TDDD93/TEN2 – Large-scale distributed systems and networks
Final Examination: 8:00-12:00, Wednesday, May 31, 2017
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

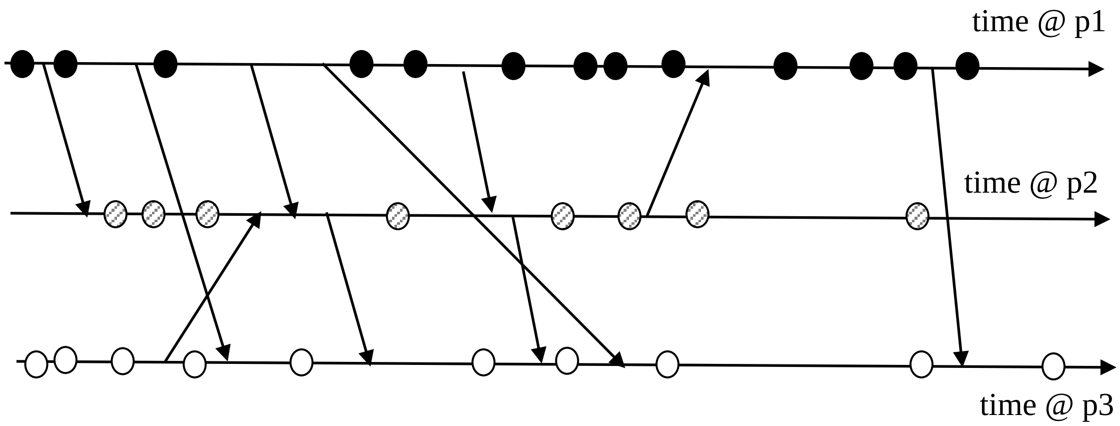
Instructions:

- 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 12+1=13 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 and utilize figures and tables to the largest extent.
- 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 (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 60ms and 30ms, 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 50ms total processing and the database requires 300ms processing to satisfy the request. The second type does not require any database access, is fully synchronized and requires 80ms 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 30% 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?

Remember to explain your answers.

Question 3 (2 points)

In the context of remote procedure call (RPC), please describe and compare at least two potential actions that can be used to deal with server orphans when the client has crashed while the server was computing.

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 9, 8 and 7 levels of interest, respectively, 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, ..., S8; and the processor speed levels P1, P2, …, P7. 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)

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 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 6 (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 7 (3 points)

Questions on parallel computer architecture.

a) What is/are the main advantage(s) and main drawback(s) of a simple (leaf-oriented) tree network if used as the main interconnection network topology for the nodes in a large cluster computer? (1p)

b) Describe the high-level architectural organization of a modern cluster supercomputer using multicore CPUs, including an annotated drawing showing computation units, memory units and interconnections for illustration. Make sure to explain which computing units can access which memory units directly, and how they can exchange data. Be thorough! (2p)

Question 8 (3.5 points)

Questions on MPI/algorithm design. You have a cluster computer running P (P>1) MPI processes. Process 0 holds in its main memory a huge array A of N elements. For simplicity, assume that P divides N.

a) Design a parallel message-passing program (using all P nodes), using explicit send() and receive() operations only, that scatters the array A, i.e. partitions it into P equally sized disjoint slices and distributes them across all P processes so that afterwards each process i, $0 \leq i \leq P$-1, shall hold, in a local array B of size N/B, the i-th contiguous partition of array A. (MPI or pseudocode is fine, explain your code). (1.5p)

b) Draw a timing diagram (Gantt chart) to illustrate the communication flow over time for your program for the case of $P = 4$ processes. (0.5p)
(Hint: A node can only send or receive one message at a time.

c) Assume that the time for sending and receiving a block of K elements is $a \times K + b$ for constant parameters a, b. Derive the asymptotic time complexity for completing the entire scatter operation (i.e., a formula in N and P, use big-O notation where appropriate). (1p)

d) Scattering a large array is a prerequisite to distribute the work of a subsequent parallelized computation. Under what condition on the subsequent computation's amount of work does it really make sense to parallelize this computation if all operand data initially resides on a single node? (General answer, use the result of (c)) (0.5p)

Question 9 (3.5 points)

Question on theory.

a) Describe the Parallel Random Access Machine (PRAM) computation model. Make sure to explain its programming model and cost model. What are the model parameters? What are the main advantages and shortcomings of the PRAM model? (2p)

b) Define the term "relative (parallel) speedup" of a parallel algorithm (commented formula). (0.5p)

c) Give an example of a (parallel) speedup anomaly. (1p)

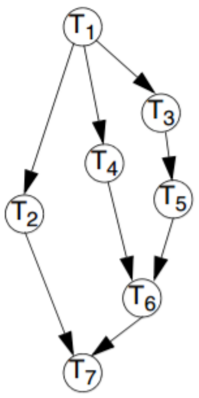
Part D: Embedded Systems

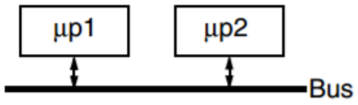
Question 10 (4 points)

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 11 (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 12 (3 points)

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

Part E: Bonus part

Question 13 (4 points)

Peer-to-peer vs client server comparison. Please derive expressions for how much time it takes to distribute file from one server to N clients/peers. You can assume that the server has an upload bandwidth U, that each peer i has an upload bandwidth of u_i, that each peer i has a download bandwidth d_i, and that the file is of size F. Then use the expressions to and plot the distribution time as a function of the number of clients/peers in the system for the two delivery models, for the case when using the normalized file size $F = 1$ (i.e., the file size is measured in units of the file size itself) and $U = u_i = 1$ (i.e., the upload rate is measured in the time units that it takes the server to upload one copy of the file and the clients have the same upload rate) and $d_i = 2$ (i.e., the maximum download rate of a client is twice its upload rate). For simplicity, assume that the file can be broken into infinitesimally small chunks and that a client can start help out as soon as it has obtained such small chunk.

Good luck!!