TDDD93/TEN2 – Large-scale distributed systems and networks
Final Examination: 14:00-18:00, Thursday, Oct. 20, 2016
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

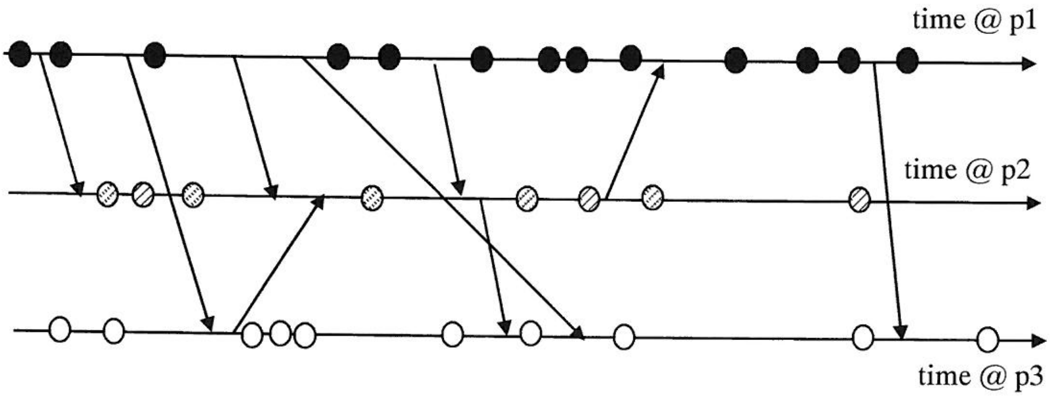
Instructions:

- 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 11 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)

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 2 (6 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 40ms and 15ms, 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 45ms total processing and the database requires 900ms 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) Assume 40% of the clients only make the first type of requests and 60% of the clients only make requests of the second type. Furthermore, assume that clients that makes the first type of requests on average makes twice as many requests as the clients that makes requests of the second type. What is the average response

time (assuming no competing jobs or other reasons for queuing) across all requests seen on the system?

(c) Please give concrete examples of two types of transparency that are provided in the above example.

Remember to explain your answers.

Part B: Methodology

Question 3 (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 9 levels (each) of interest, including identified a default request rate, job size, and processor speed. Let us call the request rate levels R1, R2, ..., R9; the job size levels S1, S2, ..., S9; and the processor speed levels P1, P2, …, P9. 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 4 (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 150ms and the average TCP window size was 40 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.)

Question 5 (3 points)

You have performed large-scale measurements and are now using scatter plots (i.e., you plot each data point individually in the x-y plane) to visualize your results. When visualizing the results you notice clear trends.

(a) What does it mean if all points end up being on a straight line on a lin-lin plot (i.e., both axes on linear scale)? Show, explain, and try to provide example equations to interpret the results.

(b) What does it mean if all points end up being on a straight line on a log-log plot (i.e., both axes on logarithmic scale)? Show, explain, and try to provide example equations.

(c) What does it mean if all points end up being on a straight line on a lin-log plot (i.e., one axis on linear and the other on logarithmic scale)? Show, explain, and try to provide example equations.

Part C: Multicore and Parallel Programming

Question 6 (4 points)

Questions on parallel computer architecture.

a) Shortly describe the high-level hardware architecture of a typical modern cluster computer used for large-scale computations in high-performance computing (such as NSC's Triolith cluster mentioned in the lecture). Make sure to name and explain its control structure and memory structure (an annotated figure). (2p)

b) What kinds of interconnection networks are suitable in large clusters that run highly parallel applications where processes need to frequently exchange data? (1p)

c) What is hardware multithreading, and why can it help to increase throughput even for a single-core processor? (1p)

Question 7 (4 points)

Questions on parallel programming.

a) What is/are the advantage(s) of writing parallel programs for a so-called parallel programming model rather than a concrete parallel computer? (1p)

b) Write a MPI message-passing program (pseudocode, using send and receive operations only) for p processes ($p>1$) to calculate, in parallel, the maximum among the N elements of a large array A of integers. Assume for simplicity that each process initially already has a copy of the entire array A. The maximum should finally be available on the process with rank 0. Explain your code, and draw an annotated figure showing the communication flow for p=4. (2p)

c) Derive the time complexity for your solution as a formula in N and p. (Use big-O notation as appropriate) (1p)

Question 8 (2 points)

Question on theory.

a) Define the term "parallel cost" as introduced in the lecture. (1p)

b) A given program for a computational problem performs 80% work that would be perfectly parallelizable over arbitrarily many processors, and 20% work that is not parallelizable at all. Give a tight upper bound for the relative parallel speedup achievable with an arbitrarily large number of processors. Explain your calculation. (1p)

Part D: Embedded Systems

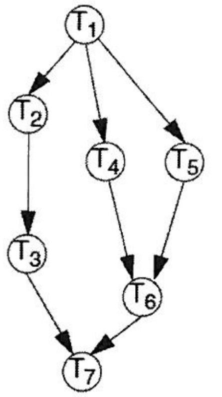
Question 9 (4 points)

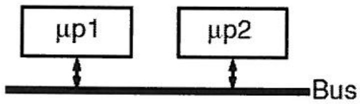
Consider an application modelled as the task graph below. Each task, when activated, consumes one message on each input edge and generates, at termination, one message on each output edge. The task graph is executed on the architecture shown in the figure. Execution times of the tasks, when executed 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	7	9
T_3	5	6
T_4	8	10
T_5	10	11
T_6	17	21
T_7	10	14

Question 10 (3 points)

In the lectures we have particularly emphasized three design steps: architecture selection, task mapping, elaboration of a schedule. Explain, in short, what each step is doing. Illustrate the three steps by a small example.

Question 11 (3 points)

Dynamic power management. How does it work? What knobs do we need from the underlying hardware platform?

Good luck!!