Distributed Software Development

Replication

Chris Brooks

Department of Computer Science
University of San Francisco

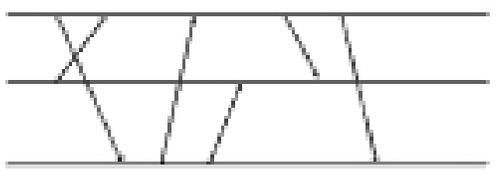
6-3: Previously on cs682: Causal delivery

- **causal delivery** says that if $send(m_1) \rightarrow send(m_2)$, then $deliver(m_1) \rightarrow (m_2)$ when different processes are sending m_1 and m_2.
- Logical clocks aren't enough to give us causal delivery.

6-3: Previously on cs682: Vector clocks

- Solution: keep a "logical clock" for each process.
- These are stored in a vector VC.
  - Assumes number of processes is known and fixed.
- Update rule:
  - $VC[p][q] = VC[q] + 1$ for send and internal.
  - $VC[p][q] = \max(VC, TS(m))$ for receive; then $VC[p][q] = VC[q] + 1$
- On receive, the vector clock takes the max on a component-by-component basis, then updates the local clock.

6-4: Previously on cs682: Vector Clock example



8-7: Why is replication useful?

- Increased performance.
  - By moving data closer to a client, latency is reduced.
  - Web caching, proxy servers are an example of this.
- Performance is improved most effectively with immutable data.
  - If the client is going to change the data and send it back, performance gains are reduced.

8-8: Why is replication useful?

- Increased availability.
- Many services need to be highly available
- Replication provides a way of overcoming server failures.
- If a server will fail with probability p then we can determine how many servers are needed to provide a given level of service.
  - $Availability = 1 - p^n$
- For example, if a server has a 5% chance of failure (i.i.d) over a given time period, and we want 99.9% availability, we need at least 4 replicas.

8-9: Client-side replication

- Note that replication is not limited to servers.
- Multiple clients may need to replicate data.
  - Shared code or documents being edited.
  - Meeting scheduling
  - Conferencing or whiteboard software.

8-10: Fault tolerance

- Highly available data may not be correct data.
  - For example, in the presence of network outages.
- Fault tolerance guarantees correct behavior in the presence of a given number of faults.
- Similar to availability, but a coordination element is also required.
- We may also want to ensure against corruption of data.

8-11: Outline

- Passive Replication
  - What problems must be solved for this?
- Active Replication
  - What problems must be solved for this?
- Lazy Replication
  - What problems must be solved for this?

8-12: Client Requirements

- Single logical copy.
- Multiple physical copies.
- Consistency
  - The details of this will depend on the application.

6-13: System Model