TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Wiskunde en Informatica

Examination Architecture of Distributed Systems (2IMN10/2II45),
on Monday January 25, 2016, from 18.00 to 21.00 hours.

Indicate on your work whether you take this exam in fulfillment of course 2IMN10 or of course 2II45.

Before you start, read the entire exam carefully. Answers to all questions must be motivated and stated clearly. For each question the maximum obtainable score is indicated between parentheses. The total score sums up to 20 points. This is a closed book exam, i.e., you are not allowed to use books or other lecture material when answering the questions.

1. (2 points) Describe the Peer-to-Peer architectural style using the appropriate vocabulary, name the concepts involved, give a motivation for its usage and mention typical usage.

   Answer. See slide 13 of the slide set on architectural styles.

2. A specific aspect of making client-server architectures fault-tolerant is dealing with orphan processes.

   (a) (1 point) Explain what an orphan process is, on which side it executes, and by which fault it is caused.

   Answer. At any moment in time, an orphan process is a process executing on the server-side that has been invoked by a remote call from a client process, whose host machine has crashed (disappeared?) sometime during the period following the time of issue of the call and the current time. A particular nasty aspect of this type of fault is that the orphan process itself may have spawned other (remote) processes whose executions also have become meaningless and a waste or resources.

   (b) (1 point) Indicate a mechanism to (partially) deal with orphans.

   Answer. A possible way to deal with orphans is to divide the life-time of a (client) machine into epochs, where an epoch is the time between any two (re)starts of the machine. Each epoch gets assigned an epoch-number in such a way that these numbers form a monotonic increasing sequence and all operations issued by a client are tagged with its current epoch number. When the server keeps track of the most recent epoch number for each client, it will be able to identify and kill orphan processes when a new epoch number. In principle, however, this action may create new orphans. The client machine can ensure killing of all offspring of the initial orphan, by broadcasting its epoch number to all machines upon a reboot. See TvS, pp 341–342, for other solutions.

3. (2 points) Describe in detail how the URL

   http://www.win.tue.nl/home/wsinmak/Education/2IMN10/ADS.html

   is resolved. In particular, indicate the closure mechanisms, and resolution procedure for the various parts of the URL.

   Answer. This URL consists of three components: a scheme **http** that names a protocol, a hostname **www.win.tue.nl** that identifies the host holding the resource, and a pathname **home/wsinmak/Education/2IMN10/ADS.html** that identifies a file (the resource) on that host. It is resolved as follows. The browser extracts the scheme and based on the value found invokes its Http_client, which is a plugin in the form of a library. This Http_client resolves the hostname by contacting its local DNS-server to obtain an IP-address. The local DNS-server may invoke other DNS-servers to assists in resolving the hostname (see TvS for a description of how this works), but in case the local DNS-server happens to be the TU/e DNS-server it will know the IP-address. Next, the Http_client will set up a connection with the host machine, construct an appropriate http-request (GET) containing the pathname and invoke the operating system to send the request to the web server at the host machine. Finally, this web server resolves the pathname to obtain the resource.

4. Interactions and interaction styles have certain qualities (for example, whether a pending interaction is persistent).

   (a) (1 point) Discuss and explain at least 4 different quality aspects (quality categories) of interactions;

   Answer. The following aspects with possible values (qualities) can be distinguished

   memory/storage: *transient* or *persistent*

   synchronization: *asynchronous*, *synchronous* or *buffered*

   units of information: *discrete* or *continuous (streaming)*

   connection: *connection-oriented* or *connection-less*

   reliability: *yes* or *no.*

   time dependence/temporal relationships: *synchronous* or *isochronous*

   For explanation on the meaning of the various qualities, see slide 9 of the slide set on interaction styles.

   (b) (1 point) What are qualities of a remote procedure call system for the quality aspects you discussed under (a)?

   Answer. transient, synchronous, discrete, reliable, connectionless, no temporal relationships

5. Consider the Chord scheme for DHTs. Assume a 6-bit identifier space, and assume that the node set N is given by $id(N) = \{3, 18, 25, 44, 50, 56\}$.

(a) (0.5 point) Give the finger table of node 18.

Answer. For a 6-bit identifier space all finger tables have 6 entries. Table FT_{18} is given by:

$FT_{18}[1] = FT_{18}[2] = FT_{18}[3] = 25$, $FT_{18}[4] = FT_{18}[5] = 44$, and $FT_{18}[6] = 50$.

(b) (1.0 point) Assume that node 56 is removed. This requires modification of the finger tables of the remaining nodes. Indicate for each of these nodes which entries of its finger table need to be modified and what their new values should be.

Answer. Since the next node on the ring after node 56 is node 3, all finger table entries whose value is 56 have to be changed to 3. No other entries need to be modified. For node p and index i with $1 \leq i \leq 6$, entry $FT_p[i] = 56$ if and only if $50 < p + 2^{i-1} \leq 56$. For nodes $3, 18, 25$, there is no index i that satisfies this criterion. For node 44 the criterion becomes $6 < 2^{i-1} \leq 12$ which has the single solution $i = 4$ and finally for node 50, the criterion becomes $0 < 2^{i-1} \leq 6$ which has solutions $1 \leq i \leq 3$.

(c) (0.5 point) Indicate a key k for which resolution starting at node 18 requires a different number of steps before and after removal of node 56. Also give the resolution sequence before and after removal. You may assume that node 18 is aware that node 3 is its predecessor.

Answer. Assuming that node 18 is aware that node 3 is its predecessor, it "knows"that it is responsible for keys k with $3 < k \leq 18$ itself, and resolves those keys in zero steps. This does not change when node 56 is removed. For keys k with $18 < k \leq 50$, the resolution process will only consult finger table entries on nodes $18, 25$ or 44 that are unmodified. Hence, for these keys, the resolution stays the same. So, the only keys for which the number of resolution steps changes are the ones for which resolution changes from $18 \to 50 \to 56 \to 3$ before removal of node 56 to $18 \to 50 \to 3$ after removal. It follows that the key should be larger than 56 but smaller than 3. For any key in this range, the number of resolution steps decreases by one. Note that for $50 < k \leq 56$ the number of resolution steps stays the same, although resolution changes from $18 \to 50 \to 56$ to $18 \to 50 \to 3$.

6. (2 points) Give the basic ingredients of an architectural description as specified by the ISO/IEC/IEEE 42010 standard. Illustrate your discussion with an appropriate UML model.

Answer. For a UML-diagram see the architectural description (meta-)model (slide 31 of the introduction slide set). Of that diagram at least the boxes labelled System of Interest, Stakeholder, Concern, Viewpoint, View, Model, and Architectural Description should be present, and of course the relationships that hold between them

7. Indicate for the following statements whether they are true or false. Motivate your

answer with a short argument.

(a) (0,75 point) In an architectural description each model SHOULD identify its stakeholders.
False. In an architectural description it are the views or rather the viewpoints to which the view conforms that should identify the stakeholders. Of course, it does no harm if this information is reiterated in the models, but as long as it is clear to which views the model belongs its stakeholders can be retrieved.

(b) (0.75 point) All DNS servers implement recursive name resolution.
False. Root servers do not.

(c) (0.75 point) (0.5 point) In the absence of caching, iterative resolution of domain names takes more time than recursive resolution.
True. At the lower levels of recursion the (sub)domains to be resolved tend to be located in geographical vicinity of each other. Hence, the RTT for those queries is shorter than a similar query from the location where the lookup of the full domain name originated.

(d) (0.75 point) Using message queues for communication provides temporal decoupling.
True. Message queues provide persistent storage, so it is not necessary for the consumer of a message to be online when that message arrives (nor when it was sent for that matter). Of course, depending on the details of the communication protocol, messages may get lost when the capacity of a queue is exceeded, so delivery is not guaranteed.

(e) (0.75 point) Top-down system design is an important aspect of component-based software engineering.
False. In component-based software engineering systems and applications are whenever possible built from predefined components that interoperate according to contractually specified interfaces. This is a bottom-up approach.

(f) (0.75 point) The following table contains the minimum, average and maximum response time of a server for 500 up to 4000 queries per second.

#queries/sec.	500	1000	1500	2000	2500	3000	3500	4000
minimum (in ms)	3.1	3.2	3.0	3.1	3.3	2.9	4.0	33.3
average (in ms)	4.1	3.9	4.0	4.2	4.1	4.2	6.5	36.5
maximum (in ms)	5.0	5.1	5.3	5.0	5.2	5.1	11.7	39.8

According to Bondi this client-server system has load scalability.
True. Bondi defines load scalability as the ability of a system to function gracefully, without undue delay or unproductive resource consumption and contention over a range of system loads. From the table, we can observe the existence of such a range. Between 500 and 3000 queries per second the response time stays essentially the same.

(g) (0.75 point) UPnP supports third-party composition.
True. This is one of the activities of a UPnP control point, e.g. video streaming in which a video-source is connected to a displaying device.

(h) (0.75 point) Mary has designed a distributed system that maintains network newsgroups. Her implementation guarantees *writes-follow-reads* consistency. Thus, she guarantees that users of her system will never see a reaction that refers to an article which they have not seen before.

True. The author of the reaction must have read the original article, and *writes-follow-reads* consistency guarantees that any write operation by a process on a data item x (here the set of messages in a newsgroup) following a previous read operation on x by the same process takes place on the same or a more recent value of x than the one that was read. Therefore the reaction is only added to copies of the newsgroup that contain the original.

8. For a certain class of problems the computational complexity of solving a problem instance of size N on a distributed system consisting of P processing elements is given by

$$T(P,N) \quad = \quad \begin{cases} \frac{N^2}{P} + N, & \text{for } P > 1 \\ N^2, & \text{for } P = 1 \end{cases}$$

(a) (0.5 point) Give the definition of the scalability metric speedup, both textual and as a formula.

Answer. Speedup is the factor by which the computation time is reduced when the problem is solved with P instead of 1 PE.

$$S(P,N) \quad = \quad \frac{T(1,N)}{T(P,N)}$$

(b) (0.5 point) Discuss the scalability of the system from the perspective of Amdahl's law.

Answer. Amdahl's law considers the situation, where a problem of fixed size N is solved by an increasingly large number P of PEs and states that the speedup is bounded from above by $1 + \frac{1}{\alpha_N}$, where α_N is the quotient of the sequential part *seq* and the parallel part *par* of the computation. From the definition of $T(P,N)$ we find $seq = N$ and $par = N^2$. So, the speedup is bounded by $1 + N$ which is independent of P. Hence, Amdahl's law tells us that the system does not scale, when the scalability criterion is defined as a growth of speedup (linearly) proportional to P.

(c) (0.5 point) Consider problem instances whose sizes depend on the number of processes used for their solution. Show that for all exponents $e > 1$

$$\lim_{P \to \infty} \frac{T(1, P^e N)}{P \, T(P, P^e N)} \quad = \quad 1$$

Answer. A simple calculation yields

$$\frac{T(1,P^eN)}{P\,T(P,P^eN)} \quad = \quad \frac{1}{1+\frac{P^{1-e}}{N}}$$

Since $\lim_{P\to\infty}\frac{P^{1-e}}{N} = 0$ for all $e > 1$ the result follows.

(d) (0.5 point) Discuss the scalability of the system from the perspective of Gustafson's law (Hint use (c)).

Answer. Gustafson's law considers the situation in which systems with more PEs are used to solve problems of increasing size. The result in part c shows that, for any $e > 1$, if the problem size grows at a rate P^e, then the speedup becomes arbitrarily close P (note the additional factor P in the denominator). Hence, from the perspective of Gustafson's law the system scales, when the scalability criterion is defined as a growth of speedup proportional to P.