TDDE35/TEN2 and TDDD93/TEN2 – Large-scale distributed systems and networks
Final Examination: 8:00-12:00, Tuesday, June 4, 2019
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

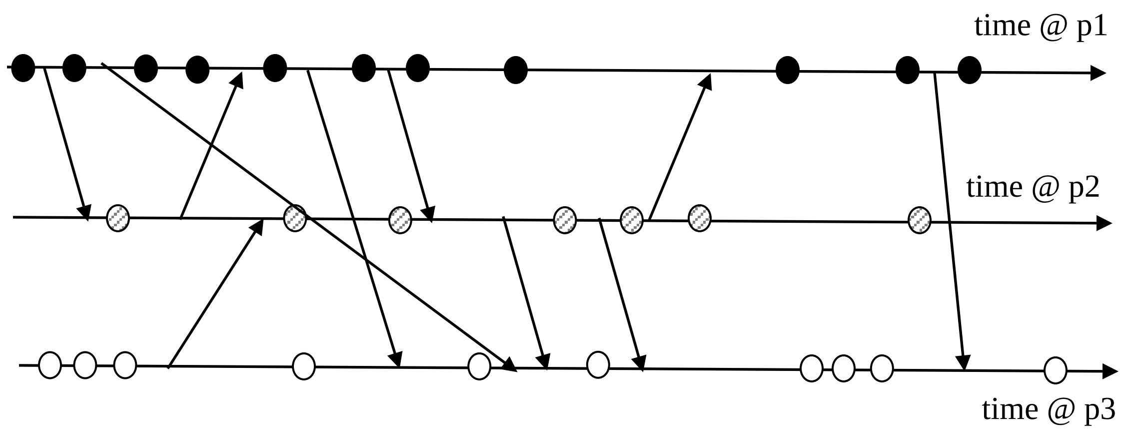
**Instructions:**

- 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 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 (3 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 28ms and 14ms, 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 48ms total processing and the database requires 32ms processing to satisfy the request. The second type does not require any database access, is fully synchronized and requires 40ms 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 that (i) 30% of the clients need to make one call of the first type, (ii) 30% need to make one call of the second type, and (iii) the rest of the clients need to make one call of each type, where the request of the second call need to include information from the response of the first call. Assuming that assuming there are no competing jobs or other reasons for queuing, what is the average response time? As always, please explain and show your work.

### Question 3 (3 points)

Mutual exclusion. Consider a simple scenario in which there are five nodes A, B, C, D, and E. First, use a figure (or a sequence of figures) to illustrate and explain the message sequences and coordination between these nodes when node A acts as a central coordinator for a shared memory resources (that all five nodes can use) and both nodes B and C almost at the same time decides that they want to write to the resource. Your figure(s) should capture the timeline of the message sequence. For this question, please assume that (i) each write access (to memory) takes 1 second, (ii) B makes its decision 10ms before C, (iii) one-way packet delays are symmetric for all node pairs, (iv) the RTT between A and B is 100ms, and (v) the RTT between A and C is 50ms.

## 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 3 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, and P3. 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 100ms 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.)

### Question 6 (3 points)

You have performed large-scale measurements and are 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) Same question as above, but now a straight line is observed on a log-log plot (i.e., both axes on logarithmic scale)?
(c) Same question as above, but now a straight line is observed on a lin-log plot (i.e., one axis on linear and the other on logarithmic scale)?

# Part C: Multicore and Parallel Programming

### Question 7 (3 points)

Questions on parallel computer architecture.

(i) Name and shortly describe an interconnection network topology that is suitable for on-chip networks in many-core processors. (1p)

(ii) How does "hardware multithreading" differ from "multicore" technically (be thorough), and what do they have in common from the software point of view? (1.5p)

(iii) What is SIMD parallelism? (0.5p)

### Question 8 (4 points)

Questions parallel programming models.

(i) What is the purpose of a thread pool? (1p)

(ii) What are "collective communication operations" in MPI? Describe one collective communication operation of your choice: its name, what are its input and output operands, what does it do with the operands (be thorough), and describe how (commented MPI pseudocode and illustrating figure) you would implement it if you were only allowed to use `MPI_Send` and `MPI_Recv` operations. (3p)

### Question 9 (3 points)

Questions on theory.

(i) A (parallel) computation model consists of a (parallel) programming model and a (parallel) cost model. What does a parallel cost model describe? What are the general requirements on parallel cost models in the early and later stages of the design and analysis of parallel algorithms? (1p)

(ii) A parallel program fulfilling the assumptions of Amdahl's Law takes 100 seconds if run on one processor and 60 seconds if run on 2 processors. What is the upper bound of the relative speedup that this program could achieve for any number of processors? Explain your answer (calculation). (1p)

(iii) What is the difference between relative and absolute (parallel) speedup? Be thorough. (1p)

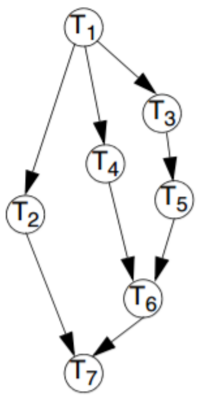
# Part D: Embedded Systems

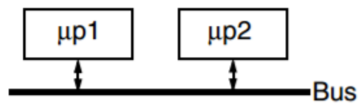
### Question 10 (4 points)

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)

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

### Question 12 (3 points)

Dynamic power management. How does it work? What knobs do we need from the underlying hardware platform? What are the main problems? What do we need to take care of in order to achieve an energy gain?

## Part E: Bonus part

### Question 13 (4 points)

Consider the scalability of two alternative server clusters. In the first design, we use a round-robin (RR) load balancer to distribute incoming jobs over $N_1$ servers with independent job queues. In the second design, we use $N_2$ servers with shared queue. For simplicity, we assume that each server only can serve one job at a time and that jobs in the queues are served using First Come First Serve (FCFS). Assuming that jobs arrive according to a Poisson process with request rate $\lambda$ and each server has a service rate $\mu$, these two systems can now be modelled as (i) $N_1$ independent $M/M/1$ queues (each with request rate $\lambda/N_1$), and (ii) one $M/M/k$ system (with $k{=}N_2$ and combined request rate $\lambda$). 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). Now, let us assume that you (or somebody else) have written a function:

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

that calculates the response times of an *M*/*M*/*k* system with request rate λ (using the known formula), where **`lambda`** is the request rate λ and **`k`** is the number of server stations *k* in the system of consideration. For this question, please use pseduocode to illustrate how you would calculate and show *how the response times scale with the overall request rate* λ for six different clusters. The first three clusters are of the first type and have sizes $N_1$=1, 4, and 16, and the last three are of the second type and have sizes $N_2$=1, 4, and 16. Also, please sketch a figure that illustrates how you expect that the final results of such performance graph could best be presented.

*Good luck!!*