M2 Mosig – Academic year 2023-2024
Cloud Computing – From Infrastructure to Applications
Final Exam (Session 1) – February 2, 2024

Part II (Renaud Lachaize) – 7 points

Reminder: Please use a distinct answer sheet for this part.

All the questions are independent. The weights of the questions are only provided for indicative purposes.

Problem 2.1 (1 point)

What are the negative consequences for a cloud tenant (i.e., an organization that uses a cloud infrastructure to host and run its application) if the elasticity manager/algorithm that is used for the compute resources (virtual machines, containers or functions) of its application is not sufficiently efficient?

Problem 2.2 (1 point)

In this question, we compare two types of cloud storage systems: a distributed file system (for example, NFS) and an object-based storage system (such as AWS S3).
Describe one advantage of object-based systems (with respect to distributed file systems) that makes them more appropriate for some types of cloud applications. Briefly justify/illustrate your answer.

Problem 2.3 (1 point)

The concept of *"sidecar"* is a popular design pattern for cloud-native applications, which is easy to implement in Kubernetes, via the notion of *"Pod"*.
Provide a brief definition of this concept of *"sidecar"*, explain why it is useful, and give one example of usage scenario.

Problem 2.4 (1 point)

In the Kubernetes architecture, what are the main components of the *control plane*? What are their respective roles?
Note: Giving the exact names of these components is not strictly required – the important point is to summarize their main purpose.

Problem 2.5 (1.5 points)

Some cloud infrastructure software layers (including the Kubernetes orchestrator) rely on the key concept of "*control loop*", itself based on the notions of "*reconciliation pattern*" and "*declarative/desired state*".
What are the benefits of this approach?

(Warning: do not forget the last exercise for this part on the next page)

Problem 2.6 (1.5 points)

One of your friends is hesitating between two main approaches (service models) to design and deploy a new application in the Cloud.
The first approach consists in using a CaaS (*Containers as a Service*) model, and more precisely a managed container orchestration platform such as GKE (*Google Kubernetes Engine*) or AWS EKS (*Amazon Elastic Container Service for Kubernetes*).
The second approach consists in using a FaaS (*Functions as a Service*) model, provided by a platform such as Google Cloud Functions or AWS Lambda.

What are the main questions that you would ask your friend (for example, regarding the application characteristics and requirements) in order to help her/him make a decision?
And what would be your suggested choice according to her/his answers to your questions?

Remarks:

- The precise choice of cloud provider (for example, Google Cloud Platform or Amazon Web Services for public cloud providers) is not relevant here. The examples mentioned above are only given for illustration purposes.
- This question is focused on technical reasons. The scope of this question does not encompass considerations such as financial costs (although theses aspects are of course important in practice).

Part III – 7 points (Thomas Ropars)
Remember to use a separate answer sheet for this part

Exercise 3.1 The scenario presented in Figure 1 describes an execution with 3 clients interacting with a service storing a piece of data. The initial value of the data is `0`. The figure describes the sequence of read and write requests executed by each client.

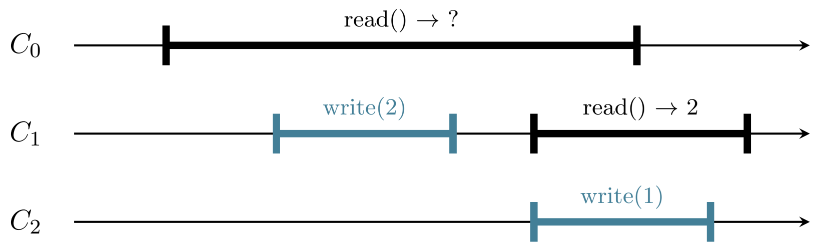

Figure 1: An execution with 3 clients

In Figure 1, what is/are the possible value(s) that can be returned by the *read()* operation of client C_0 that would make the execution linearizable?

For each value you answer, justify briefly (using a figure if it simplifies your explanation)

Exercise 3.2 In the context of linearizable replication based on Quorum Systems, we have introduced a mechanism called *Read Repair*. About *Read Repair*, we have seen that:

1. It is necessary for correctness
2. It slows down read operations
3. It helps improving fault tolerance

Explain each of these points.

Exercise 3.3 Explain why linearizable replication based on Quorum Systems can only work for $f < n/2$. (A detailed explanation is expected)

Exercise 3.4 A solution to reduce the carbon footprint of datacenters is to extend the lifetime of servers.

Discuss the possible advantages and drawbacks of such a solution. Give at least 4 advantages and drawbacks (4 in total) and justify briefly.

Tip: think about advantages/drawbacks related to Capex and Opex

!! Last exercise on next page !!

Exercise 3.5 In the context of microservices applications, we observe a large number of scientific publications in recent years that propose techniques to dynamically scale microservices beyond the default mechanisms provided by orchestration systems such as Kubernetes. Some systems are even based on machine learning models to detect automatically the best moment to add/remove replicas in one service.

Answer the following questions and justify briefly:

(a) Based on your experience in the lab and the project, explain why the Horizontal Pod Autoscaler of Kubernetes might be difficult to set up for large-scale deployments of microservices applications.

(b) Based on what you know about microservices applications, explain which characteristics of micro-services applications can make automatic scaling of services difficult. (list at least 2 characteristics)