TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Wiskunde en Informatica

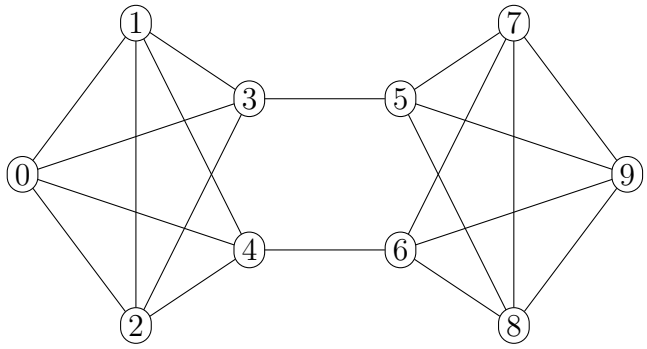
Examination Architecture of Distributed Systems (2IMN10),
on Monday January 30, 2017, from 18.00 to 21.00 hours.

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 Model-View-Controller architectural style using the appropriate vocabulary. Name the concepts and rules involved, give a motivation for its usage, and mention typical behavior and its weak points.

   Answer. See slide 18 of the slide set on architectural styles.

2. (2 points) A protocol for accessing a given replicated distributed data store with at least one correct node is *t-read-resilient* (*t-write-resilient*), when, in the presence of at most t faulty nodes, all clients that contact a correct node can perform a read (write) operation. Consider a replicated distributed data store with 10 nodes connected in the following way:



   (a) (0.5 point) Determine respectively the maximum *read-resilience* and the maximum *write-resilience* (maximum values of t), in case this data store uses a quorum-based protocol with $R = 1$ and $W = 10$. Assume that in every operation a client communicates with only a single node of the data store, which is therefore responsible for assembling the quorum. Also beware that a faulty node is *incapable* of performing routing actions necessary to assist a correct node in assembling a quorum.

   (b) (0.5 point) The same question for $R = 2$ and $W = 9$.

(c) (0.5 point) The same question for $R = 3$ and $W = 8$.

(d) (0.5 point) The same question for $R = 5$ and $W = 6$.

Answer. Due to its topology the data store has the following properties:

- In case there is only one faulty node, the data store remains connected. Hence, all correct nodes can assemble a quorum of maximal size 9.
- In case there are two faulty nodes, these can occur such that the data store becomes partitioned. Since each node has exactly four neighbours, the minimal size of a part is at least 3. The minimum occurs, e.g., when nodes 3 and 4 are faulty. Hence, the maximal size such that all correct nodes can assemble a quorum of at least that size is 3.
- In case there are three faulty nodes, these can occur such that the data store becomes partitioned. In all cases the minimal size of each part is at least 2, since all nodes have 4 neighbors, so at least one correct neighbor. A part of size 2 occurs when nodes 0, 3, and 4 are faulty. Hence, the maximal size such that all correct nodes can assemble a quorum of at least that size is 2.
- In case there are at least four faulty nodes, the maximal quorum size that can be assembled by all nodes is 1, since all nodes have four neighbours, so a correct node with only faulty neighbors can occur.

From these observations it follows that

(a) In case $R = 1$ and $W = 10$, the protocol is 9-*read-resilient* and 0-*write-resilient*.

(b) In case $R = 2$ and $W = 9$, the protocol is 3-*read-resilient* and 1-*write-resilient*.

(c) In case $R = 3$ and $W = 8$, the protocol is 2-*read-resilient* and 1-*write-resilient*.

(d) In case $R = 5$ and $W = 6$, the protocol is 1-*read-resilient* and 1-*write-resilient*.

3. For name spaces that are distributed across multiple name servers one distinguishes between iterative and recursive name resolution.

   (a) (1.0 point) Describe in some detail how each mechanism works.
   Answer. See figures 5-15 (page 206) and 5-16 (page 207) of TvS.

   (b) (0.5 point) Give an argument in favor of iterative resolution.
   Answer. In general, iterative resolution puts a lower load on non-leaf servers than recursive resolution. This is especially important for servers in the global layer, that are most frequently contacted. For root-servers this is so important that these servers do not support recursive resolution.

   (c) (0.5 point) Give an argument in favor of recursive resolution.
   Answer. With recursive resolution caching can be more effective. Resolution requests ending on the same suffix may share usage of a cached entry for that suffix. Also recursive caching leads to cheaper communication, because it involves fewer long-distance communications (see fig 5.18 TvS).

4. Component-based software engineering is an approach to system design that creates software systems from independently developed components.

   (a) (1 point) Give the (generic) definition of a component.

   Answer. See slide 10 of the slide set on component-based software engineering (CBSE).

   (b) (1 point) What is the role of a component-framework in this approach?

   Answer. See slides 12 and 13 of the slide set on component-based software engineering (CBSE).

5. Consider the Chord scheme for DHTs. Assume a 6-bit identifier space, and assume that the node set N is given by $id(N) = \{1, 7, 18, 23, 36, 55\}$.

   (a) (0.5 point) Give the finger tables for all nodes.

   Answer. For a 6-bit identifier space, all finger tables have 6 entries. Recall that $FT_p[i] = succ(p + 2^{i-1})$. Hence tables are given by:

$FT_p[i]$		p					
		1	7	18	23	36	55
	1	7	18	23	36	55	1
	2	7	18	23	36	55	1
i	3	7	18	23	36	55	1
	4	18	18	36	36	55	1
	5	18	23	36	55	55	7
	6	36	55	55	55	7	23

   (b) Next, consider the situation in which node set N is extended with node 39.

   - (0.5 point) For all $p \in N$, indicate which entries of FT_p are changed and what the new values are.
   - (0.5 point) Give a key k, $0 \leq k < 64$ and a node $p \in N$ for which the addition of node 39 increases the number of hops necessary for the resolution of key k starting in node p.
   - (0.5 point) Give a key k, $0 \leq k < 64$ and a node $p \in N$ for which the addition of node 39 reduces the number of hops necessary for the resolution of key k starting in node p.

   Answer. The only way an entry $FT_p[i]$ can change due to the addition of node 39 is that its value becomes 39. For this to happen, it must be the case that $36 < (p + 2^{i-1}) \mod 64 \leq 39$. The instances for which this condition holds are $(p, i) \in \{(7, 6), (23, 5), (36, 1), (36, 2)\}$.

   Since $FT_7[6]$ has changed from 55 to 39, it follows that key $k = 55$ is now resolved in 2 hops from node $p = 7$ ($7 \rightarrow 39 \rightarrow 55$), whereas it used to be 1 hop. Note that it is not necessary to compute FT_{39} to see this, because 55 is

the successor of 39 and is therefore reachable in 1 hop from 39.

As a consequence of the same change, it follows that key $k = 39$ is now resolved in a single hop from node $p = 7$, whereas it used to be resolved in 3 hops ($7 \to 23 \to 36 \to 55$).

6. Requirements for quality attributes are specified through quality scenarios.

   (a) (1 point) Name the ingredients of a general quality scenario.
   Answer. The inputs: stimulus and source of stimulus; the preconditions: state of the system that describes the environment of the scenario, and the resources of the systems (the artifacts) that are the subject of the scenario; the outputs: the response to the stimulus and the response measure which quantifies the extent of the response.

   (b) (1 point) Give a specific quality scenario for the real-time performance of a video conference system. For each ingredient mentioned under (a), make explicit by which part of your answer it is addressed.
   Answer. When, upon start of the video conference (environment) the chairman (source) welcomes (stimulus) the other participants, he wants them all to receive (response) that message approximately simultaneously (measure, expressed as a two-sided bound on the transmission delay). The scenario addresses the latency of the network (resources) over which the video conference is held.

7. (2 points) Name at least two of Kruchten's (library) views. For each view you mention, indicate its principle stakeholders, their concerns and the architectural issues addressed by the view.

   Answer. See slide 37 of the introductory slide set.

8. Indicate for the following statements whether they are true or false. Motivate your answer with a short argument.

   (a) (0.75 point) Triple modular redundancy is a *passive* redundancy tactic.
   Answer. False.
   In triple modular redundance, each service (computation) is performed by three independent components, whereafter the outcome is determined by majority vote. If the outcome is input to a subsequent computation also the voting elements are replicated three times. Since replicas compute in parallel, this is a form of active redundancy.

   (b) (0.75 point) Fat clients minimize client-server communication.
   Answer. True.
   Fat clients can process data before it is passed on to a server. This can be used, e.g., to check for user errors at the client side, thereby reducing the number of error messages from the server followed by corrected user input messages from the client. Also data can be compressed or aggregated into partial results before sending it to a server, thereby reducing both number and size of messages.

(c) (0.75 point) Absence of client-oriented consistency breaks replication transparency.
Answer. True.
Replication transparency means that users of a resource or service cannot detect that it is replicated. In the absence of client-oriented consistency, users (clients) of a replicated data store may observe an unexpected state. Assuming that the data store operates correctly and other clients do not intentionally provide misinformation, such as for instance replying to non-existing messages, the most plausible explanation then is that the client currently interacts with a replica different from the one with which he interacted in the paste.

(d) (0.75 point) Data compression is a tactic to achieve space scalability (in the sense of Bondi).
Answer. True.
According to Bondi, a system exhibits space scalability, if its memory requirements do not grow to intolerable levels as the number of items supported increases (memory increases sublinearly!). Data compression may indeed limit memory usage in this way, e.g., by using references or run-length encoding to store second and later occurrences of a data item in a data set. In general, these techniques get more efficient as the size of the data set grows.

(e) (0.75 point) The servers in a demilitarized zone (DMZ) are reverse proxy servers.
Answer. True.
A DMZ is a network segment set up to protect a LAN. All requests by clients targeted at some origin server in the LAN are redirected for further handling to nodes in the DMZ, which are the only ones that are allowed to contact these origin servers. So, a node in the DMZ accesses the resources of the LAN on behalf of a client, which makes it a reverse proxy server.

(f) (0.75 point) Deferred RPCs are used for latency hiding.
Answer. True.
The mechanism consists of two asynchronous RPC calls; first a request by the client, followed by a response by the server. Usually the request is made by the client to prefetch some information from the server that will speed-up subsequent synchronous RPCs by the client that without the prefetched information would block for a considerable amount of time.
(see TvS pp. 134–135 for more information)

(g) (0.75 point) mDNS + DNS-SD can be used to discover all weather (meteorological) services offered by the various airports in a country.
Answer. False.
mDNS in combination with DNS-SD is used on a local link of a single net, whereas the scenario in the statement involves internet communication.

(h) (0.75 point) Reference parameters cannot be used in RPCs.
Answer. True.

A reference parameter involves an address to a memory location. Since the caller and callee in an RPC do not share their address spaces, e.g. because they are hosted on different machines, RPCs cannot use reference parameters.