

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		10	
2		6	
3		15	
4		12	
5		8	
6		3	
7		4	
8		6	
9		6	
Total:		58/70	

Name :

Pers.no. :

Exam Rubric

A mark of 50% is required for a pass, a 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 your name and personal number on each sheet.

Students on the 4.5 (1TT835) point version of the course only do questions 1-7 all other students do all the questions.

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.

1. General Questions on Distributed Systems

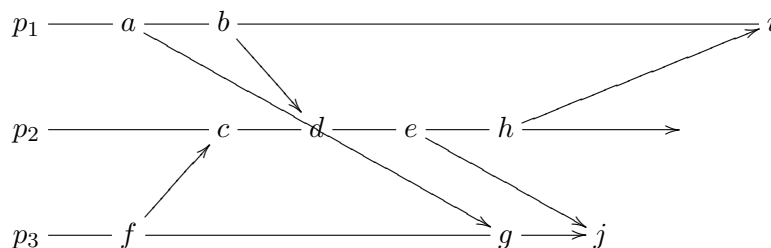
- (a) Explain the differences between a distributed system and a multi-processor system. **(2 points)**
- (b) Explain the differences between shared memory programming and message passing systems. **(2 points)**
- (c) Explain the difference between a blocking send/receive communication primitive and a non-blocking send/receive communication primitive. **(2 points)**
- (d) Explain why implementing a distributed system over a standard Ethernet network causes problems for the algorithm designer. **(2 points)**
- (e) In a client-server situation, the server is a point of failure. Explain how you would make the system more fault-tolerant in a transparent way to the client. **(2 points)**.

2. Remote Procedure Calls (RPCs).

- (a) Explain how RPCs are implemented. **(4 points)**
- (b) RPCs attempt to give the programmer the illusion that remote procedures are running locally. Is this really true? **(1 point)**
- (c) When would it be appropriate to move a remote procedure call to a local procedure call? **(1 point)**.

3. Clock Synchronization and Timestamps

- (a) What is a leap second and why is it used? **(1 point)**
- (b) Explain the Berkeley method for synchronizing clocks. **(2 points)**
- (c) Suppose that you have two clocks both with drift $\rho = 10^{-6}$ seconds/seconds. Suppose that you want the clocks synchronized within 0.25 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. Distributed Algorithms

- (a) Suppose that you wish to get an acknowledgment from all nodes in a network. Explain why using a spanning tree is more efficient. Estimate the message complexity of your algorithm using a spanning tree. (4 points)
- (b) Explain how you would convert a synchronous algorithm in an asynchronous algorithm. For full credit explain a number of different schemes and how they differ in efficiency. (4 points)

6. Distributed Shared Memory.

- (a) What is the rôle of memory consistency models in distributed shared memory? (2 points)
- (b) Define Sequential and Causal Consistency. Give an examples that shows the difference between the two. (3 points)

7. The CAP Theorem.

- (a) State the CAP theorem. (2 points)
- (b) Explain with examples how the CAP theorem influences the design of a distributed system. (2 points)

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

8. Peer to Peer
 - (a) Explain the flooding algorithm used in some versions of Gnutella. Explain its advantages and its disadvantages. **(2 points)**
 - (b) Explain how Pastry (or Chord) routing works and how it can be used to implement a distributed hash table. **(4 points)**
9. You are employed by the FRA and your job is to catch English undesirables living in Sweden. You are going to filter all emails that contain phrases such as: "Why don't they sell wine at ICA?", "English bacon is better than Swedish bacon" and "Why is the tax so high?".
Give a high-level sketch of how this could be implemented using a Map/Reduce system, assuming that you have a database of all emails sent inside Sweden since Jan. 1. 2009.
Explain how the Map Reduce framework supports fault tolerance. **(6 points)**