Lecture 09 · SENG 41283 · Distributed and Cloud Computing

Replication

Motivations · System Model · Five Phases · Ordering · Passive Replication · Active Replication

1. Motivations for Replication

Performance Enhancement

Caching avoids the latency of fetching data from the originating server. DNS round-robin returns different server addresses for the same name, distributing workload. Replication of immutable data is trivial; replication of changing data incurs overhead from consistency protocols.

Increased Availability

High availability requires the service to be accessible close to 100 % of the time. If each of n servers has an independent probability p of failure, availability = $1 – p^n$. Example: p = 0.05, n = 2 → availability = $1 – 0.05^2$ = 99.75 %.

> *★ Caches do not necessarily hold complete collections; caching does not necessarily enhance application-level availability.*

Fault Tolerance

A fault-tolerant service always guarantees strictly correct behaviour despite a certain number and type of faults. Correctness is concerned with freshness of data, effects of client operations, and timeliness of responses.

2. System Model

We assume an asynchronous system; processes may fail only by crashing; no network partitions (default). Operations are applied **recoverably**: a crash mid-operation does not leave inconsistent results.

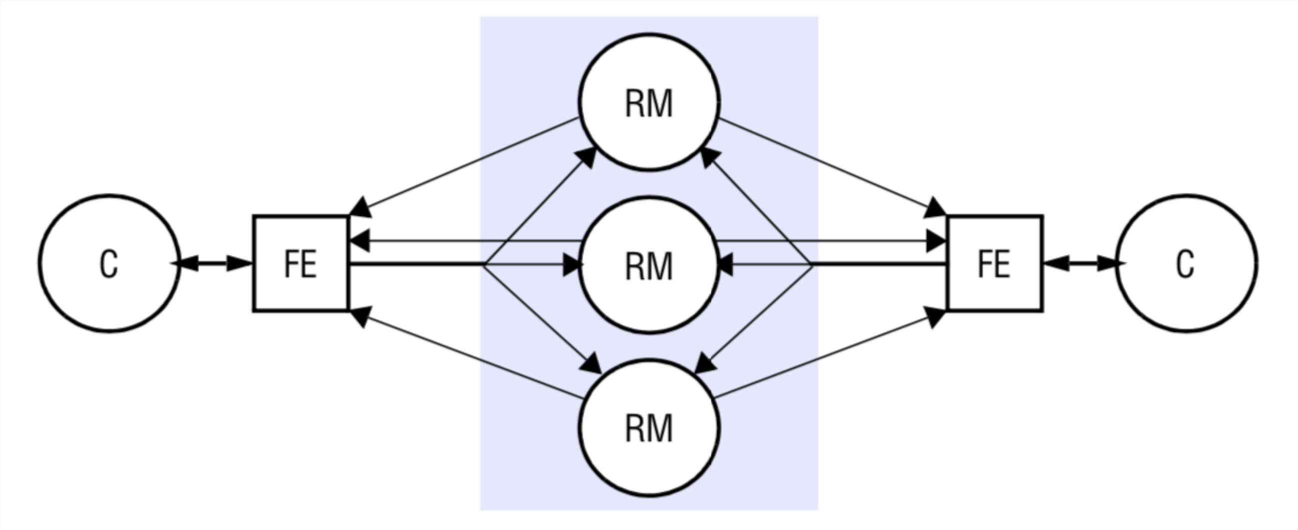

Figure 2.1 Replication system model: front ends, replica managers, and the five-phase request cycle.

The Five Phases of a Single Request

Phase	Description
1. Request	Front end issues the request to one or more replica managers (directly or via multicast).
2. Coordination	Managers coordinate: agree on whether to apply the request and on its ordering relative to others.
3. Execution	Replica managers execute the request.
4. Agreement	Managers reach consensus on the effect (e.g. abort or commit in a transactional system).
5. Response	One or more managers respond to the front end, which synthesises a single response for the client.

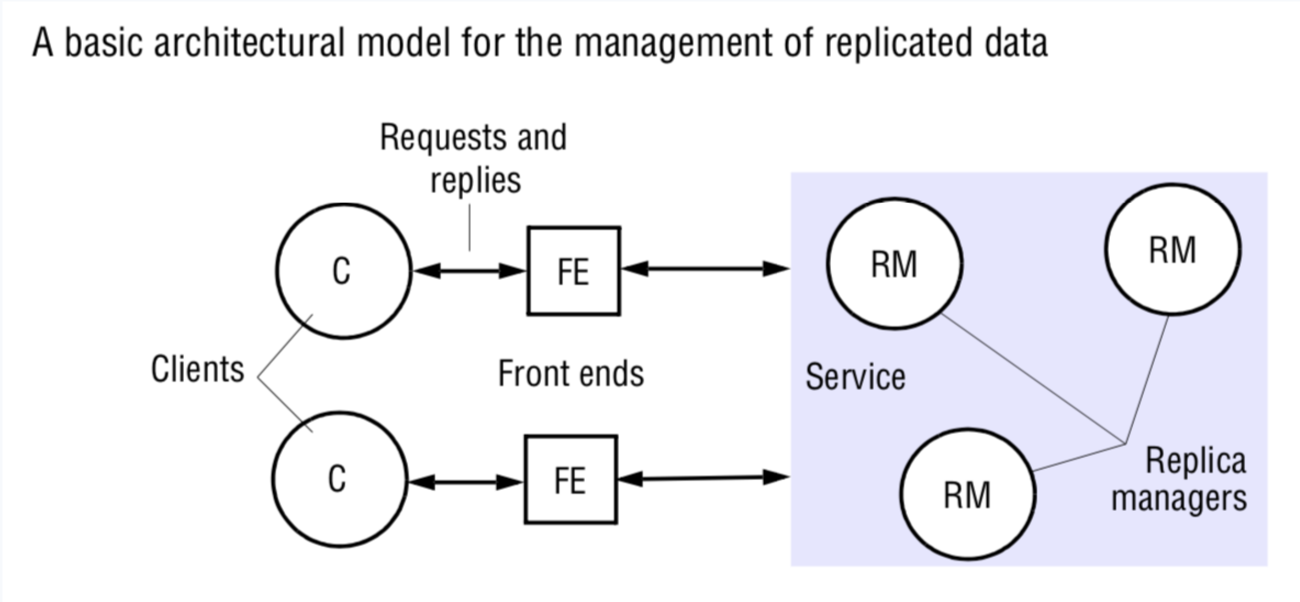

Figure 2.2 Five phases of a single request: request, coordination, execution, agreement, response.

Ordering Guarantees

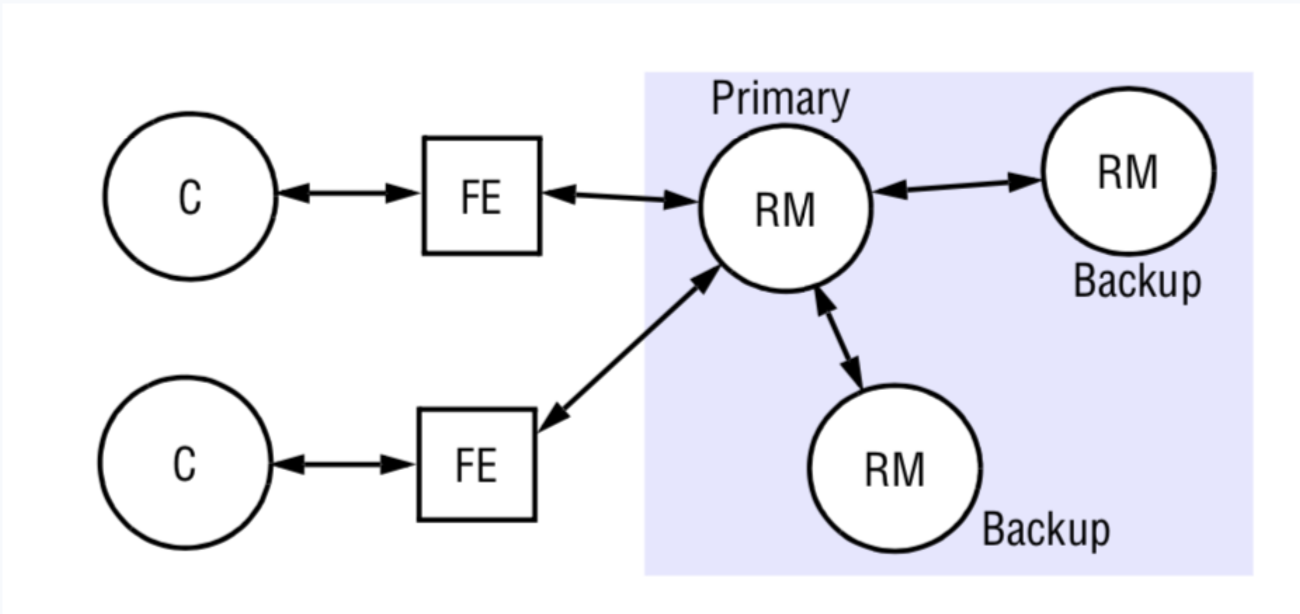

Figure 2.3 FIFO, causal, and total ordering of messages across replica managers.

Type	Description
FIFO	Messages from process P_i received at all other processes in the order they were sent.
Causal	If r happened-before r’, any manager handling r’ handles r first. Captures cross-sender causality.
Total	If any correct manager handles r before r’, every correct manager does likewise.

Central lock server: one coordinator C; processes send lock → grant if free, else queued; unlock releases the next in the queue.

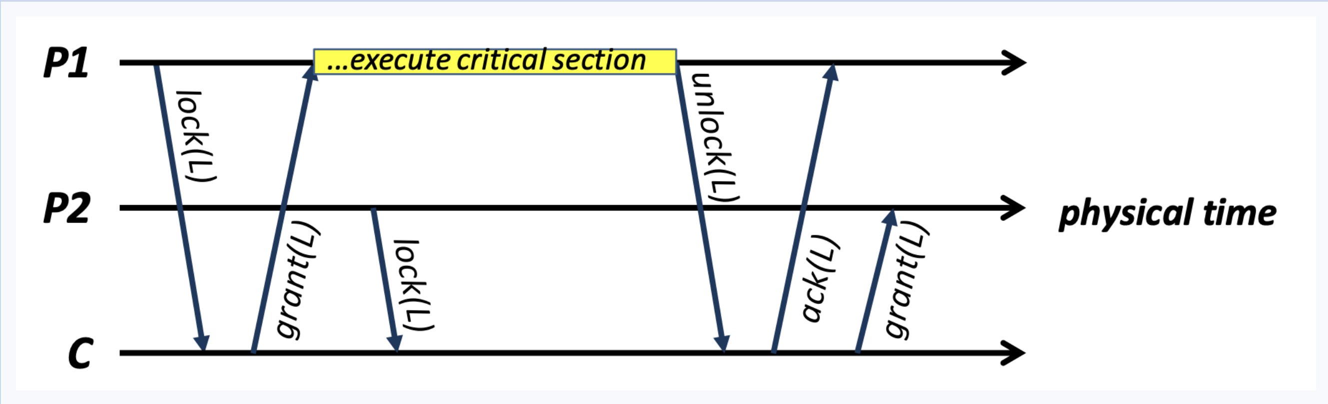

Figure 2.4 Central lock server solution for mutual exclusion and total ordering.

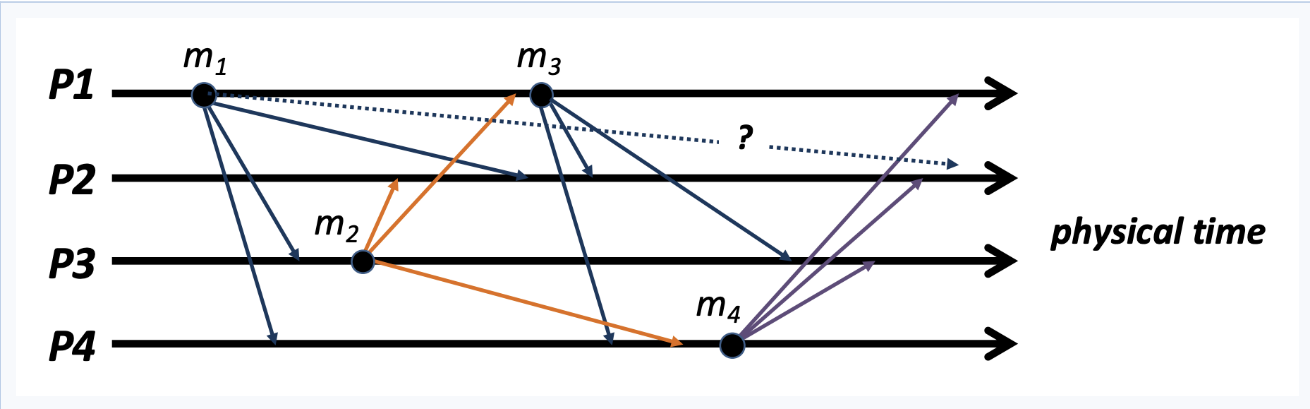

Figure 2.5 Token passing ring: token circulates; process may only access the resource when holding the token.

3. Passive (Primary–Backup) Replication

Single primary + one or more backup managers. Front ends communicate only with the primary. The primary executes operations and sends updated state to backups. If the primary fails, one backup is promoted.

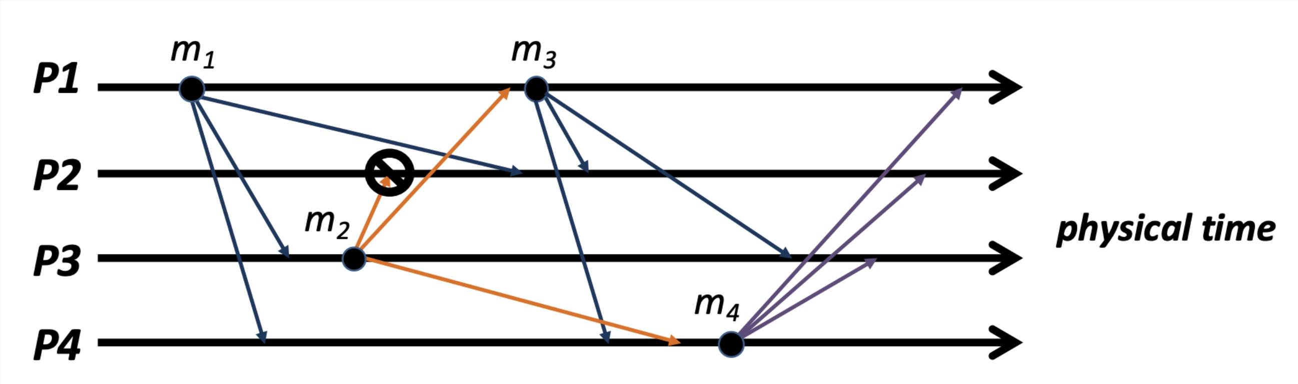

Figure 3.1 Passive replication: primary processes requests and propagates updates to backups.

Request Sequence

- 1. **Request:** front end sends request with unique identifier to primary.
- 2. **Coordination:** primary processes requests atomically in receipt order; checks for duplicates.
- 3. **Execution:** primary executes the request and stores the response.
- 4. **Agreement (if update):** primary sends updated state + response + ID to all backups; backups ACK.
- 5. **Response:** primary responds to front end.

Implements **linearisability** if the primary is correct. View-synchronous semantics ensure either all backups or none deliver a given update before delivering the new membership view.

4. Active Replication

All replica managers are state machines playing equivalent roles in a group. Front ends multicast requests to the group; all managers process independently but identically. A crashed manager has no impact.

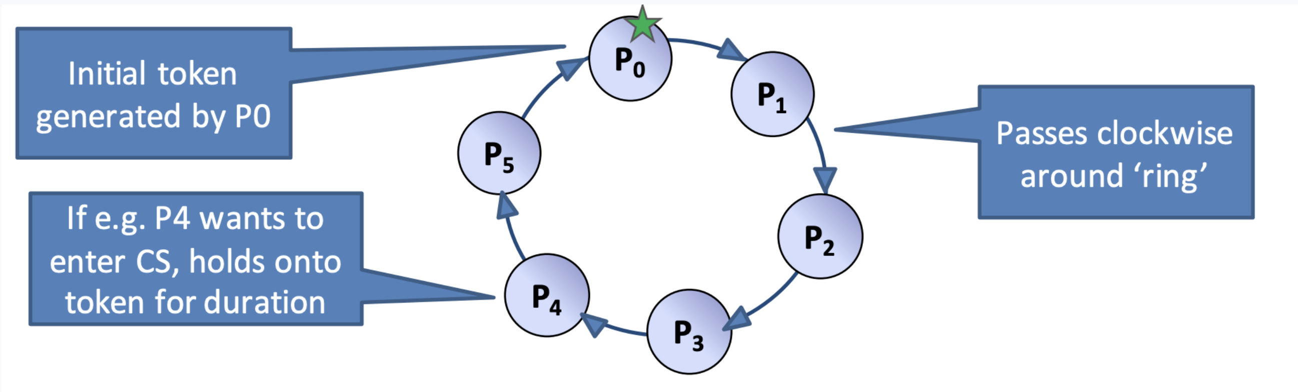

Figure 4.1 Active replication: front end multicasts to all managers; each executes identically.

Request Sequence

- 1. **Request:** front end attaches unique ID and multicasts using totally ordered reliable multicast.
- 2. **Coordination:** group communication delivers request to every correct manager in the same total order.
- 3. **Execution:** every manager executes identically (state machines + same total order).
- 4. **Agreement:** no agreement phase needed — multicast delivery semantics guarantee this.
- 5. **Response:** front end passes first response to client; discards the rest.

Guarantees **sequential consistency**: all correct managers process the same sequence of requests and end with the same state.

Transactions with Replicated Data

Two-phase commit becomes a two-level nested 2PC when replica managers are involved. Read-one/write-all: every write at all managers (write lock each); each read at a single manager (read lock). This ensures one-copy serializability.