Course: Cloud Computing

Week 8: Scalability and Performance in the Cloud

Lecturer: Ikwap Flavia Agatha
MSc. Computer Forensic
PHD in IT (Candidate)

Lecture Learning Outcome

- By the end of week 8, you will be able to:
- Understand Cloud Scalability
- Understand the Purpose of Auto scaling
- Types of scalability
- Measures used in scalability
- Understand Cloud Performance and influencing factors
- Understanding the impact of Cloud scaling to an organization

Cloud scalability

- Scaling on demand is one of the most outstanding benefits of cloud computing allowing scaling beyond the limits of the existing in-house IT resources, whether they are infrastructure (compute and storage) or applications services, therefore supporting business expansion, fluctuating workloads and unpredictable traffic patterns
- Cloud scalability: Refers to the ability of a cloud computing system to handle increasing workloads and scale resources dynamically to accommodate changes in demand without sacrificing performance, reliability, or user experience.

AUTO SCALING

- Main Purpose of Auto scaling is to preventing resource over-provisioning or under-provisioning.
- Auto-scaling is the capability in cloud computing infrastructures that enables dynamic provisioning of virtualized resources. Resources used by cloud based applications can be automatically maximized or minimized, in that way adapting resource usage to the application requirements

Characteristics of auto-scaling

- Automatically adding additional resources during increased demand (i.e., scaling out) (i.e., the automatic termination of extra unused resources when demand decreases, in order to minimize cost).
- The capability of establishing scaling rules for outbound and inbound scaling.
- It offers the facility to automatically identify and replace instances that are unreachable.
- Better availability: AS is to utilize and configure multiple Availability Zones, if it becomes unavailable, AS can launch instances in another one to compensate.

Types of Scalability

- **Vertical Scalability**: Involves increasing the capacity of individual resources, such as CPU, memory, or storage, within a single instance. While vertical scaling can address immediate resource constraints, it has limitations in terms of scalability and may lead to single points of failure. Improving the vertical scalability is important in achieving the low investment on cloud computing and virtualization.

Vertical Scalability

- Scaling up involves adding more resources to the same computing pool, (e.g., adding more RAM, disk, or virtual CPU to handle an increased application load). Vertical scaling can involve replacing the current IT resource by another one with higher capacity (scaling up) or with lower capacity (scaling down).

Advantages of Vertical Scalability

- **Cost-effective:** It much cheaper to improve or upgrade a node that already exists unlike making a brand new purchase, No need for installations and configurations since the old machine is maintained but only upgraded**.**
- **Less complex process communication** – When a single node handles all the layers of your services, it will not have to synchronize and communicate with other machines to work. This may result in faster responses.
- **Relatively easy maintenance** –Maintenance is quite cheaper and it's also less complicated because one has to deal with fewer machines.

Advantages of Vertical Scalability

- **Less need for software changes** – Software like the operating systems and virtualization software are likely not be changes since the server or device was only upgraded.
- It enhances the capacity of the existing hardware and software.
- It offers more shared resources for applications and operating system.

Disadvantages of vertical scaling

- **Higher possibility for downtime** – Unless you have a backup server that can handle operations and requests, you will need some considerable downtime to upgrade your machine.
- **Vast Failure**: Operating in single or fewer nodes like servers increases the risk of losing all data as a result a hardware or software failure.
- **Upgrade limitations** – There is a limitation to how much you can upgrade a machine. Every machine has its threshold for RAM, storage, and processing power.
- It is also slower than horizontal scaling because of the downtime required during the replacement of the resource.

Horizontal Scalability:

- Also known as scale-out, involves adding more instances or nodes to distribute the workload across multiple resources. Horizontal scalability offers better scalability and fault tolerance by leveraging distributed architectures and load balancing mechanisms.
- Horizontal cloud scalability is mainly used to connect multiple hardware or software entities, like servers, hard drives to enable them work as a single logical unit.
- Horizontal scalability is provided by means of adding or removing more individual units of resource doing the same job. In the case of servers, you could increase the speed or availability of the logical unit by adding more servers.

Advantages of horizontal scaling

- **Easy Scaling of hardware: With** horizontal scaling additional machines are added to the existing ones, there eliminates the need to analyze which system specifications you need to upgrade.
- **Fewer periods of downtime**: All current pool of machines continue running as new ones are added, minimizing down time so clients continue to access resources without interruptions.

Advantages of horizontal scaling

- **Increased resilience and fault tolerance:** In horizontal scaling, alternative devices can be used without entirely relying on one machine therefore reducing on the possibility of losing all the data and a number of operations.
- **Increased performance** – Since work load is shared and distributed among several machines hence providing better performance.

Disadvantages of horizontal scaling

- Increased complexity of maintenance and operation: Maintaining very many servers or nodes may get so costly as it will need well experienced IT personnel
- New software's like operating systems, virtualization, and Backups will be required, every time new node is added
- Need to ensure that the nodes are properly synchronizing and communicating to each.
- Increased Initial costs – Adding new servers is far more expensive than upgrading old ones.

Simple comparison of Horizontal and Vertical Scaling

	Horizontal scaling	Vertical scaling
Description	Continuous Increase or decrease of nodes in a system to manage an increase or decrease in workload	Upgrade or reduce the power of a system to manage work load expansion or workload reduction.
Example	Increasing or decreasing virtual machine instances	Add or reduce the CPU or memory capacity of the existing VM
Execution	Scale in/out	Scale up/down
Workload distribution	Shared Workload since it is distributed across multiple nodes. Parts of the workload reside on these different nodes	One node must manage the whole task.

Simple comparison of Horizontal and Vertical Scaling

Configuration	This requires modifying a sequential piece of logic in order to run workloads concurrently on multiple machines	No need to change the logic. The same code can run on a higher-spec device
Downtime	Minimized since nodes stay active	Yes! The server being upgrade has to be turned off for the period of upgrade
Load balancing	The load of work is distribute evenly across multiple nodes	Load balancing mechanisms are not required in the single node

AUTO SCALING TECHNIQUES

1. Threshold-based rules (rules)
2. Reinforcement learning (RL)
3. Queuing theory (QT)
4. Control theory (CT)
5. Time series analysis (TS)

Threshold-based rules:

- Commercial cloud providers give purely reactive AS using threshold-based rules. The scaling decisions are triggered based on some performance metrics and predefined thresholds or predefined rules. Rule-based auto-scaler is simple and easy to set-up by clients, and is easy to provide as a cloud service.

Time series analysis (TSA):

- Involves a number of techniques to detect patterns and predict future values on sequences of data points. The accuracy in the forecast value (e.g. future number of requests or average CPU utilization) depends on choosing the right technique and setting the parameters and so based on the patterns a scaling will be done.

Queuing theory (QT):

- Queuing theory has been mainly applied to computing systems, in order to discover the relationship between the jobs arriving and leaving a system. Queuing theory can be used to add capacity by analyzing and making decisions based on a queue and specifically requests queued at the load balancer. A simple approach consists in modeling each VM as a queue of requests in order to estimate different performance metrics such as the response time.

Queuing theory (QT):

- Since queuing theory only provides an estimation of performance metrics, have combined with another approaches (i.e., threshold based policies, control theory, and reinforcement learning) to deal with auto-scaling problem.
- Set Back:
- They impose non-realistic assumptions that are not valid in real scenarios
- They are not efficient for complex systems

Control theory (CT)

- **Control systems** utilize a feedback loop by changing the controller input to influence the normative output. The aim is to define a (reactive or proactive) controller to automatically adjust the required resources to the application demands. Control systems are mainly used as reactive process, but there are also some proactive approximations like Model Predictive Control, or even combining a control system with a predictive model. CT has been applied to automate management of resources in different engineering fields, like data centers, storage systems, and cloud computing platforms.

Measurements used in scalability:

- **Load scalability (LS):** It is capable of operating graceful different loads while making better usage of available resources. A few factors that can hamper load scalability is scheduling of a class of resources and scheduling of a shared resource in a way that enhances its inadequate exploitation of parallelism and own usage.
- **Space scalability**: It refers to the enlargement of memory usage compared to the scale of the system. Many different approaches like space efficient algorithms and compression can help with space scalability, but the effects (like added CPU time of compression) might reduce other types of scalability like load scalability.

Measurements used in scalability:

- **Space-time scalability:** It regards the ability of a system functions gracefully improve when the number of items it handles increase by an order of magnitude. Space-time scalability may be related to both space scalability and load scalability in that the amount of items might stem from an increased load, and the presence of these objects may use more memory and affect data structures.
- **Structural scalability**: It refers to the standards of the system and how they limit the number of item the system may handle. The prime example of structural scalability concerns the addressing of the items.

Scalability Factor

- During scaling time, it is significant to note that what percentage of resource is actually scalable whether it is servers, processors, load-balancers or storage. It is called as scalability factor.
- Four categories of scalability factor are described as follows:

- Linear scalability: It performs scalability that remains constant in spite of scaling.
- Sub-linear scalability: At this point scalability factor reduces below 1.0.
- Supra-linear scalability: It is possible to obtain better throughput performance by adding one resource.
- Negative scalability: If the performance of an application degrades when the application is scaled which is known as negative scalability.

Aspects of Scalability

- **Elasticity**: Refers to the ability to dynamically provision and de-provision resources in response to changing demand. Cloud platforms offer auto-scaling features that automatically adjust resource capacity based on predefined policies, such as CPU utilization or request rates.
- **Statelessness**: Designing applications to be stateless allows them to scale more easily by eliminating dependencies on specific instances or sessions. Stateless applications can be distributed across multiple instances, enabling seamless horizontal scalability and fault tolerance.

Aspects of Scalability

- **Microservices Architecture**: Decomposing applications into smaller, independently deployable services (microservices) facilitates scalability by allowing each service to scale independently based on demand. Microservices architecture enables better resource utilization, fault isolation, and agility in cloud environments.
- **Distributed Databases**: Traditional monolithic databases can become bottlenecks as applications scale. Distributed databases, such as NoSQL databases or shared relational databases, distribute data across multiple nodes to support high scalability and performance.

Aspects of Scalability

- **Load Balancing**: Distributing incoming traffic across multiple instances or servers using load balancers ensures optimal resource utilization and prevents individual instances from becoming overwhelmed. Load balancers dynamically route requests to the most available and responsive resources.
- **Content Delivery Networks (CDNs)**: Leveraging CDNs improves scalability by caching content closer to end-users and distributing it across multiple edge locations. CDNs reduce latency, offload origin servers, and handle spikes in traffic effectively, especially for globally distributed applications.

Aspects of Scalability

- **Serverless Computing**: Serverless platforms, such as AWS Lambda, Google Cloud Functions, or Azure Functions, abstract infrastructure management and automatically scale resources based on request volume. Serverless architectures enable high scalability, cost-efficiency, and agility for event-driven workloads.

Cloud performance

- Cloud performance refers to the speed, responsiveness, and efficiency of computing resources and services delivered over the cloud. It encompasses various aspects, including the response time of applications, throughput, resource utilization, and overall user experience. Here are key considerations and factors that influence cloud performance:

Factors that influence cloud performance

- **Response Time**: The time taken for a system to respond to a request from a user or another system. Low response times are essential for ensuring a smooth user experience and high application performance.
- **Throughput**: The rate at which a system can process a certain number of requests or transactions within a given timeframe. High throughput indicates a system's ability to handle a large volume of concurrent operations efficiently.

Factors that influence cloud performance

- **Scalability**: The ability of a system to handle increasing workloads by adding resources or scaling horizontally. Cloud platforms offer scalability through features like auto-scaling, allowing resources to be dynamically provisioned or de-provisioned based on demand.
- **Resource Utilization**: Efficient utilization of computing resources such as CPU, memory, and storage is crucial for optimal performance and cost-effectiveness. Monitoring and optimizing resource usage help ensure that resources are allocated efficiently to meet application demands

Factors that influence cloud performance

- **Network Performance**: The speed and reliability of network connections between cloud services, data centers, and end-users play a significant role in overall cloud performance. Factors like latency, bandwidth, and network congestion affect application responsiveness and user experience.
- **Storage Performance**: The speed and reliability of data storage and retrieval operations impact application performance, especially for data-intensive workloads. Cloud providers offer various storage options with different performance characteristics, such as SSDs for high-performance storage and object storage for scalability and cost-effectiveness.

Factors that influence cloud performance

- **Load Balancing**: Distributing incoming traffic across multiple servers or instances helps prevent overloading of individual resources and improves overall system performance and availability. Load balancers dynamically route requests to the most available and responsive servers.
- **Caching**: Storing frequently accessed data closer to the application or end-users reduces the need for repeated data retrieval operations, improving response times and reducing latency. Caching mechanisms like content delivery networks (CDNs) help optimize cloud performance for distributed applications.

Factors that influence cloud performance

- **Monitoring and Optimization**: Continuous monitoring of performance metrics, application behavior, and resource utilization is essential for identifying bottlenecks, optimizing configurations, and improving overall cloud performance. Automated tools and performance testing frameworks help ensure that applications meet performance objectives and SLAs.

Best Practices for optimizing cloud performance:

- **Right-sizing Resources**: Choose cloud resources (e.g., virtual machines, databases) with the appropriate CPU, memory, and storage capacities based on workload requirements. Avoid over-provisioning or under-provisioning resources to optimize cost and performance.
- **Use of Scalable Services**: Leverage cloud-native services such as AWS Lambda, Google Cloud Functions, or Azure Functions for event-driven and serverless computing. These services automatically scale based on demand, optimizing performance and cost-efficiency.

Best Practices for optimizing cloud performance:

- **Implement Auto-scaling**: Configure auto-scaling policies to automatically adjust the number of instances or resources based on workload fluctuations. This ensures that applications can handle varying traffic levels efficiently without manual intervention.
- **Optimized Storage Solutions**: Choose storage options with the right performance characteristics for your workload, such as SSDs for high-performance storage or object storage for scalability and cost-effectiveness. Utilize caching mechanisms and content delivery networks (CDNs) to optimize data retrieval and delivery speed.

Best Practices for optimizing cloud performance

- **Load Balancing and CDN**: Distribute incoming traffic across multiple servers or edge locations using load balancers and CDNs. This improves application responsiveness, reduces latency, and enhances user experience, especially for globally distributed applications.
- **Network Optimization**: Optimize network configurations to reduce latency and improve throughput. Use virtual private clouds (VPCs), CDN edge locations, and dedicated interconnects to minimize network congestion and ensure reliable connectivity between cloud services and end-users.

Best Practices for optimizing cloud performance

- **Application Performance Monitoring (APM)**: Implement robust monitoring and logging solutions to track application performance metrics, resource utilization, and user experience. Use APM tools to identify performance bottlenecks, troubleshoot issues, and optimize application code and configurations.
- **Optimize Database Performance**: Fine-tune database configurations, indexes, and queries to improve database performance. Consider using managed database services or database caching solutions to offload database management tasks and enhance scalability and performance

Best Practices for optimizing cloud performance

- **Security and Compliance**: Implement security best practices to protect cloud resources from cyber threats and ensure compliance with industry regulations. Use encryption, access controls, and security monitoring tools to safeguard data and infrastructure without compromising performance.
- **Continuous Optimization**: Continuously monitor performance metrics and resource utilization to identify optimization opportunities. Use performance testing and tuning techniques to optimize application performance, scalability, and cost-efficiency over time.

Impact of cloud scalability to an organization

- Scalability enables business and Enterprises to manage their varying Work loads
- Scalability allows business to save as much resources as possible
- It enhances the overall reliability and performance of cloud providers and cloud systems

Next lecture

- Security in the Cloud

Reference

- Dawoud, W. (2013). *Scalability and Performance Management of Internet Applications in the Cloud.* Potsdam, Germany.
- Falatah, M. M. (2014). CLOUD SCALABILITY CONSIDERATIONS. *International Journal of Computer Science & Engineering Survey (IJCSES) Vol.5, No.4*, 1-11.
- Raja, C. (2019). STUDY ON SCALABILITY SERVICES IN CLOUD COMPUTING. *JETIR June 2019, Volume 6, Issue 6*, 1-5.