Tolerância a Faltas Distribuída

2016/17

MEI/MI/MSI

Special Exam (Época Especial)

July 21, 2017

Total time: 2:30hs

Number:______ Name: ____________________________________

Instructions (please read and follow carefully):

1. This is a closed-book, closed-notes exam.
2. **Be brief** and precise in your answers. You may be penalized for unnecessarily long answers. *Hint:* Only use the space provided. Condense your answer into its key points – avoid writing long essays! Please think before you answer.
3. The total number of points is 20.
4. Some questions may have more than one correct answer.
5. Do not spend too much time on any one question. There are some simple questions that you can answer quickly. Come back to questions you cannot answer later if necessary.
6. Try to answer every question (briefly) so as to accumulate at least partial credit.
7. A blank page is attached at the end for your use as scratch paper.

For Graders' Use Only

I. ___________ / 6

II. ___________ / 14

TOTAL ___________ / 20

I. Multiple-choice questions (6 points)

For each question there is only one correct answer. Please circle the correct answer.

(Grading: a correct answer is awarded 0,6 points; an incorrect answer is awarded -0,15 points).

1. The Transmission Control Protocol (TCP) includes mechanisms for segment retransmission (to ensure reliable message transmission). These mechanisms provide fault-tolerance regarding:

   (A) Crash faults.
   (B) Omission faults.
   (C) Semantic faults.
   (D) Byzantine faults.

2. When considering the **Partially Synchronous** distributed system model (with Global Stabilization Time – GST), this means that:

   (A) Systems developed on top of this model will execute slowly.
   (B) It is possible to develop a *Perfect Failure Detector* (FD of class *P*) over this model.
   (C) Communication delays are always bounded, but the bound is unknown.
   (D) The system will be synchronous with a probability of more than 99%.

3. The assumption coverage when considering the asynchronous system model is:

   (A) Higher than when considering the synchronous system model.
   (B) Lower than when considering the synchronous system model.
   (C) The same as when considering the synchronous system model.
   (D) Zero, because the asynchronous model does not make assumptions.

4. A message delivered using a reliable causal broadcast primitive also satisfies:

   (A) Total order delivery.
   (B) FIFO order delivery.
   (C) Both total and FIFO order delivery.
   (D) None of them.

5. The properties of atomic (total order) broadcast (in asynchronous systems with crash failures) ensure that:

   (A) If a correct process delivers message M, then the sender of M must be a correct process.
   (B) If a correct process delivers M1 before M2 then all correct processes deliver M1 before M2.
   (C) If a correct process broadcasts M1 before M2 then all correct processes deliver M1 before M2.
   (D) All of the above.

6. When using a *Perfect Failure Detector* (of class *P*), it is guaranteed that:

   (A) Eventually, all correct processes permanently suspect all crashed processes.
   (B) If a process crashes then it will be immediately suspected by all correct processes.
   (C) If a correct process is suspected by mistake, the mistake will be eventually corrected.
   (D) Eventually, all correct processes will never become suspected.

7. One advantage of passive replication in comparison to active replication is that:

   (A) It is the only that can be used to tolerate value faults.
   (B) It can be used to replicate non-deterministic services.
   (C) It only needs $f+1$ replicas to tolerate f crash faults.
   (D) It has a short takeover-glitch.

8. An **atomic register** R initially holding value 33 is used by two process P and Q that perform the following operations: P executes A=read(R) during time interval [2,5] and B=read(R) during interval [6,8]; Q executes write(R,78) during time interval [4,9] (the operation overlaps in time with both reads performed by process P). In this situation, since the register provides the atomic semantics, it is guaranteed that:

   (A) The value of A will be 33.
   (B) The value of B will be 78.
   (C) If the value of A is 33 then the value of B will also be 33.
   (D) If the value of A is 78 then the value of B will also be 78.

9. A queue has consensus number:

   (A) 1.
   (B) 2.
   (C) 3.
   (D) Infinite.

10. The implementation of a durable replicated state machine involves the use of several durability techniques that can affect the performance of the system. One of such techniques is coordinated checkpointing, which fundamentally degrades the performance of the system because:

    (A) It involves the write of the whole service state to disk, which takes time due to the *bad throughput* of such devices.
    (B) It requires writing to the disk, which is slow due to the *high access latency* of such devices.
    (C) When several replicas stop to take the checkpoint at the same time, the system can't process requests and clients have to wait.
    (D) Checkpoints need to be taken frequently to avoid replaying a large log after a crash.

II. General questions (14 points)

1. [1,5] Consider the following implementation of a failure detector.

```
output_p ← ∅
for all q ∈ Π                          {Δ_p(q) denotes the duration of p's time-out interval for q}
    Δ_p(q) ← default time-out interval

cobegin
|| Task 1: repeat periodically
    send "p-is-alive" to all

|| Task 2: repeat periodically
    for all q ∈ Π
        if q ∉ output_p and
            p did not receive "q-is-alive" during the last Δ_p(q) ticks of p's clock
            output_p ← output_p ∪ {q}          {p times-out on q: it now suspects q has crashed}

|| Task 3: when receive "q-is-alive" for some q
    if q ∈ output_p                    {p knows that it prematurely timed-out on q}
        output_p ← output_p − {q}              {1. p repents on q, and}
        Δ_p(q) ← Δ_p(q) + 1                    {2. p increases its time-out period for q}
coend
```

Explain why does it implement an eventually strong failure detector ($\diamond S$) in the partially synchronous system model considered (no omissions in the network).

2. [1,5] Consider the following system model: asynchronous with crash faults. Describe in your own words an algorithm **that uses Consensus and Reliable Broadcast** as building blocks to implement Atomic (total order) Broadcast.

3. [1,5] How the Synod protocol used in the Paxos algorithm ensures that no two different operations are assigned the same sequence number for execution (i.e., same slot) even in executions with two active leaders?

4. [1,5] Consider a replicated service that becomes partitioned, leading to a situation in which there will be some clients and some replicas in each partition. Describe two possible solutions that can be used to avoid state divergence of the replicas in each partition.

5. Regarding RAID systems. **(a)** [1] What is the difference between RAID0 and RAID1? **(b)** [1,5] Let each disk of the array be able to process *w* 4kB-writes/sec and *r* 4kB-reads/sec. Fill the table bellow with the capacity in terms of number of (4kB) writes and reads per second of an array with *n* disks in these two RAID models (in function of *w*, *r* and *n*).

	RAID 0	RAID 1
WRITES		
READ		

6. [1,5] Considering concurrent objects, answer the following questions.
   a. What does it mean to say that a concurrent object is linearizable and wait-free?

   b. Is there a wait-free implementation of a *compare&swap* object by a *test&set* object? Justify.

   c. Can you implement an n-processes consensus in an asynchronous system using an augmented tuple space (as used in DepSpace)? If yes, sketch the algorithm, if no, justify.

7. [1] Zookeeper is a popular coordination framework that provides shared memory objects.
   a) Explain how is it possible to use Zookeeper to elect a leader. In your answer you do not need to provide explicit code, but you must describe the main steps of the algorithm that is executed by a process when electing the leader.

   b) How is it possible to detect that the current leader has failed in Zookeeper?

8. [1] Assume a storage system in which each data to be written is broken in k=3 data blocks of w=8 bits (values between 0 and 255) and m=2 coding blocks (tolerating two failures). Describe two equations to generate the coding blocks of this scheme.

9. [2] Consider the RAFT replication protocol, answer the following questions:

a.) During leader election, how the algorithm ensures the elected leader will never have to update itself using data from other replicas (i.e., how it is ensured information flows only from the leader to followers)?

b.) Describe one or two techniques to make a RAFT implementation minimize the performance overhead of writing log entries to disk.

c.) How can one modify the RAFT algorithm to make it implement active replication?

BLANK PAGE