International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

Serverless Computing

Presented by

Darshan.P.P

Lakshya.S

Karthik Raja.S.K

Abhishek.S

B.Tech. Artificial Intelligence and Data Science

Kongu Engineering College,Perundurai-638011.

ABSTRACT Serverless Computing is a cloud computing execution model in which the cloud provider is responsible for allocating and managing the servers, and the user is only responsible for providing and managing their application code. This allows for greater scalability and cost efficiency, as the user only pays for the resources they actually use, rather than maintaining their own infrastructure. Additionally, serverless computing allows for faster development and deployment, as there is no need to provision and maintain servers. Common examples of serverless computing include AWS Lambda, Azure Functions, and Google Cloud Functions.

Introduction

Serverless computing, also known as Function as a Service (FaaS), is a cloud-based model for building and running applications and services without having to manage and provision servers. In this model, the cloud provider is responsible for allocating, scaling and managing the servers, while the user is responsible for writing and uploading their code and providing the necessary triggers for execution.

One of the main advantages of serverless computing is that it allows for greater scalability and cost efficiency. Since the user only pays for the resources they actually use, rather than maintaining their own infrastructure, it can be a more cost-effective solution for many types of applications. Additionally, serverless computing allows for faster development and deployment, as there is no need to provision and maintain servers.

Common examples of serverless computing include AWS Lambda, Azure Functions, and Google Cloud Functions. These services allow users to write and deploy their code without having to worry about server management, and they automatically scale the number of servers based on the number of requests. This means that the user only pays for the resources they use, and they don’t have to worry about over-provisioning or under-provisioning resources.

Application:

1.E-commerce Application:

Ecommerce applications in the cloud enable users and e-businesses to respond quickly to emerging opportunities. It offers a new approach to business leaders to make things done with minimum amount and minimal time. They use cloud environments to manage customer data, product data, and other operational systems.

2. Big Data Analysis:

One of the most important applications of cloud computing is its role in extensive data analysis. The extremely large volume of big data makes it impossible to store using traditional data management systems. Due to the unlimited storage capacity of the cloud, businesses can now store and analyze big data to gain valuable business insights.

Cloud Server

A cloud server is a type of server that is hosted on a cloud computing platform, rather than on a physical server. This means that the server's resources, such as processing power and storage, are provided by a network of remote servers that are connected over the internet. This allows for greater flexibility and scalability, as the resources of the cloud server can be increased or decreased as needed.

Cloud servers can be accessed and managed remotely, making them ideal for businesses and organizations that need to scale their operations quickly and easily.

Meta Data

In serverless computing, metadata refers to information about the environment and configuration of a serverless function or service. This can include information such as the function's name and version, the memory and timeout settings, and the triggers that invoke the function. Meta data is important in serverless computing because it allows for better management and monitoring of the function or service. For example, with the function name and version, one can identify which version of a function is currently in production, and quickly roll back to a previous version if necessary. Memory and timeout settings are also important because they determine how much resources are allocated to the function, which can affect its performance and cost.Additionally, meta data also allow for better security management. For example, one can use metadata to control access to a serverless function or service by specifying the IAM roles or user groups that have permission to invoke it.In most serverless computing platforms, this meta data is provided in the form of configuration files, known as YAML or JSON, that are used to deploy and manage the functions or services. These files can be stored in a version control system such as Git or in a cloud storage bucket, which makes it easy to track changes over time and roll back to a previous version if necessary.

Architecture Of Communication Protocol

The architecture of a communication protocol refers to the overall design and structure of the protocol, including the different layers and components that make up the protocol.

A common architecture for communication protocols is the OSI (Open Systems Interconnection) model, which is a seven-layer model for describing how data is transmitted between two devices on a network. The seven layers are:

Physical layer : This layer is responsible for transmitting raw bits over a physical medium, such as a wire or wireless signal.

Data Link Layer : This layer is responsible for creating a reliable link between two devices on the network, by providing error detection and correction.

Network Layer : This layer is responsible for routing data packets through the network, by providing a logical addressing scheme.

Transport Layer : This layer is responsible for ensuring that data is delivered reliably and in order, by providing flow control and error recovery.

Session Layer : This layer is responsible for establishing, maintaining, and terminating sessions between applications on different devices.

Presentation Layer : This layer is responsible for encoding and decoding data, to ensure that it is in a format that can be understood by the application.

Application Layer : This layer is responsible for providing the interface between the application and the network, by providing services such as file transfer and email.

Another common communication protocol architecture is the TCP/IP model, which is a four-layer model. The four layers are:

Link Layer: This layer is responsible for providing a reliable link between two devices on the network, by providing error detection and correction.

Internet Layer: This layer is responsible for routing data packets through the network, by providing a logical addressing scheme.

Transport Layer: This layer is responsible for ensuring that data is delivered reliably and in order, by providing flow control and error recovery.

Application Layer: This layer is responsible for providing the interface between the application and the network, by providing services such as file transfer and email.

Both the OSI and TCP/IP models are widely used in networking and communication, and they provide a framework for understanding the different components and functions of communication protocols.

Data Security In Serverless Computing

Data security is a critical concern in serverless computing, as sensitive and personal data is often processed and stored in the cloud. Some of the key security challenges in serverless computing include,

1. Data encryption: Data should be encrypted both in transit and at rest to protect it from unauthorized access or disclosure.

2. Access control: Access to serverless functions and services should be restricted to authorized users or systems, and role-based access controls should be implemented to ensure that only authorized users have access to sensitive data.

3. Compliance: Serverless computing platforms must be able to meet regulatory compliance requirements for data security and privacy, such as HIPAA and PCI-DSS.

4. Network security: Network security measures,

such as firewalls and virtual private networks, must be implemented to protect against external threats and unauthorized access.

5. **Visibility and monitoring:** Serverless computing platforms must provide visibility and monitoring capabilities to detect and respond to security incidents in real-time.

To mitigate these challenges, serverless computing providers offer a variety of security features such as encryption, access control, and compliance management. AWS Lambda, Azure Functions and Google Cloud Functions have built-in security features, such as encryption at rest, and IAM roles to control access to the functions.

It is also important to note that the security responsibility is shared between the provider and the user, as the user is responsible for securing the data and code they upload to the cloud, while the provider is responsible for securing the infrastructure and platform.

Prototype For Serverless Computing

Prototype design in serverless computing refers to the process of creating a simplified version of a serverless application or service in order to test and evaluate its functionality and performance before fully deploying it

1. **Identify the problem or use case:** Understand the requirements and goals of the application, and identify the specific problem or use case that it is intended to solve.

2. **Design the architecture:** Determine the appropriate serverless architecture for the application, taking into account factors such as scalability, reliability, and cost.

3. **Choose the appropriate platform:** Select the serverless computing platform that best fits the needs of the application, such as AWS Lambda, Azure Functions or Google Cloud Functions.

4. **Develop the prototype:** Write the code for the serverless functions and services, and test them locally using a local development environment.

5. **Deploy and test the prototype:** Deploy the prototype to the chosen serverless platform and test it by running a series of test cases to evaluate its functionality and performance.

6. **Iterate and improve:** Based on the feedback from the tests, iterate and improve the prototype by making necessary changes to the code and design.

Main Role Of Iot In Serverless Computing

The **Internet of Things (IoT)** plays a significant role in serverless computing, as it enables the creation of connected devices that can collect, process, and transmit data without the need for a dedicated server or constant human oversight.

The main role of IoT in serverless computing is to provide a large amount of data from connected devices that can be used to trigger serverless functions. These functions can then process and analyze the data in real-time, providing valuable insights and triggering actions. This allows for the creation of highly automated and efficient systems that can respond to changing conditions and user needs.

One example of IoT and serverless computing working together is in the smart home, where connected devices such as thermostats and security cameras can collect data and send it to a serverless function that can analyze the data and make decisions about adjusting the temperature or alerting the homeowner of any security breaches.

Another example is in the

industrial sector, where connected sensors on industrial equipment can collect data and send it to a serverless function that can analyze the data and trigger maintenance or repair actions before a failure occurs.

In summary, IoT plays a critical role in serverless computing by providing a large amount of data that can be used to trigger serverless functions, allowing for highly automated and efficient systems that can respond to changing conditions and user needs.

Conclusion :

Serverless computing offers powerful, event-driven integrations with numerous cloud services, simple programming and deployment models, and fine-grained scaling and cost management. Driven by these benefits, the growing adoption of serverless applications warrants the evaluation of serverless platform quality, and the development of new techniques to maximize the technology’s potential. The performance results of our platform are encouraging and our analysis The current implementation presents many opportunities for continued development and study. We hope to see increased interest in serverless computing by academia and increased openness by the industry leaders for the wider benefit of serverless technologies. We hope that this paper added some value and accompanied you with knowledge and Information.

References

[1] Amazon Web Services, “AWS Lambda,” Available: https://aws.amazon. com/lambda/, 2017.

[2] Garrett McGrath,Paul R. Brenner,”*Serverless Computing: Design, Implementation, and Performance*” University of Notre Dame,Indiana.

[3]Serverless, Inc., “*Serverless Framework*,” Available: https://serverless. com/, 2017.