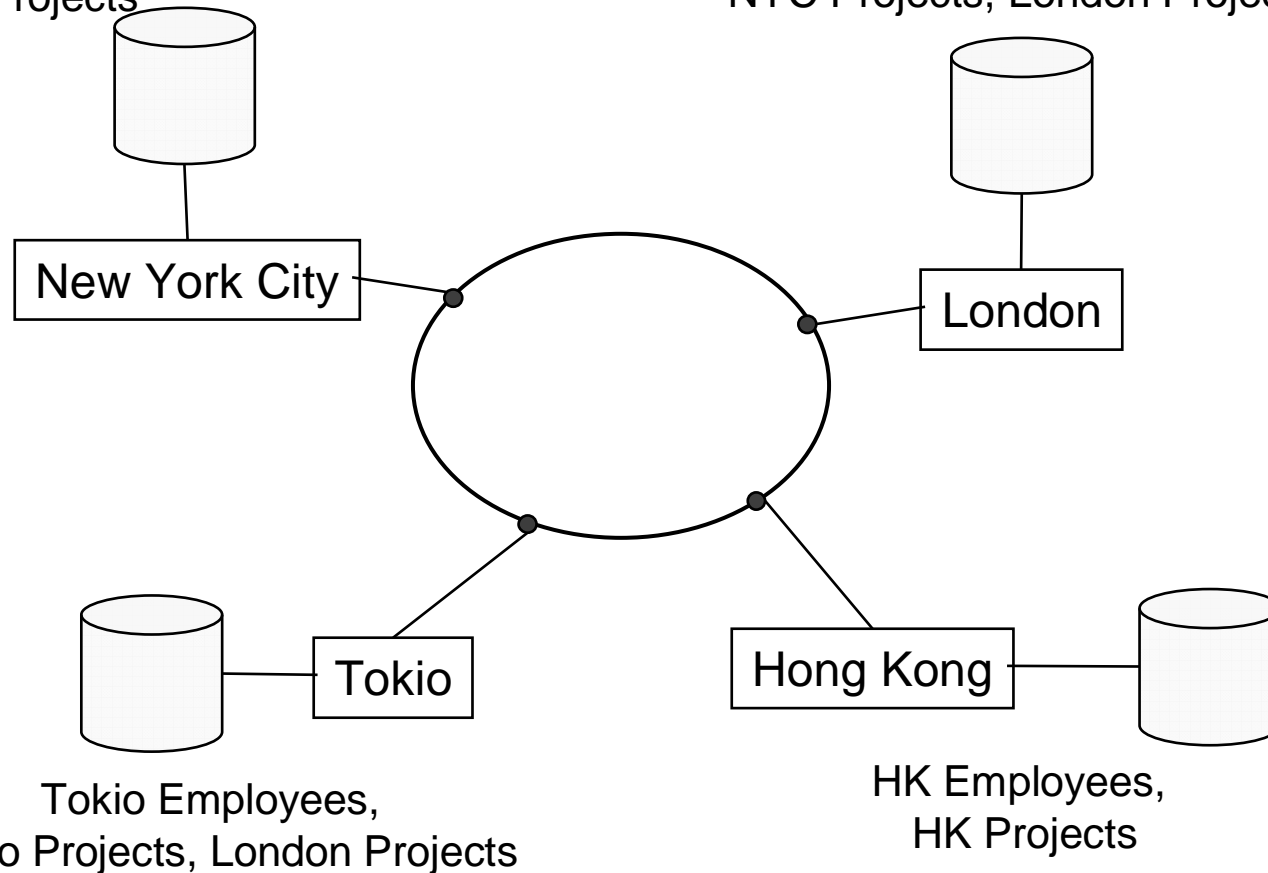


Chapter 11: Replication

Fragmentation vs. Replication – Example

NYC Employees, London Employees,
NYC Projects

NYC Employees, London Employees,
NYC Projects, London Projects



Example of distributed application.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Introduction

- Main objective: increase **availability** – replicated data are available after failure of a node as well.
- Furthermore: **better performance** wrt reads – communication is cheaper (locality), better load balancing.
(Choice among several copies.)
- In this context, full redundancy is optimal: all operations are local.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

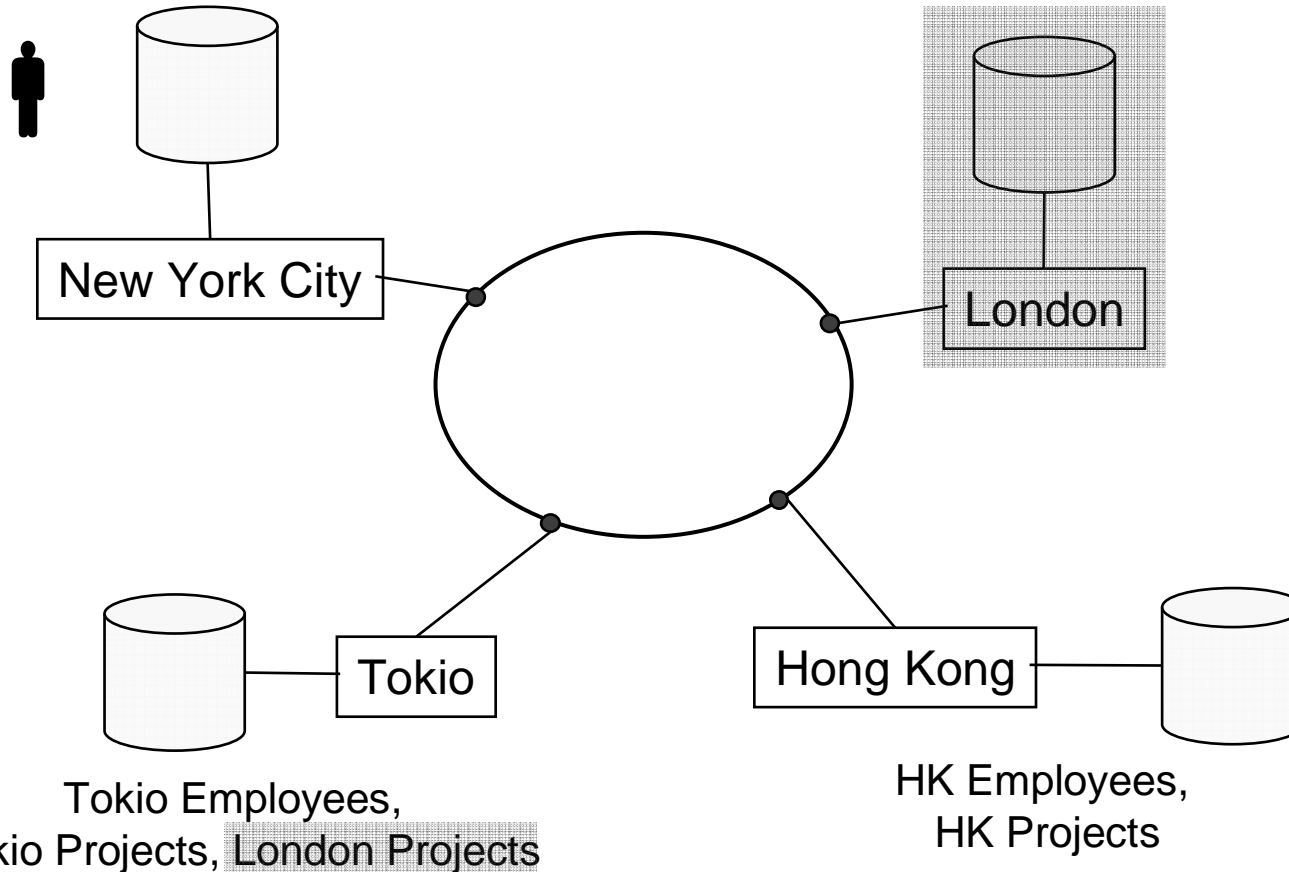
Virtual Partition

Summary

Replication – Advantages

NYC Employees, London Employees,
NYC Projects

NYC Employees, London Employees,
NYC Projects, London Projects



Example of distributed application.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Replication – Opportunities and Costs (1)

- **Disadvantages (costs):**
 - ◆ More storage space needed.
 - ◆ Modifications are more expensive:
all copies need to be updated,
expensive if distances between nodes are high.
⇒ Typically, only partial replication is feasible.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Replication – Opportunities and Costs (2)

- Significantly higher **implementation complexity**:
 - ◆ Hide existence of replicas from the user – *replication transparency*: updates go to all copies automatically.
 - ◆ Transactions shall see consistent data:
 - users shall not see copies in different states,
 - different copies shall not be modified by different transactions at the same time.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Replication – Challenges

NYC Projects

NYC Employees, London Employees,
NYC Projects

NYC Employees, London Employees,
NYC Projects, London Projects

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

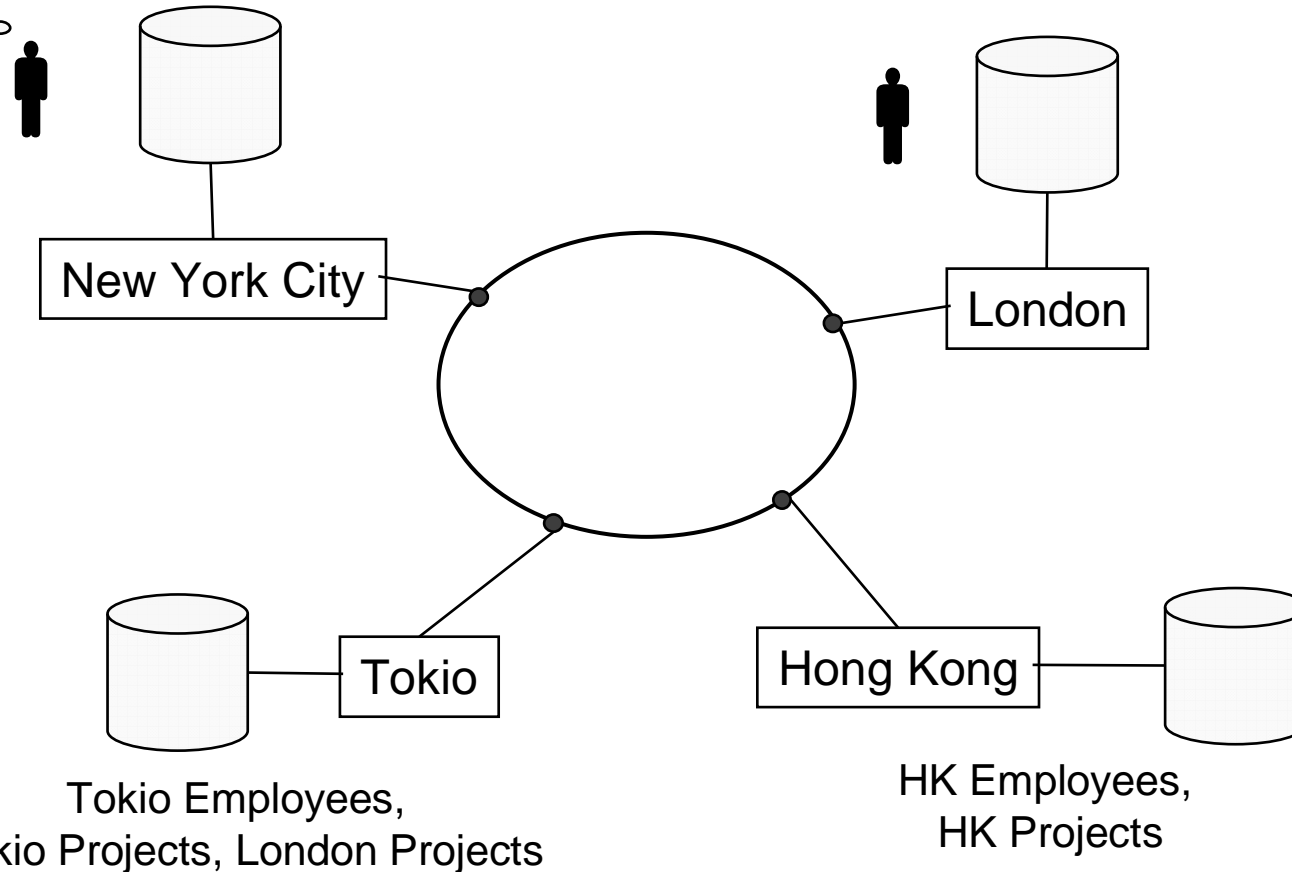
Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary



One-copy serializability

Replication – Opportunities and Costs (3)

- **Implementation complexity (cont.):**
 - ◆ ensure consistency of the database in the presence of failures:
notably network partitions –
replicas in separate partitions
may not be modified differently.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Replication – Opportunities and Costs (4)

- **Implementation complexity (cont.):**
 - ◆ Extended synchronization schemes:
correctness criterion – **one-copy-serializability**:
only allow for schedules that are equivalent
to serializable schedules
in non-redundant database.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

One-Copy-Serializability – Illustration

- Data objects a, b; copies on nodes X, Y.
- T1: r(a) w(b); T2: r(b) w(a)
- Not 1SR:
 1. T1 reads copy of a on X
 2. T2 reads copy of b on Y
 3. T1 writes both copies of b.
 4. T2 writes both copies of a.
- Local serializability.
- Equivalent execution on one node not serializable.
- 1SR in turn: Swap 2. and 3.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

One-Copy Serializability

- Atomic Commitment Protocols do not help:
 - ◆ only failures during commit processing,
 - ◆ typically no progress when component fails.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Distributed Writes (1)

- All algorithms taking replication into account must update set of copies.
- Alternatives: synchronous vs. asynchronous.
 - ◆ synchronous – within same transaction (*eager replication*),
 - ◆ asynchronous – separate transaction to update replicas (*lazy replication*); only briefly in this chapter.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Distributed Writes (2)

- Alternatives ,within‘ synchronous:
immediate vs. deferred.
- Immediate: writes are propagated right away.
- Deferred: only when transaction ends,
but before commit.
(E.g., piggybacking of write message
to other copies with VOTE-REQ Message
in Phase 1 of Atomic Commitment Protocol.)

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

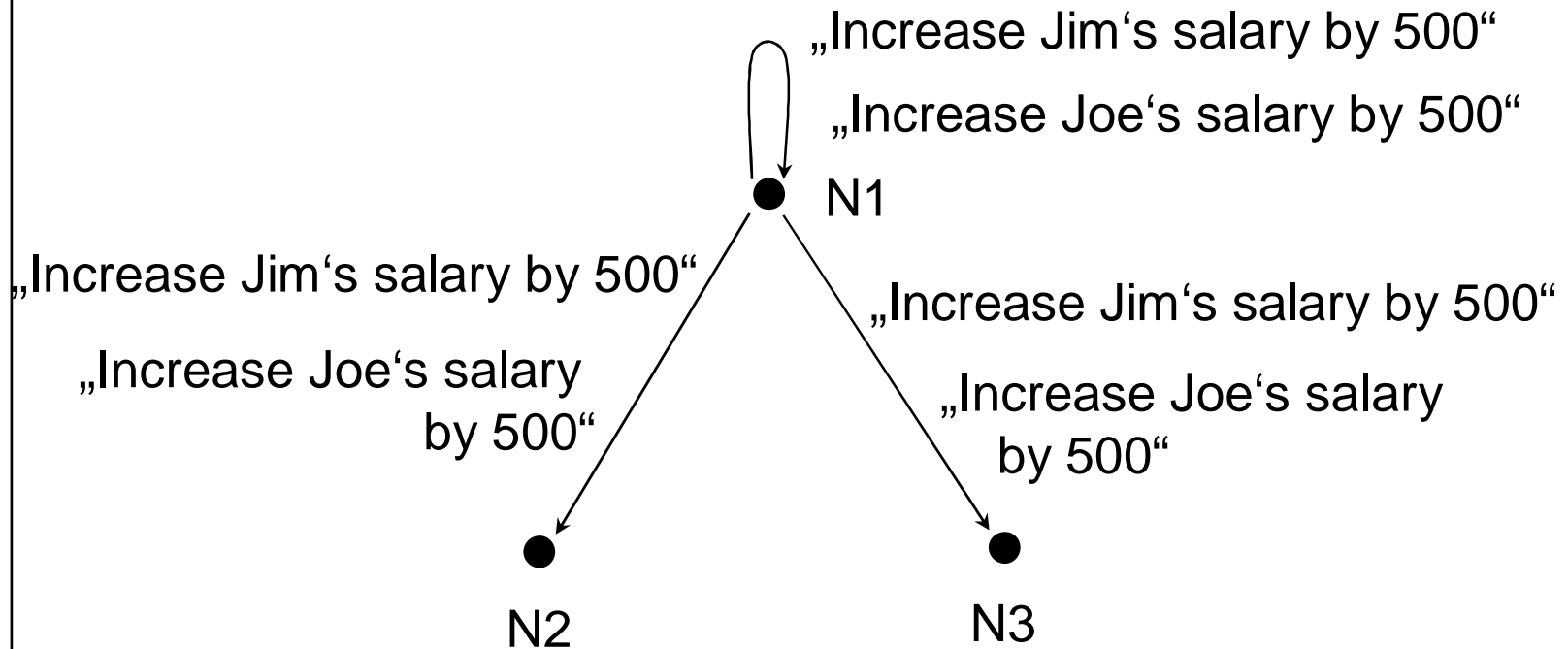
QCA

Missing
Writes

Virtual
Partition

Summary

Eager Replication, Immediate (1)



In contrast to previous chapter,
all nodes now contain the same data.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Eager Replication, Immediate (2)

Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

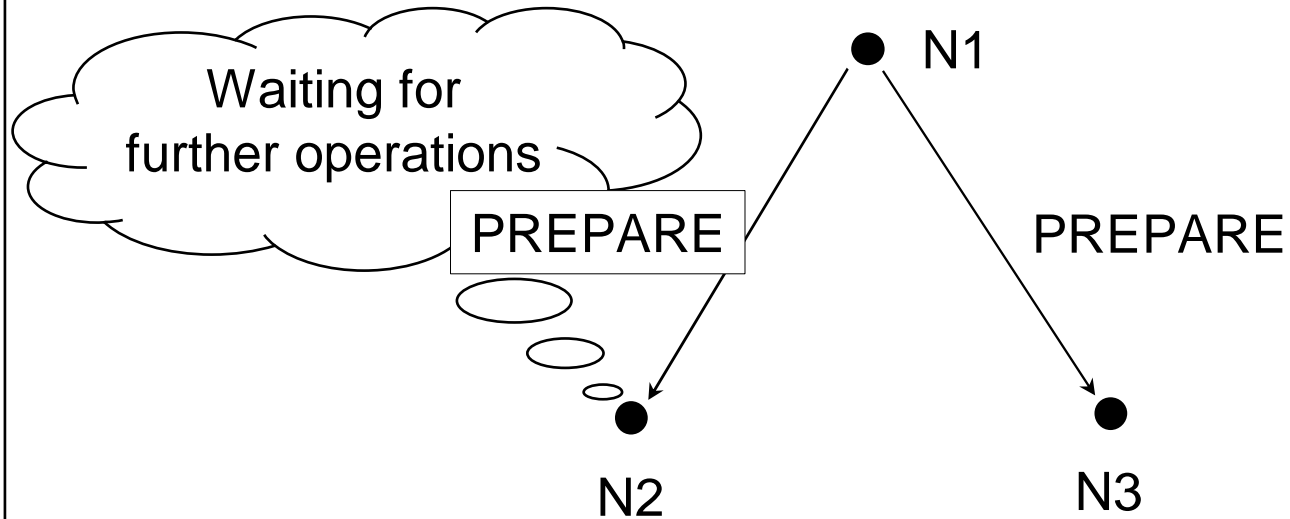
Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary



<i>Obj</i>	<i>Val</i>	
Jim	2000	L
Joe	2500	L
...	...	

Database

<i>Obj</i>	<i>Redo</i>
Jim	2500
Joe	3000
...	

Log

Eager Replication, Deferred (1)

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

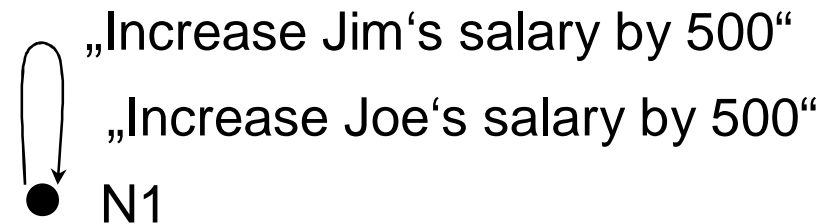
Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary



●
N2

●
N3

Eager Replication, Deferred (2)

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

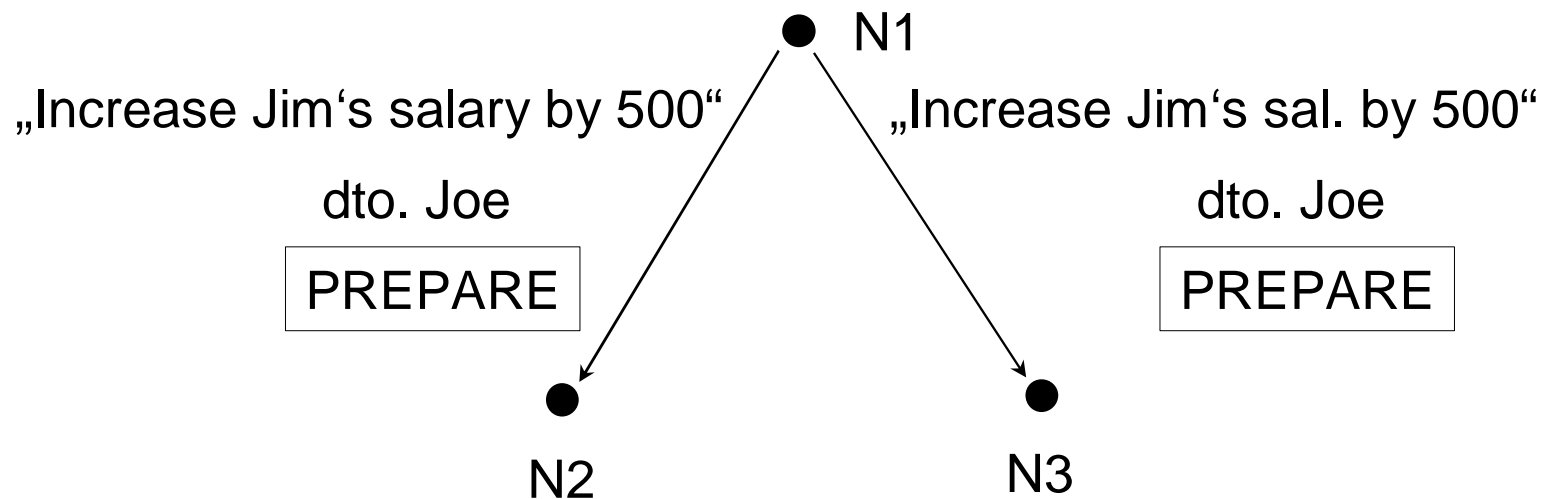
Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary



<i>Obj</i>	<i>Val</i>
Jim	2000
Joe	2500
...	...

Database

<i>Obj</i>	<i>Redo</i>

Log

Advantages/Disadvantages Immediate vs. Deferred

- Typically less messages with deferred, all deferred writes of a transaction in one message.
- Deferred tends to delay detection of conflicts – example:
 - ◆ T_1 writes x_A , T_2 writes x_B .
 - ◆ Abort only at end of execution of transaction.
- Aborts are cheaper with deferred.
Immediate: unnecessary writes of replicas that need to be undone.
- Deferred tends to delay commit.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

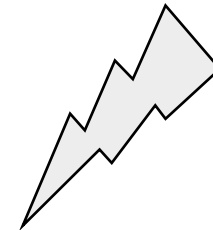
Missing Writes

Virtual Partition

Summary

Example for Deadlock with 2PL

- $T_1: r_1[x] \rightarrow w_1[y] \rightarrow c_1; T_2: w_2[y] \rightarrow w_2[x] \rightarrow c_2$
- Chronology:
 1. Both transactions do not have any locks initially.
 2. Scheduler receives $r_1[x]$ from TM. $rl_1[x]$, scheduler submits $r_1[x]$ to DM.
 3. Scheduler receives $w_2[y]$ from TM. $wl_2[y]$, Scheduler submits $w_2[y]$ to DM.
 4. Scheduler receives $w_2[x]$ from TM. $wl_2[x]$ not possible. Delay.
 5. Scheduler receives $w_1[y]$ from TM. $wl_1[y]$ not possible. Delay.
- External reset of deadlock required.



Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

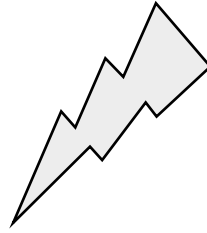
Missing Writes

Virtual Partition

Summary

Aborts Necessary not only with 2PL.

- Execution - example:
 - ◆ $r_1[x]$
 - ◆ $r_2[x]$
 - ◆ $w_2[x]$
 - ◆ c_2
 - ◆ $w_1[x]$
- Not serializable. Scheduler must reject last write operation (i.e., abort T_1).



Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Outlook

- To be presented next:
 - ◆ Primary-Copy,
 - ◆ Write-All.
- Not always useful.
- Schemes are dealt with for didactic reasons, each covers one end of the spectrum.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Primary-Copy Technique (1)

- More efficient handling of updates:
 - ◆ Only one copy (**primary copy**) is updated.
 - ◆ Replicas are updated asynchronously (i.e., by separate transaction) from the node with the primary copy, before the end of the update transaction „as soon as possible“. Inconsistencies are possible.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

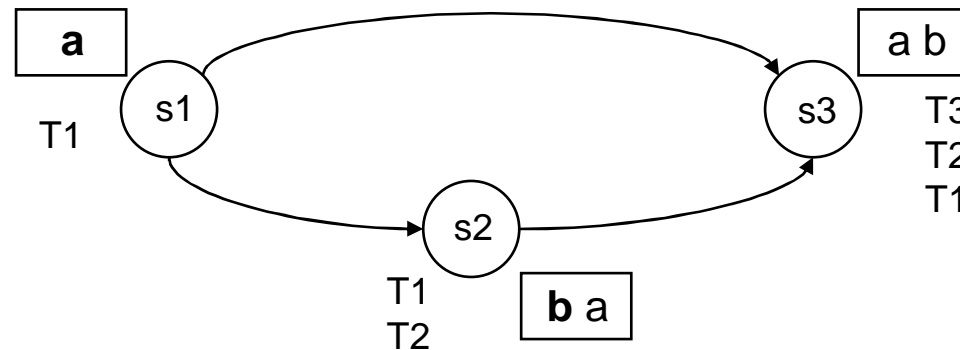
QCA

Missing Writes

Virtual Partition

Summary

Primary-Copy Technique (2)



Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

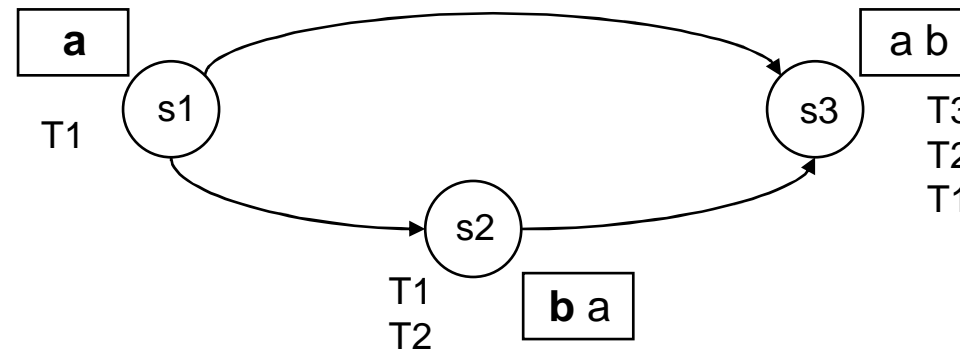
Virtual Partition

Summary

- Three transactions: T_1 on Site 1 updates a.
 T_2 on Site 2 reads a and writes b.
 T_3 on Site 3 reads a and b.
- Lazy propagation of updates – possible sequence of executions:
 - ◆ on Site 2: $w_1[a]$ $r_2[a]$ $w_2[b]$
 - ◆ on Site 3: $w_2[b]$ $r_3[a]$ $r_3[b]$ $w_1[a]$



Primary-Copy Technique (3)



- To read consistent versions of data objects, transaction would have to read a from s1 and b from s2 and would have to use 2PL.

Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Primary-Copy Technique (4)

- Save messages:
primary-copy node sends bundle
of several updates to nodes holding copies
(will be updated with a certain delay).
- *Distribute* primary copies of different objects
to avoid bottlenecks;
try to have each object at node
that updates it most frequently
→ reduced communication.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Reads with Primary-Copy Technique – Alternatives (1)

1. Read access only to primary copy.
Most current state, but what are replicas good for?

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Reads with Primary-Copy Technique – Alternatives (2)

2. Read access to local copies,
but locking at node with primary copy.
 - ◆ When lock request is processed,
check if local copy is up-to-date,
 - ◆ either wait for update
or access (non-local) copy
that has already been updated,
e.g., primary copy.
 - ◆ Interesting with large data objects,
i.e., locking takes less time than reading.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

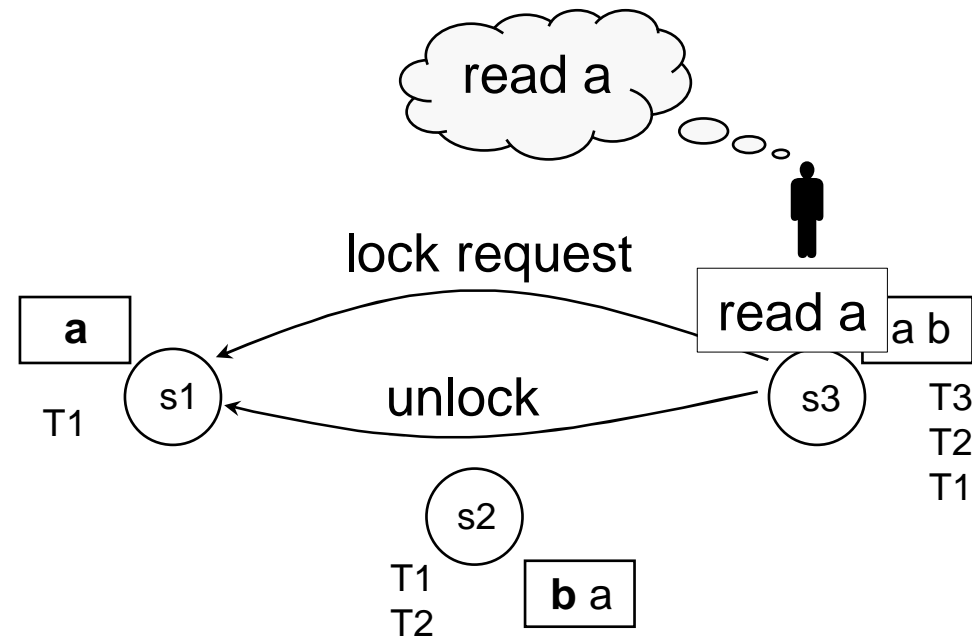
Missing
Writes

Virtual
Partition

Summary

Reads with Primary-Copy Technique – Alternatives (3)

- Illustration of second variant:



Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Reads with Primary-Copy Technique – Alternatives (4)

3. Read operations are worked off locally.
 - ◆ No synchronization via primary-copy node,
 - ◆ local copy is not necessarily up-to-date, but typically only a few seconds old.
 - ◆ Inconsistent view of the database is possible. Different database states.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

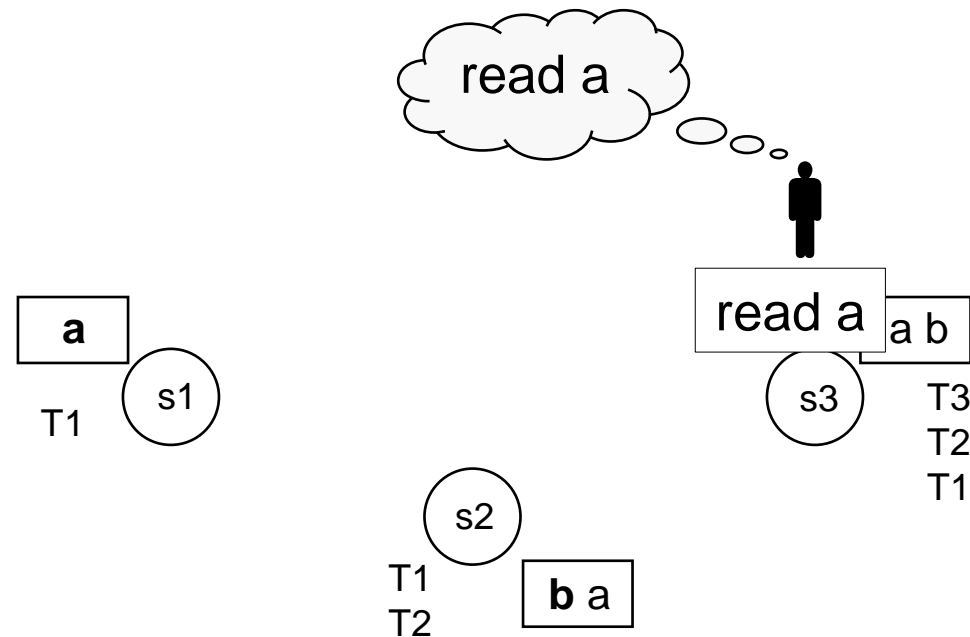
Missing Writes

Virtual Partition

Summary

Reads with Primary-Copy Technique – Alternatives (5)

- Illustration of third variant:



Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Reads with Primary-Copy Technique – Alternatives (6)

3. Read operations are worked off locally (continuation).
 - ◆ If read transaction requires consistent view:
 - resort to Alternatives 1 or 2,
 - extend synchronization scheme s.t. transaction sees outdated, but consistent database state (e.g., multi-version scheme).

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Primary-Copy Technique – Dealing with Failures (1)

- **Network partitioning:**
 - ◆ Keep on reading and updating objects whose primary copy can still be reached,
 - ◆ no access to other objects is feasible, even if some copies can still be reached, only read access to possibly outdated copy according to Alternative 3.

Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

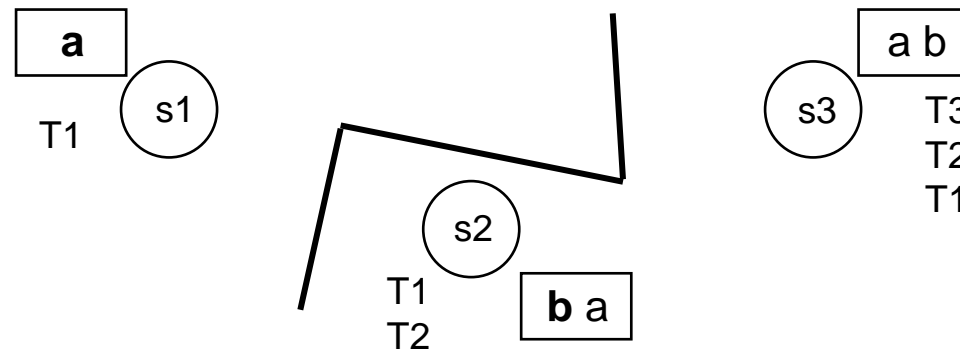
Virtual
Partition

Summary

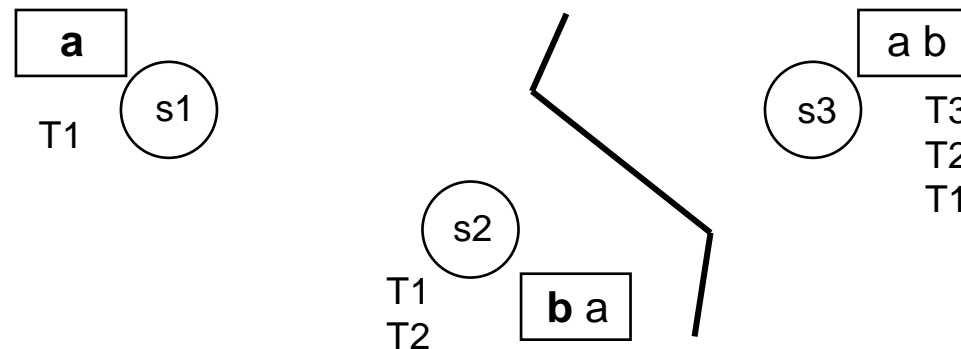
Primary-Copy Technique – Dealing with Failures (2)

◆ Illustration:

– First case: We can access b from s2 and s3.



– Second case:
We cannot access anything from s3.



Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Primary-Copy Technique – Dealing with Failures (3)

- **Failure of a node:**
 - ◆ Objects with primary copy at this node cannot be modified,
 - ◆ avoid long-lasting interruptions:
point out new primary-copy node,
 - ◆ prerequisite: new primary copy can be updated with most recent value of data object,
 - ◆ in case of network partitioning,
point out new primary copy
in at most *one* partition, e.g., in partition with more than half of the nodes.

Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Z

Write-All Approach (1)

- Most prominent variant:
Write-All-Read-Any-/
Read-One-Write-All Strategy (ROWA).
- *Synchronous* update of all replicas before update transaction commits.
- Replicas are always up-to-date, **reader** may access any copy, choice should minimize communication or should balance load;
- increased availability for read accesses: read is feasible as long as *one* copy still available.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Write-All Approach (2)

- **Updates**, however:
 - ◆ With locking, acquire locks of *all* copies before each update – significant increase of communication effort.
 - ◆ Write-All includes Strict 2PL and 2PC.
 - ◆ All affected nodes participate in commit protocol:
 - Phase 1 sends all updates to all other nodes and records them there,
 - Phase 2 updates replicas and frees locks.
 - ◆ Both immediate and deferred are possible.
 - ◆ Reduced availability: update is feasible only if all replicas can be reached.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

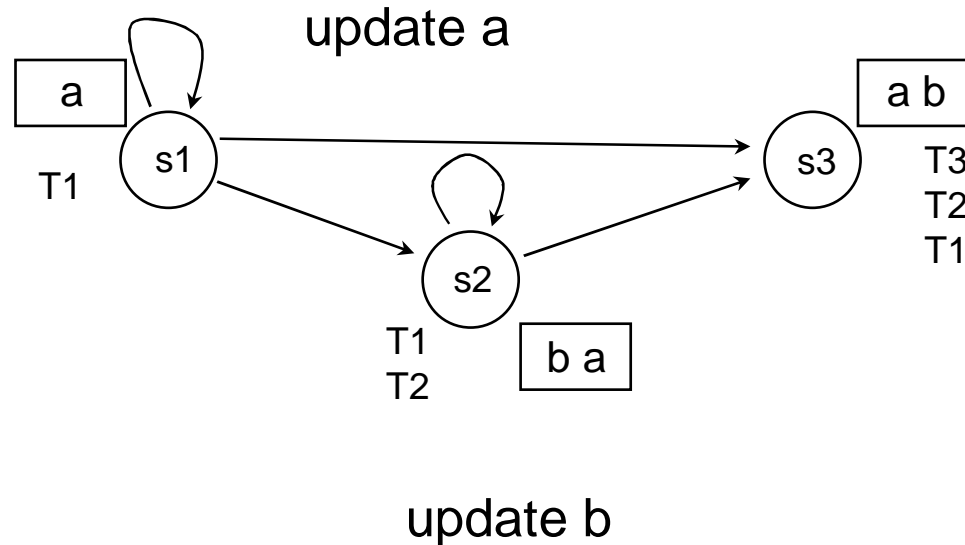
Missing Writes

Virtual Partition

Summary

Write-All Approach (3)

- Illustration (schematic, without commit processing):



- No primary copies any more.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

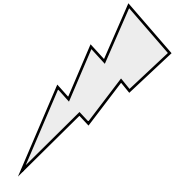
Missing Writes

Virtual Partition

Summary

One-Copy-Serializability – Illustration

- Example from above.
- Data objects a, b; copies on nodes X, Y.
- T1: r(a) w(b); T2: r(b) w(a)
- Not 1SR:
 1. T1 reads copy of a on X
 2. T2 reads copy of b on Y
 3. T1 writes both copies of b and commits.
 4. T2 writes both copies of a and commits.



Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

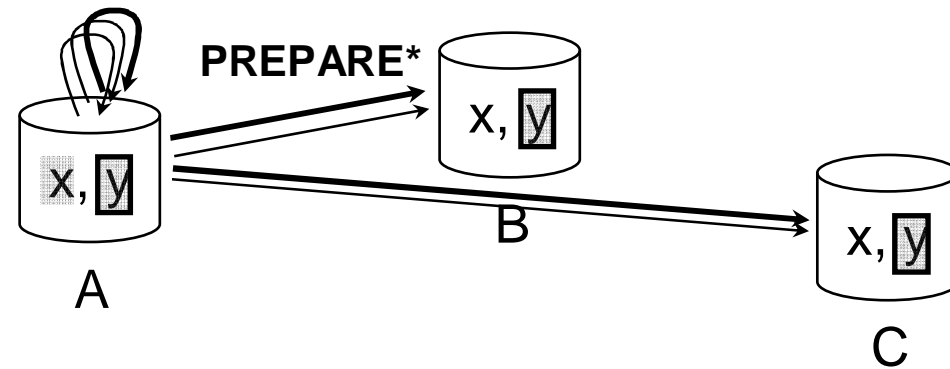
Missing Writes

Virtual Partition

Summary

Write-All Approach (1)

- $T_1: r_1[x] \rightarrow w_1[y]; T_2: w_2[y] \rightarrow w_2[x]$
- A is coordinator for T_1 ; C is coordinator for T_2 .



PREPARE* -
not only 'Prepare',
but check for all
locks required.

- First operation: $r_1[x]$.
 T_1 has the following locks: $r[x_A]$
- Next operation: $w_1[y]$.
 T_1 has the following locks $w[y_A], w[y_B], w[y_C]$
- Next operation: $w_2[y]$.
Must wait until T_1 commits and clears locks.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

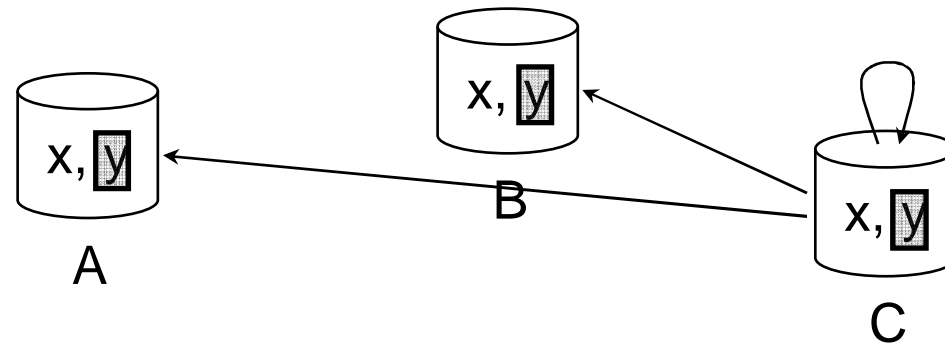
Missing
Writes

Virtual
Partition

Summary

Write-All Approach (2)

- $T_1: r_1[x] \rightarrow w_1[y]; T_2: w_2[y] \rightarrow w_2[x]$
- A is coordinator for T_1 ; C is coordinator for T_2 .



- First operation: $r_1[x]$.
 T_1 has the following locks: $r[x_A]$
- Next operation: $w_1[y]$.
 T_1 has the following locks $w[y_A], w[y_B], w[y_C]$
- Next operation: $w_2[y]$.
Must wait until T_1 clears locks needed.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

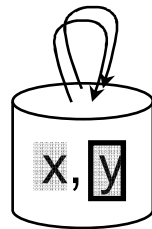
Missing Writes

Virtual Partition

Summary

Write-All Approach (3)

- $T_1: r_1[x] \rightarrow w_1[y]; T_2: w_2[y] \rightarrow w_2[x]$
- A is coordinator for T_1 ; C is coordinator for T_2 .



A

- Sequence is identical to situation with one copy.
- First operation: $r_1[x]$. T_1 has lock: $r[x_A]$
- Next operation: $w_1[y]$. T_1 has lock $w[y_A]$
- Next operation: $w_2[y]$.
Must wait until T_1 commits and clears locks.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

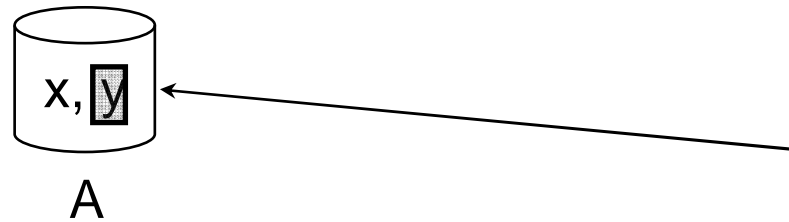
Missing Writes

Virtual Partition

Summary

Write-All Approach (4)

- $T_1: r_1[x] \rightarrow w_1[y]; T_2: w_2[y] \rightarrow w_2[x]$
- A is coordinator for T_1 ; C is coordinator for T_2 .



- Sequence is identical to situation with one copy.
- First operation: $r_1[x]$. T_1 has lock: $r[x_A]$
- Next operation: $w_1[y]$. T_1 has lock $w[y_A]$
- Next operation: $w_2[y]$.
Must wait until T_1 commits and clears locks.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

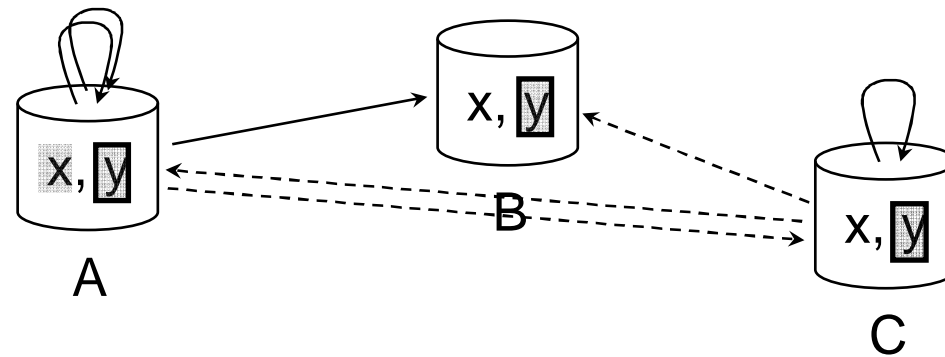
Missing Writes

Virtual Partition

Summary

Write-All Approach (5)

- $T_1: r_1[x] \rightarrow w_1[y]; T_2: w_2[y] \rightarrow w_2[x]$
- A is coordinator for T_1 ; C is coordinator for T_2 .



- First operation: $r_1[x]$.
 T_1 has the following locks: $r[x_A]$
- Next operations: $w_1[y]$ and $w_2[y]$, in parallel.
 T_1 has the following locks $w[y_A], w[y_B]$
 T_2 has the following locks $w[y_C]$

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

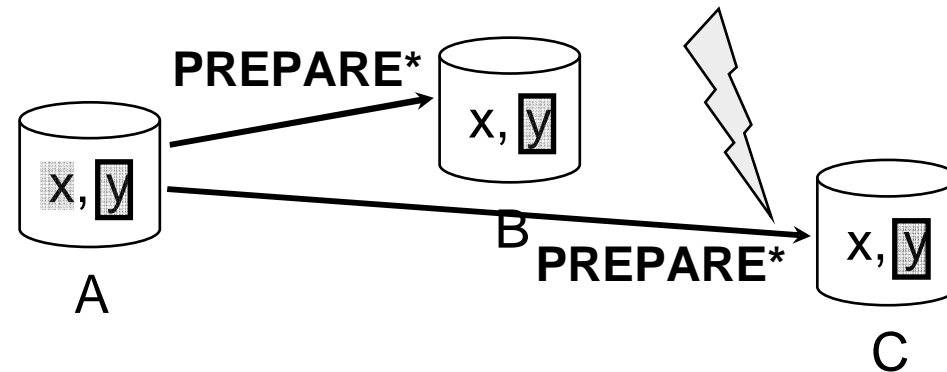
Missing Writes

Virtual Partition

Summary

Write-All Approach (6)

- $T_1: r_1[x] \rightarrow w_1[y]; T_2: w_2[y] \rightarrow w_2[x]$
- A is coordinator for T_1 ; C is coordinator for T_2 .



- T_1 cannot commit because it must have updated all copies of y .
- Similarly, T_2 .

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Write-All Approach is Correct.

- Remember that Write-All includes concurrency-control mechanisms (locking) at each node individually.
- Correctness – execution equivalent to serial execution on database without replication („1SR“).
- Each transaction always overwrites all copies of x.
- Next transaction can read arbitrary copy – always same value.
- Effect – as with execution with only one copy.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Z

Write-All-Available

- Variant **Write-All-Available**: mitigates (entschärft) availability problem.
- Only replicas on nodes that are available are locked and updated.
- ‚Write-All-Available‘ typically leads to execution that is not 1SR.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Write-All-Available – Example

- Example of execution that is not 1SR:
 - ◆ Data object x on nodes A and B, y on C.
 - ◆ Node B fails temporarily.
 - ◆ $H_1 = w_0[x_A] w_0[x_B] w_0[y_C] c_0 \underbrace{r_1[y_C] w_1[x_A] c_1 r_2[x_B]}_{\text{failure of B}} w_2[y_C] c_2$
- T_2 reads x_B from T_0 –
 T_0 is not the last transaction that has written x
 (as opposed to execution on one-copy database:
 $H_1' = w_0[x] w_0[y] c_0 r_1[y] w_1[x] c_1 r_2[x] w_2[y] c_2$
 $H_1'' = w_0[x] w_0[y] c_0 r_2[x] w_2[y] c_2 r_1[y] w_1[x] c_1$).
- Necessary: do not read copies that are not yet up-to-date after failure/recovery.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Write-All-Available – Continuation

- Write-All-Available may even result in execution that is not 1SR if there are only node failures, and we leave aside recovery (see example that follows).
- Example shows: mechanism treats data objects and copies of data objects differently. No conflict on the level of copies.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

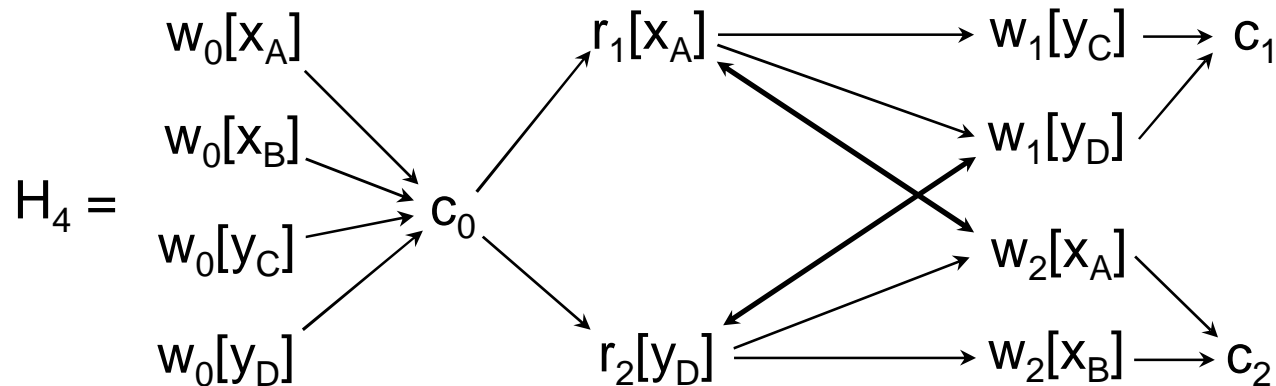
Missing Writes

Virtual Partition

Summary

Write-All-Available – Example 2 (1)

- Transactions T_0, T_1, T_2 .
- History with Write-All-Available and 2PL:



- History will not occur,
either T_1 must read from T_2 or vice versa.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

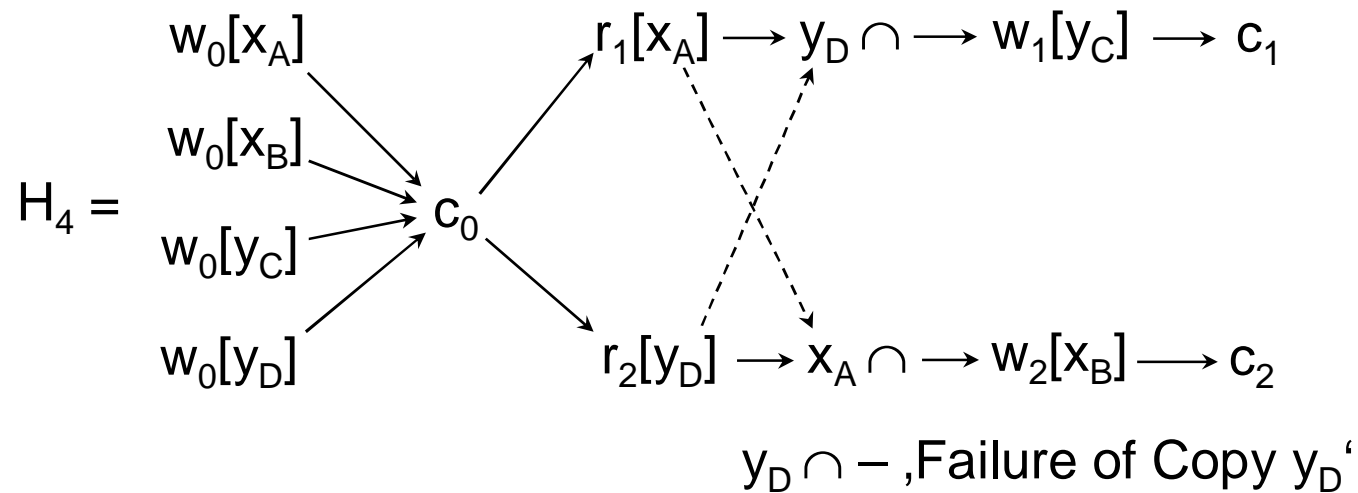
Missing Writes

Virtual Partition

Summary

Write-All-Available – Example 2 (2)

- Transactions T_0, T_1, T_2 .
- History with Write-All-Available and 2PL:



- History not 1SR,
since there is no order of T_1 and T_2 .

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Available Copies – Observation

- If failures did not happen during transaction, Write-All-Available would work!
- The following examples serve as illustration.
- Again, at most one failure, no recovery.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

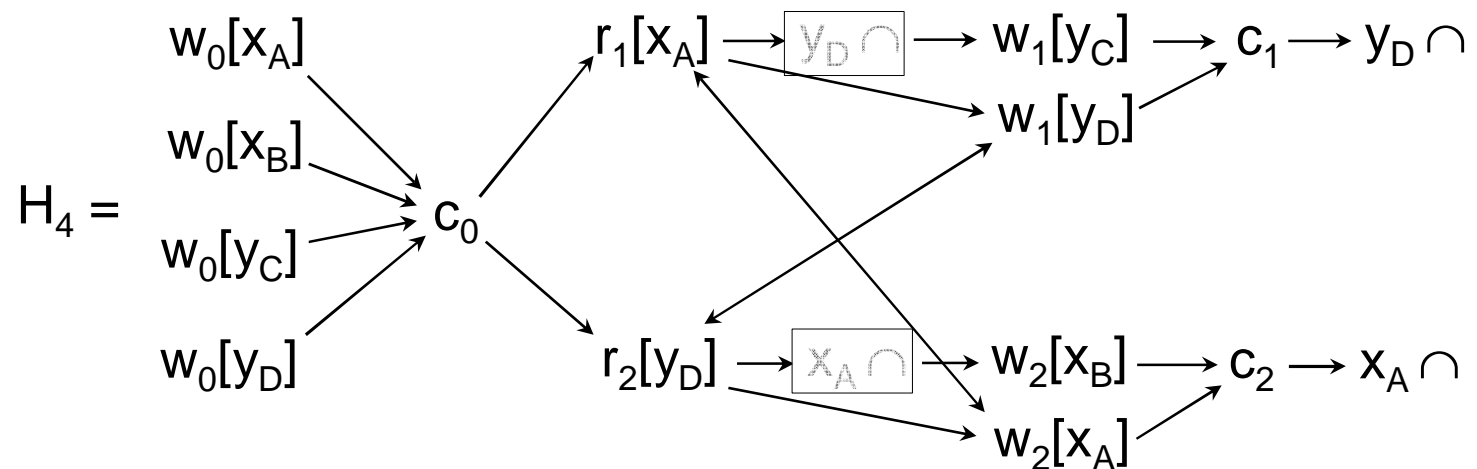
Missing Writes

Virtual Partition

Summary

Available Copies – Example 1

- Transactions T_0, T_1, T_2 .
- History with failures after transactions:



$y_D \cap$ – 'Failure of Copy y_D '

- Conflicts will be detected. How?
One transaction might be reset.
Will result in serial execution.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

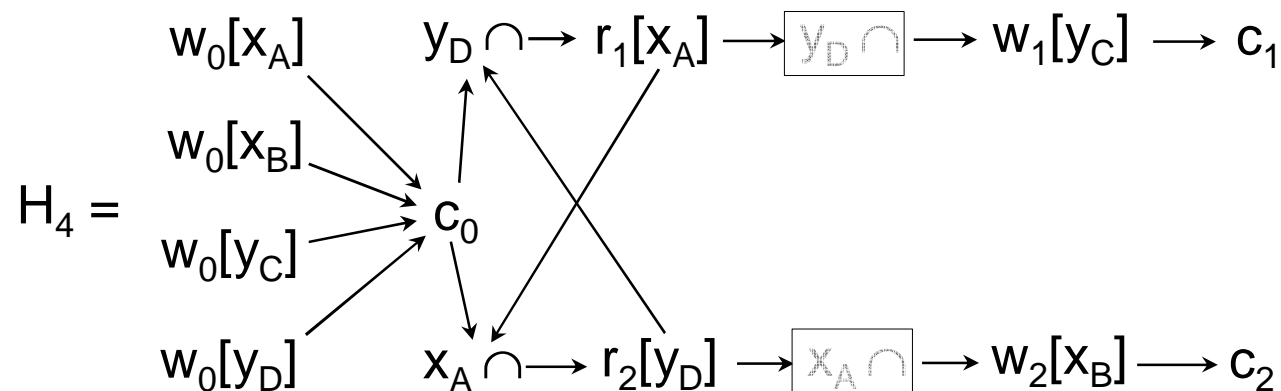
Missing Writes

Virtual Partition

Summary

Available Copies – Example 2

- Transactions T_0, T_1, T_2 .
- History with failures before transactions:



$y_D \circlearrowleft$ – 'Failure of Copy y_D '

- History now contains a cycle and cannot happen!

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

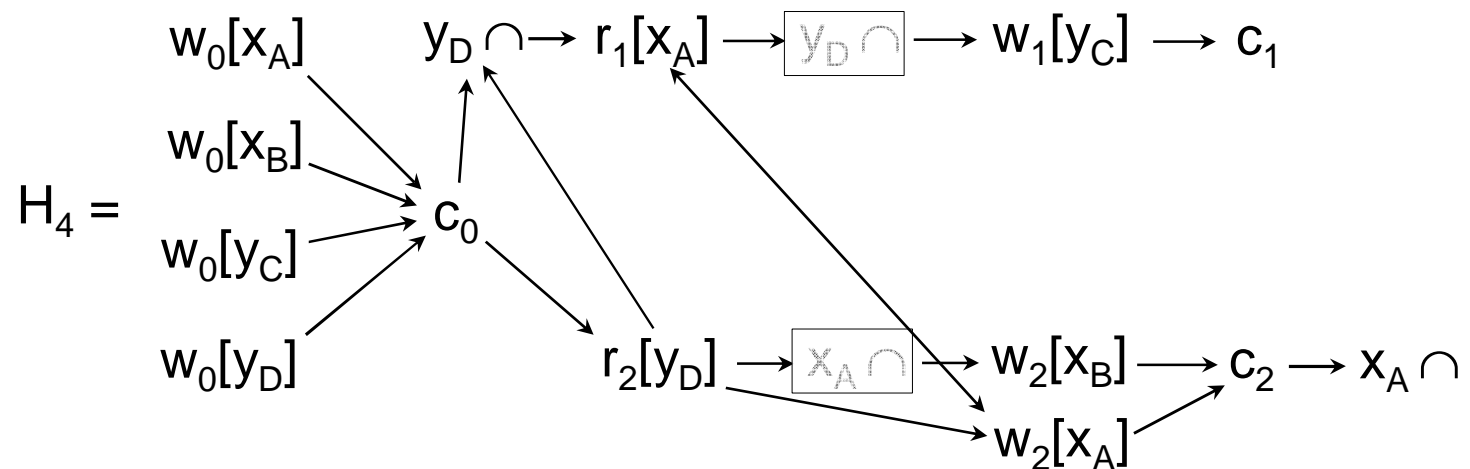
Missing Writes

Virtual Partition

Summary

Available Copies – Example 3

- Transactions T_0, T_1, T_2 .
- History with failures before or after transactions:



$y_D \cap -$, Failure of Copy y_D

- Situation is analogous to the previous one.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Write-All Available – Discussion

- Thus, Write-All Available would work if we could ensure that there are no failures DURING transaction.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Z

Available Copies Algorithm – Classification and Overview

- Refinement of write-all-available to avoid the problems sketched so far.
- Site failures, but no communication failures.
- *Available Copies* = write-all-available + **validation phase** at the end of each transaction.
- Approach:
 - ◆ Treat failures as atomic events.
 - ◆ Arrange in serialization order.
 - ◆ If failure occurs ,during‘ transaction, abort transaction.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Available Copies Algorithm – Features

- Strict 2PL.
- 2PC.
- Assumptions to ease presentation:
 - ◆ fixed set of copies,
 - ◆ each copy – at most one failure, no recovery.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Available Copies Algorithm – Processing of Reads and Writes (1)

- x_A is initialized := write of x_A has already taken place.
- read(x):
read of a (arbitrary) copy must be feasible (initialized and site not down).

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

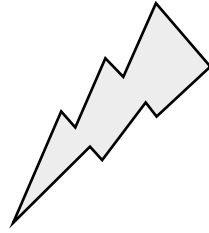
Summary

– same slide
as before –

Aborts Necessary not only with 2PL.

- Introduction
- Primary-Copy
- Write-All
- Write-All-Available
- Available Copies
- Comm. Failures
- QCA
- Missing Writes
- Virtual Partition
- Summary

- Execution - example:
 - ◆ $r_1[x]$
 - ◆ $r_2[x]$
 - ◆ $w_2[x]$
 - ◆ c_2
 - ◆ $w_1[x]$
- Not serializable. Scheduler must reject last write operation (i.e., abort T_1).



Available Copies Algorithm – Processing of Reads and Writes (2)

- write(x):
 - ◆ algorithm tries to overwrite all copies of x.
 - ◆ Reject (abort) of the write of a copy. \Rightarrow Reject. (Previous transparency – illustrates abort of a write.)
 - ◆ No write is successful. \Rightarrow Reject.
 - ◆ Otherwise: write(x) has been successful.
- In addition to processing of reads and writes, as described so far: *validation* (topic of the subsequent transparencies).

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Available Copies Algorithm – Validation (1)

Overview and classification:

- Validation is necessary to ensure correctness.
 - ◆ Failures as atomic events.
 - ◆ Allows for more precise ordering of operations and failure events.
- Validation protocol starts after acknowledgement or timeout of reads and writes of a transaction.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Available Copies Algorithm – Validation (2)

Overview and classification (cont.):

- Two phases:
 1. Missing Writes Validation,
 2. Access Validation.
- Order of these phases matters (Example 3 in what follows).

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

Missing Writes Validation

- Message *UNAVAILABLE*(x_A) to Site A that has not been available.
- Any reply \Rightarrow abort.
- In case of no reply:
We know that site is not available.
Assumption: site failures only.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Access Validation

- Message *AVAILABLE*(x_A).
- All nodes must reply, otherwise abort.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

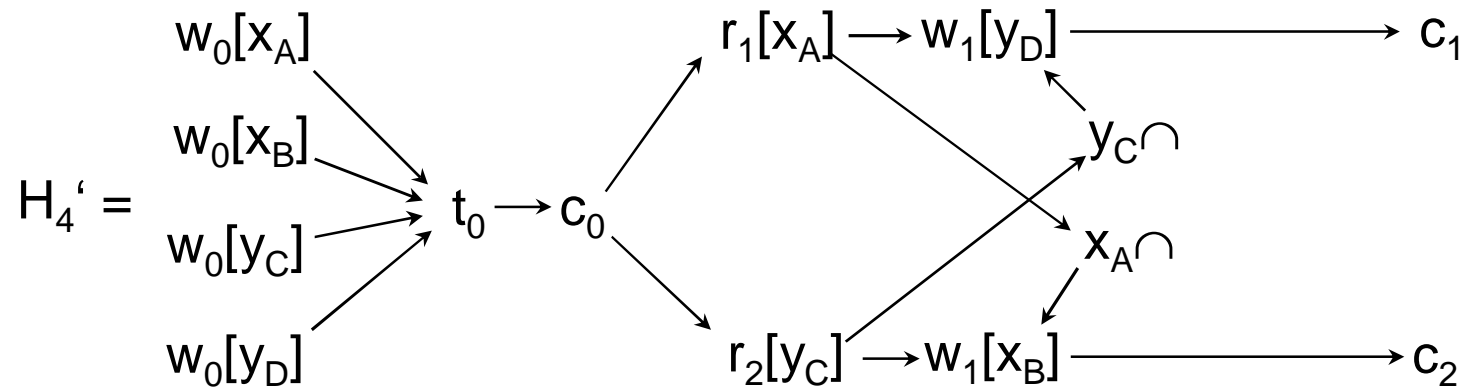
Missing
Writes

Virtual
Partition

Summary

Available Copies Algorithm – Example 1

t_i – point of time of begin of access validation step.



Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

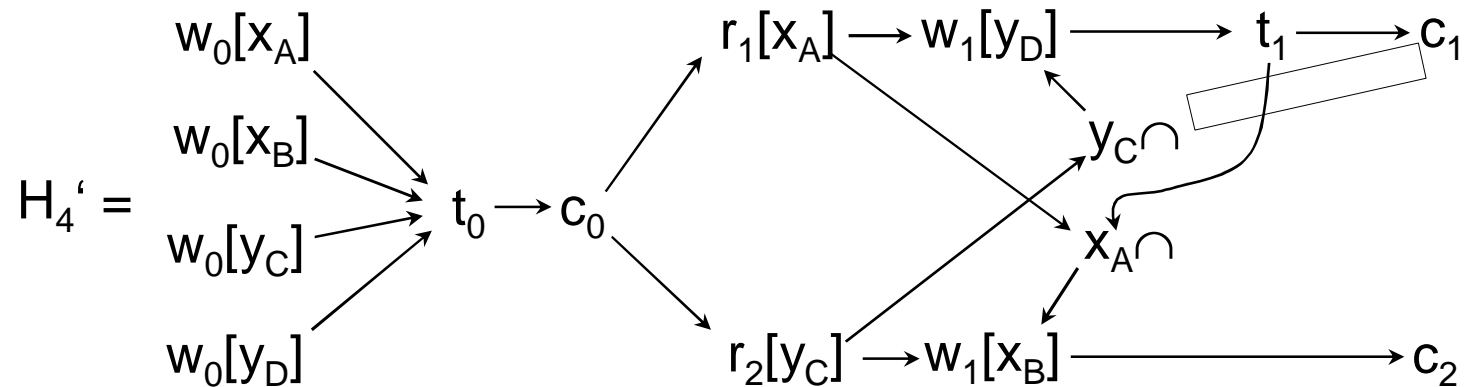
Missing Writes

Virtual Partition

Summary

Available Copies Algorithm – Example 1

t_i – point of time of begin of access validation step.



Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

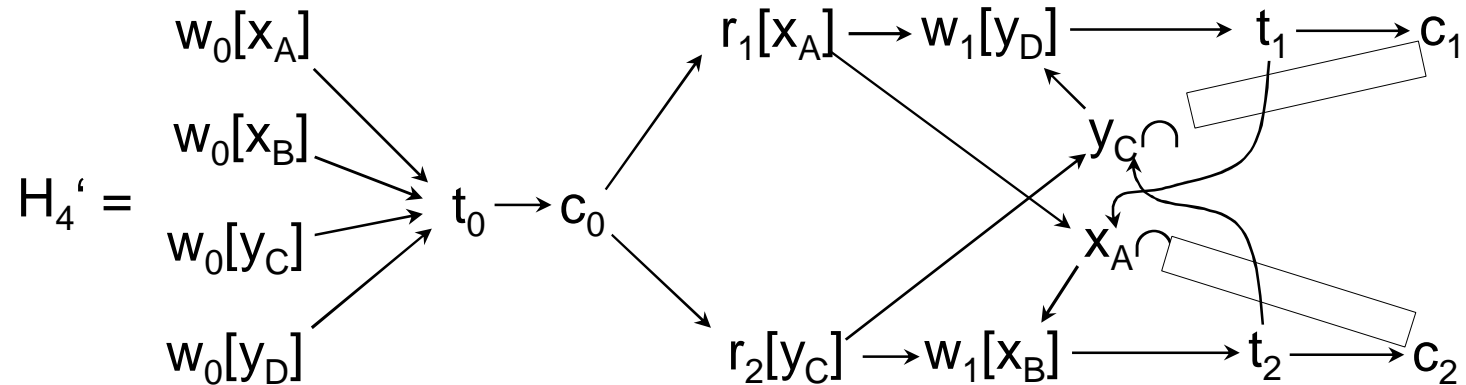
Missing
Writes

Virtual
Partition

Summary

Available Copies Algorithm – Example 1

t_i – point of time of begin of access validation step.



Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

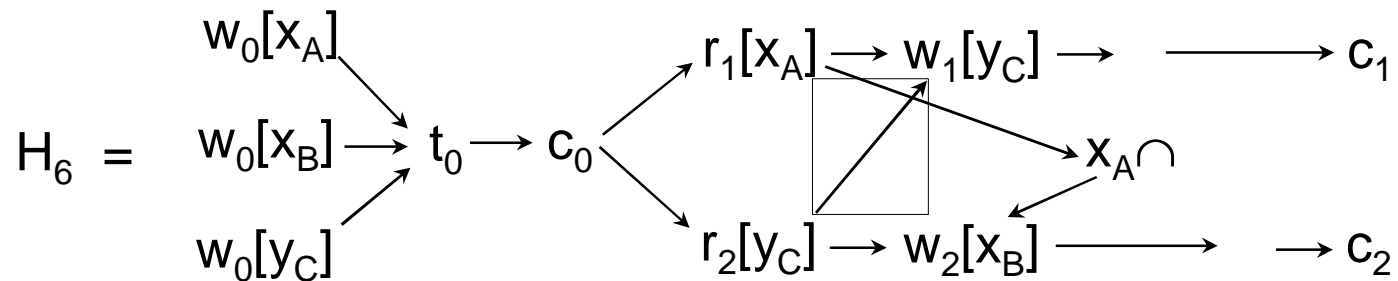
QCA

Missing
Writes

Virtual
Partition

Summary

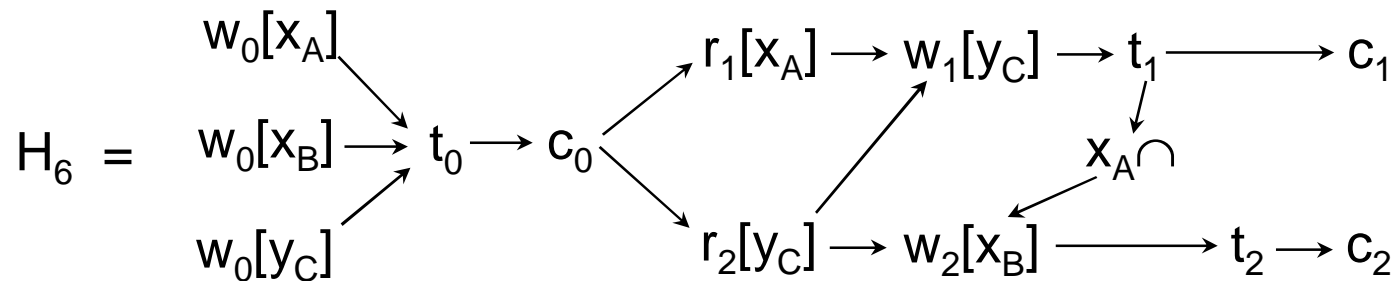
Available Copies Algorithm – Example 2



- Execution is not 1SR:
 T_1 does not read from T_2 and vice versa.
- Validation alone is not sufficient.
- Strict 2PL disallows this execution.

Introduction
 Primary-Copy
 Write-All
 Write-All-Available
Available Copies
 Comm. Failures
 QCA
 Missing Writes
 Virtual Partition
 Summary

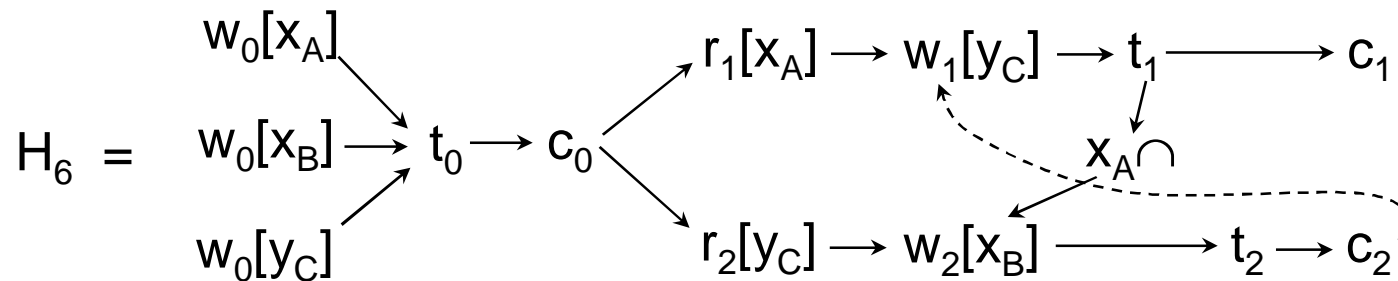
Available Copies Algorithm – Example 2



- Execution is not 1SR:
 T_1 does not read from T_2 and vice versa.
- Validation alone is not sufficient.
- Strict 2PL disallows this execution.

Introduction
 Primary-Copy
 Write-All
 Write-All-Available
Available Copies
 Comm. Failures
 QCA
 Missing Writes
 Virtual Partition
 Summary

Available Copies Algorithm – Example 2

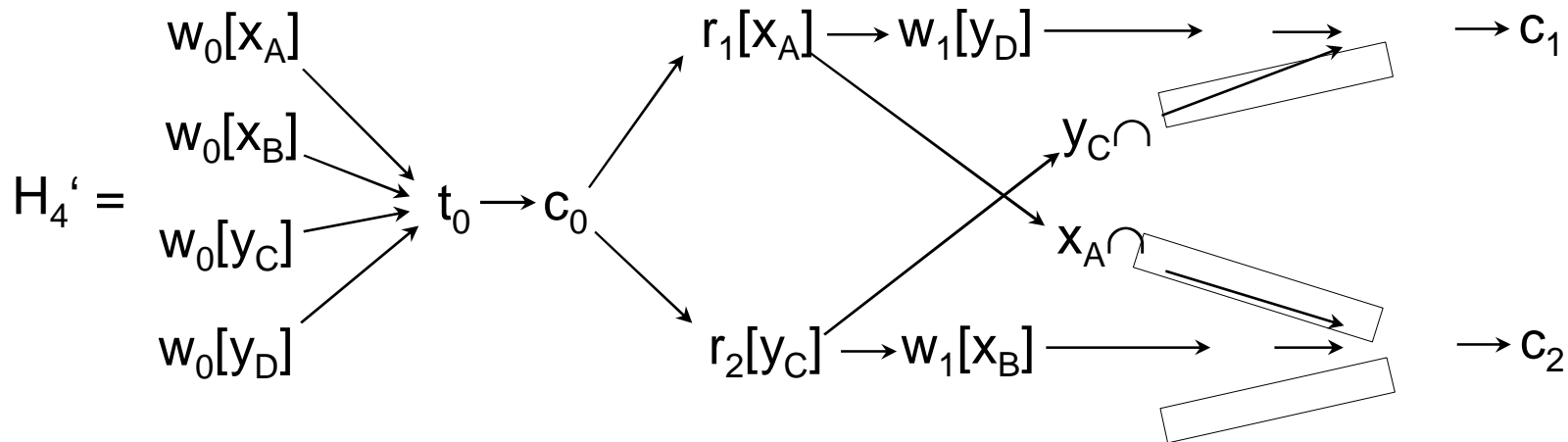


- Execution is not 1SR:
 T_1 does not read from T_2 and vice versa.
- Validation alone is not sufficient.
- Strict 2PL disallows this execution.

Introduction
 Primary-Copy
 Write-All
 Write-All-Available
Available Copies
 Comm. Failures
 QCA
 Missing Writes
 Virtual Partition
 Summary

Available Copies Algorithm – Example 3

- a_i – point of time of begin of access validation step.
- mw_i – point of time of begin of missing writes validation step.
- Available Copies – missing writes validation before access validation. Now different order:



Introduction

Primary-Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

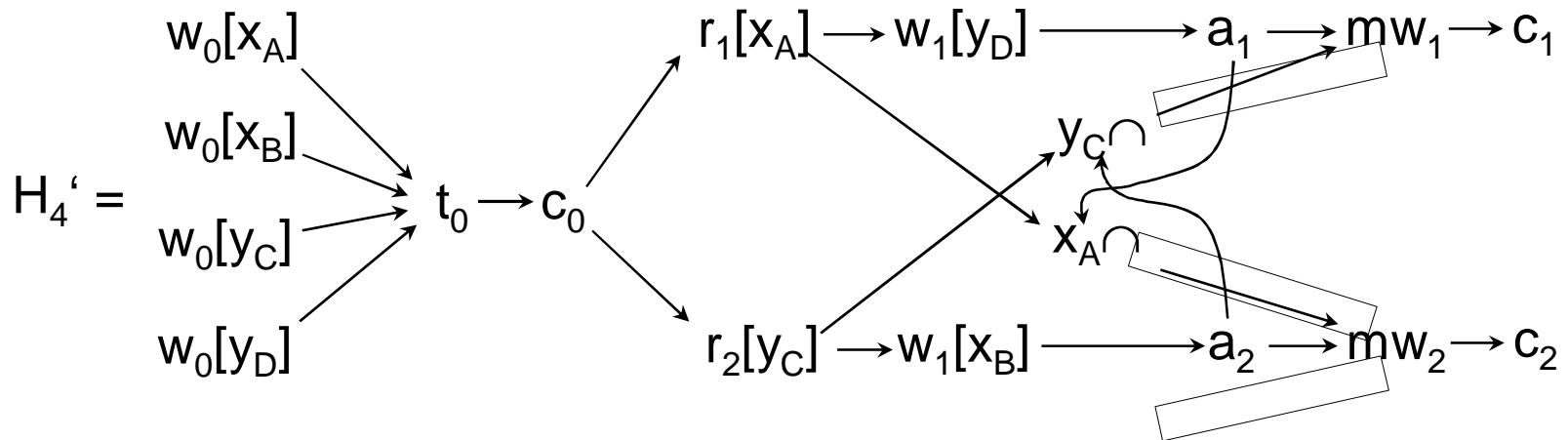
Missing
Writes

Virtual
Partition

Summary

Available Copies Algorithm – Example 3

- a_i – point of time of begin of access validation step.
- mw_i – point of time of begin of missing writes validation step.
- Available Copies – missing writes validation before access validation. Now different order:



Z

- Introduction
- Primary-Copy
- Write-All
- Write-All-Available
- Available Copies
- Comm. Failures
- QCA
- Missing Writes
- Virtual Partition
- Summary

How Expensive is Available-Copies Algorithm, after all? (1)

At first sight extensive communication effort, but:

- comprise all *UNAVAILABLE-/AVAILABLE* messages for all data objects of a site in one message.
- *UNAVAILABLE* messages are not required if all sites are available (this is what we typically expect).
- *AVAILABLE* message together with *PREPARE* message of Atomic Commitment Protocol.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

How Expensive is Available-Copies Algorithm, after all? (2)

- Better than Write-All
 - we do not need to write ALL copies.
Sufficient to write available copies.
- Better than Primary Copy, if you wish
 - we do not need to write one specific copy.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Z

Components

- Component (a.k.a. partition) := set of sites that can communicate with each other.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

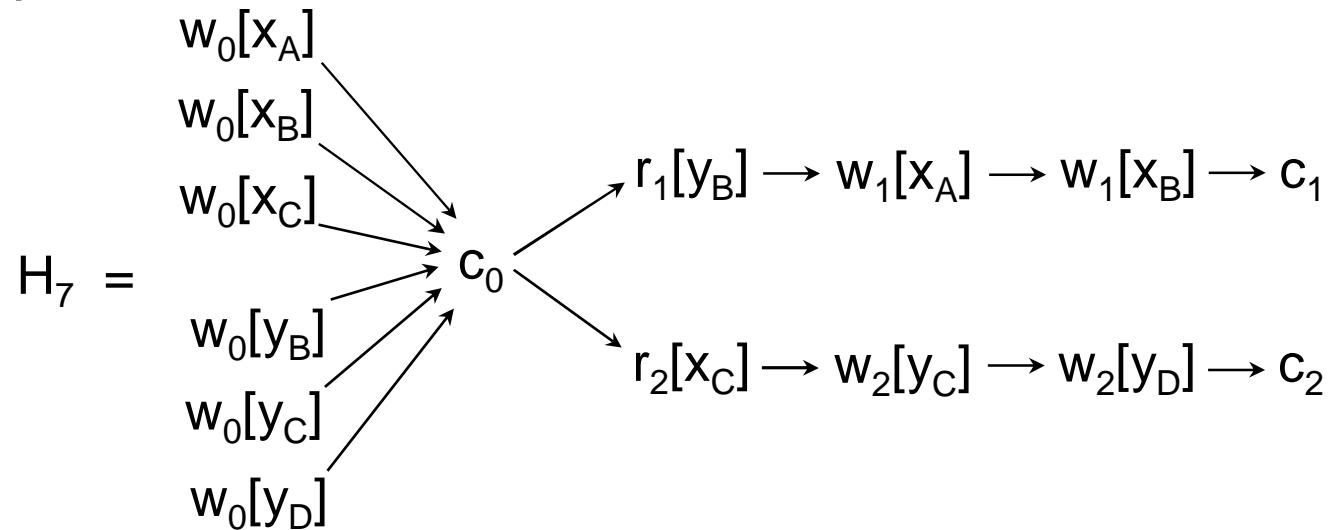
Missing
Writes

Virtual
Partition

Summary

Communication Failures

- Available-Copies algorithm does not work in presence of communication failures.
- Example:



- Two *components* $P_1=\{A, B\}$, $P_2=\{C, D\}$.
Communication failure right after c_0 .
- No communication between components in example.
No failure during T_1 or T_2 .

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Site Quorums

- Each component must decide for itself if it may process transaction.
- Each site has a weight.
- *Quorum*: set of sites with sum of weights $> \frac{1}{2} \cdot$ total weight.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Site Quorums – Comments

- Total failure of system is feasible.
E.g., if no component has a quorum.
- Weights – significance of component.
- Observation: quorum rule
not necessary for non-replicated data objects.
- In what follows, we work with copy quorums
instead of site quorums.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

CC Algorithms

Coping with Communication Failures

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

- *Quorum Consensus Algorithm*
Each transaction must be able to access quorum of copies.
- *Virtual Partition Algorithm*
Each site knows set of available sites and compares them to the ones of other sites in due course.
- Comparison:
 - ◆ failure handling – less overhead with QCA,
 - ◆ QCA is more expensive as long as there are no failures.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

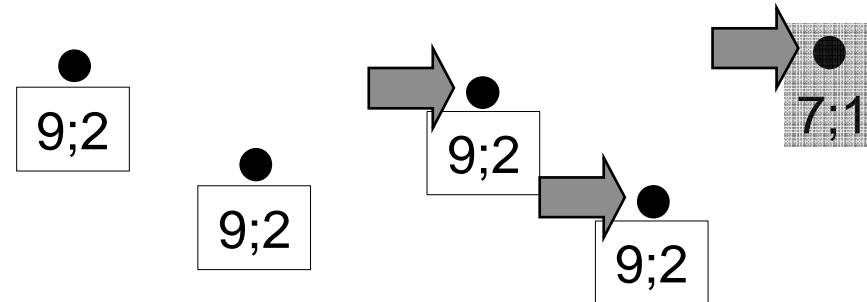
Summary

Quorum Consensus Algorithm (1)

- Threshold values RT , WT , s.t.
 - ◆ $2 \cdot WT > \text{total weight}$,
 - ◆ $RT + WT > \text{total weight}$.
- Read- and write quorum.
- Observation:
read and write quorum always overlap.
- TM transforms reads and writes of data objects to reads and writes on copies:
 - ◆ $\text{write}(x) \rightarrow$ overwrite all copies of the quorum,
 - ◆ $\text{read}(x) \rightarrow$ read all copies of the read quorum, return value of most recent copy.

QC Algorithm – Illustration

- 5 nodes; WT=3, RT=3



value; version number

- write(9)
- read

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Quorum Consensus Algorithm (2)

- Versions are numbered –
write(x) consists of two steps:
 1. Get all version numbers of copies of x in the write quorum.
 2. As part of the Atomic Commitment Protocol, send new version number and new value to agents.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Quorum Consensus Algorithm (3)

Effect of QCA is identical with the one of execution over 1C database.

- T_j reads x from the most recent transaction T_i .
- Namely,
 - ◆ T_i has written write quorum.
 - ◆ T_j reads read quorum.
 - ◆ Intersection is not empty.
 - ◆ T_i has furnished all copies in write quorum with maximal version number.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Z

Quorum Consensus – Choice of Weights

- High flexibility** in fixing the costs of read and write operations and the availability in case of failures.
- **High weight of a copy:**
access to this node is relatively fast.
Why is this the case?
 - **Small RT/WT:**
processing of read/write is relatively easy, since less votes need to be collected;
higher availability for same reason.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Quorum Consensus – Choice of Weights: Example (1)

- Object A replicated in four nodes K_1 to K_4 , choice of weights $\langle 2, 1, 1, 1 \rangle$, total weight = 5.
- $RT = 3, WT = 3$:
 - ◆ pull in two to three nodes for read- or write access;
 - ◆ preferential treatment of K_1 : only one node more;
 - ◆ if a node fails, operations are still possible; if K_1 available, system is operational even if two nodes fail.

Quorum Consensus – Choice of Weights: Example (2)

- Object A replicated in four nodes K_1 to K_4 , choice of weights $\langle 2, 1, 1, 1 \rangle$, total weight = 5.
- RT = 2, WT = 4:
 - ◆ Preferential treatment of readers – read accesses in K_1 feasible locally, but more than two nodes necessary when writing;
 - ◆ if K_1 fails, object cannot be modified any more.

Choice of Weights with QCA: Special Cases (1)

- Mimick ROWA and primary-copy technique by appropriate choice of parameters.
- **ROWA:**
 - ◆ One weight per copy, $RT = 1$, $WT = \text{total weight}$.
 - ◆ Allows for local read operations.
- **Primary-Copy:**

Primary copy has one weight, all other replicas *have weight zero*, $RT = WT = \text{total weight}$.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Choice of Weights with QCA: Special Cases (2)

- **Only two copies:**
 - ◆ Different from majority votes, i.e., same weight for all,
 - ◆ e.g., weights $\langle 2, 1 \rangle$ and $RT = WT = 2$,
 - ◆ Read and write quorum can be achieved locally in first node.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Quorum Consensus Algorithm – Discussion (1)

- Advantage:
No implementation of recovery is necessary.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

QC Algorithm – Illustration

- 5 nodes; WT=3, RT=3

●
9;2

●
9;2

●
9;2

●
9;2

●
7;1

value; version number

- write(9)
- read
- write(10)
- write(28)

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Quorum Consensus Algorithm – Discussion (2)

- Disadvantage:
 - ◆ Read of several copies.
 - ◆ Many copies necessary to tolerate individual site failures (with majority votes).
 - Three copies for one failure.
 - Two copies do not yield an improvement.
 - ◆ Topology of the system may not change dynamically (i.e., we do not talk about this issue here).

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Z

Missing Writes Algorithm (1)

- QCA is rather expensive, in particular if there are no failures.
- Principle of *Missing Write Algorithm*:
 - ◆ As long as there is no failure, read only one copy and write all copies (ROWA; *normal mode*),
 - ◆ In case of failure, i.e., current transaction knows about missing write, shift to QC (*failure mode*).

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Missing Writes Algorithm (2)

- *Transaction T_i knows about missing write. :=*
 - ◆ No acknowledgement of write of a copy by T_i , or
 - ◆ missing write of the copy is known to T_j , and \exists path from T_j to T_i in serialization graph.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

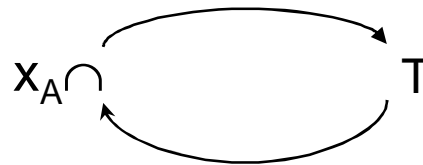
Missing Writes

Virtual Partition

Summary

Missing Writes Algorithm (3)

- Transaction T must abort if it knows about MW of a copy that it has read.



T knows about MW

T has read x_A

I.e., transaction may not have read current value; in any case, unclear which version it has seen.

- If transaction knows missing write – processing in failure mode.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

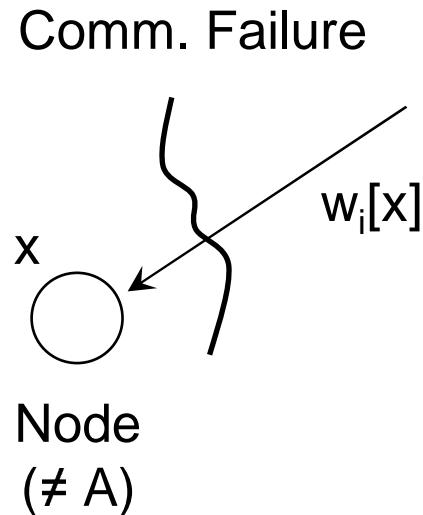
QCA

Missing Writes

Virtual Partition

Summary

Missing Writes Algorithm (2a)



- T_i has read x_A .
- Maybe the version of X on the current node is more recent.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Missing Writes Algorithm (3)

Implementation of ‚MW Awareness‘:

- If transaction knows about a MW:
annotate all data objects accessed
with known MWs.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

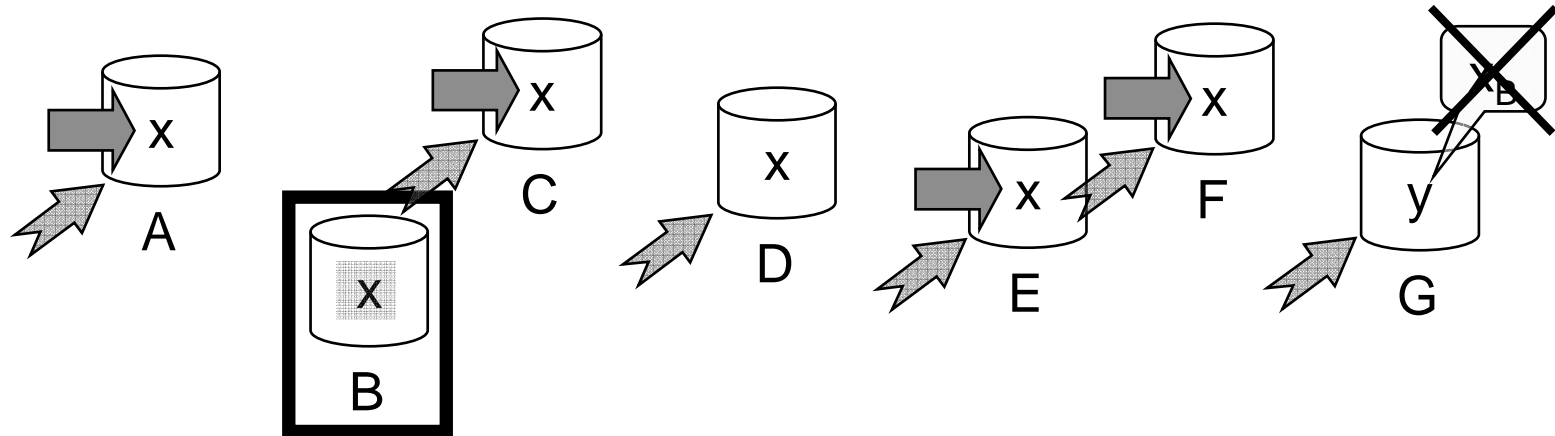
QCA

Missing
Writes

Virtual
Partition

Summary

Missing Writes Algorithm (4)



Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

- Recovery:
 - ◆ update copies (obtain a read quorum and write the copy; blue arrows),
 - ◆ remove these copies from those annotations.
 - Messages to all sites (green arrows),
 - version numbers, if new failure in the meantime.

Z

Virtual Partition Algorithm

- Read only one copy.
- Weights of sites, read- and write threshold value.
- *View $v(A)$ of a Site A :=* set of sites of which A believes that they form its component.
- $v(A)$, $\text{home}(T)$, $v(T)$
 $v(T) = v(\text{home}(T))$
- $v(T)$ must comprise read-/write quorum for reading or writing.
- T aborts if $v(T)$ changes before commit.
- Write-all within $v(T)$.
- All nodes in view must have the same view.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

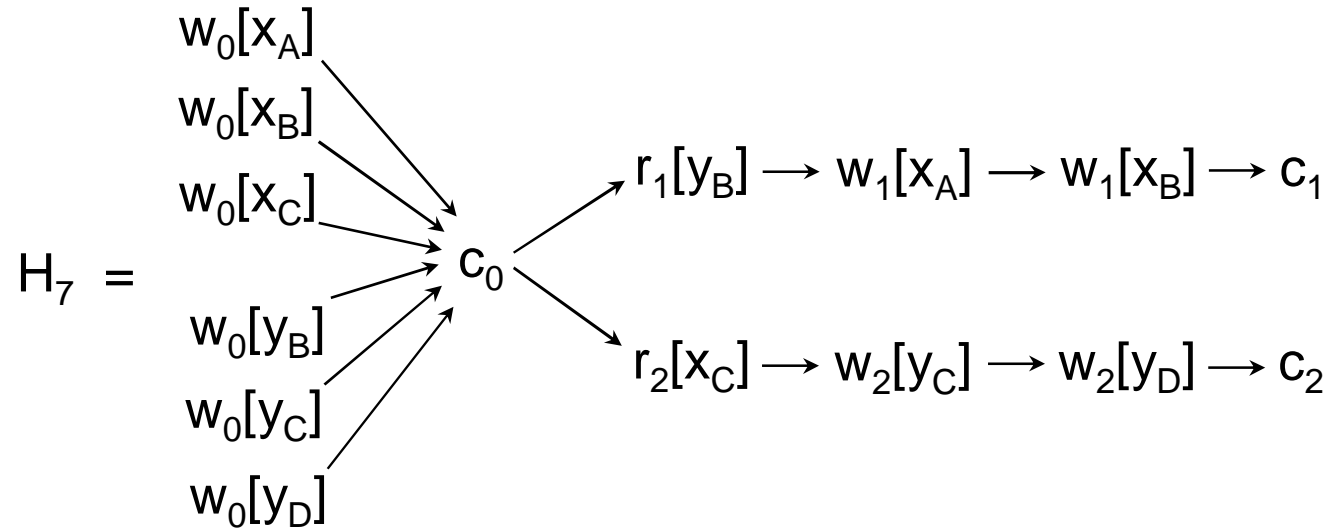
QCA

Missing Writes

Virtual Partition

Summary

Virtual Partition Algorithm – Example



- Components $\{A, B\}$ and $\{C, D\}$
- $\text{home}(T_1)=A$, $\text{home}(T_2)=C$
- T_1 and T_2 cannot have all necessary quora at the same time.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Out-of-date Views

- Two possible situations:
 - ◆ Transaction T tries in vain to address site in its view.
 - ◆ T communicates with site with different view.
- T must abort, likewise all active transactions T_i with $\text{home}(T_i) = \text{home}(T)$.
(All transactions with $\text{home}(T)$ thought that they had the necessary quorum, but this might not be the case.)

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

View Update Transaction (1)

- System starts *view update transaction*, if a view obviously is not up-to-date any more.
- Objectives:
 - ◆ update views of sites in the new view,
 - ◆ Update all copies of data objects with a read quorum in the new view.
- What happens if several sites initiate view update transactions at more or less the same time?

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

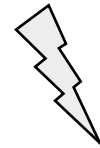
Missing Writes

Virtual Partition

Summary

View Update Transaction (2)

- Example:
 - ◆ four sites A, B, C, D,
 - ◆ initially: $v(B)=\{B\}$, $v(D)=\{D\}$, $v(A)=v(C)=\{A, C\}$,
 - ◆ Then: B learns that it can communicate with A and C; same for D at the same time.
 - ◆ If coordination is insufficient:
 $v(A)=v(B)=\{A, B, C\}$, $v(C)=v(D)=\{A, C, D\}$,
and H_7 would be allowed.



Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

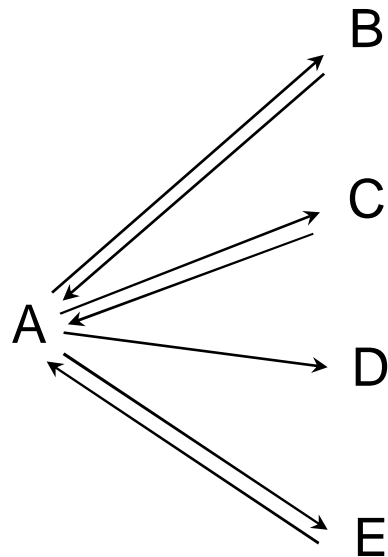
QCA

Missing Writes

Virtual Partition

Summary

View Formation Protocol (1)



JOIN-VIEW
(+ VID)

View-ID of B
< VID

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

View Formation Protocol (2)

- Each view has view identifier (VID).
- New view $v(A)$
→ generate *newVID*,
with higher value than current VID.
- Protocol itself (first part):
 - ◆ *JOIN-VIEW* message from Site A
to all sites in the new view.
Message contains *newVID*.
 - ◆ Current VID of a site $< newVID$
⇒ positive acknowledgement,
otherwise negative (or no) acknowledgement.

Introduction

Primary-
Copy

Write-All

Write-All-
Available

Available
Copies

Comm.
Failures

QCA

Missing
Writes

Virtual
Partition

Summary

View Formation Protocol (3)

- Protocol itself (continuation):
 - ◆ A can
 1. abort protocol and try again (later), or
 2. form smaller view than originally intended.

Case 2.: *VIEW-FORMED* message from A containing new view v' .
 v' will also be new view of the receiving site.
- In example (with H_7):
A and C join the same views,
depending on the VIDs.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Structure of View Identifier

- Pair (c, s) – c : counter, s : site-ID
- $(c, s) < (c', s') \iff c < c' \text{ or } (c = c' \wedge s < s')$
- Illustrations:
 - ◆ $(11, 7) < (12, 4)$ because $11 < 12$
(4, 7 – site-IDs)
 - ◆ $(11, 7) < (11, 8)$ because $7 < 8$
(7, 8 – site-IDs)
- c – counter that is incremented each time when s wants to create a new view.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Z

Conclusions

- Techniques for transactional guarantees in presence of replication.
- The more sophisticated algorithms are more ,interesting‘.
- Different settings:
 - ◆ Site failures vs. communication failures,
 - ◆ robust vs. error-prone environments,
 - ◆ read-intensive vs. write-intensive.

Introduction

Primary-Copy

Write-All

Write-All-Available

Available Copies

Comm. Failures

QCA

Missing Writes

Virtual Partition

Summary

Potential Exam Questions

- Be able to explain the various protocols presented in this chapter. Make sure that you know in which situations exactly the various protocols are beneficial, and what their advantages/disadvantages are.
- What are the advantages/disadvantages of immediate/deferred?
- In which situations might Primary Copy be useful?
- Why is Write-All Available not an acceptable solution?
- How does recovery with Quorum Consensus work?
- How can Quorum Consensus be refined in order to reduce communication costs?