TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Wiskunde en Informatica

Examination Architecture of Distributed Systems (2II45),
on Wednesday January 22, 2014, from 09.00 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 parenthesis. The total score sums up to 25 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) Give the basic ingredients of an architectural description as specified by the ISO/IEC/IEEE 40210 standard. Illustrate your discussion with an appropriate UML model.

   Answer. For a UML-diagram see the architectural description (meta-)model (slide 40 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.

2. Consider a client-server architecture that uses RPCs.

   (a) (1 point) Indicate the software components, both on the client and on the server side, that implement the RPC mechanism and explain their role.
   Answer. On both the client and the server side there is a stub. The stub at the client side packs the call (procedure name + parameters) into a message which it hands to the client OS for transmission. The stub at the server side unpacks the message and invokes the procedure on the server. When the client and server run distinct OSs also marshalling and unmarshalling of data may be involved. For the return values the reverse process is followed. See also fig. 4.7 on page 130 of the book by Tanenbaum and van Steen (TvS).

   (b) (1 point) A large part of this implementation can be automatically generated from an interface description using a framework such as, e.g., DCE-RPC. Describe in some detail the compilation procedure that needs to be followed to obtain these components.
   Answer. See slide 35 of the slide set on interaction styles.

3. (1 point) Explain the difference between an architectural view and a viewpoint.

   Answer. A view is a collection of models that conforms to a viewpoint. This viewpoint defines stakeholders and their concerns. It provides the concepts and vocabulary used in the view to express stakeholder concerns. Moreover, it may provide methods and techniques for their analysis.

4. (1.5 points) In a given distributed system, a shared file store is used for background storage. For the architecture of the file store we may choose between a centralized Client- Server or a distributed Peer-to-Peer solution (in which each client stores part of the file system). Which of these two styles is more appropriate considering

   (a) performance
   Answer. P2P, as it allows parallel processing of both individual as well as multiple simultaneous requests;

   (b) maintainability
   Answer. Centralized CS, single source. Note: maintainability $\neq$ availability.

   (c) information protection (access control)
   Answer. Centralized CS, it is easier to regulate access at a single access point. On the other hand, if in a P2P solution the protection fails at a single peer, only part of the information is compromised.

   If you think there is no clear winner, mention an advantage for each architectural style

5. (2 points) Describe the REST architectural style using the appropriate vocabulary, name the concepts involved, give a motivation for its usage and mention typical usage.

   Answer. See slide 23 of the slide set on architectural style.

6. (2 points) Consider the Chord scheme for DHTs. Assume an 8-bit identifier space, and assume there are 4 nodes with equidistant identifiers that store resources,e.g., $id(N) = \{0, 64, 128, 192\}$. Assume look-ups of resources are uniformly hashed, i.e., each 8-bit key is generated with equal probability.

   (a) Give the finger table of node 128.
   Answer. $FT_{128}[1] = \cdots = FT_{128}[7] = 192$ and $FT_{128}[8] = 0$.

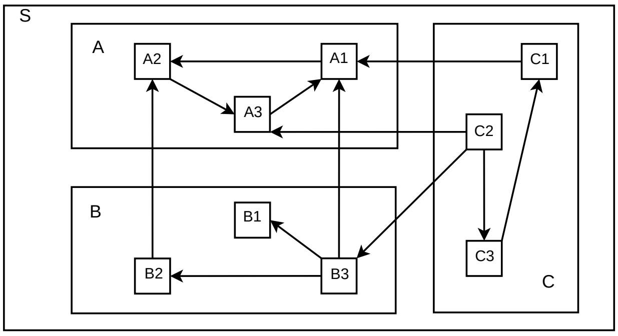
   (b) What is the average number of hops taken to look up a resource (resolve a key)?
   Answer. Since the configuration is completely symmetric w.r.t. all nodes, it suffices to determine the average number of hops needed to resolve a key for a single node only, say node 0. To begin with, note that $FT_0[1] = \cdots = FT_0[7] = 64$ and $FT_0[8] = 128$. For key k, where $0 \leq k < 256$ we therefore find:

      0 hops, for $192 < k \leq 256 = 0$, because the resources with those keys reside on node 0.
      1 hop, for $0 < k \leq 128$, because the corresponding resources reside either on node 64 or on node 128, and the locations of both these nodes are in FT_0.
      2 hops, for $128 < k \leq 192$, because the corresponding resources reside on node 192. Resolution from 0 first visits node 128, where we find 192 in the finger table (see part (a)).

So the average number of hops is $1 = \frac{1}{4} \cdot 0 + \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2$.

7. (1 point) Describe the TCP-handoff mechanism. Why is it used?
   Answer. See slide 8 of the slide set on naming, or pages 94–95 of TvS.

8. Consider a system whose module view is given by



   (a) (1 point) Give its *part-of* relation P and its *uses* relation U.
   Answer.

$$\begin{aligned} P &= \{\langle A_1, A\rangle, \langle A_2, A\rangle, \langle A_3, A\rangle, \langle B_1, B\rangle, \langle B_2, B\rangle, \langle B_3, B\rangle \\ &\quad , \langle C_1, C\rangle, \langle C_2, C\rangle, \langle C_3, C\rangle, \langle A, S\rangle, \langle B, S\rangle, \langle C, S\rangle\} \\ U &= \{\langle A_1, A_2\rangle, \langle A_2, A_3\rangle, \langle A_3, A_1\rangle, \langle B_2, A_2\rangle, \langle B_3, A_1\rangle, \langle B_3, B_1\rangle, \\ &\quad , \langle B_3, B_2\rangle, \langle C_1, A_1\rangle, \langle C_2, A_3\rangle, \langle C_2, B_3\rangle, \langle C_2, C_3\rangle, \langle C_3, C_1\rangle\} \end{aligned}$$

   (b) (0.5 point) Indicate how relation algebra can be used to determine that this system can is non-strictly layered, i.e., the modules A, B and C can be ordered such that all components from a layer only use components of a lower layer.
   Answer. The system is indeed non-strictly layered with top-layer C, middle layer B and bottom layer A. For this layering order of the modules, the set F of forbidden uses-relations between layers is given by $F = \{(A, B), (A, C), (B, C)\}$. Using this set, the layering rule expressing that the system is non-strictly layered becomes $(F \downarrow P) \cap U = \emptyset$. You were not required to actually compute the left hand side of this equation to verify its validity.

9. (2 points) Give a (quantitative) definition of modifiability. Name at least two tactics that can be applied to achieve this quality and give an example of each.

   Answer. Modifiability is the quality attribute that addresses the effort and cost involved in system change. The effort can be expressed in all kinds of metrics, from lines or modules of code that have to be changed, via the man-power involved in doing that, to, in the end, simply money. Modifiability tactics aim at reducing these costs, e.g., by reducing the time it takes to (re)deploy the system. Three broad categories of tactics are: *localize modification, prevent ripple effects* and *defer binding time.* Examples of specific tactics within these categories can be found on slides 49–51 of the slide set on quality attributes.

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

    (a) (0.5 point) Recursive resolution of domain names uses less network bandwidth than iterative resolution.
    Answer. True. At the lower levels of recursion the domains to be resolved are in geographical proximity, and therefore less bandwidth of the global internet is used. Of course, caching may blur this effect, and scenarios of sequences of lookups are imaginable in which iterative resolution generates less network traffic.

    (b) (0.5 point) Distributed hash tables can be used to implement application level multi-casting.
    Answer. True, the DHTs contain routing information that can be used to compute a minimal spanning tree necessary for implementing multi-casting through a P2P routing overlay. Note, however, that optimal routing at the overlay level does not imply optimal usage of the underlying physical network.

    (c) (0.5 point) Streaming data in synchronous transmission mode ensures a two-sided bound on the end-to-end delay (a.k.a. bounded jitter).
    Answer. False. Two-sided bounds are obtained using isochronic transmission mode.

    (d) (0.5 point) Proxies can be used to manage the life cycle of an object by means of activation policies.
    Answer. False. Although occasionally a component performing life cycle management tasks such as, e.g., creation and destruction of objects is (erroneously) referred to as a proxy, the proper name for such a component is adaptor. Usually, proxies are located on a machine different from the one that hosts the original object for which they stand in, thus making it difficult, if not impossible, to manage the life cycle of such an object.

11. (1.5 points) Give the three ingredients of a scalability framework. Illustrate their usage with an example.

    Answer. First, there are the *scale parameters*, that are used to indicate the load on and size of the system. To better study the scalability, these parameters may be related via a *scale factor* that is used to couple their growth rates. Second, there are the *scale metrics* that indicate the type of load or size and that quantify their magnitudes. Finally, there are the *scalability criteria* that are used to compare the scalability metrics of various architectures, or to set an absolute target for a metric. See the slide set on scalability for an illustration of the framework by means of a matrix multiplication example or the GFS.

12. (1 point) In distributed systems we can distinguish between referentially (space) coupled or decoupled communication/coordination. Explain what this distinction is about. Give an example of both types of communication.

Answer. In referentially coupled interaction the communication/processing entities are aware of each others identity. In referentially decoupled interaction parties do not know each other and are often not even aware of each others existence.

Coupled interaction: RPC-interaction with known Web service, email.

Decoupled interaction: Publish-subscribe using topics.

13. (1.5 points) Describe the primary back-up protocol, i.e., the participating entities and actions these entities perform (including the order in which they are performed)upon a read and a write operation.
Indicate the type of consistency obtained by this protocol and explain why this is the case.

    Answer. Participants are at least client and servers (primary and back-up) In a more detailed architecture, servers may run front-ends and replica managers that manage local replicas. Clients then always contact a front-end which, depending on the particular protocol, contacts one or more replica managers. One can distinguish three versions, of which two are mentioned in the book by TvS (pages 308–310). Primary-read primary-write, which is not very efficient but provides linearizability, primary-read local-write, which provides sequential consistency, and local-read and local-write which provides eventual consistency. For the protocols see TvS. For this exercise, describing any of the three protocols was sufficient.

14. (2 points) Name two architectural styles that are used in the OSAS system. Motivate their usage.

    Answer.

    Service-oriented. Sensors offer their sensing capabilities as services for usage by other sensors and applications. In this way it is easy to compose applications that rely on data aggregated by many sensors of multiple types.

    Virtual machine. Sensors interpret byte code. This style facilitates efficient code distribution and node customization.

    Publish-subscribe. Clients can subscribe to and customize sensor services, e.g., receive a notification when the temperature rises above 30 degrees Celsius, or determine sampling rate. Customization may vary between clients.

15. (2 points) Relays (or routers) are special queue managers in a message-queueing system that forward incoming messages to other queue managers. Describe at least two reasons for their existence.

    Answer. (see slide 42 of the slide set on communication, or the book by TvS p. 147 - 149):

    - scalability: only few routers (rather than all queue managers) need to know (and therefore to be updated upon changes to) the network topology and queue-to-location mapping.

- secondary processing: e.g. logging of messages for security or fault tolerance, or transforming messages;
- multi-casting: sending a message to multiple queues.