TSR **Second Parcial** **Units 4,5,6** **2019-01-09** **A**

This test provides 2 points (20%) of the global grade in NIST and it consists of 20 questions. Each question has 4 choices and only one of them is correct. Each correct answer provides 0.1 points and each error -0.033 points. Answers must be given in a separate answer sheet. The exercise mark is 1 point and its answer must be given in the blank sheets delivered with this wording.

1 *Let us assume a service with N replicas that follows the active replication model. Such replication introduces these benefits:*

- **a** The service capacity is N times higher.
- **b** Its consistency becomes eventual and this increases the service throughput.
- **c** A single replica failure may be easily handled.
- **d** The deployment of those N replicas is automated.

2 *The passive replication model has the following disadvantages:*

- **a** It may lose some requests in case of a primary replica failure.
- **b** It requires that all operations are deterministic.
- **c** It requires that all operations are commutative.
- **d** It does not tolerate failures in backup replicas.

3 *Which of these statements on faults and failures is TRUE?*

- **a** Replicated services never have faults or failures.
- **b** A process may fail without showing any previous fault.
- **c** A faulty process (i.e., a process with a fault) always fails subsequently.
- **d** After a fault, a failure may arise.

4 *Let E be an execution that respects the causal consistency model. Then, we may state that E also respects this other consistency model:*

- **a** FIFO.
- **b** Sequential.
- **c** Processor.
- **d** Cache.

5 *We have implemented a service with update operations that, on average, require 15 ms in order to be run and modify 100 MB of the service state. That service will be deployed on a system with a LAN of 100 Mbps and 0.5 seconds of transmission delay. What is the most convenient replication model if that service requires: (1) sequential consistency, (2) tolerance of up to two simultaneous process failures, and (3) minimal response time?*

- **a** Passive replication.
- **b** Active replication.
- **c** Multi-master replication.
- **d** No replication. A single server should be used.

6 *Which of the following statements about multi-master replication is FALSE?*

- **a** Every client may choose a different master replica for each of its requests.
- **b** It ensures sequential consistency.
- **c** The master replica may send a reply to the client before sending the updates to other replicas.
- **d** Read-only operations may be managed by a single replica.

7 *A distributed service is elastic when:*

a It is replicated and it uses eventual consistency.

b It is replicated and it uses Docker for managing its deployment.

c It is scalable and it adapts its amount of replicas to the current workload in an autonomous way.

d A fail-stop model has been assumed in the service design stage.

8 *Which of these statements about network disconnection failure management and the CAP theorem is TRUE?*

a If we use the primary partition model, then we renounce to strong consistency.

b If we use the primary partition model, then we renounce to network partition tolerance.

c If we use the partitionable model, then we renounce to availability.

d If we use the partitionable model, then we renounce to strong consistency.

9 *Which are the advantages provided by the cluster module in NodeJS?*

a Multiple NodeJS server processes may be bound to the same port.

b This module ensures server elasticity in a direct and autonomous way.

c With it, every NodeJS program may be run by multiple activities that share memory.

d This module automates the deployment of a set of NodeJS programs in a cluster of computers.

10 *How does MongoDB enhance its scalability?*

a With multi-master replication and eventual consistency.

b With sharding (i.e., data distribution) among multiple server processes.

c With a very efficient management of long transactions.

d With active replication in the `mongod` processes.

11 `EXPOSE 8000 8001` *in a* `Dockerfile` *means that…*

a Port `8000` in the container corresponds to port `8001` in the host.

b Port `8001` in the container corresponds to port `8000` in the host.

c The container exposes ports `8000` and `8001`.

d The container exposes ports `8000` and `8001` and they correspond to those same ports in the host.

12 *In order to build an image with the current contents of a container, we should use:*

a `docker build containerName imageName`

b `docker commit containerName imageName`

c `docker build -t imageName containerName`

d `docker commit imageName containerName`

13 *Let us assume that we should deploy one broker, several clients and several workers, each one in a different container and all they in the same host. The broker uses a ROUTER socket as its frontend (to interact with clients) and a DEALER socket as its backend (to interact with workers). The broker uses bind() on each socket. Each client uses a* `REQ` *socket to interact with the broker. Each worker uses a* `REP` *socket to interact with the broker. In order to deploy these components in a correct way, their appropriate deployment order is:*

- **a** Clients, broker, workers.
- **b** Broker, workers, clients.
- **c** Workers, broker, clients.
- **d** Anyone.

14 *Dependency injection is generally used when:*

- **a** A message is received using a zmq socket.
- **b** An event is processed.
- **c** Multiple instances of several cooperating and interdependent service components are deployed using docker-compose.
- **d** This command is run: `docker run -i -t tsr1718/centos-zmq bash`

15 *In regard to the* `ENTRYPOINT` *instruction in a Dockerfile, which of these statements is the most appropriate?*

- **a** There may be, at most, one instruction of that kind in the Dockerfile.
- **b** There may be more than one.
- **c** If there is a `CMD` instruction, `CMD` prevails over `ENTRYPOINT.`
- **d** When there are more than one, only the first one is considered.

16 *On the replication concept:*

- **a** Replication complicates failure recovery.
- **b** Replication is needed for providing failure transparency.
- **c** Replication improves throughput in update operations.
- **d** Replication may only be used in combination with strict consistency.

17 *On the eventual consistency model:*

- **a** It specifies that the states of replicas converge when there are long intervals without new read operations.
- **b** It is equivalent to strict consistency.
- **c** It is the model generated when FIFO and cache models are combined.
- **d** It specifies that the states of replicas converge when there are long intervals without new write operations.

18 *A distributed system execution generates the following trace* `1:1Wx 2:3Wy 3:3Ry 1:3Wx 2:4Wx 4:4Rx 1:2Wy 3:2Ry 4:3Ry 4:1Rx 3:4Rx 3:1Rx 4:3Rx 3:3Rx 4:2Ry`, *where* `i:nRv` *means that process* `i` *reads value* `n` *from variable* `v` *and* `i:nWv` *means that process* `i` *writes value* `n` *to variable* `v.` *Choose the consistency model respected by that trace:*

- **a** Cache
- **b** Causal
- **c** FIFO
- **d** Processor

19 *In regard to the tools for scaling on multiple computers:*

a Only `node-http-proxy` implements a broker-worker pattern.

b Only `HAProxy` implements a broker-worker pattern.

c Only `nginx` implements a broker-worker pattern.

d All these choices implement a broker-worker pattern.

20 *On the MongoDB configuration servers:*

a They do not maintain data or metadata: they behave as client request routers.

b They maintain metadata, but they do not maintain data.

c They are a replica-set whose w (write-concern) value is 1.

d They maintain data, but they do not maintain metadata.

NIST – 9th January 2019. EXERCISE

Let us consider these two programs. They implement a chat service:

```
// Client.js
const zmq=require('zmq')
const ps=zmq.socket('push')
const sub=zmq.socket('sub')

ps.connect('tcp://127.0.0.1:8000')
sub.connect('tcp://127.0.0.1:8001')

sub.subscribe('')
sub.on('message', (msg) => {
        console.log(msg+'')
})

let userid=process.argv[2] || 'user'+process.pid

let conMsg={type:'connect',user:userid}
ps.send(JSON.stringify(conMsg))

process.stdin.resume()
process.stdin.setEncoding('utf-8')
process.stdin.on( 'data', (line) => {
        let chatMsg = {type:'chat', user:userid, data:line}
        ps.send(JSON.stringify(chatMsg))
})

function disconnect() {
        let disMsg = {type:'bye', user:userid}
        ps.send(JSON.stringify(disMsg))
        setTimeout( process.exit,500)
}

process.on('SIGINT', disconnect )
process.stdin.on('end', disconnect )
```

```
// Server.js
const zmq=require('zmq')
const pull=zmq.socket('pull')
const pub=zmq.socket('pub')

pull.bindSync('tcp://*:8000')
pub.bindSync('tcp://*:8001')

pull.on('message', (msg) => {
    let m=JSON.parse(msg)
    let answer=''
    switch(m.type) {
        case 'connect':
            answer='Server> chat-client '+m.user+' connected!'
            break
        case 'chat':
            answer=m.user+'> '+m.data
            break
        case 'bye':
            answer='Server> chat-client '+m.user+' disconnected!'
    }
    pub.send(answer)
})
```

Please, answer the following questions related to the deployment of those elements using Docker. To this end, let us assume that the host computer has a local Docker image based on "centos:latest" with the **node** and **npm** commands, the ZeroMQ library, and the **zmq** NodeJS module properly installed. The name of that image is "**exercise02**":

1. Revise the code of both programs, specifically the URLs used in the bindSync() and connect() operations, and describe whether any change is needed in order to allow their deployment in containers. If so, write and explain the program fragment needed for appropriately dealing with those bind/connect steps (2 points).
2. Write a single Dockerfile for deploying the "Server.js" component. Assume that such a Dockerfile is in the same folder where the file "Server.js" is placed (2 points).
3. Write a command for building an image called "server" using that Dockerfile (1 point).
4. Write a command to run a container that uses the "server" image (1 point).
5. Let us assume that such server is running in a container whose IP address is 172.17.0.3. Write a single Dockerfile for deploying the "Client.js" component. It should be able to interact with that running "server" component (2 points).
6. Write and explain the commands needed to run a container that uses the "client" component generated from the image built with the Dockerfile written in part 5. Take care of the "-i" and "-t" options (1 point).
7. **Discuss** which actions are needed in some of the previous parts in order to deploy each one of those components (server and multiple clients) in different hosts. To this end, let us assume that the IP address of the host where the chat server is running is 192.168.0.10 (1 points).

NOTE: In this exercise, **docker-compose** is not needed. All parts can be solved using the **docker** command and any of its available subcommands (i.e., **build**, **run**, **commit**, **ps**, **images**, **rm**, **rmi**...)

This test provides 1 point (10%) of the global grade in NIST and it consists of 10 questions. Each question has 4 choices and only one of them is correct. Each correct answer provides 0.1 points and each error -0.033 points. Answers must be given in a separate answer sheet.

1 *In order to find out the IP address of a docker container…:*

a We may use the command `docker ps`

b We may use the command `docker images`

c We may use the command `docker inspect`

d There is no way to get such information, since that is not a property of a docker image.

2 *The command docker-compose:*

a Uses as its input a `Dockerfile` where we specify relations among docker images.

b Allows the execution of containers in remote computers. To this end, it includes remote connection directives in its configuration file.

c Allows the execution of multi-container docker applications.

d All the statements are true.

3 *In Lab 3, we have built a docker application in which one of its images saves events in a log file. That component is known as* `logger.` *Choose the correct alternative:*

a The log file to be generated may become inconsistent due to race conditions in the logger.

b With docker-compose, all components are connected to the logger. Thus, every component may access the log file, and all they may write on the file concurrently.

c The log file may be accessed from the host machine, since our configuration allows this.

d The logger component uses a `REP` socket in order to receive the log messages.

4 *If the logger component was placed in a host different to the rest of the* `cbw` *system, choose which statement would be true:*

a The `yml` deployment file should be divided into two, with the logger-related parts in a separate file (with the rest of needed instructions).

b The tools used in Lab 3 do not allow any automated solution for that deployment.

c An automated deployment is feasible if the `cbw_ftcl` host admits connections from other computers that it routes to the containers.

d An automated deployment is feasible if the new logger host admits connections from other computers that it routes to the appropriate container.

5 *The end of part 2 suggests to think about the possibility of deploying* `cbw_ftcl` *in two specific scenarios. Choose the correct statement:*

a A deployment with two brokers is not possible since both bind to the same URL.

b In a deployment with two brokers, only one of them may interact with workers, clients and logger.

c In a deployment with two loggers, broker messages are alternatively (i.e., in round-robin) forwarded to those loggers.

d In a deployment with two loggers, docker-compose cannot easily assign any value to variable `LOGGER_URL` since it depends on `log.`

6 *In lab 3, in regard to an external client, it cannot be deployed jointly with* `cbw_ftcl` *because…*

a In the Linux desktop computers, Docker has not been installed.

b Deployments with docker-compose cannot handle multiple hosts.

c The external client is incompatible with regular clients.

d This wording is false: those two elements can be deployed jointly.

7 *Let us imagine a succesful execution of the command* `docker build -t provisional .` *; then, choose the correct statement:*

a In the current directory, there is now a new file called `provisional`.

b If we run now `docker inspect provisional`, then we can obtain its IP address (almost at the end of its output), among other things.

c We may run `docker run provisional`, even in other directories.

d The `Dockerfile` we have used certainly includes a `CMD` or `ENTRYPOINT` instruction.

8 *In the system where an external client is considered:*

a The access to the broker container is made through a host port.

b External clients may directly interact with the logger.

c In order to forward client requests to the broker a `ROUTE ip_localhost:ip_container` line is needed in the broker's Dockerfile.

d The IP address of the broker container is static and accessible from the computer where the remote client runs.

9 *In the system with a* `worcli` *component, its provided program accepts as parameters:*

a One broker URL.

b Two URLs from two brokers and the job class to be processed.

c Two URLs from two brokers and a delay.

d Two URLs from two brokers, a delay, and the job class to be processed.

10 *In regard to the external client scenario (*`4_CBW_FTCL_CEXT`*) and its provided* `docker-compose.yml`

a Such `docker-compose.yml` allows external clients to directly interact with the broker.

b If we remove the `expose:` section from `docker-compose.yml`, we interconnect broker and external clients.

c We should add a section `ports:` mapping the port to be used by clients from the host to the container (`9999:9999`)

d We should remove the `expose:` section.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

A

ID

0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

ETSINF - Tsr

Second Partial + Lab 3 - 01/09/2019

Surname ....

Name ....

Mark this way

DO NOT mark this way

DO NOT ERASE, do corrections with Typex

Theory

1	a	b	c	d
2	a	b	c	d
3	a	b	c	d
4	a	b	c	d
5	a	b	c	d
6	a	b	c	d
7	a	b	c	d
8	a	b	c	d
9	a	b	c	d
10	a	b	c	d
11	a	b	c	d
12	a	b	c	d
13	a	b	c	d
14	a	b	c	d
15	a	b	c	d
16	a	b	c	d
17	a	b	c	d
18	a	b	c	d
19	a	b	c	d
20	a	b	c	d

Lab

1	a	b	c	d
2	a	b	c	d
3	a	b	c	d
4	a	b	c	d
5	a	b	c	d
6	a	b	c	d
7	a	b	c	d
8	a	b	c	d
9	a	b	c	d
10	a	b	c	d