**Homework-2025-04-01**

**1. Horizontal Scaling Example Service**

Horizontal scaling definition: To manage increased load, horizontal scaling—also referred to as scaling out—involves adding extra machines or instances. To balance the demand, workloads are usually divided among several servers or instances. Horizontal scaling adds more units to handle the demand, as opposed to vertical scaling, which upgrades the CPU and RAM of a single computer.

Its Distinction from Vertical Scaling

Increasing the number of machines to spread the load is known as horizontal scaling. By adding more instances in parallel, the system expands.

Vertical scaling is the process of increasing an existing machine's power (CPU, RAM). By increasing the capabilities of a single instance, the system expands.  
  
Cloud Service Supporting Horizontal Scaling: Google Kubernetes Engine (GKE) is an example of a cloud service that supports horizontal scaling.Depending on demand, GKE can dynamically scale your clusters up or down. It also lets you run apps in containers across numerous instances. By varying the number of nodes in the cluster, GKE attains scalability while maintaining high availability and efficient use of resources.

An Example from the Real World Where Horizontal Scaling Helps: There are times when demand is strong (like during live events or viral material) and times when demand is low (like during off-peak hours) on a social media platform. Without being restricted by the restrictions of a single computer, horizontal scaling enables the platform to dynamically add or remove servers to address traffic spikes.

**2. An Illustration of a Vertical Scaling Service**

Vertical scaling is defined as: Adding extra CPU, RAM, or storage to a single computer in order to handle heavier workloads is known as vertical scaling, or scaling up. Instead of installing additional servers, this approach usually entails improving the current server.

Comparative Analysis of Horizontal Scaling

Vertical scaling is the process of increasing a single machine's capacity to manage a greater load by upgrading its resources.

By distributing the load among more machines, horizontal scaling enables the system to expand beyond the constraints of a single machine.

Cloud Service Supporting Vertical Scaling: By modifying the resources, such as CPU and RAM, Google Compute Engine (GCE) offers virtual machine (VM) instances that can be vertically scaled. Users can improve performance during times of high load by resizing their virtual machines (VMs) to higher performance tiers as needed with GCE.

**Benefits and Drawbacks of Vertical Scaling in Cloud Settings:**

* Benefits:
* less complicated to administer and execute than horizontal scaling (load balancing is not required).
* Perfect for applications that are difficult to spread or demand strong single-machine performance.

**Restrictions:**

* A single machine can only be scaled up to a certain physical extent.
* Because larger instances are more expensive, scaling up can be costly.
* Fault tolerance could be a problem since performance might be affected by a single point of failure.

**3. Example for Object Storage**

What is Object Storage? Rather than storing data as files or blocks, object storage stores data as objects. Every object has a unique identifier, metadata, and the actual data. It’s highly scalable and accessible over the internet, making it suitable for unstructured data, such as media files, backups, and web data.

The distinction between block storage and file storage

Applications that need an orderly file system (like NFS or SMB) can benefit from file storage, which stores data in a hierarchical structure (files and directories).

Block storage is perfect for databases and applications that demand high performance because it stores data in blocks that are managed and displayed as raw storage volumes.

**Object storage**: Designed for large-scale storage requirements, it stores data as unstructured objects that are accessible over HTTP/S protocols and frequently contain copious amounts of metadata.

**Cloud Object Storage Service**: One well-known example of cloud object storage is Google Cloud Storage. It offers extremely safe, long-lasting, and scalable object storage. Features include data lifecycle management, automated redundancy across various sites, and connectivity with other Google Cloud services.

**Real-World Use Case for Object Storage:** For a media streaming platform, object storage is perfect for holding vast amounts of media content, including audio, video, and picture files. Global access to these files necessitates scalable, reasonably priced storage options with high redundancy and fast access speeds.

**4. Use of Cloud Filestores**

Cloud Filestore: What is it? Applications that need a file system interface and shared access can store their files on Cloud Filestore, a fully managed network-attached storage (NAS) service. It is appropriate for workloads requiring shared storage across several instances because it supports common file protocols like NFS.

Comparing Block and Object Storage:

Usability and Performance:

For file-based applications, Cloud Filestore offers better speed, especially when shared file system access is necessary. It is optimized for low-latency file access.

Object Storage is more scalable and better suited for unstructured data with high availability and redundancy, but it lacks the functionality needed for file-based workloads (e.g., NFS or SMB support).

Block Storage is ideal for high-performance applications needing low-level storage access but doesn’t offer the file system features or scalability of object storage.

Situation Where Cloud Filestore is the Best Solution: Cloud Filestore is ideal for a high-performance computing (HPC) environment where multiple virtual machines (VMs) need shared access to a common file system for processing large datasets, such as in scientific research or video rendering.

**5. Difference Between Zone and Region Definition:** A zone is a deployment area within a region; it is essentially a data center with its own independent power, cooling, and networking, offering isolation and fault tolerance.

A region is a larger geographical area that contains multiple zones; it permits distributed computing across zones. The reason cloud providers organize infrastructure into zones and regions is to provide high availability and redundancy, meaning that services can continue to operate from another zone or region in the event of a zone or region failure. Additionally, by providing multiple geolocated options, regions aid in ensuring compliance with data residency laws.

Impact of Choosing a Specific Region on Service Availability and Performance: Choosing a region closer to the end-users or where the application needs to meet data residency requirements will optimize performance (lower latency) and availability (fewer risks of service disruption). Additionally, regions often have different availability of services, so choosing the right region is key for service selection.

6. GCP's Definition of IaaS and FaaS and Those Services IaaS vs. FaaS:

Infrastructure as a Service, or IaaS, offers virtualized computer resources like networking, storage, and virtual machines via the internet. Users manage the operating systems, applications, and runtime.

FaaS (Function as a Service): A serverless computing model that allows users to run individual functions without managing the underlying infrastructure. The cloud provider takes care of scaling and resource allocation.

Examples in Google Cloud Platform:

IaaS: Google Compute Engine (GCE) allows users to deploy and manage virtual machines.

FaaS: Google Cloud Functions is a fully managed environment for running event-driven functions.

**Situations for Selecting IaaS over FaaS:**

If you have certain software or performance needs and demand complete control over the infrastructure, go with IaaS. Running a legacy application that needs a certain operating system or special settings is one example.

If event-driven applications require a serverless architecture where you only pay for execution time and don't need to manage infrastructure, go with FaaS. Processing real-time data from a social media stream with little administration overhead is an excellent example.