TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Wiskunde en Informatica

Examination Architecture of Distributed Systems (2II45),
on Monday October 27, 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 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 SOA architectural style using the appropriate vocabulary, name the concepts involved, give a motivation for its usage and mention typical usage.

   Answer. See slide 21 of the slide set on architectural style.

2. Both Amdahl's law and Gustafson's law address scalability by looking at the speedup that can be achieved by the replication of processing elements.

   (a) (0,5 point) Give a definition of this scalability metric (i.e., speed-up).
   Answer. The speed-up S is given by the formula

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

   where $T(P,N)$ is the execution time a problem of size N using P processing elements.

   (b) (1,5 point) Explain the difference in scalability perspective offered by the two law's. In particular, indicate the underlying assumption that makes Gustafson's law more optimistic, i.e., explain why it considers more systems scalable than Amdahl's law.
   Answer. Amdahl's law considers the situation of solving a *fixed-size* instance of a problem with an increasing amount of PEs. In this case, the ratio between the size of sequential part of the instance and the part of the instance amenable to parallelization places an upper bound on the attainable speed-up. Gustafson's law, on the other hand, considers the case where the *instance size is increased* with the number of PEs, and states that speed-up proportional to the number of PEs is obtained as long as the instance size is chosen such that the ratio of the two parts is kept constant. As long as this does not lead to overwhelming instance sizes, scalability is maintained.

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 for iterative resolution.
Answer. In general, iterative resolution puts a lower load on non-leaf servers than recursive resolution. This 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 for 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. (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 29 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. A similar part of the conceptual model of the IEEE 1471 standard, as given on slide 21, will also be accepted as a proper answer.

5. Consider the Chord scheme for DHTs. Assume an 6-bit identifier space, and assume that the node set N is given by $id(N) = \{3, 15, 18, 30, 37, 52, 59\}$.

   (a) (1 point) Give the finger tables of nodes 15 and 52.
   Answer. For a 6-bit identifier space all finger tables have 6 entries. Table FT_{15} is given by:

   $FT_{15}[1] = FT_{15}[2] = 18$, $FT_{15}[3] = FT_{15}[4] = 30$, $FT_{15}[5] = 37$, and $FT_{15}[6] = 52$.

   Table FT_{52} is given by: $FT_{52}[1] = FT_{52}[2] = FT_{52}[3] = 59$, $FT_{52}[4] = 3$, $FT_{52}[5] = 15$, and $FT_{52}[6] = 30$.

   (b) (1 point) Give the keys that are resolved in three or more steps starting at node 52? (You may assume that node 52 is aware that node 37 is its predecessor).
   Answer. Keys 19 up to and including 29. To be precise, for key k where $0 \leq k < 64$, routing from node 52 is as follows:

0 steps, for $37 < k \leq 52$, because the resources with these keys are hosted on node 52, and it is assumed that node 52 is aware that node 37 is its predecessor.
1 step, for $53 < k \leq 59$, because the corresponding resources are hosted on node 59, which is reached in 1 step
2 steps, for $59 < k < 3$, because the corresponding resources are hosted on node 3 and routed via node 59.
1 step, for $k = 3$, because the corresponding resource is hosted on node 3, which is reached in 1 step.
2 steps, for $3 < k < 15$, because the corresponding resources are hosted on node 15 and routed via node 3.
1 step, for $k = 15$, because the corresponding resource is hosted on node 15, which is reached in 1 step.
2 steps, for $15 < k \leq 18$, because the corresponding resources are hosted on node 18 and routed via node 15.
3 steps, for $18 < k < 30$, because the corresponding resources are hosted on node 30 and routed via nodes 15 and 18.
1 step, for $k = 30$, because the corresponding resource is hosted on node 30, which is reached in 1 step.
2 steps, for $30 < k \leq 37$, because the corresponding resources are hosted on node 37 and routed via node 30.

6. Consistency models can be divided into data-centric and client-centric models

   (a) (0,5 point) Name at least three client-centric models.
   Answer. The four client-centric models mentioned in TvS (Section 7.3) are: *Monotonic Reads, Monotonic Writes, Read your Writes* and *Writes Follow Reads*

   (b) (1.5 point) For each model, give its definition and a scenario where it is desired.
   Answer. For precise definitions see TvS (Section 7.3). The corresponding scenario's are:
   Monotonic Read is desirable for mobile readers that read data, e.g. e-mail, from successive distinct locations. Data seen the first time will then also be seen the second time.
   Monotonic Write is desirable, e.g., when updating a library (at distinct locations) by the same writer. It ensures that the updates are performed in the order as issued, ensuring that the result is always the most recent version.
   Read Your Writes is desirable when updating documents (hosted) at one location and subsequently inspecting them at another location. Thus, the writer always sees the effect of his latest update. An example would be updating a web-page on a remote server and subsequently inspecting it through the local browser. (Note that to obtain Read Your Writes, the page entry in browser

cache must be marked stale in time)
Writes Follow Reads is desirable in case of reading items from a topic-oriented medium such as a news group. It ensures that references in a more recent item to earlier items can be understood, because these are also present in the (consulted replica of) the news group.

As an alternative to the verbally described scenarios above, sequence diagrams, as presented on slides 34 – 43 of the slide set on replication, are also acceptable as an answer.

7. Reliability is an important quality for systems that need to be dependable.

   (a) (0,5 points) Give a (quantitative) definition of reliability.

   Answer. Reliability is the length of time a service may be expected to function properly, i.e., provide its functionality at the contractually agreed quality level. It is expressed as the mean time to failure (MTTF).

   (b) (1.5 point) Give three examples (of distinct flavor) used to achieve reliability.

   Answer.

   i. Usage of error-correcting codes to repair stored or transmitted data that has been corrupted.
   ii. Usage of a spare servers to replace crashed ones.
   iii. Retransmission of idempotent requests by clients after time-out to mask request omissions by servers.

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

   (a) (0.5 point) mDNS + DNS-SD can be used to discover all wheather (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.

   (b) (0.5 point) RPCs can be used for eventing.
   Answer. True. In particular, asynchronous RPCs can be used for notification, as is done, e.g., in OSAS.

   (c) (0.5 point) TCP hand-off is necessary to achieve load balancing.
   Answer. False. TCP hand-off is involved when providing load balancing at the transport level for a cluster of servers that share a single access point. However, load balancing can also be done by handling requests at the application level, in which case TCP hand-off is not involved.

   (d) (0.5 point) Architectural Description Languages (ADLs) can be used for simulation.

Answer. Not really, so false. Although some ADLs, such as AADL, offer support for analysis, they usually do not support the executable models necessary for simulation. Sometimes, however, extensions and additional tooling is developed to make simulation possible.

9. Often, replication is applied in a system architecture to meet quality requirements.

   - (1 point) Name four quality drivers for the usage of replication.

     Answer. Four important drivers are: *availability, reliability, performance and scalability*

   - (1 point) For each of the named qualities indicate a concrete context in which it is used, and explicitly identify what is replicated.

     Answer.

     availability and reliability. Redundancy allows switching to a backup server in case the primary server fails. This increases both availability and reliability. In this case both the primary server and its state need to be replicated. Another reason to temporarily switch to a backup server is to do periodic maintenance or update servers. So maintainability, updatability and serviceability can also be seen as a quality driver.

     performance. By replication of processing elements and distributing, but not necessarily replicating, the data to be processed, performance, in particular throughput, is increased by concurrency. As another example, sending a query for domain name resolution to the geographically nearest DNS root server (replica), diminishes the round trip communication time and therefore response time.

     scalability. In any P2P system, each peer, in its role as client, increases system load, but also, in its role as server, provides resources. Hence the balance between system load and system capacity (i.e., amount of resources) is maintained, and thus scalability is realized.

10. (2 points) An important issue in the architecture of the Open Service Architecture for Sensors (OSAS) is the program life cycle.

    (a) (0.5 point) Describe the translation chain from "network program"to "runnable". Give both the translation methods and the (intermediate) program forms.

    (b) (0.5 point) Discuss the places in this chain where configuration information can be incorporated.

    (c) (1.0 point) Identify the three kinds of (intermediate) code in this chain that are eligible for deployment. Discuss the pros and cons for each deployment option.

    Answer. See slide 31 of the slide set on OSAS.