CSE5306, Distributed Systems
Fall 2017, Homework 1
Due date: Oct. 17, in class, hard-copy submission ONLY

Please read this:

Homework assignments are to be completed individually, no teamwork.
Total points possible: 100 pts.
Please add the following statement in the beginning of your submission.
I have neither given or received unauthorized assistance on this work
Signed: Date:

1. What are the differences between distributed and parallel systems? Why the boundary between the two is blurring?

Distributed (D) and parallel (P) systems are different in that

- Design objectives: scalability, fault tolerance and resource sharing capability (D) vs concurrent performance (P)
- Data distribution: entire file on a single node (D) vs striping over multi nodes (P)
- Symmetry: machines acting as server and client (D) vs service separated from clients (P)
- Fault-tolerance: designed for fault-tolerance (D) vs relying on enterprise storage (P)
- Workload: loosely coupled, distributed apps (D) vs coordinated HPC apps (P)
- Architecture: distributed system components (D) vs simultaneous operations (P)
- Dependency: loosely or tightly coupled (D) vs tightly coupled, coordinated (P)
- System size: multiple computers (D) vs only a single computer (P)
- Operations: performed by different computers (D) vs by multiple processors (P)
- Synchronization: no central clock, synchronization required (D) vs master clock (P)
- Resource sharing: distributed memory (D) vs shared or distributed memory (P)
- Communication: message passing between computers (D) vs bus between processors (P)

The boundary is blurring because many parallel and distributed systems borrow idea from each other, narrowing the differences between them.

2. Discuss the three techniques for scaling distributed systems and compare their advantages and disadvantages.

a. *Hide and reduce latency*:
This technique tries to mitigate the waiting time between a remote service request and its response with two main features:
i. using asynchronous communication.
ii. moving part of the computation to the client if applications can't use asynchronous communications efficiently.
b. *Distribution*:
Distribution aims to avoid a single server having to deal with all requests by splitting a component into smaller parts, and subsequently spreading those parts across the system.
c. *Replication*:
Replication not only increases availability, but also helps to balances the load between components for better performance. However, there is one serious

drawback that may adversely affect scalability. Since we now have multiple copies of a resource, modifying one copy makes that copy different from the others. Consequently, replication leads to consistency problems.

3. Discuss the differences among transient and persistent communications, asynchronous and synchronous communications.

Transient vs persistent communications:

a. Transient communication:
In a transient communication, a message is stored by the communication system only if the sending and receiving applications are executing, so the middleware cannot deliver a message due to a transmission interrupt, or if the sending and receiving application are currently not active, the message will simply be discarded.

b. Persistent communication:
In a persistent communication, message that has been submitted for transmission is stored by the communication middleware as long as it takes to deliver to the receiver, so the middleware will store the message at one or several of the storage facilities. As a consequence, it is not necessary for the sending application to continue execution after submitting the message. Likewise, the receiving application need not be executing when the message is submitted.

Asynchronous vs synchronous communications:
a. Asynchronous communication:
In an asynchronous communication, the sender resumes immediately after sending a message, which means that the message is temporarily stored by the middleware.

b. Synchronous communication:
In a synchronous communication, the sender is blocked after sending a message until its request is known to be accepted.

4. Discuss the differences between threads and processes, and the benefits of multi-threading.

Threads:
a. generally maintains only the minimum information to allow CPU to be shared by several threads,
b. Separates concurrency from protection,
c. Maintains sequential execution stream of instructions, and
d. Shares address space with other threads.

Processes:
a. can currently share the same CPU and other hardware resources with other processes, and
b. when created, requires the operating system to create a complete independent address space at a high cost.

Benefits of multi-threading include:
a. blocking system call does not stop a process,
b. parallelism is exploited in a multiprocessor system,

c. it is useful in cooperating programs: different parts of an application need to talk to each other (pipes, message queues, and shared memory segments), and
d. it is easier to develop a program using a collection of threads.

5. Discuss the challenges to achieve passing reference parameters in remote procedure calls.

a. It's difficult to pass pointers or references,
b. forbidding pointers and references parameters is highly undesirable and not necessary,
c. changing the server makes using the pointer directly affect the message buffer inside the client stub,
d. call-by-copy/store is not always identical,
e. the stubs need to know whether the buffer is an input parameter or an output parameter to the server to eliminate one of the copies, and
f. we still cannot handle the most general case of a pointer to an arbitrary data structure such as a complex graph.

6. Discuss the advantages and disadvantages of RPCs and message-oriented communication, respectively.

RPCs:
Advantages: RPC hide communication in distributed system to enhance access transparency.

Disadvantages: Unfortunately, neither mechanism is appropriate at all times. In when it cannot be assumed that the receiving side is executing at the time a request is issued, alternative communication services are needed. Likewise, the inherent synchronous nature of RPCs, by which a client is blocked until its request has been processed, sometimes needs to be replaced by something else.

Message-oriented communication:
Advantages: Transport layers offers a simple message-oriented communication model as well as many applications built on top of such a model. A typical example is socket.

Disadvantages: Sockets were deemed insufficient for two reasons. First, they were at the wrong level of abstraction by supporting only simple send and receive primitives. Second, sockets had been designed to communicate across networks using general-purpose protocol stacks such as TCP/IP.

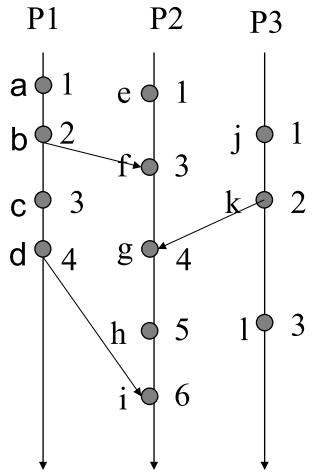
7. Discuss how message-queuing model can help simplify the design of distributed systems.

a. Messages are inserted in specific queues.
b. These messages are forwarded over a series of communication servers.
c. The sender is generally given only guarantees that its message will eventually be inserted in the queue, without guaranteeing when, if ever, the message will actually be read.

8. Briefly discuss how to causally order events on distributed machines using logical clocks.

It is sufficient that all nodes agree on a current time, which does not have to be the real time. If two processes do not interact, it is not necessary that their clocks be synchronized. Therefore, it is adequate that all processes agree on the order in which events occurs.

9. Identify all the causally related events and some concurrent events in the following figure:



Causally related events: all events on P1, P2, and P3, and events on the path of message passing.
Concurrent events: (a, e, j), (c, f, j), (d, h, l)

10. When using Lamport's logical clocks to derive a total order of events in distributed systems, why does each message need to be acknowledged?

Since Lamport's logical clocks are unable to order concurrent events, the totally ordered multicasting algorithm uses a predefined order for concurrent events. Acknowledgements are needed to ensure such a predefined order.