Team2. Hyunsuk Lim(hyunsukl@andrew.cmu.edu)

**LG Software Architectures Training Program: Assignment A3: Performance, Modifiability, and Availability**

For the scenarios below, identify two tactics/patterns that could be used to support the given requirement. For each selected tactic/pattern, explain how it would support the scenario and identify any gaps in support that *could* exist (15 pts each).

1. The system experiences a server failure under normal operating conditions, the system is able to continue to operate without losing any requests and without degrading average case response time
2. Warm standby: We can use this pattern to support that scenarios. But it needs to be considered that failover time might take too long which can makes losing request.
3. Hot standby: If this system doesn’t manage any state (stateless instance), then we can use active redundancy without degrading average case response time. It also ensures that the system works without any loss of requests.
4. Data is corrupted due to a disc error. The system is able to detect the corruption and recover the data without losing any data.
5. Checksum: We can detect that corruption using checksum tactic. Although the checksum reveals the data in which the corruption occurred, the corruption may also appear in the checksum itself. We also need additional resources to calculate and store the checksum data. A checksum cannot recover data, so it needs a way to recover it.
6. Redundant spare (Hot spare): We can recover the data using one of redundant spare tactics. In this scenario, we can choose ‘Hot spare’ to recover the data without losing any data. This makes development and maintenance more expensive including system resources but is applicable in systems where data’s reliability is important.
7. If a node becomes slow to respond under normal operating conditions, the system will continue to operate without the response time increasing more than the expected average case response time.
8. Voting: This can be used to detect the faulty node by checking respond time for each node and comparing result with other nodes. However, if this is a problem with the algorithm implemented by the node, errors cannot be detected through simple voting. But we can overcome that by using a different algorithm implementation for each node.
9. Hot standby: We can use active redundancy strategy to replace the node responding slowly. But it may increase average case response time for synchronization if node has many data to synchronize.
10. Identify all of the information you’d need in order to determine the average wait time for Task A from the system shown below (15 pts):
    1. Expected load for process A, E, B, D.
    2. Execution time for process A, E, B, D.
    3. Scheduling strategy for queue on the process A, E, B, D.
    4. Execution time for Database.
    5. RPC processing time between process A and E, E and Database, Database and E, D and B.

Legend

Database

Process

Task A

RPC

Task B

Queue

C

B

A

E

D

1. For the system shown in the diagram above, identify at least 6 specific faults that could disrupt the operation of Task A (15 pts).
2. Process A crashed
3. Process B hung
4. Data corruption in Database
5. Queue buffer full issue from queue of process E
6. A race condition with Task B occurred in Database
7. A race condition with Task B occurred in Process D
8. Transaction fails in Database related with Task A
9. Write a well formed, actionable availability scenario based on one of the faults from the previous question (10 pts).

When Process E tries to read a configuration file that has corruption in the database, it should be able to detect it and replace it with a backup configuration file within 1ms.

**Grading:**

15 points overall depth and quality

85 points for the questions (as marked)

**Submission Instructions:** Name your file *A3andrewID* (e.g. A3mbass.docx). Upload your answers in .doc(x) or pdf format to canvas prior to the due date/time. If you’re unable to upload your document to canvas, you can email your submission prior to the due date/time to all of the following: Paulo Merson ([pmerson@andrew.cmu.edu](mailto:pmerson@andrew.cmu.edu)), Mona Elsing ([mona.elsing@gmail.com](mailto:mona.elsing@gmail.com)), and Matt Bass ([mbass@andrew.cmu.edu](mailto:mbass@andrew.cmu.edu)).