




AWS Lambda and Serverless Computing

Veröffentlicht: 27. August 2016



Mark Looi
President

[Folgen](#)

 18

 1

 4

Serverless computing is purported to be the next big thing. This is a brief overview focusing on Amazon's AWS Lambda (courtesy of an Amazon webinar and some research).

AWS Lambda is a logical progression in enterprise-class computing, certainly in the Amazon model. When AWS first burst upon the scene a decade ago, it was innovative, but not primarily for technical breakthroughs. In fact, hardware virtualization, which it depended upon, was well underway in enterprise computing, with very successful companies such as VMWare leading the way. What was innovative was Amazon's ability to aggregate, meter and sell server capacity on-demand and with apparently limitless scale. This was the non-trivial innovation. And although customers were freed from the problems of dealing with physical hardware, management of servers still consumed human and other resources, with under-utilization a chronic issue.

Fast forward to the serverless present and not only is metering of computing resources at finer and finer granularities possible, all the burden and overhead of managing a server-based environment (virtual or no) is magically lifted. In a strange way, it's a throwback to the happy days of monolithic mainframe computing, when a programmer (we were programmers then, not developers or engineers) focused on code functionality and resources like memory, and let systems administrators worry about all the environmental issues. It's a liberating paradigm, but also terrifying to some, who have only gotten used to giving up physical control of a server!

Some key points (art courtesy of Amazon):

- Monolithic architectures dominated the first generation of applications development.
- By the late 1990's, the Service Oriented Architecture emerged. It broke down the monolith into three or more tiers: for example, presentation, logic, data, with suitable abstraction between them to reduce or eliminate sources of tight coupling:
- What seemed like a good thing eventually lead to the micro-services architecture (MSA), with much more of the above:
- Enabling such MSAs today are a panoply of frameworks, code libraries, services, etc., to facilitate the decoupled, service oriented architecture. Some of these are readily available as services from Microsoft, Amazon, Google and others. We are in the midst of a mature MSA paradigm, building and delivering sophisticated solutions relatively quickly.
- But, the dependency on servers still remains. In fact, an accompanying proliferation of specialized servers now becomes the pressing issue, with managing, balancing, tuning, forecasting, administering, etc., the different types of servers emerging as a key competence that some engineering teams either don't think about or want to think about.
- The ever decreasing size and scope of server granules coupled with their increasing specialization anticipates the move to serverless architectures.
- Some obvious advantages of serverless-ness: no provisioning, no administration, focus on business application, less code.
- Enter AWS Lambda: a serverless, event-driven compute service.
- Developers focus on
 - Writing a Lambda function (yes, it's a function) in Java, NodeJS, Python.
 - Defining the role that the code assumes during execution.
 - Specifying the memory allocated.
- Event sources trigger the function's execution; events could be driven by AWS services such as S3, DynamoDB, via an API, etc.
- AWS Lambda services manage the execution of these custom code functions, including providing an API to trigger them.
- Functions' networking environment can be customized, e.g., private IPs, etc.
- Lambda provides abstraction at the code/function level, a security model, and pricing at very fine levels of granularity.
- Scalability is unlimited, at least given enough lead time.
- First million function executions per month on AWS Lambda are free!

The disadvantage of serverless computing, in this model at any rate, is the almost certain dependency and tie-in to the AWS platform. It would be nearly impossible for an enterprise to deracinate and move an application. Additionally, though the cost can be very low at small scale, it will grow with volume and usage. There will also be some retraining as developers adapt to the different programmatic model. Inevitably, the first few functions will become monolithic! As noted, it's a bit of a throwback!

A key difference between a serverless Lambda function and traditional architectures is statelessness. This means applications have to be designed with this in mind, using perhaps cross-application caches or networked storage. Other constraints include startup latency and function execution duration (though these are likely to be ameliorated as the services mature).

AWS Lambda compares favorably with Google's Google App Engine, which attempted to abstract computing away from a pure server paradigm. However, though a bold and breakthrough approach at the time, it didn't offer the scope of AWS services or the granularity that Lambda now does.

In the market today, AWS Lambda is not really challenged by Microsoft Azure as yet (there are, however, Azure Functions). Google has an early offering, Cloud Platform. AWS again leads the way with the most complete and comparatively mature services. In this respect, its conservative evolution from virtualized servers in EC2 to serverless computing has been what the relatively risk-averse enterprise computing market has been willing to tolerate.


Serverless computing has arrived and while much more refinement will be seen in the coming years, especially around tools and removing some of the constraints, it will alter mostly for the better computing as we know it.







Getaggt in: [server architecture](#), [computing](#), [amazon web services \(aws\)](#)




Mark Looi
President


[22 Artikel](#)

[Folgen](#)

1 Kommentar Neueste





[Loggen Sie sich ein](#), um einen Kommentar zu hinterlassen

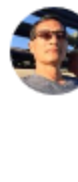


Raphael Morozov
Technologist, Entrepreneur, Dad

Interesting write up, though your implication that a microservices based architecture is by definition not serverless is inaccurate. Amazon's API Gateway in combination with Lambda can be used to accomplish just that. A relatively comprehensive example of it can be found on the AWS blog here: <https://aws.amazon.com/blogs/compute/microservices-without-the-servers/>

I also... [Mehr anzeigen](#)

Gefällt mir Antworten |  2  1



Mark Looi
President

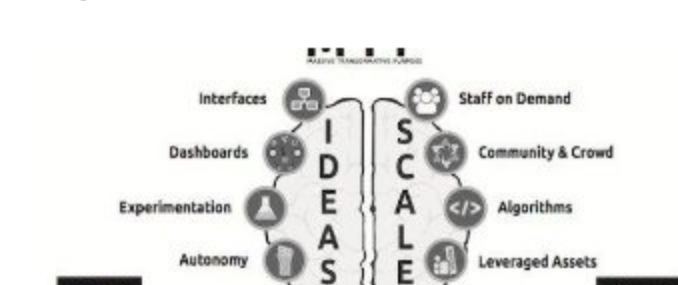
Thanks for your comments; it's great to see your thinking. Appreciate your taking the time to read my post in the first instance!

As to your point about using the API Gateway and Lambda to build a serverless micro-service is interesting, but even more telling that it's built on top of a serverless service itself!


Indeed, it's early in the serverless chapter of computing. I... [Mehr anzeigen](#)

Gefällt mir Antworten


Verpassen Sie keine Artikel mehr von Mark Looi



Organizations of the Future
Mark Looi auf LinkedIn



Leading Digital Transformation at L'Oréal
Mark Looi auf LinkedIn



What Makes a New Market?
Mark Looi auf LinkedIn

Sie möchten noch weitere aktuelle News auf LinkedIn sehen?

[Mehr Artikel entdecken](#)