Serverless Computing: Economic and Architectural Impact

2017

Authors

Gojko Adzic
Neuri Consulting LLP

Robert Chatley
Imperial College London

Group 3:
Yizhou Pan
Jujie Bai
Yicun Zeng

Talk Outline

- Paper Overview
- Background
- Related Work
- Benchmarks
- Author's Evaluation (Case Studies)
- Author's Conclusions
- Critique
- Questions

Paper Overview: *Serverless computing - economics and architecture impact*

- Introduction of serverless computing and AWS Lambda
- Economics and architectural impacts
- Two companies case studies
- Potential opportunities created by serverless computing
- Limitations of serverless computing
- Conclusion

Paper Overview (Continued)

Compare with traditional client/server architecture

- Economics Impact:

  Lower cost than EC2 for renting the same memory

  Different Services Billed According to different utilisation metrics

Service instance	Billable unit	Unit cost (USD)	Fail-over costs (%)	Cost of 12 x 200ms exec'ns	% reference price
Lambda (128 MB)	100 ms	$0.000000208	included	$0.000004992	24.94%
Lambda (512 MB)	100 ms	$0.000000834	included	$0.000020016	100.00%
Heroku Hobby (512 MB)	1 month	$7.00	100%	$0.0097222222	48572.25%
AWS EC2 t2.nano (512 MB)	1 hour	$0.0059	100%	$0.0118	58952.84%
AppEngine B1 (128MB)	1 hour	$0.05	100%	$0.1	499600.32%
AppEngine B4 (512MB)	1 hour	$0.20	100%	$0.4	1998401.28%

Paper Overview (Continued)

- Architectural Impact:

  Have to apply distributed, request-level authorization.

  Allow client applications to directly access “back-end” resources

Background

“Serverless” Computing

“A new generation of platform-as-a-service offering by major cloud providers.”

Infrastructure providers take responsibility for receiving client requests and responding to them, capacity planning, task scheduling and operational monitoring

Developers - only worry about the logic for processing client requests and pay for CPU time when functions are executing

Background (Continued)

Lambda - Function as a service (FaaS)

First announced at the end of 2014, and saw significant adoption in mid to late 2016

Other similar services: Google Cloud Functions, Azure Functions and IBM Cloud Functions

Lambda provides function instances that exist only for the duration to run a function (low memory instances.)

Resources will scale up automatically

Developers have no control of the server process

Related Work

-- Gojko Adzic. 2017. The key lesson from our serverless migration. Accessed: 2017-04-20

-- Mario Villamizar, Oscar Garces, Lina Ochoa, Harold Castro, Lorena Salamanca, Mauricio Verano, Rubby Casallas, Santiago Gil, Carlos Valencia, Angee Zambrano, and Mery Lang. 2017. Cost Comparison of Running Web Applications in the Cloud Using Monolithic, Microservice, and AWS Lambda Architectures. *Service Oriented Computing and Application*

Benchmarks

Case Studies:

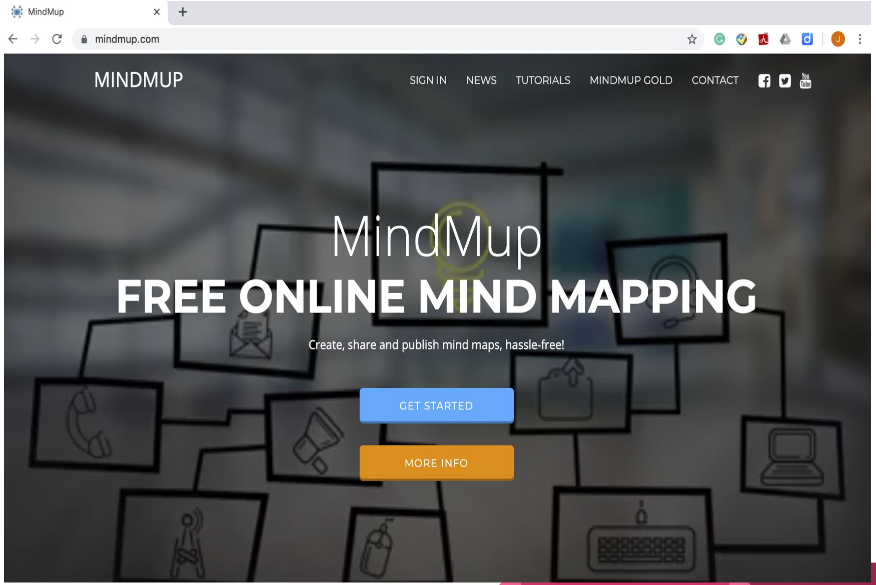
MindMup - A collaboration platform that moved from Heroku to AWS Lambda in 2016

Yubl - London-based social networking company which had migrated large parts of their backend systems to Lambda in 2016

Case Study: MindMup - Introduction

- A commercial mind mapping application (https://mindmup.com)
- A open source project at the beginning (2013 - 2015)
- Has Google Drive connection and Social Network ports

Case Study: MindMup - Introduction



Case Study: MindMup - Major Modifications

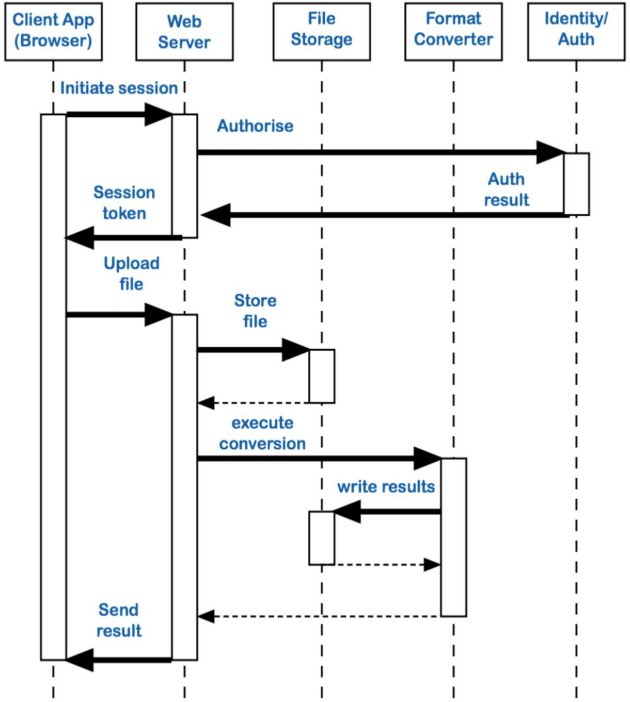


Figure 1: MindMup file conversion with a server

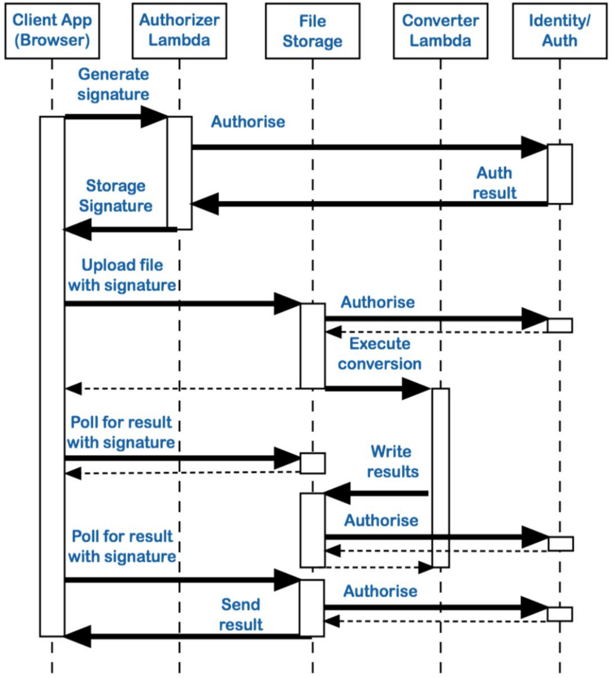


Figure 2: MindMup file conversion with Lambda

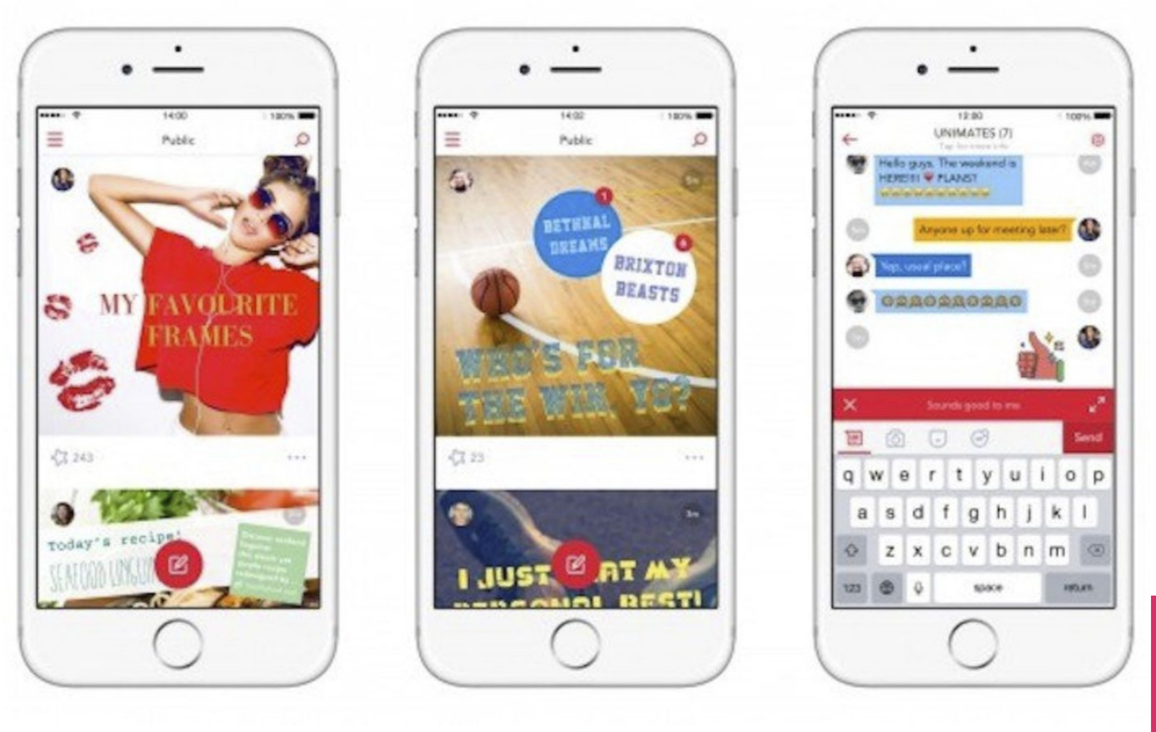
Case Study: MindMup - Impacts

- **Engineering Impacts:**
  - Reduced boiler-plate code
  - Reduced service bundling
- **Economic Impacts:**
  - Reduced hosting cost - by 66%, while user increase by 50%

Case Study: Yubl - Introduction

- A social networking mobile application
- Used to be favoured by teenagers and TV stars, but shutdown in 2016.

Case Study: Yubl - Introduction



Yubl

Case Study: Yubl - Major Modifications

- From EC2 to Lambda
- From Monolithic to Microservice

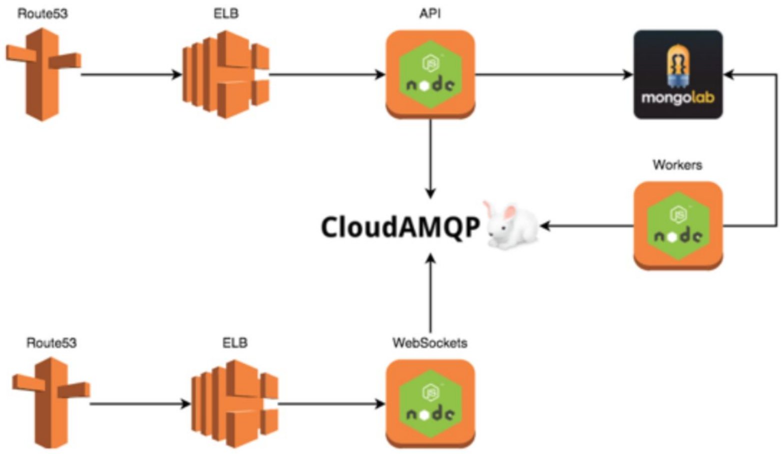


Figure 3: Yubl system architecture, before migration.

Case Study: Yubl - Impacts

- **Engineering Impacts:**
  - Improves teamwork: enables development of functions in parallel
  - Makes feature deployment easier and faster
  - Makes auto-scaling easier and more cost-effective
- **Economics Impacts:**
  - Reduced hosting cost by 95%, from $5000 to less than $200 per month

Conclusions: Impacts

- **An overall much lower hosting cost**
    - Services were billed according to different Metrics
    - Distributed request-level authorization

- **Less Incentives for bundling**
- **Less Barriers for versioning**
- **Less Workload: auto-scaling, auto failover**

Conclusions: Engineering Impacts

- **Less incentives for bundling**
- **Less barriers for versioning**
- **Less workload: auto-scaling, auto failover (also less cost)**

Author’s Conclusions: Opportunities created

- An overall much lower hosting cost
  - Services were billed according to different Metrics
  - Distributed request-level authorization
- Remove incentives for bundling
  - No virtual machine provisioning, scaling and monitoring
- Removing barriers for versioning
  - A/B Testing based on different version of lambda function

Author’s Conclusions: Weakness of AWS Lambda

- No Strong service-level agreement
  - 99.95% - 99.0% - 10% credit
  - 99.0% - 95.0% - 25% credit
  - <95.0% - 100% credit
- Potentially high latency
- No compliance(SOC PCI FedRamp HIPAA)
- Relatively short life-span
- No local execution environment
- Vendor lock-in

Critique: Strengths

- Concrete economic comparison between serverless computing and on-premise.
- Summary or architectural change before and after AWS Lambda.
- Actual study case in production environment, MindMup and Yubl

Critique: Weaknesses

- Paper is more about learning AWS Lambda, serverless computing is a more general idea of back-end service
- Study case lacks metrics comparison so that advantage of Lambda usage is not clear

Critique: Evaluation

Paper gives a detailed explanation of AWS Lambda’s economic and architecture influence. This is helpful to potential Lambda users in application design stage.

As paper points out, SLA(service-level agreement), high latency, compliance, short life-span, lack of local execution environment, vendor lock-in are the weaknesses of AWS Lambda.

Critique: GAPS

AWS Lambda is part of serverless computing.

If the title is “serverless”, paper should introduce more tools in serverless computing.

Future Work

- Comparison with Google Cloud Functions and Azure Functions
- Gather more study cases from industry to compare between Lambda and client-server architecture.

Questions