

Replication in distributed systems

Distributed Software
Systems

Replication

■ Motivation

- Performance Enhancement
- Enhanced availability
- Fault tolerance

■ Requirements

- Replication transparency
- Consistency
 - | Depends upon application
 - | Different clients making (read/write) requests to different replicas of the same logical data item should not obtain different results

Request ordering

- Replica consistency depends upon request ordering
 - Asynchronous model (e.g. USENET news)
 - Totally synchronous model
 - All requests are totally ordered
 - Intermediate orderings, e.g. causal ordering
- Key approaches for achieving satisfactory tradeoffs between consistency, availability, and performance
 - Quorum-based schemes, e.g. majority voting
 - Causality

Figure 11.1 Smith's display from bulletin board reader.

Bulletin board: <i>os.interesting</i>		
Item	From	Subject
23	A.Hanlon	Mach
24	G.Joseph	Microkernels
25	A.Hanlon	Re: Microkernels
26	T.L'Heureux	RPC performance
27	M.Walker	Re: Mach
end		

Figure 11.2 Jones' display from bulletin board reader.

Bulletin board: <i>os.interesting</i>		
Item	From	Subject
20	G.Joseph	Microkernels
21	A.Hanlon	Mach
22	A.Sahiner	Re: RPC performance
23	M.Walker	Re: Mach
24	T.L'Heureux	RPC performance
25	A.Hanlon	Re: Microkernels
end		

Architectural models

- Architectural components
 - Client, Front end, replica manager
- Architectures
 - Gossip architecture
 - Primary copy model
 - Groupware model
 - Client and replica manager in same process

Figure 11.3 A basic architectural model for the management of replicated data.

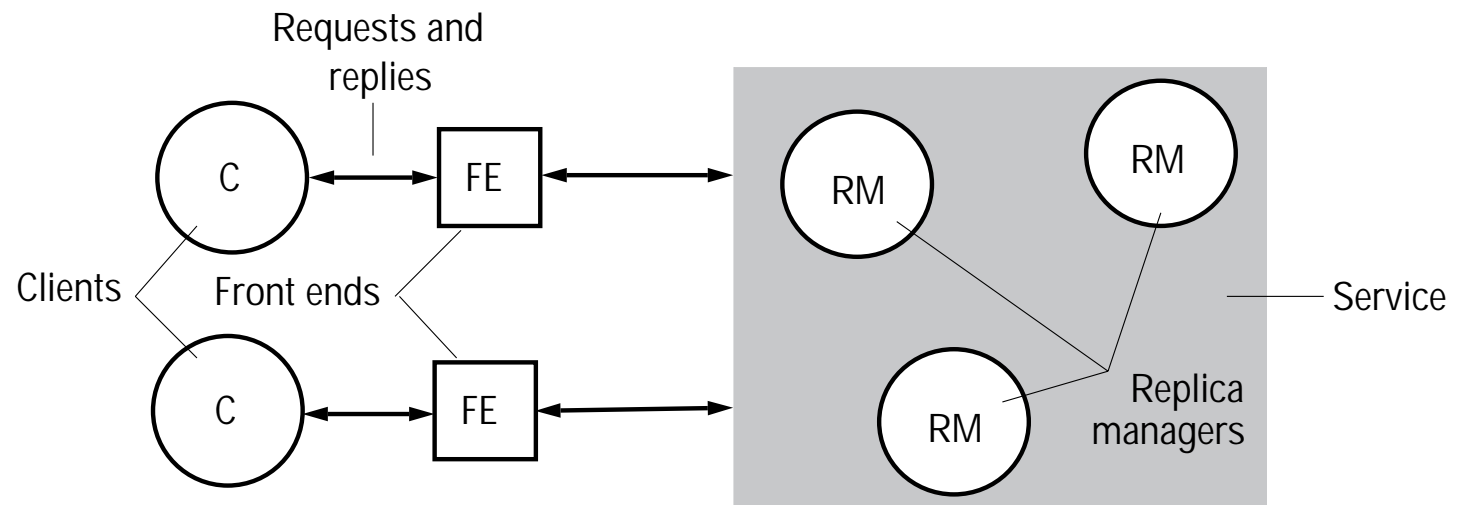
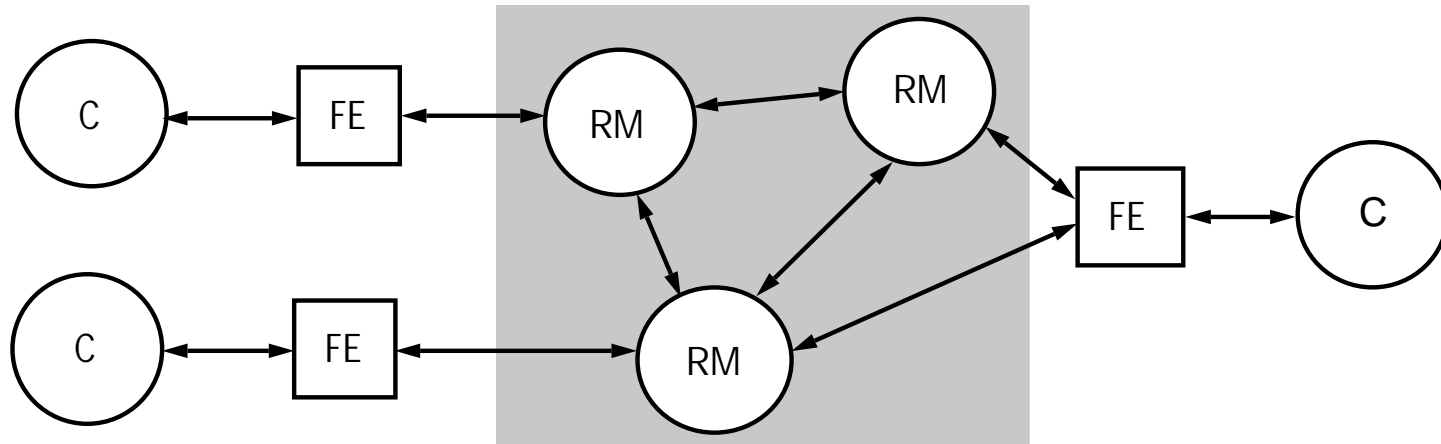
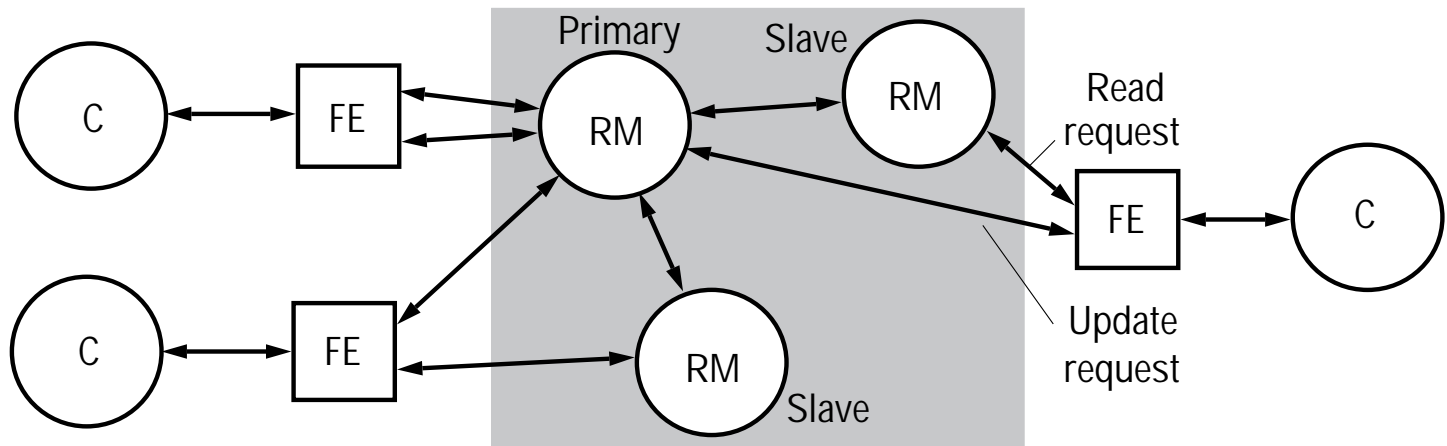


Figure 11.4 The gossip architecture.



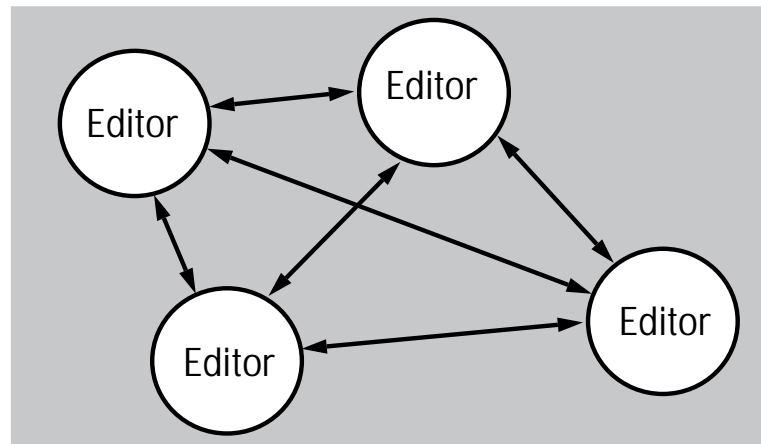
Front ends may communicate with different replica managers.

Figure 11.5 The primary copy model for replicated data.



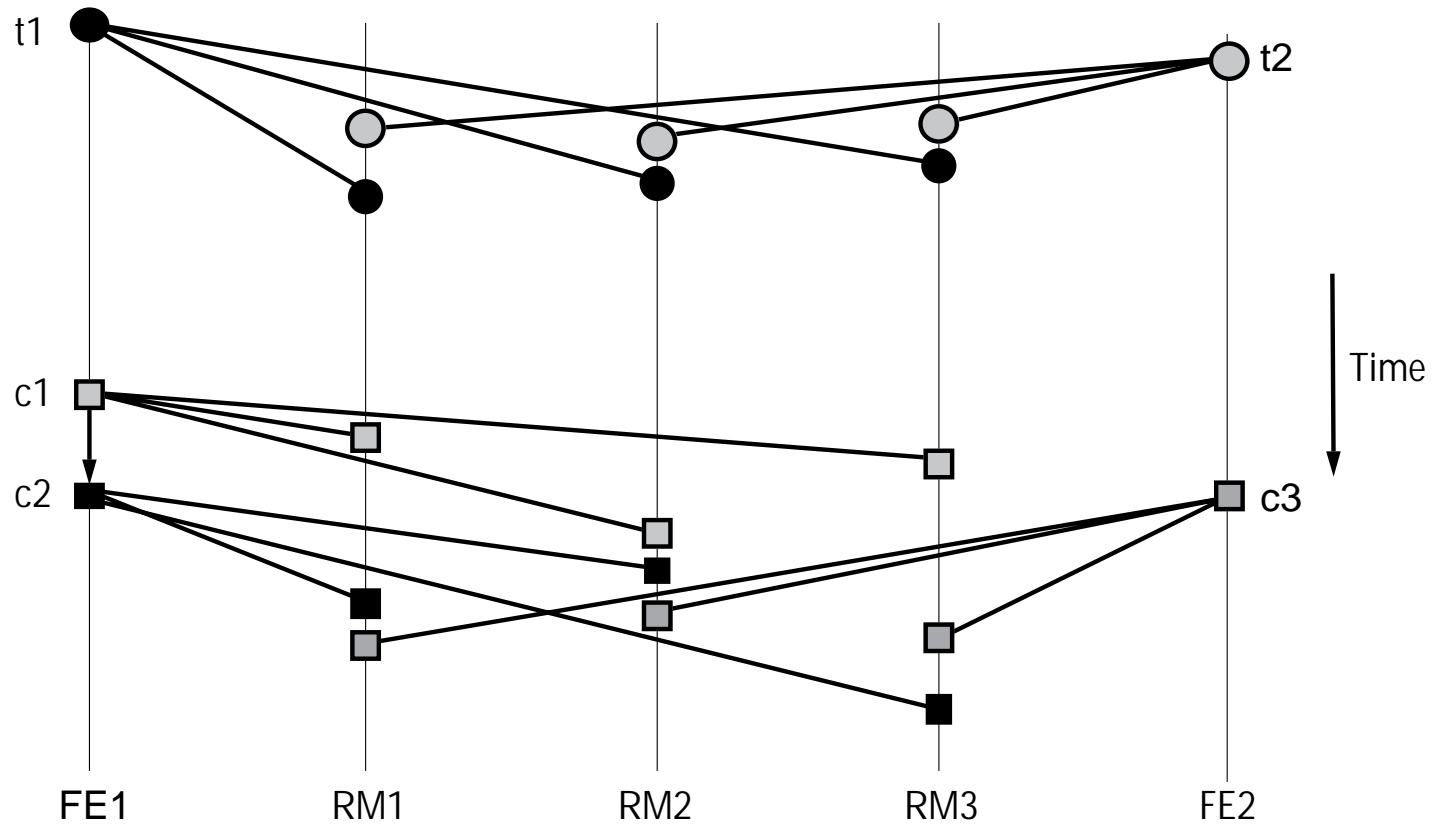
The primary server is accessed for all update requests. It propagates updates to its slaves.

Figure 11.6 The architecture of a multi-user editor.



Each circle is a combined client, front end and replica manager (one per user).

Figure 11.7 Totally and causally ordered request processing.



Notice the consistent ordering of $t1$ and $t2$, the consistent ordering of the causally related operations $c1$ and $c2$, and the arbitrary relative ordering of $c2$ and $c3$, which are unrelated.