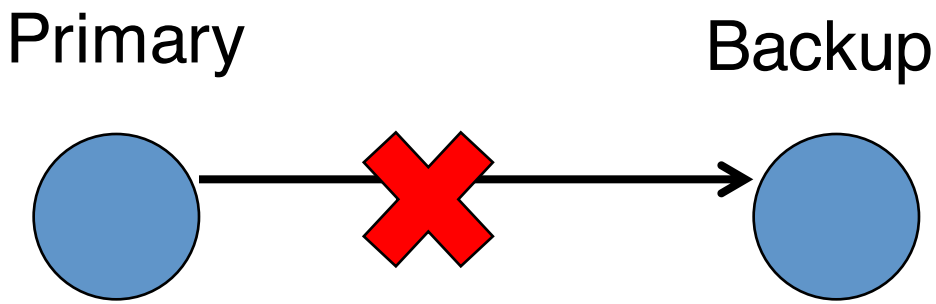
Discussion

- If primary fails, backup becomes primary
  - But how do we know primary fails
- Classic approach: heartbeat and timeout
  - In an asynchronous environment, can B conclude A fails if B does not receive the heartbeat in time?

Go back to primary backup

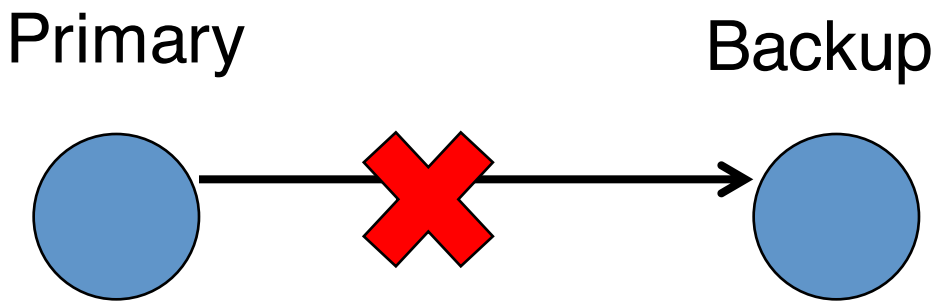


Go back to primary backup



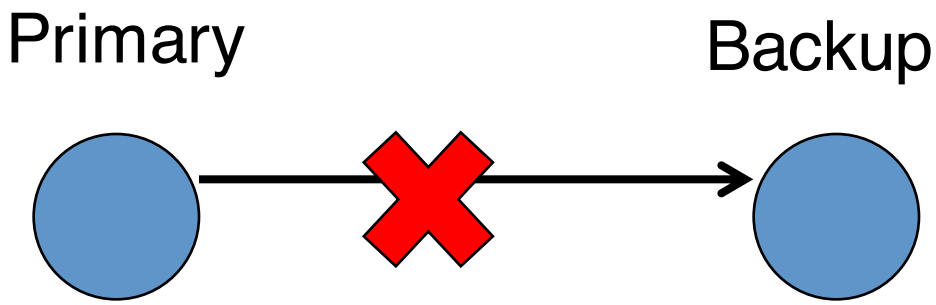
Can backup be promoted as the primary?

Go back to primary backup



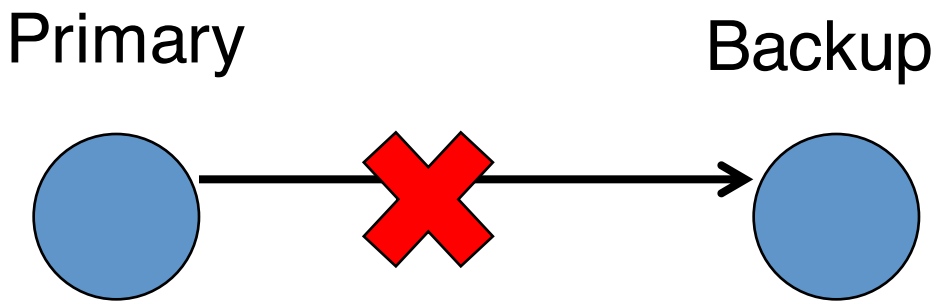
Can backup be promoted as the primary?
No. If primary has not failed, doing this may violate “agreement”.

Go back to primary backup



Can backup just wait?

Go back to primary backup



Can backup just wait?
No. If primary has actually failed, doing this may violate “termination”

Can you think about a solution?

Impossibility of consensus

- FLP theorem: it's impossible to solve consensus in an asynchronous environment
  - M.J. Fischer, N.A. Lynch, and M.S. Paterson. Impossibility of Distributed Consensus with One Faulty Process. E.W. Dijkstra Prize, 2001.
- Weakest failure detector to solve consensus
  - Chandra and Toueg. Unreliable failure detectors for reliable distributed systems.
- We don't have time to cover these topics.

It’s proved to be impossible.
What can we do?

Paxos' guarantees

- **Termination**: Every correct process decides some value when there are no asynchronous events.
- **Validity**: If all processes propose the same value v, then all correct processes decide v.
- **Integrity**: Every correct process decides at most one value, and if it decides some value v, then v must have been proposed by some process.
- **Agreement**: Every correct process must agree on the same value.
- Validity and integrity are trivial in fail-stop model. Let's focus on termination and agreement.