Exam in Distributed Systems

Justin Pearson

March 9th, 2009

Cover Sheet

This sheet should be handed in together with the exam.

Each problem must be solved on a separate sheet. Write your name on each sheet. Indicate below which questions you have answered.

Problem no.	Solution provided	Max	Your points
1		4	
2		10	
3		17	
4		14	
5		10	
6		4	
7		11	
	Total:	55,59,70	

Name : .....

Pers.no. : .....

Exam Rubric

A mark of 50% is required for a pass, marks for 4 and 5 are distributed evenly. Answers can be in English or Swedish.

Each full answer should be started on a separate sheet. Please write you name and personal number on each sheet.

I will not be able to come to the exam. If you are unsure about how to answer a question. Make some reasonable assumptions, state your assumptions and answer the exam.

Hjälpmedel:

Pen, pencil, ruler, rubber (eraser for people from the Antipodes or from across the pond) calculator, and a dictionary.

1. General Questions on Distributed Systems

   (a) Explain, with an example, the client-server paradigm. (**2 points)**.

   (b) When using a client-server system there can be errors due to network faults. Suppose that a confirmation is lost during a client-server interaction. With an examples, explain some strategies for dealing with errors. Explain how the strategies depend on the nature of transaction. (**2 points)**.

2. Remote Procedure Calls (RPCs). When implementing remote procedure calls or remote object calls you have to solve two related problems: first data has to be serialized so that it can be sent over a network connection and second different computer architectures have to be connected.

   (a) Explain the process of data-marshaling and the role it plays in executing a RPC. (**2 points)**.

   (b) Explain at least *two* different strategies for translating data between different computer architectures in a RPC system. (**4 points)**

   (c) Pointers are hard to implement in a RPC system. Explain why and givetwo strategies for implementing them. (**4 points)**.

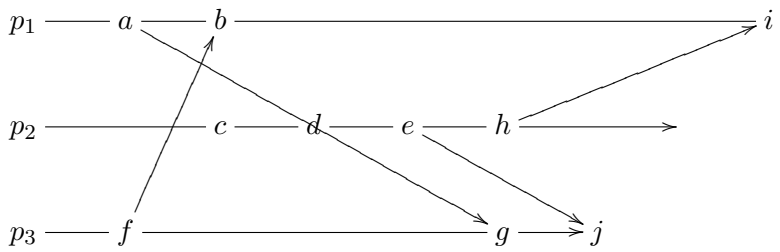
3. Clock Synchronization and Timestamps

   (a) What is a leap second and why is it used? (**1 points)**

   (b) Explain the Cristian's method for synchronizing clocks. Further explain the assumptions that are required for the algorithm to work.(**4 points)**

   (c) Suppose that you have two clocks both with drift $\rho = 10^{-8} seconds/seconds$. Suppose that you want the clocks synchronized within 0.01 of a second. How often should the clocks be resynchronized to achieve this? (**2 points)** (Obs. Show your workings, no workings no points for a correct or incorrect answer)

   (d) Consider the following three processes p_1,p_2 and p_3 with the following pattern of communication:



   Label the each event with a normal Lamport timestamp (**2 points)** and a vector timestamp (**2 points)**

(e) What does it mean for two events to be concurrent and what is the relation of the Lamport timestamps of the two events. (**2 points)**.

(f) Is it possible for two events to have the same Lamport (non vector) timestamp? If it possible give an example, if it is not possible argue why it not possible. (**2 points)**.

(g) Why are vector timestamps required? (**2 points)**

4. Cuts

   (a) Define a consistent cut in a distributed system (you may use the notation used in the slides or the notation used in the book). If you are not able to define it formally you will still get some points if you define it informally. (**4 points)**

   (b) Motivate the above definition: that is, explain why it is useful. (**2 points)**

   (c) What assumptions are required on the channels in a system for the Chandy-Lamport snapshot algorithm to work. (**2 points)**

   (d) Explain with examples the Chandy-Lamport snapshot algorithm. (**6 points)**

5. The CAP Theorem.

   (a) State the CAP theorem. (**2 points)**

   (b) You are two build a distributed system for an online music and video service. Premium customers have access to the service advert-free. As part of the licensing agreement with the content providers you are to provide details to the content owner and how often a particular track is played.

   Your system will consist of many components including: a system to handle payments; a set of servers to deliver content to users and logging servers so as to send data back to the content provider. You aim to make the system with as little downtime as possible for the customers.

      i. Sketch an outline of a design for the system. Describe how the system is going to be partitioned and how your design will support fault tolerance. (**4 points)**.

      ii. The CAP theorem has implications for system design, in particular it forces certain design compromises. Give at least two examples in your above design[1] where you have to be aware of the CAP theorem. (**4 points)**.

[1] If you can't think of at least two examples in your design, then your design is not complicated enough.

Stop here if you are only doing the 4.5 point version of the course

6. Peer to Peer

   (a) Explain the flooding algorithm used in some versions of Gnutella. Explain its advantages and its disadvantages. (**2 points)**

   (b) One of the aims of using a peer-to-peer system is to achieve some sort of load balancing, explain how the Bit-Torrent protocol achieves this. (**2 points)**.

Only answer the next question of you are registered on an old version of the course with more than 5 points.

7. Transactions.

   (a) With examples define what a transaction is. (**2 points)**

   (b) Explain what problems can happen there is no concurrency control where multiple transactions are being executed at the same time. (**3 points)** (Obs. **Give examples**).

   (c) Define what it means for an interleaving of two transactions to be serially equivalent? (**2 points)**

   (d) What is the two-phase commit algorithm? Explain it in some detail. (**3 points)** (With **examples!!!**).

   (e) Why is the two-phase commit algorithm used in distributed transactions? (**1 points)**