Slide 0

Window-Consistent Replication for Real-Time Applications

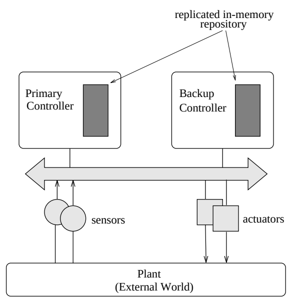
Jennifer Rexford, Ashish Mehra, James Dolter and Farnam Jahanian

May 19, 1994

Real-Time Computing Laboratory
The University of Michigan
Ann Arbor, Michigan

Slide 1

Application Data Requirements



- High rate of data access (query/modify)
- Bounded time-inconsistency of data
- Timely availability

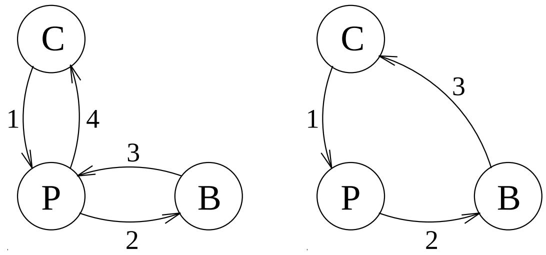
Slide 2

Related Work

- Replication schemes
  - Primary-backup (passive) [Alsberg76, Bartlett81, Budhiraja92]
  - State machine (active) [Schneider90, Birman91]
- Weak consistency models
  - Real-time transactions [Davidson88, Pu91, Lin92, Kuo93]
  - Imprecise computations [Liu91, Liu94]
- Real-time scheduling [Liu73, Lehoczky89, Han92]

Slide 3

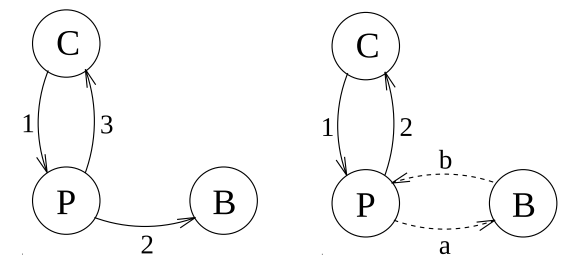
Primary-Backup Models: Blocking



Real-Time Computing Laboratory
The University of Michigan

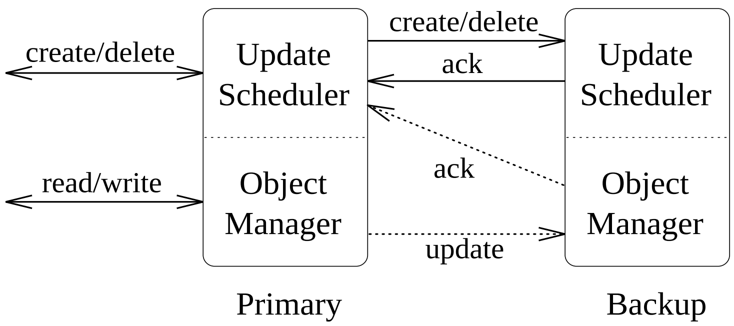
Slide 4

Primary-Backup Models: Non-blocking



Slide 5

Window-Consistent Primary-Backup Model



Slide 6

Window Consistency for an Object

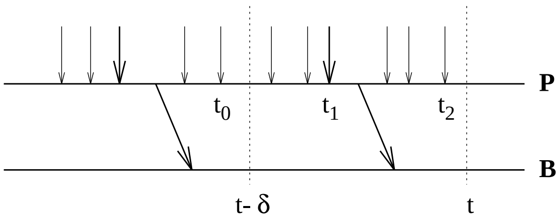
B must have a copy that P believed no more than δ time units ago.

- **Relaxing data consistency**: Don't send every update to the backups $\rightarrow$ can handle higher rate of client operations
- **Bounding time consistency**: Must schedule updates to the backups $\rightarrow$ maintain window consistency

Real-Time Computing Laboratory
The University of Michigan

Slide 7

Window-Consistency Semantics



Real-Time Computing Laboratory
The University of Michigan

Slide 8

Modes of Operation

Primary	Backup	Operating Mode
Valid	Valid	P schedules updates to maintain valid B
Valid	Restricted	P schedules updates to attain valid B
Valid	Invalid	P schedules updates to integrate new B
Invalid	Valid	B detects failure and performs failover
Invalid	Not valid	B does not perform failover

Slide 9

Real-Time Scheduling Framework

Schedule update transmissions from P to B for each object O_i to maintain window consistency.

- Finite processing and network resources
- In absence of performance failure, communication latency $< d$
- Object O_i with window δ_i and service time e_i

$\rightarrow$ object priorities

Slide 10

Applying Rate-Monotonic Scheduling

- Consecutive updates for O_i separated by at most δ_i
- Period $p_i = (\delta_i - d)/2$ for O_i
- Object admission from schedulability criterion
- Compressed schedule to improve utilization and object consistency
- Distance-constrained scheduling for improved schedulability

Slide 11

System Model and Failure Assumptions

Failure assumptions

- Processor crash (P or B crash)
- Link performance failures (communication delay $> d$)
- Link omission failure (lost update or ack message)

The update and ack messages between the servers provide:

- Synchronized clocks $\rightarrow$ common time base
- Heartbeats $\rightarrow$ failure detection
- Knowledge of window consistency $\rightarrow$ determine operating mode of B

Slide 12

Failure Detection and Failover

Backup must balance the likelihood of false failure detection with the possibility of becoming window-inconsistent before failover.

- B has not heard from P in the last β time units
- B may become window inconsistent in γ time units

Adaptive timeout: trigger failover as β grows and $\gamma \to 0$

Slide 13

Current Status and Future Work

- Prototype on Sun workstations running Solaris 1.1
- Evaluation on a real-time platform (pSOS+, 68040 boards, VME bus)
- Handling temporal consistency between related objects
- Incorporating a variety of scheduling algorithms
- Developing protocols for the case of multiple backup sites
- Allowing dynamic creation/deletion of objects

Slide 14

Conclusions

- Decouple client requests from communication within the service
- Schedule selective transmissions within the replication service
- Bounded temporal inconsistency between the servers

$\rightarrow$ window consistency offers a framework for designing replication protocols with predictable timing behavior

Real-Time Computing Laboratory
The University of Michigan