Information page for written examinations at Linköping University

Examination date	2016-06-04
Room (2)	G33 G35
Time	8-12
Course code	TDDD93
Exam code	TEN2
Course name **Exam name**	Large-Scale Distributed Systems and Networks (Storskaliga distribuerade system och nätverk) Written examination (Skriftlig tentamen)
Department	IDA
Number of questions in the examination	13
Teacher responsible/contact person during the exam time	Niklas Carlsson (Christoph Kessler may also visit, if time)
Contact number during the exam time	013-282644
Visit to the examination room approximately	ca 10:00
Name and contact details to the course administrator (name + phone nr + mail)	Elin Brödje elin.brodje@liu.se 013-284767
Equipment permitted	None (with hardcopy exception of dictionary, as explained on the exam cover page).
Other important information	
Number of exams in the bag	

Information page for written examinations at Linköping University

Examination date	2016-06-04
Room (2)	G33 G35
Time	8-12
Course code	TDDD93
Exam code	TEN2
Course name **Exam name**	Large-Scale Distributed Systems and Networks (Storskaliga distribuerade system och nätverk) Written examination (Skriftlig tentamen)
Department	IDA
Number of questions in the examination	13
Teacher responsible/contact person during the exam time	Niklas Carlsson (Christoph Kessler may also visit, if time)
Contact number during the exam time	013-282644
Visit to the examination room approximately	ca 10:00
Name and contact details to the course administrator (name + phone nr + mail)	Elin Brödje elin.brodje@liu.se 013-284767
Equipment permitted	None (with hardcopy exception of dictionary, as explained on the exam cover page).
Other important information	
Number of exams in the bag	

TDDD93/TEN2 – Large-scale distributed systems and networks
Final Examination: 8:00-12:00, Saturday, June 4, 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 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 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 (3 points)

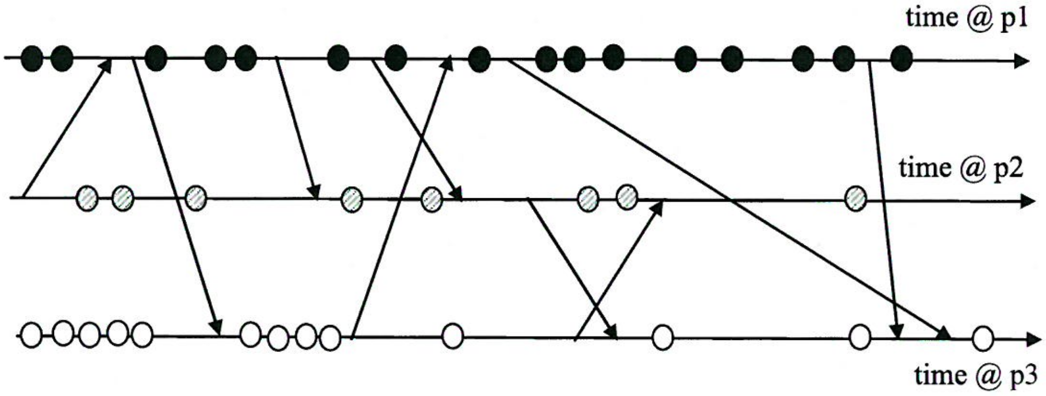
Transparency plays a central role in some distributed systems. Consider a simple multi-tier system with three levels: a user interface, an application server, and two replicated database servers. Assume these layers are implemented as a distributed cloud service at different geographic locations and that the average round trip time (RTT) between the machines used in the consecutive layers (starting with the top-tier layer) is 30ms and 20ms, 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 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).

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 both internal in-process events (shown as circles) and messages being sent between the processes (shown as arrows). 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 (3 points)

Mutual exclusion. Consider a simple scenario in which there are five nodes A, B, C, D, and E. Use 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. You can assume that each write access (to memory) takes 1 second and that there is 10ms between the times when B's and C's decisions, and that the round trip times between the nodes are random in the approximate range 50-100ms.

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, 6, and 4 levels of interest, respectively. The researcher has also 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, ..., S6; and the processor speed levels P1, P2, …, P4. 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)

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.

Question 6 (3 points)

Consider a system with two states: "on" and "off". Assume that the system is "on" whenever there are jobs to serve and the system instantaneously can go between the "on" and "off" states whenever a new job arrive to an empty system or when the system is done serving all jobs, respectively. Furthermore, assume that the system only can serve one job at a time (as with any G/G/1 queue system), on average 100 jobs/second arrive to the system, each job on average takes 20ms to serve, and each job stays in the system for on average 60ms.

a) How many jobs are on average in the system?
b) Assuming that the "on" state consumes 100 Watts and the "off" state 10 Watts. What is the average power consumption of the system, given the described workload and system characteristics?

Part C: Multicore and Parallel Programming

Question 7 (2 points)

Questions on parallel computer architectures.

a) Name and shortly describe an interconnection network topology that is suitable for on-chip networks in many-core processors. (1 point)
b) What is hardware multithreading, and why can it help to increase throughput even for a single-core processor? (1 point)

Question 8 (5 points)

Questions on MPI/algorithm design.

Given is a huge array *A* of *N* characters stored in the main memory of one node of a cluster computer with *P* (*P*>1) nodes. Given is also a short constant array *S* of *M* (*M*<<*N*, can be considered constant) characters obtained as an input argument (`argv[1]`) that is available on every node at program start.

a) Write a message-passing parallel program (MPI or pseudocode is fine, explain your code) that uses all *P* nodes in order to count the total number of (contiguous) occurrences of the string *S* in *A*, i.e., the number of all positions $i<N-M$ of *A* where, for all $j=i,\ldots,i+M-1$, $S[j]==A[j]$. Use a simple brute-force parallel algorithm for string matching like the one that we discussed in the lecture. Be thorough, and explain your code carefully. (2.5 points)
b) Which type of parallelism did you use in your program in (a)? (0.5 point)
c) Make sure to explain all communication operations in (a). Draw a figure for *P*=4 (fully connected) nodes showing the distribution of *A* and the flow of messages between the nodes. Which of these communication operations are point-to-point communications and which ones are collective communication operations? (1 point)
d) Derive the asymptotic worst-case parallel execution time for your algorithm as an expression (big-O notation) in *N*, *M* and the number *P* of nodes. (1 point)

Remark: If you do not know how to write message passing parallel programs, you could instead solve (a), (b) and (d) for the shared memory (multithreaded) programming model, though with half the amount of points each because it is much easier.

Question 9 (3 points)

Question on Theory.

The following execution times (in seconds) have been measured for two parallel programs A and B solving the same problem with the same problem size on the same parallel computer system:

Number of processors	1	2	4	8	16
Program code A	200	105	45	30	20
Program code B	20	15	12	10	9

(a) Which code scales better (i.e., relative speedup)? (1 point)
(b) Which code is more cost-efficient, and why? (1 point)
(c) Which (if any) of these codes has a speedup anomaly, and if so, where? (1 point)

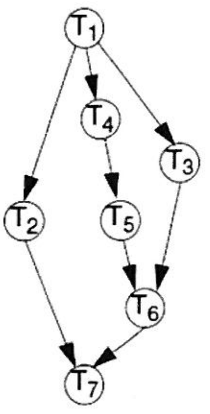
Part D: Embedded Systems

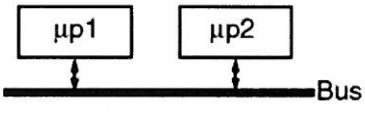
Question 10 (4 points)

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	10	11
T_4	5	6
T_5	3	4
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?

Bonus (only on original exam)

Question 13 (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!!

TDDD93/TEN2 – Large-scale distributed systems and networks
Final Examination: 8:00-12:00, Saturday, June 4, 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 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 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 (3 points)

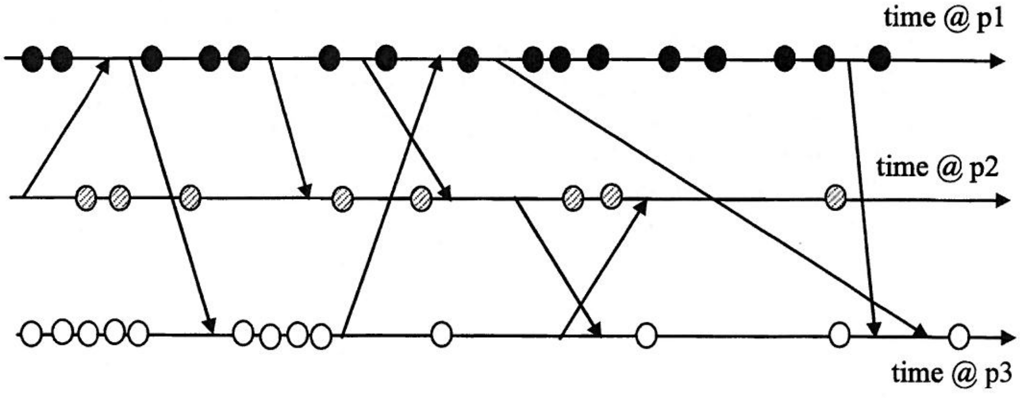
Transparency plays a central role in some distributed systems. Consider a simple multi-tier system with three levels: a user interface, an application server, and two replicated database servers. Assume these layers are implemented as a distributed cloud service at different geographic locations and that the average round trip time (RTT) between the machines used in the consecutive layers (starting with the top-tier layer) is 30ms and 20ms, 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 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).

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 both internal in-process events (shown as circles) and messages being sent between the processes (shown as arrows). 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 (3 points)

Mutual exclusion. Consider a simple scenario in which there are five nodes A, B, C, D, and E. Use 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. You can assume that each write access (to memory) takes 1 second and that there is 10ms between the times when B's and C's decisions, and that the round trip times between the nodes are random in the approximate range 50-100ms.

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, 6, and 4 levels of interest, respectively. The researcher has also 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, ..., S6; and the processor speed levels P1, P2, …, P4. 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)

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.

Question 6 (3 points)

Consider a system with two states: "on" and "off". Assume that the system is "on" whenever there are jobs to serve and the system instantaneously can go between the "on" and "off" states whenever a new job arrive to an empty system or when the system is done serving all jobs, respectively. Furthermore, assume that the system only can serve

one job at a time (as with any G/G/1 queue system), on average 100 jobs/second arrive to the system, each job on average takes 20ms to serve, and each job stays in the system for on average 60ms.

a) How many jobs are on average in the system?

b) Assuming that the "on" state consumes 100 Watts and the "off" state 10 Watts. What is the average power consumption of the system, given the described workload and system characteristics?

Part C: Multicore and Parallel Programming

Question 7 (2 points)

Questions on parallel computer architectures.

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

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

Question 8 (5 points)

Questions on MPI/algorithm design.

Given is a huge array *A* of *N* characters stored in the main memory of one node of a cluster computer with *P* (*P*>1) nodes. Given is also a short constant array *S* of *M* (*M*<<*N*, can be considered constant) characters obtained as an input argument (`argv[1]`) that is available on every node at program start.

a) Write a message-passing parallel program (MPI or pseudocode is fine, explain your code) that uses all *P* nodes in order to count the total number of (contiguous) occurrences of the string *S* in *A*, i.e., the number of all positions *i*<*N*–*M* of *A* where, for all *j*=*i*,...,*i*+*M*–1, *S*[*j*]==*A*[*j*]. Use a simple brute-force parallel algorithm for string matching like the one that we discussed in the lecture. Be thorough, and explain your code carefully. (2.5 points)

b) Which type of parallelism did you use in your program in (a)? (0.5 point)

c) Make sure to explain all communication operations in (a). Draw a figure for *P*=4 (fully connected) nodes showing the distribution of *A* and the flow of messages between the nodes. Which of these communication operations are point-to-point communications and which ones are collective communication operations? (1 point)

d) Derive the asymptotic worst-case parallel execution time for your algorithm as an expression (big-O notation) in *N*, *M* and the number *P* of nodes. (1 point)

Remark: If you do not know how to write message passing parallel programs, you could instead solve (a), (b) and (d) for the shared memory (multithreaded) programming model, though with half the amount of points each because it is much easier.

Question 9 (3 points)

Question on Theory.

The following execution times (in seconds) have been measured for two parallel programs A and B solving the same problem with the same problem size on the same parallel computer system:

Number of processors	1	2	4	8	16
Program code A	200	105	45	30	20
Program code B	20	15	12	10	9

(a) Which code scales better (i.e., relative speedup)? (1 point)
(b) Which code is more cost-efficient, and why? (1 point)
(c) Which (if any) of these codes has a speedup anomaly, and if so, where? (1 point)

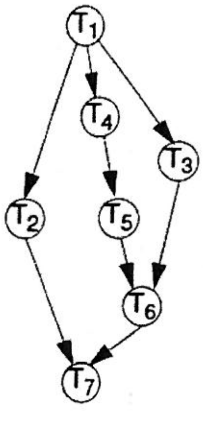
Part D: Embedded Systems

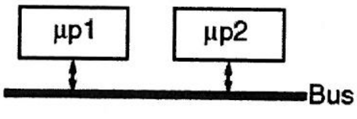
Question 10 (4 points)

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	10	11
T_4	5	6
T_5	3	4
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?

Bonus (only on original exam)

Question 13 (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!!