Serverless Computing

Summarized by: Natasha Kubiak

ECE 531

SUMMER 2022

Introduction to Serverless Computing

A serverless computer architecture is one in which a third-party BaaS and custom code run on a FaaS platform.This type of architecture is agnostic towards the typical server and benefits from lower costs, less complexity, and reduced engineering lead time. However the consequence is a reliance on these third party vendors. The three big vendors today include Amazon, Google, and Microsoft, and they have taken over the market.

What is it?

Serverless computing architecture can be described in two main themes: *Backend as a Service* (BaaS) and *Functions as a service* (Faas). We can think of BaaS as “rich client” application, which has cloud accessible database authentication services. FaaS can be thought of as being defined as cloud computing services in which a platform is provided where users can develop, run and manage their applications but without dealing or maintaining infrastructure, instead a third party vendor would manage that. In this summary the focus will mostly be on FaaS.

> “Fundamentally, FaaS is about running backend code without managing your own server systems or your own long-lived server applications. “ [1]

Looking further into FaaS we want to observe some key characteristics. FaaS functions has limitations in regards to state; it is not guaranteed a state will persist over multiple invocations, or be available from one invocation to another. FaaS can be described to be stateless, and any state of FaaS should be externalized from the FaaS function if it wants to be persistent. FaaS functions also have limits on run time, and long-lived executions may not be suitable. It’s also notable to define FaaS start up time, as start up and initialization may time time, or vary significantly depending on different variables.

Benefits

As mentioned in the summary, Serverless allows a user to reduce their own costs of operation. Essentially this is like outsourcing to a provider, which then does the server, database, and application management for you. A vendor is also able to provide lower costs because a user will be sharing the same resource with other users, in terms of infrastructure, and gain of labor back on the user since they do not need to do the management.

Drawbacks

Since serverless computing requires we outsource and consume from a third-party vendor, there can be some risks and drawbacks. When outsourcing to a vendor, this means a lot of the infrastructure and management is not directly controlled by you, but instead by the vendor. This control can come in the form of changes in price, loss of function, API upgrades, or limitations of use. Using serverless computing and leaving it up to the vendors can also have security risks, once your application is in the cloud it can be like a black box to the consumer as to what is going on behind the scenes. The vendors may be vulnerable to attack, and you may not be aware of who can see and have access to your data or not.

Conclusion

To conclude, Serverless computing can be beneficial when reducing costs and wanting less infrastructure and management on the user side. The benefits of reduced operation costs and development costs, plus operational management and possible reduced environmental impact make this type of computing very alluring. This may not be for everyone, as it could mean an overhaul of infrastructure with difficulty leaving or moving your environment once started. This type of computing is still new and in development and may prove to come with many new and enticing improvements in the future.

Sources:

[1] https://martinfowler.com/articles/serverless.html