TDDD93/TEN2 – Large-scale distributed systems and networks
Final Examination: 8:00-12:00, Friday, June 5, 2015
Time: 240 minutes
Total Marks: 40
Grade Requirements: Three (20/40); four (28/40); and five (36/40).
Assistance: None (closed book, closed notes, and no electronics)
Instructor: Niklas Carlsson

Instructions:

- Read all instructions carefully (including these)!!!! Some questions have multiple tasks/parts. Please make sure to address *all* of these.
- The total possible marks granted for each question are given in parentheses. The entire test will be graded out of 40. This gives you 10 marks per hour, or six minutes per mark, plan your time accordingly.
- This examination consists of a total of 13+1=14 questions. Check to ensure that this exam is complete.
- When applicable, please explain how you derived your answers. Your final answers should be clearly stated.
- Write answers legibly; no marks will be given for answers that cannot be read easily.
- Where a discourse or discussion is called for, be concise and precise.
- If necessary, state any assumptions you made in answering a question. However, remember to read the instructions for each question carefully and answer the questions as precisely as possible. Solving the *wrong* question may result in deductions! It is better to solve the *right* question incorrectly, than the *wrong* question correctly.
- Please write your AID number, exam code, page numbers (even if the questions indicate numbers as well), etc. at the top/header of each page. (This ensures that marks always can be accredited to the correct individual, while ensuring that the exam is anonymous.)
- Please answer in English to largest possible extent, and try to use Swedish or ”Swenglish” only as needed to support your answers.
- If needed, feel free to bring a dictionary from an official publisher. Hardcopy, not electronic!! Also, your dictionary is not allowed to contain any notes; only the printed text by the publisher.
- Good luck with the exam.

Part A: Distributed Systems

Question 1 (4 points)

Transparency plays a central role in some distributed systems. Consider a simple multi-tier system with three levels: a user interface, an application server, and two replicated database servers. Assume these layers are implemented as a distributed cloud service at different geographic locations and that the average round trip time (RTT) between the machines used in the consecutive layers (starting with the top-tier layer) is 20ms and 10ms, respectively. Consider a workload (set of calls) with two different types of "jobs" (call types). The first type results in fully synchronized calls in which the application server requires 70ms total processing and the database requires 100ms processing to satisfy the request. The second type does not require any database access, is fully synchronized and requires 50ms processing at the application server.

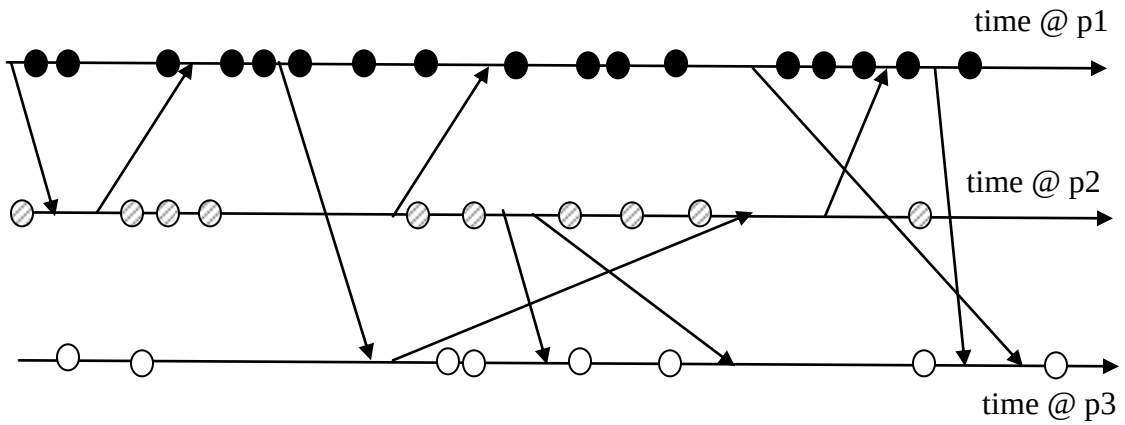
(a) For each of the two types of jobs, how much time is the client process locked from the moment it makes the request to the application server? You can assume that no large data is transferred between the layers such that the call and responses fits within a single package, and that messages do not need to be acknowledged. Please explain your answer and illustrate with a figure.

(b) Assuming 50% of the clients make each type of requests, what is the average response time (assuming no competing jobs or other reasons for queuing).

(c) Please give concrete examples of two types of transparency that are provided in the above example. Remember to explain your answers.

Question 2 (4 points)

Assume that you have three processes p1, p2, and p3 which are implementing Lamport's clocks. There are many events that take place at these processes, including some messages being sent between the processes. In the figure below we use circles and arrows to specify in-processor events and messages being sent between processes, respectively. Please provide the logical timestamps associated with each event. You can assume that all three clocks start at zero, at the left-most point in time. (Also, explain how the processes would adjust their clocks if using Lamport's logical clocks.)



Question 3 (2 points)

Please explain the advantage of asynchronous calls compared to synchronous calls. Also, please give a concrete example where such functionality could be beneficial.

Part B: Methodology

Question 4 (4 points)

When designing experiments, it is important to carefully identify the most appropriate factors, levels, and metrics to consider. Consider a researcher wanting to assess the performance of a webserver. The researcher has identified three factors of interest: (i) the request rate, (ii) the job size, and (iii) the processor speed. For each of these factors, the researcher has identified 10 levels (each) of interest, including identified a default request rate, job size, and processor speed. Let us call the request rate levels R1, R2, ..., R10; the job size levels S1, S2, ..., S10; and the processor speed levels P1, P2, …, P10. Please estimate the number of experiments that the researcher would need to perform if performing (a) one factor experiments with the default scenario as baseline, (b) two factor experiments with the default scenario as baseline, and (c) full factor experiments. Also, please explain which experiments would be performed in each case.

Question 5 (3 points)

Describe the main difference between active measurements and passive measurements. Explain which approach introduce the most traffic overhead and use a concrete example to explain how these techniques can be used to build a map of the Internet topology. In your explanation, please use existing protocols/techniques and how they can be used for active and passive measurements.

Question 6 (3 points)

Consider a long duration video streaming session between a client and a server, for which it was observed that the average round trip time (RTT) between the client and server was 200ms and the average TCP window size was 50 packets, each of which is 1.5kB. It was also measured that each video frame is buffered on average 10s at the player. Please estimate the average video encoding and buffer occupancy measured in bytes? (**Hint:** You may want to use Little's law twice.)

Part C: Multicore and Parallel Programming

Question 7 (2 points)

Questions on parallel computer architecture concepts

a) Explain the concept of MIMD parallel computing, and give one example of a parallel computer architecture type where the concept is used. (1 point)
b) Name and briefly describe one type of interconnection network where the maximum distance between any two nodes (in terms of network hops) grows

asymptotically much slower than linear in the number of nodes. Motivate your answer. (1 point)

Question 8 (4.5 points)

Questions on thread programming

a) The dot product of two vectors x, y (stored as float-arrays) of N elements each is defined as $\sum_{i=0}^{N-1} x[i] \cdot y[i]$. Write a shared-memory parallel program using threads (pseudocode is fine, explain your code) that calculates the dot product. Make sure that your program is free of race conditions and try to achieve cache-friendly memory access patterns to the shared operand arrays. Explain your solution. (3 points)
b) Derive the asymptotic worst-case parallel execution time, parallel work, and parallel cost for your algorithm as functions (big-O notation) in N and the number P of processors, using the (EREW) PRAM model. (1.5 points)

Question 9 (1 points)

Question on MPI

a) Name and explain one collective communication operation in message passing / MPI programming. (1 point)

Question 10 (2.5 points)

Questions on Design and Analysis of Parallel Algorithms

a) A long-running program is known to have a perfectly parallelizable part that accounts for 90 percent of its sequential execution time. The rest is inherently sequential. If we parallelize the program, how much parallel speedup can we expect with 9 processors? Explain your calculation. (1 point)
b) What is a parallel speedup anomaly, in general? (0.5 point)
c) Explain the parallel algorithmic design pattern "Parallel Divide-and-Conquer". (1 point)

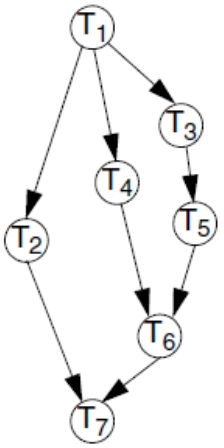
Part D: Embedded Systems

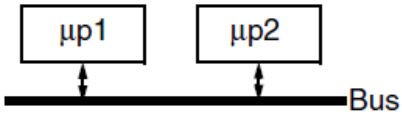
Question 11 (4 points)

Consider an application modelled as the task graph below. Each task, when activated, consumes one message on each input edge and generates, at termination, one message on each output edge. The task graph is executed on the architecture shown in the figure. Execution times of the tasks, when executed on the corresponding processor, are shown in the table. All messages transmitted over the bus, between tasks mapped on different processors, consume 2 time units to reach the destination. Communication between tasks mapped to the same processor is considered to not consume any time.

Propose an efficient task mapping (indicate on which processor each task is executed) and a corresponding static (nonpreemptive) schedule for the application. Illustrate your schedule as a Gantt chart (similar to the way we captured schedules in Lecture 1&2).

Try to achieve a maximum delay (the time interval between the start of T1 and the finishing of T7) of 46 time units!





Task	WCET	
	µp1	µp2
T_1	5	6
T_2	12	15
T_3	5	6
T_4	8	10
T_5	5	5
T_6	17	21
T_7	10	14

Question 12 (3 points)

What is an Embedded System? What makes it different from other applications? Why is it difficult to design?

Question 13 (3 points)

Think at the sources of power dissipation as we discussed at the lectures. What are main opportunities to reduce power consumption?

Bonus (only on original exam)

Question 14 (4 points)

Consider the scalability of two alternative server clusters. In the first design, we use a round-robin (RR) scheduler and all N_1 servers have independent job queues. In the second design, we use a common shared queue in which jobs wait for anyone of the N_2 servers to become free. For simplicity, we assume that each server only can serve one job at a time and that jobs in the queue are served using First Come First Serve (FCFS). Assuming that jobs arrive according to a Poisson process with request rate λ and each server has a service rate μ, these two systems can now be modelled as (i) N_1 independent $M/M/1$ queues (each with request rate λ/N_1), and (ii) one $M/M/k$ system (with $k= N_2$ and combined request rate λ). For both these systems there exists closed form equations for the response times R as a function of the request rate on each queuing system (N_1 independent systems in the first case and 1 system in the second case). Let us assume that the response times of an $M/M/k$ system with request rate λ can be calculated using the function:

```
double response_k(double lambda, int k),
```

where **`lambda`** is the request rate λ and **`k`** is the number of server stations *k* in the system of consideration. Please use pseduocode to illustrate how you would calculate and show *how the response times scales with the overall request rate* λ for six different clusters with N_1=1, 4, and 16 and N_2=1, 4, and 16. Also sketch a figure that illustrates how you expect that the final results could be presented.

Good luck!!