TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Wiskunde en Informatica

Examination Architecture of Distributed Systems (2II45),
on Wednesday January 21, 2015, 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 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 styles.

2. (2 points) Describe the problem of service discovery and discuss its solutions.

   Answer. Service discovery addresses the problem how parties (clients and servers) that are unaware of each other's location or even existence can establish interaction. Key activities in this process are: service advertisement, service requests and the matching of the two. Solutions to this problem can be distinguished into two categories: using a mediator (broker, repository) or immediate. In the first case, there is a central party known to both clients and servers, where servers register the services they provide and where clients query for services they require. In immediate discovery, either service advertisements or service queries or both are broadcasted over a (local) broadcast domain. Clients may store advertisements for later usage, or servers may listen for and respond to broadcasted service requests.

3. There exist various primary-based protocols for (data) consistency.

   (a) (1 point) Describe the local-read, primary-write variant (aka as primary-backup protocol). Illustrate your discussion with a diagram.

   Answer. See TvS, Section Remote-Write protocols, pp. 308-309. Your answer must discuss the order in which the communications between the primary and backup servers take place. In particular, it should be clear that a write operation blocks until replicas have updated their value.

   (b) (0.5 point) What type of consistency is obtained with this protocol?

   Answer. Sequential consistency. Operations originating from the same client, which arrive at the same replica, are executed in the order issued by the client.

   (c) (0.5 point) In general, a read operation on the store need not provide the most recent version of a data object. For the primary-backup protocol, how many versions can a read operation lack behind in the worst case?

   Answer. The version read from a local replica can be at most 1 version behind, because the write operations are sequentialized by the primary. Hence all replicas are updated before the next write can start.

4. (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 31 of the introductory slide set.

5. Consider the Chord scheme for DHTs. Assume a 5-bit identifier space, and assume that the node set N is given by $id(N) = \{7, 15, 31\}$.

   (a) (1 point) Give the finger tables of all nodes.

   Answer.

   i. $FT_7[1] = FT_7[2] = FT_7[3] = FT_7[4] = 15$, and $FT_7[5] = 31$.
   ii. $FT_{15}[1] = FT_{15}[2] = FT_{15}[3] = FT_{15}[4] = FT_{15}[5] = 31$.
   iii. $FT_{31}[1] = FT_{31}[2] = FT_{31}[3] = FT_{31}[4] = 7$, and $FT_{31}[5] = 15$.

   (b) (1 point) Compute the average number of steps to resolve a key (where the average is taken over all keys and all nodes).

   Answer. For node 7 we have: 0 steps for keys 0–7, 1 step for keys 8–15, 31, and 2 steps for keys 16–30. So, on average, $\frac{8\cdot 0+9\cdot 1+15\cdot 2}{32} = \frac{39}{32}$ steps. By similar analysis, we find for node 15, the average of $\frac{8\cdot 0+16\cdot 1+8\cdot 2}{32} = \frac{32}{32}$ steps, and for node 31, the average of $\frac{16\cdot 0+9\cdot 1+7\cdot 2}{32} = \frac{23}{32}$ steps.
   Averaging once more over all nodes, we find a total average of $\frac{39+32+23}{3\cdot 32} = \frac{94}{96}$.

6. (1 point) Explain the difference between remote procedure calls (RPCs) and remote method invocations (RMIs).

   Answer. Remote method invocations are associated with distributed objects, i.e. objects whose interfaces are made available on a machine other than the one at which the object's state and code is located. As such, this is not different from remote procedures. However, middleware implementing distributed objects, keeps track of object references and offers services for locating these objects. As a consequence, since method parameters are objects, RMIs also support call-by-reference, whereas RPCs only support call-by-value.

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

   (a) (0.5 point) Give a (quantitative) definition of availability.

   Answer. Availability addresses the issue of finding the system ready for correct service. It is expressed by the quantity $\frac{MTTF}{MTTF+MTTR}$, where $MTTF$ is the mean time to failure and $MTTF$ is the mean time to repair.

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

   Answer.

   i. Use hardware/software redundancy to mask faults, e.g., triple modular redundance.
   ii. Use redundant bits (error codes) to check and repair mutilated data.
   iii. Use checkpointing during operation and, upon detection of an error, rollback to a previous state without error (consistent cut).

iv. Use a well-established coding techniques and patterns during development to prevent faults (typed languages, transactions, etc.)

See slides 32-36 of the slideset on quality attributes for more examples.

8. (2 points) Give the definition of a component as used within component-based software engineering. Name three reasons for using this engineering style.

   Answer. Following the definition by Szyperski: "A software component is a unit of composition with contractually specified interfaces and explicit context dependencies, i.e., no dependencies other than through interfaces. Moreover, a software component is independently deployable and subject to composition by third parties".
   Reasons for using this style can be found on slides 5 and 6 of the slideset on CBSE.

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

   (a) (0.5 point) For software aboard of a spacecraft reliability is a more important quality attribute than availability.

   Answer. True.
   Reliability addresses continuity of correct service, whereas availability addresses readiness for correct service. In circumstances where system repair is impossible or extremely difficult, as in spacecrafts, failure usually amounts to loss of the system. Reliability, which postpones failure, is therefore the more important quality attribute.

   (b) (0.5 point) Scalability is determined by the process view of an architecture description.

   Answer. False.
   Also the process (deployment) view covers important aspects of scalability.

   (c) (0.5 point) The Publish & Subscribe architectural style provides both temporal and referential decoupling.

   Answer. False.
   According to TvS page 591 "Most publish/subscribe systems require that communicating processes are active at the same time, hence there is a temporal coupling". However, systems that use *brokers* and come with some form of persistent memory for messages, also provide temporal decoupling. So under this provision the answer "True", will also be accepted.

   (d) (0.5 point) Using a push-based protocol between a client cache and origin server increases client response time.

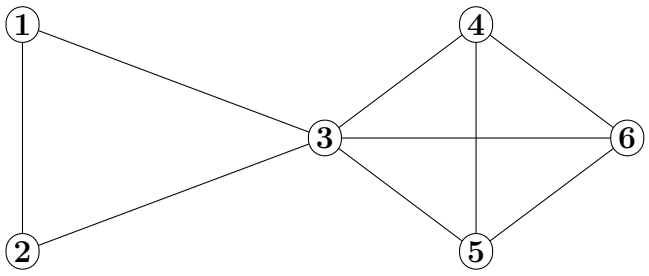
   Answer. False.
   When the server is pushing fresh data to the client immediately when it becomes available, the client is less likely to find a stale cache item. Hence its response time improves (on average).

   (e) (0.5 point) Each DNS zone is served by a single (physical) name server.

   Answer. False.

First of all, in the global layer, there are many root servers. But also, at the administrational level, e.g. the Dept. of Math. and Comp. Sc. of the TU/e, there is a primary and a secondary name server.

10. Consider a replicated distributed data store with 6 replicas each managed by an individual replica manager (RM). For intra data store communication, the RMs are connected according to the following network topology.



Clients access the data store via an RM. Each client always contacts the same RM, but distinct clients may contact different RMs. To increase availability under network partitions, while maintaining eventual consistency, the data store uses Gifford's quorum protocol, with read quorum N_R and write quorum N_W.

(a) (1.5 point) Explain how Gifford's quorum protocol works and indicate the constraints that need to be imposed on the quorum values 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 N_R and for write operations by N_W. Upon writing, the data object is updated in all replicas of the 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 replica that holds the most recent time stamp. For a data store of N replicas, the quorum sizes need to satisfy two constraints:

i. $N_R + N_W > N$, to prevent read-write conflicts,
ii. $N_W > N/2$, to prevent write-write conflicts.

(b) 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, and at any moment in time at most one RM is crashed. Consider the quorum-value pairs $(N_R, N_W) = (3, 4)$, $(N_R, N_W) = (2, 5)$, and $(N_R, N_W) = (1, 6)$. Show that

- (0.5 point) for read operations, quorum-value pair $(N_R, N_W) = (3, 4)$ performs worse than the other two.

  Answer. If an RM other than 3 is crashed, a connected data store with

5 RMs is left. Since $N_R <= 5$ in all pairs, all clients (that access a non-crashed RM) can obtain a read quorum. If, on the other hand, RM 3 is crashed, then the store breaks up into two parts containing 2 respectively 3 RMs. For pairs with $N_R \leq 2$, again all clients can obtain a read quorum, but in case $N_R = 3$, only clients that access an RM in the largest partition can obtain a read quorum. Hence, the availability of the data store is less for pair $(N_R, N_W) = (3, 4)$ than for the other two.

- (0.5 point) for write operations, quorum-value pair $(N_R, N_W) = (1, 6)$ performs worse than the other two.

  Answer. If $N_W = 6$, then no client can obtain a write quorum, irrespective which RM has crashed. If $4 \leq N_W < 6$, then all clients can obtain a write quorum, unless RM 3 is crashed, in which case no client can obtain a write quorum, since the largest partition only has 3 RMs.

Hint. Distinguish between RM 3 is crashed and another RM is crashed.