Distributed Systems

03r. Assignment 1 review

Paul Krzyzanowski

Rutgers University

Fall 2016

Question 1 (a)

Read Intel's *An Introduction to the Intel® QuickPath Interconnect*.

(a) What is a home agent responsible for?

The home agent is responsible for managing a portion of system memory.

Intel QuickPath implements a NUMA (Non-Uniform Memory Access) architecture – each processor (agent) is connected to a region of memory

Question 1 (b)

Read Intel's *An Introduction to the Intel® QuickPath Interconnect*.

(b) What is the purpose of the directory in the home snoop coherency protocol?

For a home agent to identify a caching agent that may have a copy of the requested memory.

A *directory* is a table that identifies which agent has the latest copy of a region of memory.

Question 1 (c)

Read Intel's *An Introduction to the Intel® QuickPath Interconnect*.

(c) What are the advantages and disadvantages of a Source Snoop coherency protocol over Home Snoop?

Advantage: fewer message hops (lower latency). The caching agent sends a request to the home agent **AND** sends snoop requests to all other caching agents at once so data can be delivered immediately to the requesting caching agent.

Disadvantage: The caching agent has to send a message to all the other caching agents, thus increasing overall bandwidth in the switching fabric.

Question 2 (a)

Read *The Design Philosophy of the DARPA Internet Protocols*
and *End to End Principle in Internet Architecture as a Core Internet Value*.

(a) The end-to-end principle is a core design principle of the Internet. What is the end-to-end principle?

"whenever possible, communications protocol operations should be defined to occur at the end-points of a communications system, or as close as possible to the resource being controlled."

This recommends that the network itself be “dumb”. For example, routers shouldn’t be responsible for reliability or message ordering.

Question 2 (b)

Read *The Design Philosophy of the DARPA Internet Protocols*
and *End to End Principle in Internet Architecture as a Core Internet Value*.

(b) "Fate-sharing" is an example of the Internet's end-to-end principle. What is meant by the term fate-sharing and how does it address fault tolerance?

Fate sharing: "It is acceptable to lose the state information associated with an entity if, at the same time, the entity itself is lost."

Example: A TCP driver in the operating system keeps track of message ordering via sequence numbers on packets. It's ok if this information is lost if the machine crashes because all processes that care about this die as well.

Compare this with having routers keep track of message ordering. If state is lost in the router, the computers will not get sequenced data. Also, if the computers die, the routers are storing state that will never be used.

Question 2 (c)

Read *The Design Philosophy of the DARPA Internet Protocols*
and *End to End Principle in Internet Architecture as a Core Internet Value*.

(c) What was observed to be the greatest source of jitter (the variation in delay) on IP-based communication networks?

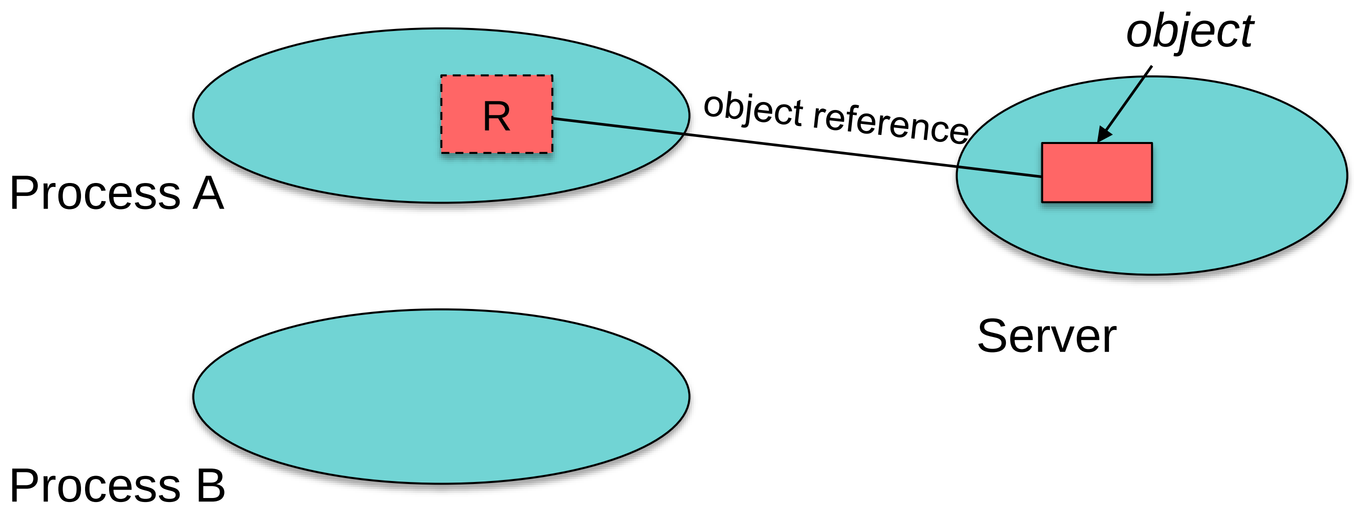
The mechanism to provide reliable delivery is the greatest source of jitter.

Consider the extra delay of timing out and realizing that you have not received a packet and requesting a retransmission. This delay is usually much greater than any delays in queuing packets at routers or transmission delays.

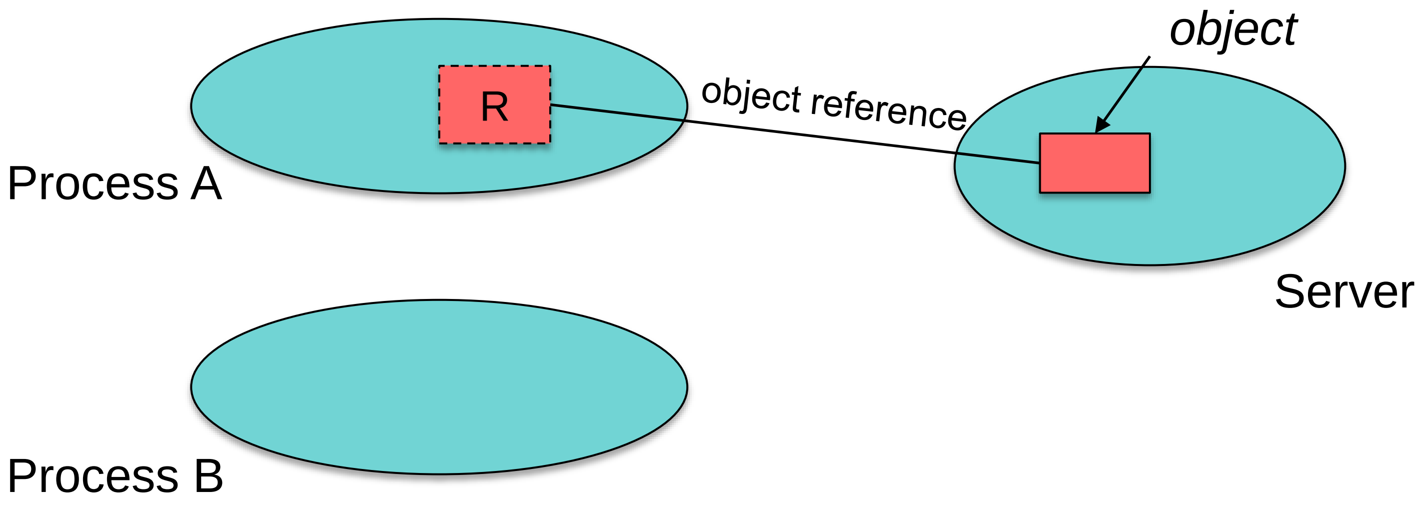
Question 3

Read *Read Distributed Garbage Collection for Network Objects*

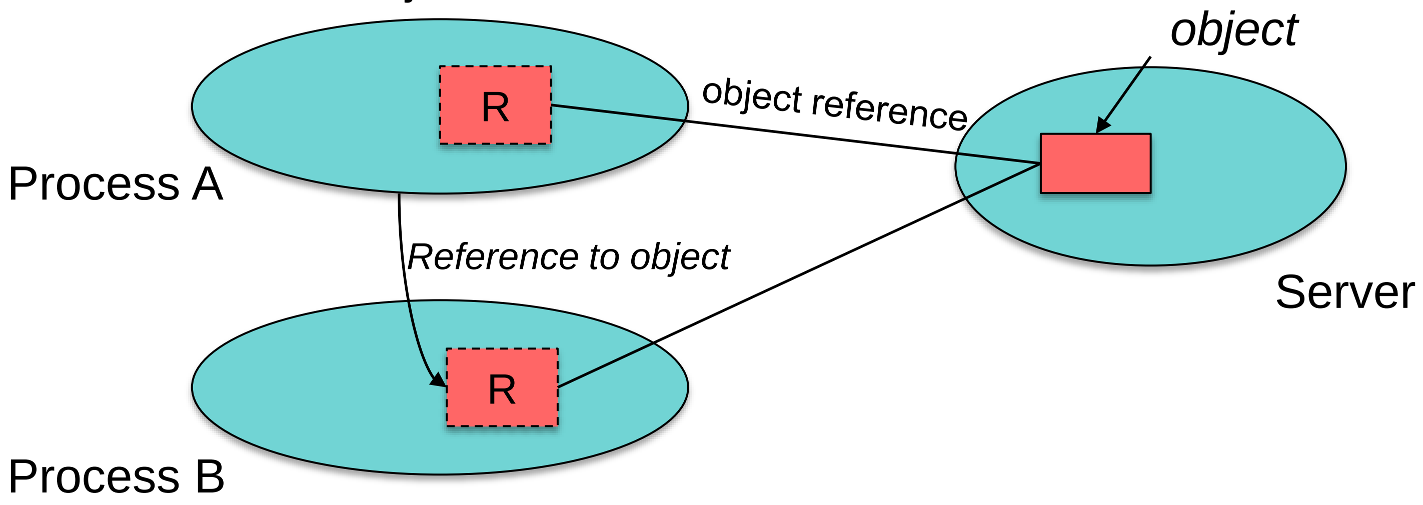
A proposal is introduced for managing remote object references by having a server maintain a dirty set per object: a list of active remote references to a particular object. When a local garbage collector at a client determines that the client has no more references to a remote object, it sends a clean message to the server to remote the reference from the dirty set. One problem that arises is the situation where one process, A, passes an object reference to another process, B. It is possible that the garbage collector on A will send a *clean* message to the server before B's *dirty* message is received. This will cause the object to be destroyed even though B still needs it. Explain how this situation is handled. Assume neither process A nor process B is the owner of the object. A simply passes the object reference to B.



Question 3

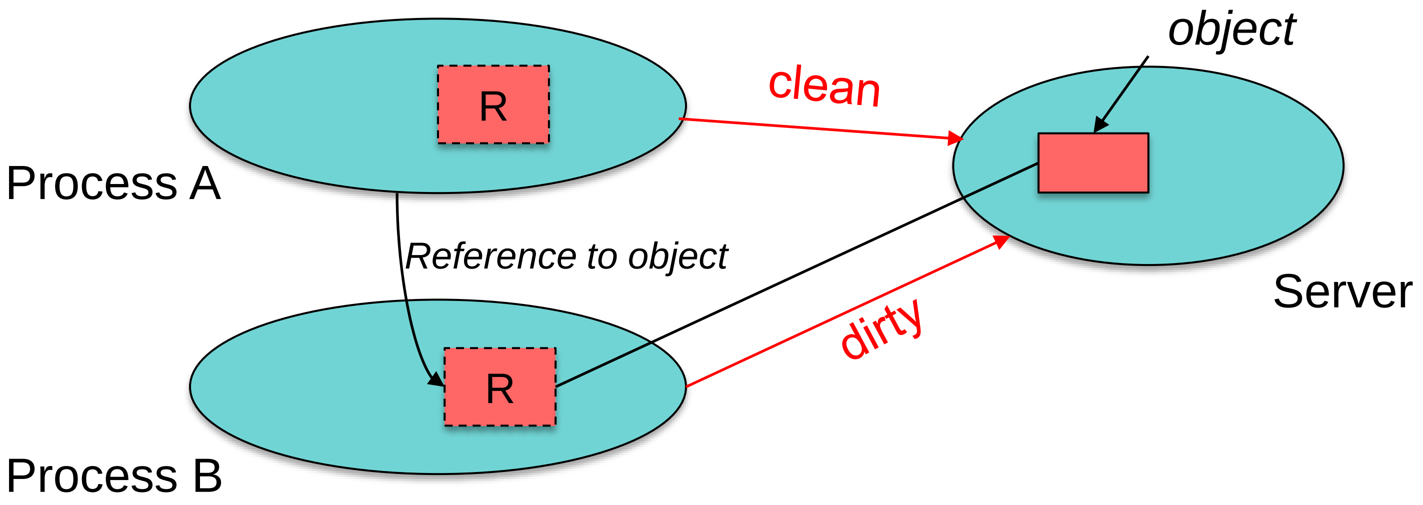


Process A passes a reference to a remote object to Process B and deletes its reference to the object.



Question 3

Since *A* no longer needs the object reference, it sends a *clean* message to the server. At around the same time, *B* sends a *dirty* message to tell the server that it is using the object.



The problem is that if the server receives the *clean* message first and has no more references for the object, it may delete it before it receives the *dirty* message from process B.

Question 3

Solution:
Process A needs to know that it passed an object reference to process B.

It will then wait for an acknowledgement from B that indicates that the dirty call has been made.

The garbage collector on A will not send a *clean* message to the server until this acknowledgment is received.

Process A

1. Send object reference to B
2. Wait for acknowledgement from B
3. Send *clean* message to server

Process B

1. Receive object reference from A
2. Send a *dirty* message to server
3. Send *acknowledgement* message to A

The End