TSR / NIST – Second Partial

This exam consists of 20 multiple choice questions. In every case only one answer is correct. You should answer in a separate sheet. If correctly answered, they contribute 0.5 points to the exam grade. If incorrectly answered, the contribution is negative: -0.167. So, think carefully your answers.

THEORY

1. In the scope of deployment configuration, this is an example of "dependency injection":

a	To throw an exception when the port to be used is already assigned to another (i.e., to a different and already running) server.
b	To start another server replica when the workload is increasing; i.e., to manage a scale-out action.
c	To dynamically learn (or even update) which is the endpoint of another service component that will be used later on.
d	Some (optional) component of the service is not deployed (i.e., it has no deployed instance).

2. Which of the following tasks IS NOT a deployment-related task:

a	Component installation in the target hosts.
b	Application debugging.
c	Dependency resolution.
d	Software upgrading.

3. Let us imagine that we have written an algorithm AL1 in order to reach consensus in a distributed system assuming the Byzantine (or arbitrary) failure model. Then:

a	AL1 is useless, since no real system behaves in that way (i.e., in a way that matches the Byzantine failure model).
b	AL1 needs to replicate all its processes using the passive model, since the Byzantine failure model can only be managed using that replication model.
c	AL1 is easier to port to a real system than another algorithm AL2 that assumes the stop failure model, since the assumptions of AL2 are hard to reach in real systems.
d	AL1 will be one of the simplest algorithms, since the Byzantine failure model simplifies failure handling in the algorithms that assume it.

4. The active replication model...

a	…may handle replica failures faster than the passive replication model.
b	…is unable to handle the Byzantine failure model.
c	…cannot ensure sequential consistency.
d	…is used by default in the MongoDB replica sets.

TSR / NIST – Second Partial

5. Considering the constraints of the CAP theorem, if we manage network partitions using the primary partition model then...

a	…sequential consistency may be ensured in minor process subgroups.
b	…sequential consistency may be ensured in the process subgroup with a majority of processes.
c	…availability is ensured, and replica consistency is relaxed.
d	…availability is lost in the major process subgroup.

6. Considering the constraints of the CAP theorem, if we must ensure availability in a service that is deployed in multiple datacentres where network partitioning should be tolerated, then…

a	…we may use sequential consistency among all service replicas.
b	…we should use a relaxed consistency model; e.g., FIFO consistency.
c	...there is no constraint on replica consistency.
d	...service availability cannot be ensured in a scenario like that, consisting of multiple datacentres geographically dispersed.

7. Which of these services is the most elastic?

a	One that adds a replica per hour, independently on the workload at that moment.
b	One that reports its currently supported workload, remaining capacity and amount of replicas to the system administrator, who decides the scaling action to be applied.
c	One that autonomously adapts its number of replicas to the current workload, maintaining an appropriate quality of service (as stated in its SLA).
d	One that never monitors its resource usage.

8. Which of these approaches DOESN'T increase the scalability of a service?

a	Task distribution..
b	Data distribution.
c	To take decisions using voting techniques among all participating processes.
d	To use caches.

TSR / NIST – Second Partial

SEMINARS

9. Which of these characteristics distinguishes virtual machines from Docker containers?

a	Some applications to be run on them may require libraries that are not present in the host operating system.
b	The need of a host operating system.
c	The isolation from other containers and/or virtual machines.
d	Virtual machines can run applications that cannot be run in the host operating system.

10. With the "docker commit" command we can...

a	End a pull- or push-repository transaction. In those transactions, a Docker image is either downloaded from or uploaded to a repository.
b	Create a Docker image, taking a Dockerfile as its base.
c	Start a Docker container, taking a Docker image as its base.
d	Create a Docker image, taking a Docker container as its base.

11. Let us assume that this Dockerfile is correct (i.e., no error arises when it is processed):

```
FROM zmq
COPY ./myProgram.js /server.js
EXPOSE 8000 8001
CMD node /server.js
```

Which of the following sentences is FALSE?

a	There is a "zmq" Docker image, either in the local or the global repository.
b	The resulting components will run a "server" process bound to port 8001 in its container. Such a port corresponds to port 8000 in its host.
c	The image to be generated from this Dockerfile may run other programs besides "node server.js".
d	When an image is generated from this Dockerfile, a file called "myProgram.js" should be in the folder where such Dockerfile is placed.

TSR / NIST – Second Partial

12. Which of the following sentences is FALSE about the docker-compose command?

a	It assumes that a docker-compose.yml file exists in the folder where that command is run.
b	It always needs that a Dockerfile exists for each one of the components to be run.
c	It may be used for controlling a service based on multiple components. All instances for all those components are run in the same host.
d	Its "scale" action may be used for changing the amount of instances of a given component in the service.

13. Which of these sentences is FALSE about "fast" consistency models?

a	When a read or write action is run by a process, that action may return control to its invoker without exchanging any message with other processes.
b	FIFO is a fast consistency model.
c	Causal is a fast consistency model.
d	Sequential is a fast consistency model.

14. In order to implement the FIFO consistency model (using ZMQ on TCP), we need:

a	A sequencer process that implements a total order.
b	That each process multicasts its writes using a PUB socket and receives the writes from all other processes using a single SUB socket.
c	That each process multicasts its writes using a PUB socket per each variable it is able to write and a single SUB socket for receiving the writes from all other processes.
d	To use vector clocks and add the current vector clock to each written value when it is multicast to the other processes.

15. The "cluster" module in NodeJS…

a	…makes possible that a set of containers is deployed in more than one host computer.
b	…extends docker-compose commands for deploying NodeJS programs faster than without such "cluster" module.
c	…makes possible that NodeJS programs be multi-threaded.
d	…makes possible that a set of worker NodeJS processes share the ports that have been bound by the master NodeJS process.

TSR / NIST – Second Partial

16. We are writing a NodeJS program with the 'cluster' module. It uses as many workers as processors and the master process reports every second to the user how many requests have been served up to now by all workers.

```
var cluster = require('cluster');
var http = require('http');
if (cluster.isMaster) {
  var numReqs = 0;
  setInterval(function(){ console.log("numReqs =", numReqs);}, 1000);
  function messageHandler(msg) {
    numReqs++;
  }
  var numCPUs = require('os').cpus().length;
  for (var i=0; i < numCPUs; i++) cluster.fork();
  /* (1) Worker message management instructions should be here. */
  /* (2) Worker regeneration code should be here, if any.       */
} else {
  http.Server(function(req, res) {
    res.writeHead(200); res.end('hello world\n');
    process.send({ cmd: 'notify' });
  }).listen(8000);
}
```

Choose which instructions are needed by the master process to manage (1) the worker messages:

a	`cluster.on('message', messageHandler);`
b	`for (var i in cluster.workers)` `  cluster.workers[i].on('message', messageHandler);`
c	`for (var i in workers)` `  workers[i].on('notify', messageHandler);`
d	`cluster.on('notify', messageHandler);`

17. In the program shown in question 16, we want to keep the amount of workers constant. To this end, if any of them dies, it should be replaced by a new worker. Which are the instructions (2) that implement such a functionality?

a	`cluster.on('exit', function(worker, code, signal) {`
	`  cluster.fork();`
	`});`

b	`cluster.on('exit', function(worker, code, signal) {`
	`  worker.restart();`
	`});`

c	`cluster.on('exit', function(worker, code, signal) {`
	`  worker.process.fork();`
	`});`

d	`cluster.on('exit', function(worker, code, signal) {`
	`  worker.process.start();`
	`});`

TSR / NIST – Second Partial

18. MongoDB uses the following scalability mechanisms:

a	Active (or state-machine) replication and task distribution.
b	Primary-backup replication and data distribution (horizontal partitioning).
c	Active replication and data distribution (horizontal partitioning).
d	MapReduce.

19. Which statement is FALSE about "mongos" processes:

a	They are request forwarders.
b	They hold a cache of the MongoDB sharding configuration metadata.
c	They are placed in client computers.
d	They may be replaced by a "replica set".

20. Which statement is FALSE about the MongoDB "write concern":

a	It states how many replicas should persist the modifications caused by a write, delete or update operation.
b	Lower values in the "write concern" reduce the write service time, but endanger data persistency in case of failure.
c	Its recommended value is "majority", since it provides a good compromise between performance and failure tolerance.
d	It needs a confirmation from the client for every demanded operation.