

**DI/FCT/UNL**  
**Mestrado Integrado em Engenharia Informática**

**Cloud Computing Systems**  
**1st Semester, 2019/2020**

**Midterm Test (26/October/2019)**

- 1) "One day, all (business) applications will run in (public) cloud platforms".  
a) Give reasons why the previous sentence could become true.

Cloud platforms offer a wide range of services required to run business applications. Those services can be deployed at any moment without any upfront cost and can automatically scale to meet any peaks in demand so that businesses don't have to worry about over/under provisioning and only pay what they need. Cloud platforms also take care of a lot of tasks like keeping servers up to date and managing machine failures. The scale of these cloud providers also means they have a lot of machines spread around the world, allowing businesses to provide the best experience independent of user location.

- b) Give reasons why the previous sentence will never be true.

It is not true that all businesses applications will run on cloud platforms because some applications might have specific requirements that are not met by public cloud platforms. Some businesses might also be big enough to justify running their own infrastructure. For other businesses the cost of cloud platforms might be too high, even though cloud platforms allow them to pay for what they use. These businesses may have a harder time scaling up but it's possible that they just don't need to and are served well enough by their current infrastructure.

- 2) Vendor lock-in (i.e. the difficulty of changing of cloud provider) is a problem in the cloud computing market? Justify your answer.

Yes, because... / No, because... (select your answer)

Yes because every cloud provider uses different APIs to access and manage their services. Each provider might have similar services but they won't work exactly the same and each will have its own upsides/downsides. Even something as common as SQL database, that can be queried the same independent of the provider, still needs to be deployed using each provider's webui/cli/dashboard/... and will have different configuration options. All of these reasons, and more, mean that once a business has built their software on top of these services there will be a lot of things that need to change in order to transition to another provider's services. The exception to this are applications that are self-contained and don't rely on the services provided by these cloud platforms.

3) App Service systems, such as Azure App Service, typically support scaling an application by scale out (run in more machines) and scale up (run in more powerful machines). Discuss the advantages/problems of each approach.

Scaling out has the potential to handle a higher combined load, at lower prices compared to scaling up, because at some point it becomes a lot harder and more expensive to obtain more powerful machines.

Scaling up can have benefits if the application being run is not easy to distribute across more machines and in this case fewer more powerful machines might be more efficient.

4) Azure Blob Storage supports three blob types. Explain the difference between block blobs and append blobs and give an example that is better supported with each type of blob.

Azure block blobs can be written to and deleted at any offset and are optimized for bulk data storage. Azure append blobs are optimized for append only workloads, these blobs can only be appended to.

Something like storing video files is better supported in block blocks.

Append blobs are better at things like keeping logs, since logs are usually append-only.

5) Azure Cosmos DB supports different consistency models: strong consistency, bounded staleness, session, consistent prefix and eventual.

a) Assuming a geo-replicated deployment, what are the advantages of selecting a weaker consistency model (for an application).

Using a weaker consistency has the advantage of lowering the latency required for a database operation to complete since a stronger consistency might require machines in different regions to communicate before accepting the operation.

b) "An application that uses the strong consistency level does not need to use the conditional update mechanism". State if you agree or disagree with this sentence and justify.

True, because... | False, because... (select the appropriate)

False, because strong consistency only means that readers will see the latest write but there is still the problem that if an application wants to modify a value it needs to read it, modify it and write it to the database. Between the read and the write there could have been another write by someone else and for some applications, writing over this new value could be a problem so a conditional update mechanism is required so that we only write if the data we are writing over is the one we read initially.

c) The recovery point objective of strong consistency is 0 for a deployment in multiple regions with a single master. Discuss what are the implications of this requirement for the implementation of the system and whether the implication could be relaxed by making the value of RPO larger.

Note: **Recovery point objective (RPO)** is the period of time for which updates might get lost in a failure.

The implication is that write has to be committed to at least another region so that RPO is 0.

Increasing the RPO has no effect when using strong consistency since the operation still has to be committed on all replicas regardless of RPO.

- 6) Consider the following example of the use of Memcached (from the slides). Explain why the data is stored in Memcached with the key "userrow:" + **userid** and not only **userid**.

```
function get_foo(int userid) /* first try the cache */
    data = memcached_fetch("userrow:" + userid)
    if not data /* not found : request database */
        data = db_select("SELECT * FROM users WHERE userid = ?",
                        userid)

        /* then store in cache until next get */
        memcached_add("userrow:" + userid, data)
    return data
```

Because in memcached all keys are stored in the same namespace so it is very likely that there are multiple values that need to be obtained from the userid and if we only used the userid as the key there could only be one value. By adding a prefix we allow the userid to be the key of different values.

- 7) Consider the project you are developing in the course – a clone of reddit.
- a) Give two examples of data that you can cache in Azure Redis for Cache, the first in which there is no advantage of using Redis over Memcached and the other in which using Redis is better. Justify briefly.

The first example could be something like storing body of a post in cache. This is just a blob of data and both cache services can handle that well.

The second example could be something like keeping a list of the most recently created posts. For this use case redis is better since it has native support for lists, and many other datatypes, unlike memcached which is mostly just an in memory key/value store.

b) For making your system available in the web, it would be necessary to develop a web interface – web interfaces typically consist of HTML pages, Javascript files and images.

For making these resources available to applications, they would have to be stored in one of the following alternatives:

- in the war files that is deployed to the App Service service, being served as files in a normal web site;
- on Azure blob storage;
- in Cosmos DB.

Discuss what would be the best alternative to store each of these resources. Justify.

CosmosDB is not a good option since it is a database and is not designed for this type of data storage.

Storing the those files in the war file has the advantage that we can just deploy our app and be certain that all files are available and are the most up to date version. The downside of this method is that files cannot be modified without redeploying the app and the war file that each app server will need will be much bigger than if it was just code.

Blob storage has the advantage that these files can be created/modified/deleted at any time while the application is running without the need to redeploy it while also making the war file much smaller. These last 2 services support the use of CDN's.