AWS Lambda and #serverless. What’s all the fuzz about?

Joint work with
Erik Wittern and Waldemar Hummer
Josef Spillner

Philipp Leitner
philipp.leitner@chalmers.se
@xLeitix
We are hiring a postdoc!!

https://icet-lab.eu

@IcetLab
Function-as-a-Service?
Function-as-a-Service?
Serverless?
Upload your code to AWS Lambda

Source: https://aws.amazon.com/lambda/
**Source:** https://aws.amazon.com/lambda/
Upload your code to AWS Lambda

Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity

Lambda runs your code only when triggered, using only the compute resources needed

Source: https://aws.amazon.com/lambda/
Source: https://aws.amazon.com/lambda/
An Example: **Real-Time Tweet Processing**

Source: https://github.com/aws-samples/lambda-refarch-streamprocessing
Internally

Source: loosely based on IBM’s OpenWhisk architecture
No state
No state

Hard cap on max. execution time
No state

Hard cap on max. execution time

Extreme tail latency
A Mixed-Method Empirical Study of Function-as-a-Service Software Development in Industrial Practice

Philipp Leitner\textsuperscript{a,}\textsuperscript{*}, Erik Wittern\textsuperscript{b}, Josef Spillner\textsuperscript{c}, Waldemar Hummer\textsuperscript{b}

\textsuperscript{a}Software Engineering Division, Chalmers / University of Gothenburg, Sweden
\textsuperscript{b}IBM Research, Yorktown Heights, New York, USA
\textsuperscript{c}Service Prototyping Lab, Zurich University of Applied Sciences, Switzerland

Abstract

Function-as-a-Service (FaaS) describes cloud computing services that make infrastructure components transparent to application developers, thus falling in the larger group of “serverless” computing models. When using FaaS offerings, such as AWS Lambda, developers provide atomic and short-running code for their functions, and FaaS providers execute and horizontally scale them on-demand. Currently, there is no systematic research on how developers use serverless, what types of applications lend themselves to this model, or what architectural styles and practices FaaS-based applications are based on. We present results from a mixed-method study, combining interviews with advanced practitioners, a systematic analysis of peer literature, and a Web-based survey. We find that successfully adopting FaaS requires a
Methodology

- Interviews (n = 12)
- Analysis of Grey Literature (n = 50)
- Web Survey (n = 182)
Main Findings

Compositional Application Model

“Microservices on steroids”
Recall: Real-Time Tweet Processing

Source: https://github.com/aws-samples/lambda-refarch-streamprocessing
Compositional Application Model

"I think the term 'application' is oftentimes not really that applicable anymore (...) it's really hard to say, like, what is the application anymore [and what is part of the cloud or infrastructure]." -I6

"AWS API Gateway, S3, Kinesis, SNS, DynamoDB, Step-Functions, or their Azure and GCP siblings — are at play with any serverless solution" -A9
Main Findings

Compositional Application Model

Use Cases
Main Findings

But:
many challenges in user-facing apps
## Use Cases - Backend

### What do you use FaaS for in the backend?

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process application data (e.g., transform images)</td>
<td>72</td>
<td>76%</td>
</tr>
<tr>
<td>Perform scheduled jobs (e.g., backups, notifications)</td>
<td>61</td>
<td>64%</td>
</tr>
<tr>
<td>Process monitoring or telemetry data</td>
<td>37</td>
<td>39%</td>
</tr>
<tr>
<td>I’m not using it for backend tasks</td>
<td>7</td>
<td>7%</td>
</tr>
<tr>
<td>Other</td>
<td>6</td>
<td>6%</td>
</tr>
</tbody>
</table>
Main Findings

Compositional Application Model

Use Cases

Challenges
### Challenges

Which of the following do you consider significant challenges for using FaaS services?

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Lack of tooling (e.g., testing, deployment)</td>
<td>51 / 55%</td>
</tr>
<tr>
<td>2: Integration testing</td>
<td>37 / 40%</td>
</tr>
<tr>
<td>3: Vendor lock-in</td>
<td>30 / 32%</td>
</tr>
<tr>
<td>4: Container start-up latency</td>
<td>27 / 29%</td>
</tr>
<tr>
<td>5: Managing state in functions</td>
<td>25 / 27%</td>
</tr>
<tr>
<td>6: Unit testing</td>
<td>17 / 18%</td>
</tr>
<tr>
<td>7: Little support for reusing functions</td>
<td>13 / 14%</td>
</tr>
<tr>
<td>8: Lack of documentation</td>
<td>12 / 13%</td>
</tr>
<tr>
<td>9: Finding/hiring developers familiar with FaaS</td>
<td>11 / 12%</td>
</tr>
<tr>
<td>10: Little support for composition of functions</td>
<td>11 / 12%</td>
</tr>
<tr>
<td>11: CPU or processing limitations</td>
<td>8 / 9%</td>
</tr>
<tr>
<td>12: Memory limitation</td>
<td>5 / 5%</td>
</tr>
<tr>
<td>13: Other</td>
<td>3 / 3%</td>
</tr>
</tbody>
</table>
The Future (as we see it)

“Faster horses” vs Different Services

(end user view)      (provider view)
The Future (as we see it)

Different Services

“Function-as-a-Service as the assembly language of the cloud”
The Future (as we see it)

Different Services

“Function-as-a-Service as the assembly language of the cloud”

"We will have languages that compile something that you can execute in a serverless platform." -I1
A mixed-method empirical study of Function-as-a-Service software development in industrial practice

Philipp Lethner, Erik Witsen, Josef Spillner, Waldemar Hummer

June 26, 2018

Author and article information

Abstract

Function-as-a-Service (FaaS) describes cloud computing services that make infrastructure components transparent to application developers, thus allowing for dynamic scaling and pay-as-you-go billing. The FaaS development process is emerging as a new software development paradigm, and this paper presents a mixed-method empirical study of FaaS software development in industry.

https://peerj.com/preprints/27005/