CprE 450/550X
Distributed Systems and Middleware

# Consistency & Replication

Yong Guan
3216 Coover
Tel: (515) 294-8378
Email: guan@ee.iastate.edu

April 15, 2003



## Readings for Today's Lecture

- References
  - Chapter 6 of "Distributed Systems: Principles and Paradigms"

- **Announcement:**

  **Next class will be at:**

  April; 17, Thursday - 3:40 pm - 5:00 p.m. - Carver 232
  Martin Nystrom, Cisco, Raleigh, NC - Web Security

3

## Consistency Protocols

- We have studied various consistency models.
- Today, we will focus on issues of implementation of consistency models:
  - Whether or not there is a primary copy of the data to which all write operations should be forwarded.
  - When no such primary copy exists, a write operation can be initiated at any replica.
- Primary-based protocols
- Replicated-write protocols
- Cache-coherence protocols

4

## Primary-based protocols

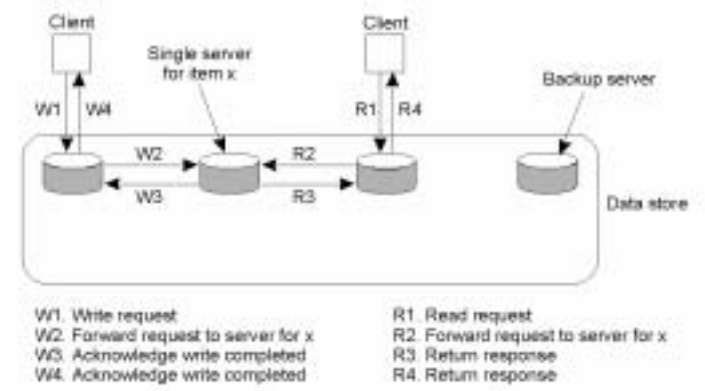
Each date item x has an associated primary for coordinating write operations on x.

Depend on whether primary is fixed or movable.

- Remote-write protocols
  - No replication
  - All read and write operations are carried out at a (remote) single server.
- Local-write protocols
  - Fully-migrating approaches: keeping track of data item
  - Primary-based approaches

5

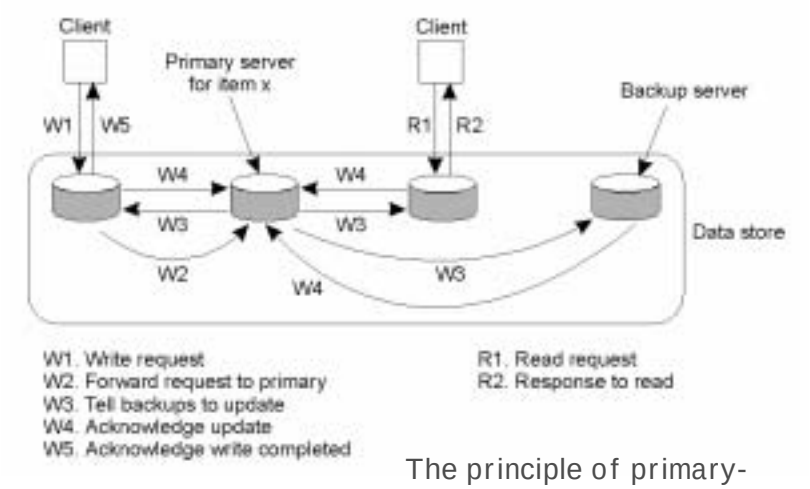
## Remote-Write Protocols (1)




Primary-based remote-write protocol with a fixed server to which all read and write operations are forwarded.

6

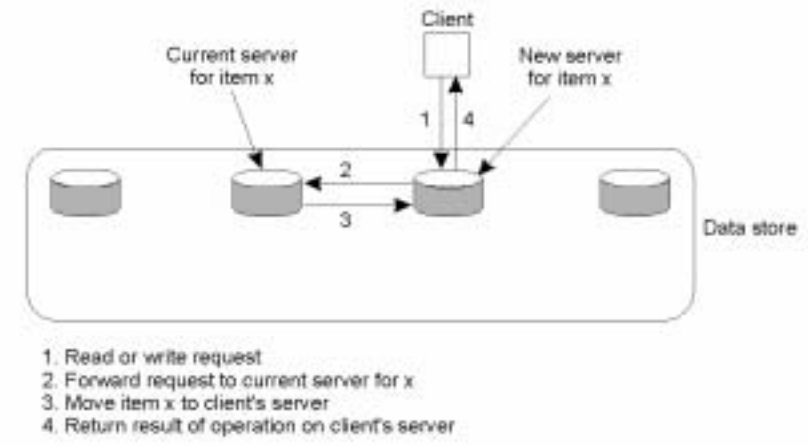
## Remote-Write Protocols (2)




The principle of primary-backup protocol.

7

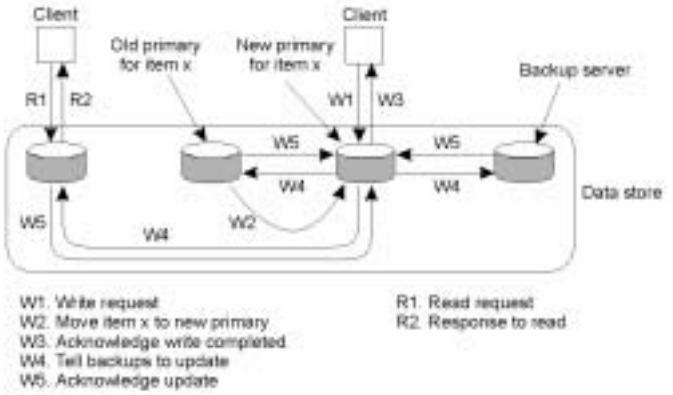
## Local-Write Protocols (1)




Primary-based local-write protocol in which a single copy is migrated between processes.

8

## Local-Write Protocols (2)




Primary-backup protocol in which the primary migrates to the process wanting to perform an update.

9

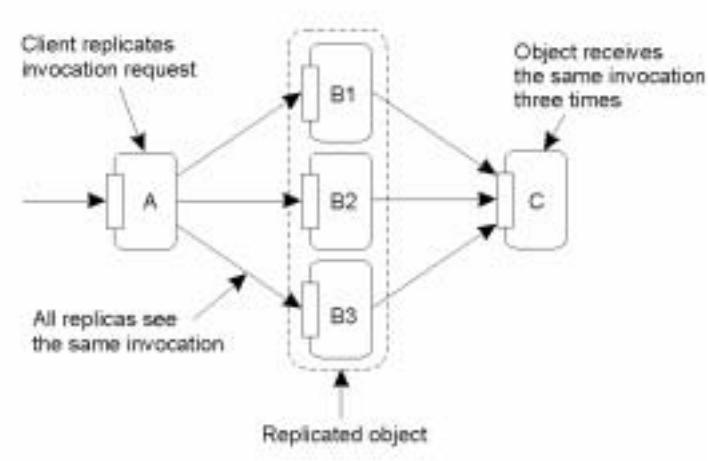
## Replicated-write protocols

Write operations can be carried out at multiple replicas instead of only one.

- Active replications
  - An operation is forwarded to all replicas
- Consistency protocols based on majority voting

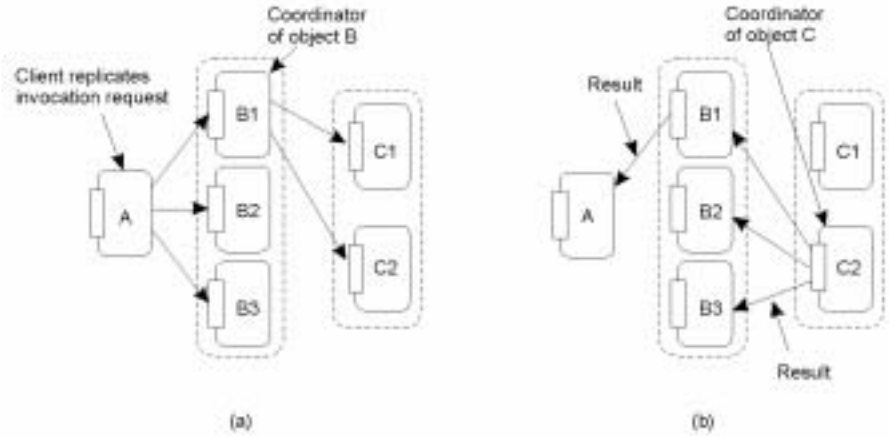
10

## Active Replication (1)




The problem of replicated invocations.

## Active Replication (2)




a) Forwarding an invocation request from a replicated object.
b) Returning a reply to a replicated object.



## Quorum-Based Protocols (1)

- The basic idea is to require clients to request and acquire the permission of multiple servers before either reading or writing a replicated data item.
- Gifford's scheme:

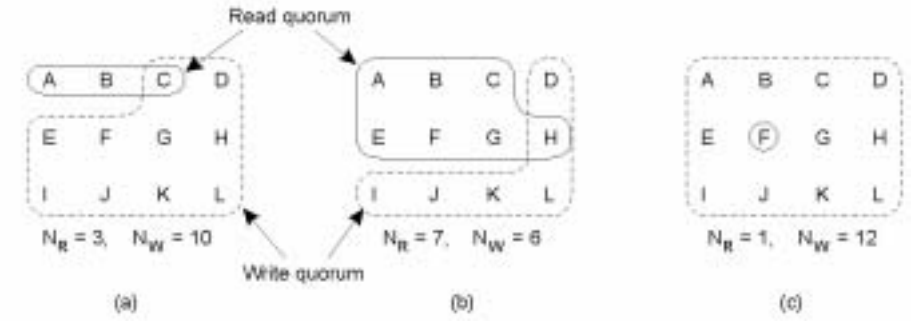
  **Nr: read quorum**

  **Nw: write quorum**

  **Two conditions:**

  1. Nr+Nw > N
  2. Nw > N/2

# Quorum-Based Protocols (2)




Three examples of the voting algorithm:

a) A correct choice of read and write set
b) A choice that may lead to write-write conflicts
c) A correct choice, known as ROWA (read one, write all)



# Cache-Coherence Protocols

- Cache: A special form of replication
- Controlled by clients, not servers
- Three approaches:
- Coherence detection strategy
- Optimistic approach
- Verify whether the cached data were up to date only when the transaction committed.

- Coherence enforcement strategy
- Write-through caches: allow clients to directly modify the cached data and forward the update to the servers.
- Write-back cache: Delay the propagation of updates by allowing multiple writes to take place before informing the servers.

15

Next, we are going to study Security.

16

# Orca

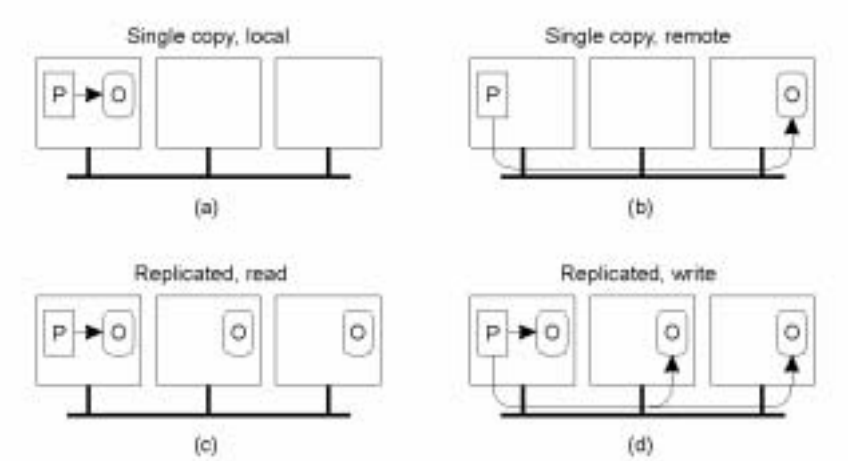
```
OBJECT IMPLEMENTATION stack;
  top: integer;                                  # variable indicating the top
  stack: ARRAY[integer 0..N-1] OF integer        # storage for the stack

  OPERATION push (item: integer)                 # function returning nothing
  BEGIN
    GUARD top < N DO
        stack [top] := item;                     # push item onto the stack
        top := top + 1;                          # increment the stack pointer
    OD;
  END;

  OPERATION  pop():integer;                      # function returning an integer
  BEGIN
    GUARD top > 0 DO                             # suspend if the stack is empty
         top := top – 1;                         # decrement the stack pointer
        RETURN stack [top];                      # return the top item
    OD;
  END;
BEGIN
  top := 0;                                      # initialization
END;
```

A simplified stack object in Orca, with internal data and two operations.

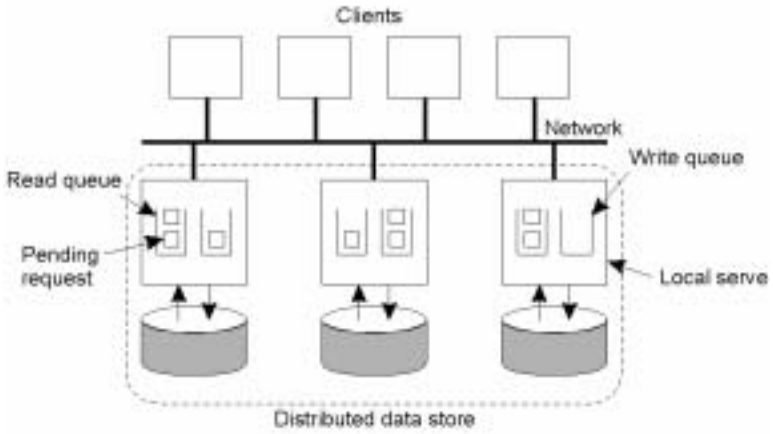
# Management of Shared Objects in Orca




Four cases of a process *P* performing an operation on an object *O* in Orca.



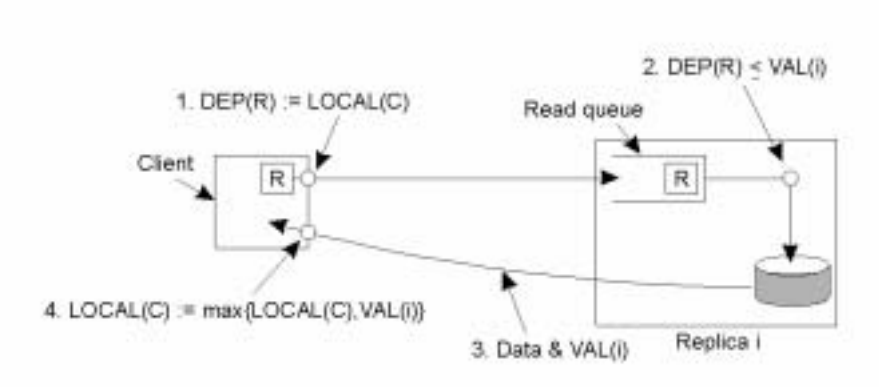
# Casually-Consistent Lazy Replication




The general organization of a distributed data store. Clients are assumed to also handle consistency-related communication.

19

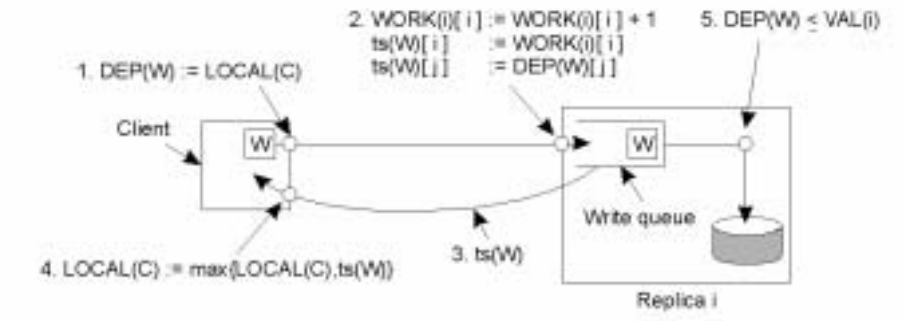
## Processing Read Operations




Performing a read operation at a local copy.

20

## Processing Write Operations




Performing a write operation at a local copy.


21