A

This exam consists of 20 multiple choice questions. In every case only one of the answers is correct. You must answer in a separate sheet. If correctly answered, each question contributes 0,5 points to your grade. If the answer is wrong, the contribution is negative: -0,167 points. In the answer sheet, fill carefully your chosen slot. To this end, use a dark pen or pencil.

THEORY

1. In the deployment plan of a distributed application...

a	The plan template states how to connect the components, listing both the dependencies to be resolved and the exposed endpoints.
b	The configuration templates for each component maintain the same values for every component instance.
c	Several instances of a component may have unresolved dependencies at deployment time. They will be resolved later, while the service is running.
d	The plan template states the location (nodes) for every component instance respecting the constraint that, at each node, one and only one instance of each component must be run.

2. Considering failure models for distributed systems...

a	In both the ***stop*** and ***crash*** failure models, the failure of a process may always be detected by other processes.
b	It is easy to implement a distributed system whose failures match the ***stop*** or ***crash*** failure models.
c	A ***network partition*** model is equivalent to the ***general omission*** (i.e., send-omission and receive-omission) failure model.
d	When failure transparency is ensured assuming a ***Byzantine*** failure model, then it is also ensured for all the remaining failure models.

3. Regarding replication models...

a	The passive replication model supports the ***Byzantine*** failure model but it does not support the ***crash and link*** failure model.
b	In the passive replication model, update propagation in total order is very difficult to implement.
c	In the active replication model, reconfiguration after a replica failure is a very difficult procedure.
d	The active replication model demands more replicas for supporting ***Byzantine*** failures than for supporting ***crash*** failures.

4. NoSQL datastores are easily scalable because...

a	They guarantee referential integrity.
b	They reduce the redundancy of the stored data.
c	They simplify or eliminate transactions.
d	They guarantee that network partitions never happen.

5. According to the CAP theorem, it is possible that when a network partition arises...

a	...all system subgroups go on, answering client requests, and replicated services will ensure at least sequential consistency.
b	...a partitionable model is assumed and replicated services ensure eventual consistency.
c	...the primary component model must be assumed and every service replica must answer client requests.
d	...all service replicas must be stopped until the network connectivity is repaired. No service activity is tolerated in that interval.

6. Contention (i.e., performance bottlenecks that compromise service scalability) may be caused by...

a	The usage of decentralised algorithms for managing heavy tasks.
b	A wrong distribution of the resouces that originates a dense network traffic.
c	The usage of an asynchronous communication middleware.
d	The replication of the components that balance the load among the worker processes.

7. General principles for achieving scalability:

a	To increase the concurrency degree without bound.
b	To reach a strong consistency.
c	To avoid inter-agent synchronisation as much as possible.
d	To save all service data in secondary storage in order to ensure its persistency.

8. Peter is a security expert that works in company B. He used yesterday a packet sniffer and obtained the ID and password of a system administrator from company A. He also found the public address of one of the A's servers where company administrators may remotely control the other servers in the company. For company A, the current scenario is an example of...

a	...a denial of service attack.
b	...a potential external threat.
c	...a weakness in the routing protocols.
d	...a physical security mechanism.

SEMINARS

9. Given this sequence of Docker commands executed from the CLI:

```
docker pull fedora
docker run --name fedora fedora dnf install –y nodejs
docker commit fedora node
docker push node
```

Which of the following actions hasn't been done?

a	Download an image (*fedora*) from the public repository (i.e., from the Docker Hub).
b	Upload an image (*node*) to the Docker Hub.
c	Create a container and run the *node* command in it.
d	Create a container, modify it and create a new image from that container.

10. Considering this Dockerfile:

```
FROM zmq
RUN mkdir /zmq
COPY ./worker.js /zmq/worker.js
WORKDIR /zmq
CMD node worker $BROKER_PORT_8001_TCP
```

If we generate a Docker image from it using the following command:

```
docker build –t worker .
```

Which is the FALSE sentence?

a	The *worker* image is a modification of the *zmq* image.
b	The working directory for the CMD command is */zmq.*
c	If a container is created from the *worker* image, it will run the programme cited in the CMD command.
d	If a container is created from the *worker* image, it won't run any programme since programmes should be specified using ENTRYPOINT instead of CMD.

11. Assume that we have a default Docker installation in our computer where we have created a "node2" image able to run the node interpreter from the command line (i.e., "node2" is an image, not a container). Imagine that we want to execute a node.js programme that we have in our computer (e.g., the file "/tmp/example.js") in a Docker container. To this end, among other actions, we should...

a	Use **docker run node2** from the command line, passing the programme pathname as its last argument; i.e. **docker run node2 /tmp/example.js**
b	Nothing can be done since the files in the host computer cannot be used by a running container and there is no way to transfer those files to an image.
c	Copy that file into a new image based on the **node2** one. To this end we may use the COPY command in a Dockerfile.
d	Use **docker cp /tmp/example.js node2**.

12. Considering this Dockerfile...

```
FROM fedora
RUN dnf install -y nodejs
RUN dnf install -y zeromq-devel
RUN dnf install -y npm
RUN dnf install -y make
RUN npm install zmq
```

The following sentences are true:

a	This Dockerfile doesn't make sense since it has no ENTRYPOINT or CMD commands in it. It does nothing at all.
b	This Dockerfile doesn't work since it fails in its second line. There is no "dnf" instruction in Docker.
c	The name of the image being created with this Dockerfile is "zmq".
d	This Dockerfile creates a new image based on the "fedora" one. The new image has added at least 4 fedora packages onto the base "fedora" image.

13. The implementation of a weak consistency model (Activity 1 from Seminar 5, source files: shared1.js and proc1.js) using a PUB/SUB communication pattern guarantees the following consistency model:

a	Sequential since the writes of every process are immediately forwarded to other processes using a multicast from the PUB socket.
b	Causal since every process reads the values written by the remaining processes before starting its own writes on the shared variable.
c	Cache since the subscription to the writes on each variable guarantees that all processes read the same sequence of values from each variable.
d	FIFO since each process propagates its writes in order using its PUB socket and the protocol being used (TCP) respects FIFO order.

14. Considering the implementation of a replication protocol based on a sequencer process, as that described in the bulletin from Seminar 5...

a	When we run the sequencer process in the fastest node of our system, the resulting consistency model is fast.
b	If we use a different sequencer process for each variable, we will provide cache consistency without ensuring sequential consistency.
c	If we use the same sequencer process for all variables, we will provide sequential consistency without ensuring cache consistency.
d	If we use the same sequencer process for all variables, we will provide sequential consistency without ensuring FIFO consistency.

15. Considering this node.js programme...

```
var cluster = require('cluster');
var http = require('http');
var numCPUs = require('os').cpus().length;
if (cluster.isMaster) {
  for (var i=0; i < numCPUs; i++) cluster.fork();
  cluster.on('exit', function(who, code, signal) {
    console.log('Process '+ who.process.pid +' died');
  });
} else {
  http.createServer(function(req, res) {
    res.writeHead(200);
    res.end('hello world\n');
  }).listen(8000);
}
```

The following sentences are true:

a	This programme fails when the second worker is created since all workers are trying to use the same port (8000) and that port is already in use.
b	The master process successfully creates as many worker processes as CPUs (or cores) exist in the local computer.
c	The "cluster" module may deploy each generated worker process in a different computer.
d	This programme prints a message when the master dies.

16. Regarding the programme shown in question 9, the following sentences are true...

a	The first worker prints a message each time it receives a message.
b	The answer being returned by the worker depends on the contents of the HTTP request sent by the client.
c	Master-worker communication has been implemented using a REQ/REP communication pattern.
d	Each worker is an HTTP server process.

17. In order to enhance its scalability, MongoDB uses...

a	...the active replication model.
b	...a partitionable model for dealing with network partitions.
c	...database sharding, combined with primary-backup replication.
d	...transactions that respect the four ACID properties.

18. When a MongoDB database is partitioned, all its shards need to have a similar size. This is achieved using...

a	...“*journaling*”.
b	...a “chunk migration” algorithm.
c	...normalization.
d	...a mutual exclusion algorithm.

19. Regarding the thematic classification of security vulnerabilities...

a	“Phishing” attacks exploit “social engineering” vulnerabilities (i.e., the exploited vulnerabilities belong to the “social engineering” class in this case).
b	“Protocol design” vulnerabilities belong to the “social engineering” class.
c	“Internal espionage” vulnerabilities belong to the “software error” class.
d	“Personal protection” faults belong to the “software error” class.

20. The origin-based classification of vulnerabilities...

a	...identifies four vulnerability classes: social engineering, software errors, faults in security policies and general weaknesses.
b	...also considers the time needed to exploit vulnerabilities (and react to it in case of attack) and the degree of interaction demanded for that exploitation.
c	...considers that the origin of vulnerabilities will provide us with a guide to remedy them.
d	...states that the attacks are the cause (i.e., the origin) of vulnerabilities.