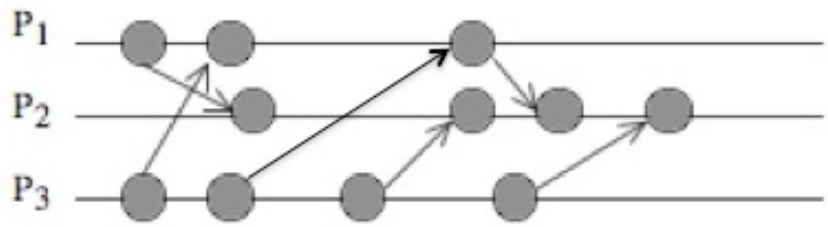
Some Exercises – Distributed Systems and Cloud Computing

Topics:

Introduction to Distributed Systems and Cloud Computing
IPC
Mechanisms fro RMI, RPC, indirect communications
Synchronization and coordination in distributed systems
Shared data, limited to transaction concurrency and distributed transactions

- Consider the **two phase commit protocol**, give an algorithmic description and discuss its design objectives, properties, and its computational complexity. Describe the advantages and limitations of the protocol.
- Discuss the design issues of a **notification service with a mailbox** for subscriber requests that can specify when to receive notification. How to design a system that can manage faults such as subscriber's crash and subscriber temporary inactivity.
- Discuss the **transparency** property in distributed systems and in cloud computing. (a) Which **type** of transparency can be designed, (b) discuss whether an it is relevant to have the **maximum** transparency (c) give two examples of possible **drawbacks** of high transparency.
- Give examples of application of the **mutual exclusion problem** in distributed systems and cloud computing. Describe an algorithm to solve the mutual exclusion problem, its algorithmic description, its assumptions, advantages and drawbacks with respect to other possible choices.
- Define the various types of g**roup communication** in distributed systems and discuss the main design issues.
- Consider a distributed computation shown in the following spatial-temporal diagram.



- Assign to each event a value for **the Logical Clock and for the Vector Clock**. Provide two examples of concurrent events and two of not concurrent event. Explain whether the Logical Clock always allows to identify concurrent events and why. Explain the difference between Logical clock and vector clock.
- Give an example of application of an **election** algorithm in distributed systems. Describe the main goals, the advantages and drawbacks, the number of messages and time to solve the problem. Discuss whether it is more appropriate to use an election algorithm to solve a mutual exclusion problem.
- Discuss the problem of **clock synchronization** in asynchronous distributed systems. Describe and compare the main methods, their advantage and limits.
- What is and to what aim is defined a **consistent global state** in a distributed system? Give an example and explain how it can be used for distributed computation.
- A) Show two examples interleaving of two transactions that are in one case **serially equivalent** and in the case they are not. Discuss how to guarantee or check serial equivalence. B) Then explain in details whether serial equivalent transactions can incur also in **dirty reads** and **premature writes.**
- Give the main **goals** of a distributed system design, and provide an application example for each of them. Discuss whether the design of special distributed systems has different goals: consider
  mobile distributed system
  wireless distributed system
  real-time distributed systems
- Specify how the selection of a **request reply** protocol affects the possible **semantics** of inter-process communication in distributed systems. Give an example of selection of a protocol and its impact on the semantics of RMI and RPC.
- What is the difference between atomic, reliable and ordered **multicast**? What are the main design choices for an ordered multicast? Define the various types and give two examples.
- Describe the ring algorithm for **mutual exclusion** in distributed system. Discuss its advantages and drawbacks with respect to other approaches. Specify under which conditions it should be effectively applied to manage mutual exclusion.
- For customers applications at the office or at home discuss what are the main problems that can arise. Consider the design requirements and focus on transparency, availability, security and performance.
- What are a ***logical clock*** and the *logical time*? Define the various possible ordering relations and the logical clock defined by Lamport. Give an example, for each ordering type, of events that satisfy the ordering relations and events that do not satisfy it. Discuss the motivation of logical clock and their application in distributed system design.
- What are the ***two-phase locking*** and the ***two-phase commit*** algorithms? Describe the algorithms and explain

the motivation, the application and give an application example of each of them.

- Describe the mechanism of ***Remote Procedure Call*** in distributed systems; discuss the software architecture design and the main problem in its implementation. Discuss the type of parameter passing. Give an example.
- Discuss and describe the basic design principle of the **Ricart-Agrawala** algorithm for mutual exclusion in distributed systems. Give an algorithmic description. Discuss the computational complexity and the advantages and limits with respect to other possible solutions.
- What is **inter process communication** and what are the main solution?
- What is the **publish-subscribe** paradigm? Discuss the types of objects that can be defined and what are the various types of observers that can be used and their meaning.
- How do you define **serial equivalence** of transactions in distributed systems?
  Describe the main problems in the design of concurrency control and how they can be solved.
- What is the difference between concurrent transactions and concurrent **distributed transactions**? Discuss whether the solutions proposed above (a) can be applied also for distributed transactions and give examples.
- Describe the algorithm by **Maekawa** for mutual exclusion in distributed systems. Give an algorithmic description. Discuss the computational complexity and the advantages and limits with respect to other possible solutions.
- Consider the **design of an operating system** for distributed systems. Describe the **main types** of operating system and their software architectural components in distributed systems. Discuss the relative merit of the various alternative designs, their properties. Discuss the issues related to **fault tolerant** design.

_____

Topics:

Fundamental of Cloud Computing
Cloud service and application
Distributed Operating Systems
Replication and Consistency
Distributed file systems

- What are the **main characteristics of cloud computing**? Describe what is service and the main service models of cloud computing. Define the **delivery and the deployment model**.
- In the design of a **distributed file syste**ms describe the main components, their functionalities and the mail software architecture, properties and design characteristics. What are the main differences with respect to centralized file systems? There are special problems that derive from distribution? Explain with details and provide an example.
- Discuss the design of a **load-balancing scheme** for a set of processes in a distributed system. Assume that the processes cannot be migrated. Discuss and define: (a) the goal and the requirements to be satisfied, (b) whether the scheme is designed for some specific types of system or applications , (c) how to measure system load and with what accuracy, (d) how to monitor the load, choose and assign the location for a new process.
- Describe the types of **scaling** that can be adopted **in cloud computing models** and discuss their comparison. Give two examples of application of scaling.
- What is a **notification** service? Describe the design of notification services in distributed systems and cloud computing. Provide two examples. Then discuss manage **faults management** and specifically how to deal with a subscriber's crash.
- What is the difference between **delivery model and deployment model** in cloud computing? Give two examples.
- Consider the design of **mobile distributed systems.** Compare the main paradigms for code mobility in distributed systems and how they affect the main design goals.
- What are the main **characteristics of cloud computing**? Describe what is service and the main service models of cloud computing.
- Describe the project of **scheduling and migration** in a distributed system. Discuss the main difference in the scheduler definition and design with respect to centralized systems. Give two examples.
- Consider ***peer-to-peer*** systems. Describe their main characteristics and compare with client-server architecture.
- Consider communication in distributed systems: consider a **web application for data sharing** (with various kinds of files) that provide service to a many customers. Describe a design of the system architecture that satisfies requirements of high reliability, scalability and performance. Assume that a high number of files can be loaded and that can be read from many devices. Discuss the characteristics of the proposed solution and its advantages and limitation.
- Describe three main **models for replication** management in distributed systems and their relative advantages and drawbacks.
- Describe the **worker-pool multi thread**, thread-per request and thread per object architectures. Explain what are the meaning and goal and the conditions under which each of them can be appropriately applied.
