**TECHNISCHE UNIVERSITEIT EINDHOVEN**
**Faculteit Wiskunde en Informatica**

*Examination Architecture of Distributed Systems (2IMN10),*
*on Thursday, November 8, 2018, from 9.00 to 12.00 hours.*

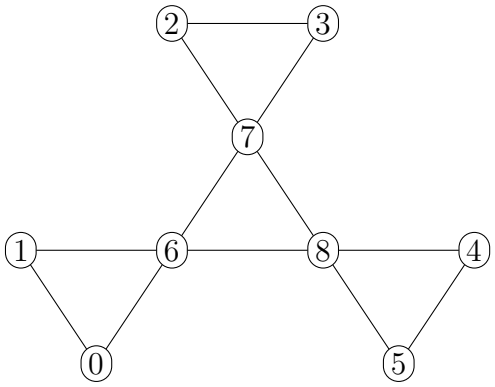
---

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 Virtual machine 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 27 of the slide set on architectural styles.

2. Consider a replicated distributed data store with 9 replicas each managed by an individual replica manager (RM). For communication within the data store, the RMs are connected according to the following network topology.




   Clients access the data store via an RM. Each client always contacts a single RM, but may contact distinct RMs in successive operations. To increase availability under network partitions, while maintaining eventual consistency, the data store uses Gifford's quorum protocol, with read quorum size $NR$ and write quorum size $NW$.

   (a) (1.0 point) Explain how Gifford's quorum protocol works and indicate the constraints that need to be imposed on the quorum sizes in order for the protocol to achieve consistency.

   **Answer.** To execute an operation the data store needs to establish a subset of RMs, called a quorum, that is capable to engage in the operation, i.e., is reachable from the RM where the operation is submitted to the store. For read

operations, the size of the set is given by $NR$, and for write operations by $NW$. Upon writing, the data object is updated in all replicas of the write quorum and is given a unique time stamp (version number) that is more recent than any time stamp handed out in an earlier update. Upon reading the value returned to the clients is the value from the replica that holds the most recent time stamp. For a data store with $N$ replicas, the quorum sizes need to satisfy two constraints:

- $NR + NW > N$, to prevent read-write conflicts,
- $NW > N/2$, to prevent write-write conflicts.

(b) (0.5 point) For the data store described above, assume that the RM network obeys the following fault model. Links are fully reliable, i.e., no messages are lost or corrupted, and at any moment in time at most one RM is crashed. To operate the store, consider choosing between the pairs $(NR, NW) = (4, 6)$ and $(NR, NW) = (5, 5)$ of quorum sizes. Is there a difference in availability of the store between these pairs?

**Answer.** No. Under the given fault model, there are only two topological distinct configurations in which a node is crashed. In the first configuration, a node from the set $\{0, 1, 2, 3, 4, 5\}$ is crashed, in which case the remaining nodes form a single connected set of 8 nodes that is available both for read and write operations, irrespective which pair of quorum sizes is chosen. In the second configuration, a node from the node set $\{6, 7, 8\}$ is crashed, in which case the store is partitioned into a set of 6 nodes and a set of 2 nodes. No matter which pair of quorum sizes is chosen, in the larger part both operations are possible and in the smaller part neither of them.

(c) (0.5 point) Answer the same question, but now for a choice between quorum size pairs $(NR, NW) = (2, 8)$ and $(NR, NW) = (3, 7)$.

**Answer.** Yes, there is a distinction regarding availability of read operations, in case the store is partitioned into two parts. For $NR = 3$, read operations are only available to clients that connect to an RM in the larger part. For $NR = 2$, read operations are available to all clients. Note that no part of the data store is available for write operations under either quorum size pair.

3. RPCs are a form of direct communication in distributed systems.

(a) (1 point) Name the architectural elements involved in this interaction style and describe their responsibilities.

**Answer.**

| element | responsibilities |
|---|---|
| client process | calls the procedure in the normal way on the interface offered by the client stub and receives the result |
| client stub | marshals the call into a message, resolves the location of the server stub and hands the message to the client OS for transmission<br>unmarshals the result delivered by the OS and forwards it to the client process |
| client OS | sends the message to the server OS at the location of the server stub<br>receives the message containing the result and delivers it to the client stub |
| server OS | receives the message containing the call and delivers it to the server stub<br>sends the result to client OS |
| server stub | unmarshals the message and calls the corresponding local procedure on the server process<br>marshals the result into a message and hands it to the server OS for transmission |
| server process | executes the invoked procedure and returns the result to the server stub |

(b) (1 point) Explain the difference between synchronous and asynchronous RPCs. For both variants, give an example of their usage.

**Answer.** In both variants the interaction between client and server consists of a request followed by a reply. In a synchronous RPC, the client performing the call (request) is blocked until the result (reply) has been received. Upon receipt of the request the server calls the local procedure and replies with the result. In an asynchronous RPC, the server, upon receipt of the request, immediately sends an acknowledge to client (reply) before it proceeds to call the local procedure. In this case the client is only blocked until the acceptance message is received. In case a result is required, it is returned by the server at a later stage by invoking a callback. When the result message arrives the client is interrupted and processes the result using the callback. The combination of an asynchronous call with a callback is called a deferred synchronous RPC. Common usage of Web-services involves synchronous RPCs (e.g. using SOAP over HTML). Asynchronous RPCs are used for eventing. In particular, one-way RPCs are used in multicasting to a group of servers, when a reply is needed from only a subset of the group.

4. Requirements for quality attributes, such as modifiability, can be specified through scenarios and realized by the application of tactics.

(a) (0.5 point) Give a definition of modifiability.

**Answer.** Modifiability is concerned with the cost of system change: with the extent of the modification and with the time and effort to do it. For details, see slide 46 of the slideset on quality attributes.

(b) (0.5 point) State a modifiability requirement and give a scenario that specifies it.

**Answer.** Recall that a scenario consists of 6 ingredients (see slide 48 of the slideset) A change request (stimulus) from the owner of a system (source of stimulus) to expose functionality currently (so, state of the system is äfter release") only accessible via a command line interface (CLI) also via a graphical user interface GUI. So, affected artifacts are code; in particular interfaces, and modules for runtime configuration. The response consists of designing and implementing the new interfaces and configuration code (hopefully reusing the current CLI-backend), testing and integrating it in the system and deploying the extended system. The updated system should be available within a month at a prefixed price. From the price and time a budget (measure) in man-months can be determined.

(c) (1 point) Mention at least two tactics that improve modifiability.

**Answer.** Localize modifications, prevent ripple effects, defer binding time. For details, see slides 53–55 of the slideset on quality attributes.

5. Consider the Chord scheme for DHTs. Assume an 8-bit identifier space, and assume that the node set $N$ is given by $id(N) = \{32 \cdot i \mid 0 \leq i < 8\}$.

(a) (0.5 point) Give the finger table of node 160.

**Answer.** For an 8-bit identifier space, all finger tables have 8 entries, where for node $p$ table $FT_p$ is given by $FT_p[i] = succ((p + 2^{i-1}) \mod 256)$, for $1 \leq i \leq 8$. Thus we obtain

$FT_{160}[1] = FT_{160}[2] = \ldots = FT_{160}[6] = 192$, $FT_{160}[7] = 224$, and $FT_{160}[8] = 32$.

(b) (1.0 point) What is the maximum number of steps necessary to resolve a key? For your answer, you may assume that each node is aware of the identity of its predecessor and, consequently, resolves all keys for which it is responsible in zero steps.

**Answer.** As can be seen from $FT_{160}$ (the other tables are similar), each time the resolution process forwards the lookup query from node $n$ to node $(n+x) \mod 2^8$, the increment $x$ satisfies $x \in \{32, 64, 128\}$. Furthermore, in any sequence of forwarding operations, the increments are decreasing with exception of the last elements, which may all be 32. For $x = 128$ the next increment is smaller, because otherwise the lookup query travels at least once around the ring (since there are only $2^8 = 2 * 128$ identifiers) which cannot be the case. For $x \in \{64\}$, the next increment is smaller, because otherwise the query would have been forwarded from $n$ to $(n + 128) \mod 2^8$. So a maximal length sequence starts

with a step of size 128, followed by a step of size 64, followed by $\ell \geq 1$ steps of size 32. Since $128 + 64 + \ell \cdot 32$ must be smaller than $2^8 = 256$, it follows that $\ell \leq 1$. Hence, $\ell = 1$ and the maximum number of steps is 3.

(c) (0.5 point) Give a key that requires the maximum number of steps to be resolved, when starting the resolution at node 160. Indicate the nodes that are visited when resolving the given key.

**Answer.** Starting at node 160, any key $k$ with $1 \leq k \leq 31$ or $65 \leq k \leq 95$ or $97 \leq k \leq 127$ will need 3 steps to be resolved. E.g., for $k = 97$, the path traversed is $160 \rightarrow 32 \rightarrow 96 \rightarrow 128 = succ(97)$. Note that the algorithm described in TvS also takes 3 steps for key 128, but since node 160 knows that node 128 is its predecessor, it could take advantage of that knowledge and forward the query directly to its predecessor.

6. A consistency model is a contract between a data store and its clients.

(a) (1 point) What are the guarantees given by a sequential consistent data store?

**Answer.** That the results of any execution is the same as if the (read and write) operations by all processes (clients) on the data store were executed in some sequential order (single server paradigm) and that the operations of each individual process appear in this sequence in the order specified by its program.

(b) (1 point) Explain how the primary backup protocol for fault tolerance realizes this form of consistency.

**Answer.** In a primary-based protocol, such as the primary backup protocol, all writes are performed by the primary, which subsequently propagates the resulting updated object to all secondaries and waits for their acknowledgement, before replying (acknowledging) the write request to the client. This ensures that all replicas are updated in the same total order determined by the primary. Moreover, since the interaction between the data store and every individual client is synchronous, i.e., an alternation of requests and replies, the operations originating from a single client are performed by the data store in the order in which they were issued by that client. Note that since reads are done locally, no RM will see all reads. Nevertheless, they can be interleaved with the writes such that a single server illusion is created.

7. In the lecture notes, a scalability framework has been introduced to make the notion of scalability quantifiable.

(a) (0.5 point) State the generic components of that framework.

**Answer.** A scale parameter, a scalability metric and a scalability criterion. For details, see slide 8 of the slideset on scalability.

(b) (0.5 point) Illustrate its usage on a concrete example system. In particular, show for each of its components how it can be instantiated in your example.

**Answer.** A possible example is a client-server system, where the server is implemented as a cluster of identical server nodes. The load on the system will be quantified as the number of queries/second issued by all clients combined, and the capacity of the system will be quantified as the number of server nodes in the cluster. For the scalability metric of interest, we take the average response time. Furthermore, we define $R$ to be the average response time of a cluster consisting of a single server at a typical load of $N_0$ queries (reference for normalization). Then the system scaled by a factor $k$ (the scale parameter) would consist of a cluster of $k$ server nodes receiving a load of $k \cdot N_0$ queries. As the scalability criterion, we require that the average response time at scale $k$ is, e.g., between $0.9R$ and $1.1R$, for all $k$ up to 10000.

(c) (1 point) Amdahl's law and Gustafson's law both consider the scalability of a system. Explain the difference between the two perspectives on scalability expressed by these laws.

**Answer.** Both laws are concerned with the effect of increasing the computational resources of a system in response to the growth of the computational load on that system, as measured by speed-up. Amdahl's law considers the situation of a fixed load and states that if there is a part of the computation that is intrinsically sequential, i.e., cannot be distributed amongst compute resources, the system does not scale. Gustafson's law, on the other hand, states that if the computational load can be increased in correspondence with the increase in computational resources in such a way that the intrinsically sequential part stays at a fixed fraction of the load per compute resource, then the system is scalable.
Paraphrasing, Amdahl's law says that there is a limit on how fast we can solve a fixed problem, no matter how large a system we use, whereas Gustafson's law says that bigger systems should be used to solve larger problems, not for solving a fixed problem faster.

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) A name service provides a lookup mechanism based on (attribute,value)-pairs.
**Answer.** False.
A name service looks up the attributes of an object based on its name. Services

that lookup the name of an object based on (attribute,value)-pairs are called directory services.

(c) (0.75 point) Load balancing improves the availability of a system.
**Answer.** False.
Load balancing improves the response time of a system. Although a client waiting for a response has no way of knowing, whether the system is unavailable or merely slow this is not the same. Availability is defined as the probability to find a system is ready for service, but a high availability does not imply a rapid reply.

(d) (0.75 point) Using a Pipes & Filters architectural style supports flexibility.
**Answer.** True.
Pipes, in a sense, provide standard interfaces for passing data between filters. Moreover, all dependencies between filters are given by the pipes that connect them. Hence we can easily reorder or replace filters, to get better performance or even different functionality as long as the replacements adhere to the standards imposed by the pipes.

(e) (0.75 point) Proxies provide access transparency.
**Answer.** True.
Although reasons for their existence vary wildly, all proxies share the fact that they are representations for objects at other locations. They are designed in such a way that third parties requiring a service from the original object can obtain that same service in the same way from the proxy. So, in terms of interaction, proxies are indistinguishable from the original, which is referred to as access transparency.

(f) (0.75 point) Using a push-based protocol between a client cache and origin server decreases client response time.

**Answer.** True.
When the server is pushing fresh data to the client immediately when it becomes available, the client is less likely to find a stale cache item. Hence its response time improves (on average).

(g) (0.75 point) An architectural viewpoint is a collection of models
**Answer.** False.
A viewpoint is a collection of patterns, templates, and conventions for constructing models that address a specific (set of) concern(s). In addition, a viewpoint identifies stake-holders that have an interest in those concerns. A collection of models constructed according to the principles and guidelines of a particular viewpoint, on the other hand, is called a view.

(h) (0.75 point) For a specific application it is essential that the client side knows whether it communicates with the origin server or not. For such a situation a REST-based architectural style would be a good choice.
**Answer.** False.

Two of the main features of REST are the usage of a uniform interface for communication and the usage of caching to improve performance. Both features make it hard to determine the origin of a reply to queries.