Part II – 14 points (Thomas Ropars)
Remember to use a separate answer sheet for this part

1 Replication (4 points)

1.1 The scenario presented in Figure 1 describes an execution with 3 clients interacting with a replicated service supporting 3 operations on objects:

- *read()*: returns the most recent value of the object
- *write(X)*: update the value of the object to X
- *add(X)*: Add X to the value of the object (valid only if the object is of `Integer` type)

We consider an Integer object, with `0` as initial value. The figure describes the sequence of requests executed by each client, with the value returned by each read request.

Answer the following questions:

(a) Name 2 replication approaches that would be suitable to implement this service assuming that we target linearizability. (Justify your answer)

(b) Is the execution presented in Figure 1 linearizable? Answer YES or NO, and:

- If your answer is yes, draw a graph with the equivalent sequential execution that shows that the execution is linearizable.
- If your answer is no, explain which operation(s) is/are not linearizable and why.

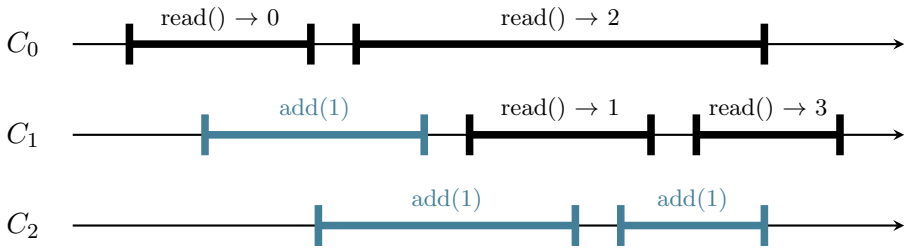

Figure 1: An execution with 3 clients

1.2 The scenario presented in Figure 2 describes another execution with 3 clients interacting with the same service as in the previous exercise. We still consider an Integer object with initial value `0`.

Is the execution presented in Figure 2 linearizable? Answer YES or NO, and:

- If your answer is yes, draw a graph with the equivalent sequential execution that shows that the execution is linearizable.
- If your answer is no, explain which operation(s) is/are not linearizable and why.

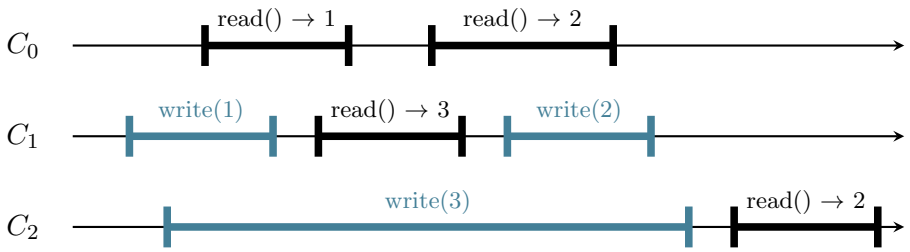

Figure 2: Another execution with 3 clients

1.3 We want to implement replication based on quorum systems. We have chosen a replication degree of 5. But we hesitate between multiple configurations for the quorums.

We note R, the number of replicas that must be accessed during a read operation. We note W, the number of replicas that must be updated during a write operation.

Here are the configurations that we consider:

(a) $R = 2$ and $W = 3$

(b) $R = 3$ and $W = 3$

(c) $R = 4$ and $W = 2$

(d) $R = 5$ and $W = 1$

For each of the following configuration, explain the advantages and drawbacks of the configuration (a detailed explanation is expected – make your assumptions explicit if you make any)

2 Resource Management (5 points)

2.1 The article *What Serverless Computing Is and Should Become* presents the graph described in Figure 3 to illustrate the impact of the serverless approach on the resources billed to Cloud users. The figure consider 3 approaches for accessing/allocating computing resources: *on-premise*, *serverful*, and *serverless*.

Provide a detailed explanation and analysis of this figure. Your analysis should focus on the following points:

(a) The expected benefits in terms of resource usage of the different approaches for the cloud users.

(b) The limitations/drawbacks of the different approaches in terms of resource usage for the cloud users.

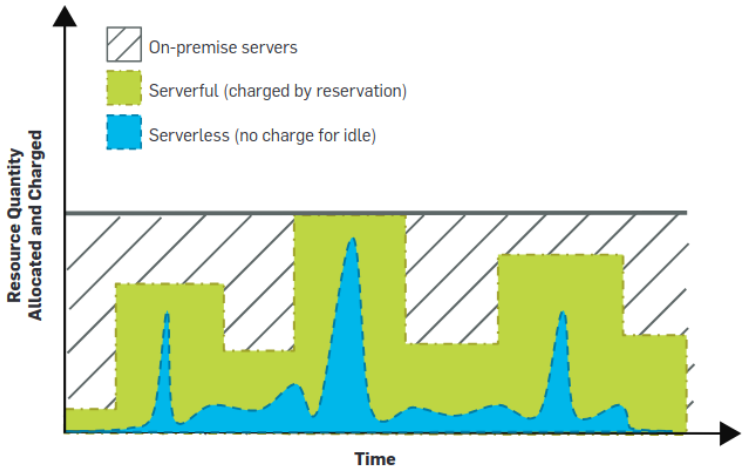

Figure 3: Serverless vs Serverful cloud computing

(c) The impact of the different approaches on the expected hardware resource usage (here, you should consider the point of view of the entity in charge of administrating the hardware resources)

2.2 In the paper describing `Prequal`, the load balancing strategy used for the `Youtube` service, we can read the following observation: *"Each replica runs inside its own virtual machine (VM), which has access to a guaranteed portion of the CPU cycles on its host machine; this amount is called its (CPU) allocation. The fraction of its allocation that a replica uses over any given time period is its (CPU) utilization, which could be more or less than 100%, since the allocation is just a guaranteed minimum"*

Answer the following questions and justify:

A) Based on this observation, the authors of the paper explain that although balancing CPU utilization between replicas seems to be an obvious goal for a load balancing strategy, it can lead to bad results in terms of tail latency. Explain the problem and propose an alternative strategy.

B) Could the same problem occur in the context of resources managed using Kubernetes?

3 Containers and Orchestration (5 points)

3.1 One of the core approaches for designing Cloud-Native applications is *Health reporting*. The designers of Borg and Kubernetes identify this approach has a key to the success of *Infrastructure as software*.

A) Describe what *Health reporting* refers to in this context.

B) Explain why this approach is important to be able to implement *Infrastructure as software* efficiently

3.2 During the labs, you have manipulated several Kubernetes concepts related to the execution of applications. In this question, we consider 3 of these concepts:

- *Deployment*
- *ReplicaSet*
- *Service*

Explain each of these concept in the context of Kubernetes, their purpose, and how they relate to each other.

3.3 During the labs, you deployed micro-services in a GKE cluster and made these micro-services accessible on Internet.

Draw a figure that represents the main components involved in the execution of a micro-service in a GKE cluster. We assume this micro-service is accessible through a public IP address.

In this figure, at least the following *concepts/components* should be represented:

- Master nodes
- Worker nodes
- Virtual Machines
- Containers
- Pods
- ReplicaSet
- Load-balancer
- Kubelet
- Kube-controller-manager

Additionally, your figure should show:

- Which components interact between each other
- Which components are directly visible from the cloud user and which components are fully managed by the cloud provider

You can add comments to your figure if you are not sure your representation is clear enough.