Distributed Computing

CMSC 433, March 27th

Is distributing computing different?
• What kinds of distributed computing environments exist?
• Ways in which distributed computing is different
  – Addressing objects
  – Latency
  – Partial failure
  – Concurrency

Distributed computing environments
• Usually refers to multiple CPU’s
• Shared address space or message passing?
• On same chip, in same room, or across the internet?
  – Latency, failure modes

Existing environments
• Seti @ Home
• Server for search engine
• My laptop, PDA, cell phone, MP3 player and digital camera

Types of failure
• Machine sleeps
  – wakes up, recovers state
• Machine crash or failure
  – machine may reboot and rejoin
• Network partition
  – network may heal

Uniform view of distributed objects
• Some objects are remote, some are local
  – Doesn’t really matter to user of object
  – Objects might transparently migrate
• Design doesn’t have to take object distribution into account
• Failure and performance issues don’t belong in the design
• The interface doesn’t change if an object is remote
Uniform view

- not appropriate for
  - wide area networks,
  - consumer electronics,
  - portable devices
- appropriate for some local area networks
  - but robust distributed applications plan for failure
  - even if local

Memory access

- Can we make the fact that an object is remote transparent?
- Perhaps for objects
  - What about int’s?
  - What about char *’s?
- If you can’t directly access fields and create pointers to them,
  - not transparent

Partial failure

- Computers fail
- OS’s crash
- Networks fail
- PDA’s get turned off or taken out of the room
- Often no warning

Queue example

- Want to add x to remote queue q
  - q.enqueue(x)
- Operation could fail
- Want to reliably enqueue x

Queue example

- while (true) {
  try {
    q.enqueue(x);
    break;
  }
  catch (RemoteException e) {} 
}

Partial failure

- Object was enqueued, but failure occurred during return message
- Could enqueue x multiple times
- How to fix?
  - Need a request tag so that duplication enqueue requests can be detected
Concurrency

- Distributed computations mandates concurrency

Latency

- Remote calls are much more expensive than local calls

Latency

- Making a call to an object on a remote machine is expensive