Chapter 14
Applications of Serverless Computing:
Systematic Overview

A. Kathirvel
https://orcid.org/0000-0002-5347-9110
Panimalar Engineering College, India

ABSTRACT

Applications that are serverless can be distributed (many services are connected for smooth operation), elastic (resources can be scaled up and down without limit), stateless (interactions and data aren't stored), event-driven (resources are allocated only when triggered by an event), and hostless (apps aren't hosted on a server). Serverless computing is becoming more and more popular as cloud adoption rises. In many respects, serverless computing unleashes the entire potential of cloud computing. we pay only for the resources consumed, and resources are allocated, increased, or decreased dynamically based on user requirements in real-time. It makes sure that when there are no user requests and the application is effectively dormant, resources are immediately scaled to zero. More scalability and significant cost reductions are the outcomes of this. According to research by Global industry Insights, the serverless industry is expected to reach $30 billion in market value by the end of the forecast period, growing at an above-average rate of 25% between 2021 and 2027.

1. INTRODUCTION

Serverless computing (SC) is a method of providing backend services on a need-based basis. Through serverless providers, users can create and execute applications without having to consider the underlying infrastructure. When a business uses a serverless vendor for backend services, the cost is determined by the vendor's computation; no set bandwidth or server count needs to be reserved or paid for because the service is auto-scaling. Even though it has a different name, the physical server is still used; Developers don't necessarily know this.

DOI: 10.4018/979-8-3693-1682-5.ch014

When the Web was born, anyone who wanted to build a Web application had to buy the expensive and time-consuming physical hardware needed to run a server.

In cloud computing, where one could rent a set number of servers or a set amount of server space remotely. In order to make sure that a surge in activity or traffic won't surpass their monthly allotment and disrupt their applications, developers and businesses that rent these fixed units of servers typically overbuy. It can therefore be the case that a large portion of the paid server space is wasted. To alleviate this problem, cloud manufacturers have come up with auto-scaling models. However, even with auto-scaling, unintended increases in activity, such as DDoS attacks (Bhupathi & Kathirvel, 2022; Kathirvel & Pavani, 2023; Kathirvel & Shobitha, 2023; Sudha & Kathirvel, 2022a; Sudha & Kathirvel, 2023), can impose significant costs. Developers use pre-packaged functions such as Google Cloud Functions and Microsoft Azure Functions to operate so-called serverless applications that either work Function as a service (FaaS) or computation as a service (CaaS).

1.1 Fundamentals of SC

Developers can purchase back-end services using serverless computing on a flexible "pay-as-you-go" basis, meaning they only pay for the services they need. This is like switching from your cell phone data plan that has a set monthly limit to a plan that only charges for the actual amount of data used.

Although these backend services are still provided by the server, the term "serverless" is a bit confusing because the provider manages all aspects of the infrastructure and server space. Developers don't have to worry about servers at all when their work is serverless.

1.2 Techniques in Serverless Computing

The two domains of application development are often separated into 1. frontend and 2. backend. The portion of the program, people can view and interact with SC, (i.e) the layout visually, is called the frontend (Kathirvel & Pavani, 2023). Portion of the program that is hidden from the user's view is called the backend, and it consists of the database and server that store user information and business logic as shown in the Figure 1.

Figure 1. Front end and back end server request