Cloud Computing
Serverless Computing

Minchen Yu
SDS@CUHK-SZ
Fall 2024

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen
SCHOOL OF DATA SCIENCE
數據科學學院

Outline

- Introduction to serverless
- Evolution of virtualization
- Case study
- Limitations

Why serverless?

In traditional IaaS, e.g., AWS EC2

EC2Instances.info Easy Amazon **EC2** Instance Comparison

Tweet | Star

Proudly sponsored by Vantage

Last Update: 2021-04-10 00:48:35 UTC

EC2 | RDS

Region: US East (N. Virginia) | Cost: Hourly | Reserved: 1-year - No Upfront | Columns | Compare Selected | Clear Filters | CSV

Filter: Min Memory (GiB): 0 Min vCPUs: 0 Min Storage (GiB): 0 **Search:**

Name	API Name	Memory	vCPUs	Instance Storage	Network Performance	Linux On Demand cost	Linux Reserved cost	Windows On Demand cost	Windows Reserved cost
Search	Search	Search	Search	Search	Search	Search	Search	Search	Search
M5DN Extra Large	m5dn.xlarge	16.0 GiB	4 vCPUs	150 GiB NVMe SSD	Up to 25 Gigabit	$0.272000 hourly	$0.171000 hourly	$0.456000 hourly	$0.355000 hourly
M5A Double Extra Large	m5a.2xlarge	32.0 GiB	8 vCPUs	EBS only	Up to 10 Gigabit	$0.344000 hourly	$0.217000 hourly	$0.712000 hourly	$0.585000 hourly
R5B Extra Large	r5b.xlarge	32.0 GiB	4 vCPUs	EBS only	Up to 10 Gigabit	$0.298000 hourly	$0.187740 hourly	$0.482000 hourly	$0.371740 hourly
R5N 12xlarge	r5n.12xlarge	384.0 GiB	48 vCPUs	EBS only	50 Gigabit	$3.576000 hourly	$2.253000 hourly	$5.784000 hourly	$4.461000 hourly
R5AD Extra Large	r5ad.xlarge	32.0 GiB	4 vCPUs	150 GiB NVMe SSD	Up to 10 Gigabit	$0.262000 hourly	$0.165000 hourly	$0.446000 hourly	$0.349000 hourly
R5N Extra Large	r5n.xlarge	32.0 GiB	4 vCPUs	EBS only	Up to 25 Gigabit	$0.298000 hourly	$0.188000 hourly	$0.482000 hourly	$0.372000 hourly
R5DN Extra Large	r5dn.xlarge	32.0 GiB	4 vCPUs	150 GiB NVMe SSD	Up to 25 Gigabit	$0.334000 hourly	$0.210000 hourly	$0.518000 hourly	$0.394000 hourly
I2 Extra Large	i2.xlarge	30.5 GiB	4 vCPUs	800 GiB SSD	Moderate	$0.853000 hourly	$0.424000 hourly	$0.973000 hourly	$0.565000 hourly
M5N 16xlarge	m5n.16xlarge	256.0 GiB	64 vCPUs	EBS only	75 Gigabit	$3.808000 hourly	$2.399000 hourly	$6.752000 hourly	$5.343000 hourly
T2 Micro	t2.micro	1.0 GiB	1 vCPUs for a 2h 24m burst	EBS only	Low to Moderate	$0.011600 hourly	$0.007200 hourly	$0.016200 hourly	$0.011800 hourly
D2 Eight Extra Large	d2.8xlarge	244.0 GiB	36 vCPUs	48000 GiB (24 * 2000 GiB HDD)	10 Gigabit	$5.520000 hourly	$3.216000 hourly	$6.198000 hourly	$3.300000 hourly
INF1 Extra Large	inf1.xlarge	8.0 GiB	4 vCPUs	EBS only	Up to 25 Gigabit	$0.368000 hourly	$0.232000 hourly	unavailable	unavailable
R6GD 16xlarge	r6gd.16xlarge	512.0 GiB	64 vCPUs	3800 GiB (2 * 1900 GiB NVMe SSD)	25 Gigabit	$3.686400 hourly	$2.322400 hourly	unavailable	unavailable
X1E 16xlarge	x1e.16xlarge	1952.0 GiB	64 vCPUs	1920 GiB SSD	10 Gigabit	$13.344000 hourly	$8.223000 hourly	$16.288000 hourly	$11.167000 hourly
R5N 24xlarge	r5n.24xlarge	768.0 GiB	96 vCPUs	EBS only	100 Gigabit	$7.152000 hourly	$4.506000 hourly	$11.568000 hourly	$8.922000 hourly
I2 Eight Extra Large	i2.8xlarge	244.0 GiB	32 vCPUs	6400 GiB (8 * 800 GiB SSD)	10 Gigabit	$6.820000 hourly	$3.392000 hourly	$7.782000 hourly	$4.521000 hourly
R5A Eight Extra Large	r5a.8xlarge	256.0 GiB	32 vCPUs	EBS only	Up to 10 Gigabit	$1.808000 hourly	$1.139000 hourly	$3.280000 hourly	$2.611000 hourly
R6G Medium	r6g.medium	8.0 GiB	1 vCPUs	EBS only	Up to 10 Gigabit	$0.050400 hourly	$0.031800 hourly	unavailable	unavailable
R6G 12xlarge	r6g.12xlarge	384.0 GiB	48 vCPUs	EBS only	20 Gigabit	$2.419200 hourly	$1.524100 hourly	unavailable	unavailable
A1 Metal	a1.metal	32.0 GiB	16 vCPUs	EBS only	Up to 10 Gigabit	$0.408000 hourly	$0.257000 hourly	unavailable	unavailable
T4G Micro	t4g.micro	1.0 GiB	2 vCPUs for a 2h 24m burst	EBS only	Up to 5 Gigabit	$0.008400 hourly	$0.005300 hourly	unavailable	unavailable
R5B Large	r5b.large	16.0 GiB	2 vCPUs	EBS only	Up to 10 Gigabit	$0.149000 hourly	$0.093870 hourly	$0.241000 hourly	$0.185870 hourly
I2 Double Extra Large	i2.2xlarge	61.0 GiB	8 vCPUs	1600 GiB (2 * 800 GiB SSD)	High	$1.705000 hourly	$0.848000 hourly	$1.946000 hourly	$1.131000 hourly
M5A Extra Large	m5a.xlarge	16.0 GiB	4 vCPUs	EBS only	Up to 10 Gigabit	$0.172000 hourly	$0.108000 hourly	$0.356000 hourly	$0.292000 hourly
P3 Double Extra Large	p3.2xlarge	61.0 GiB	8 vCPUs	EBS only	Up to 10 Gigabit	$3.060000 hourly	$2.088000 hourly	$3.428000 hourly	$2.456000 hourly
C6GN Eight Extra Large	c6gn.8xlarge	64.0 GiB	32 vCPUs	EBS only	50 Gigabit	$1.382400 hourly	$0.872510 hourly	unavailable	unavailable

Challenges for cloud users

- What type of instances to use?
- How many to spin up?
- What base image?
- How to configure environment?
- What price spot?

……

Cloud users (e.g., developers) need to work on many server-related issues, making it hard to deploy applications in the cloud

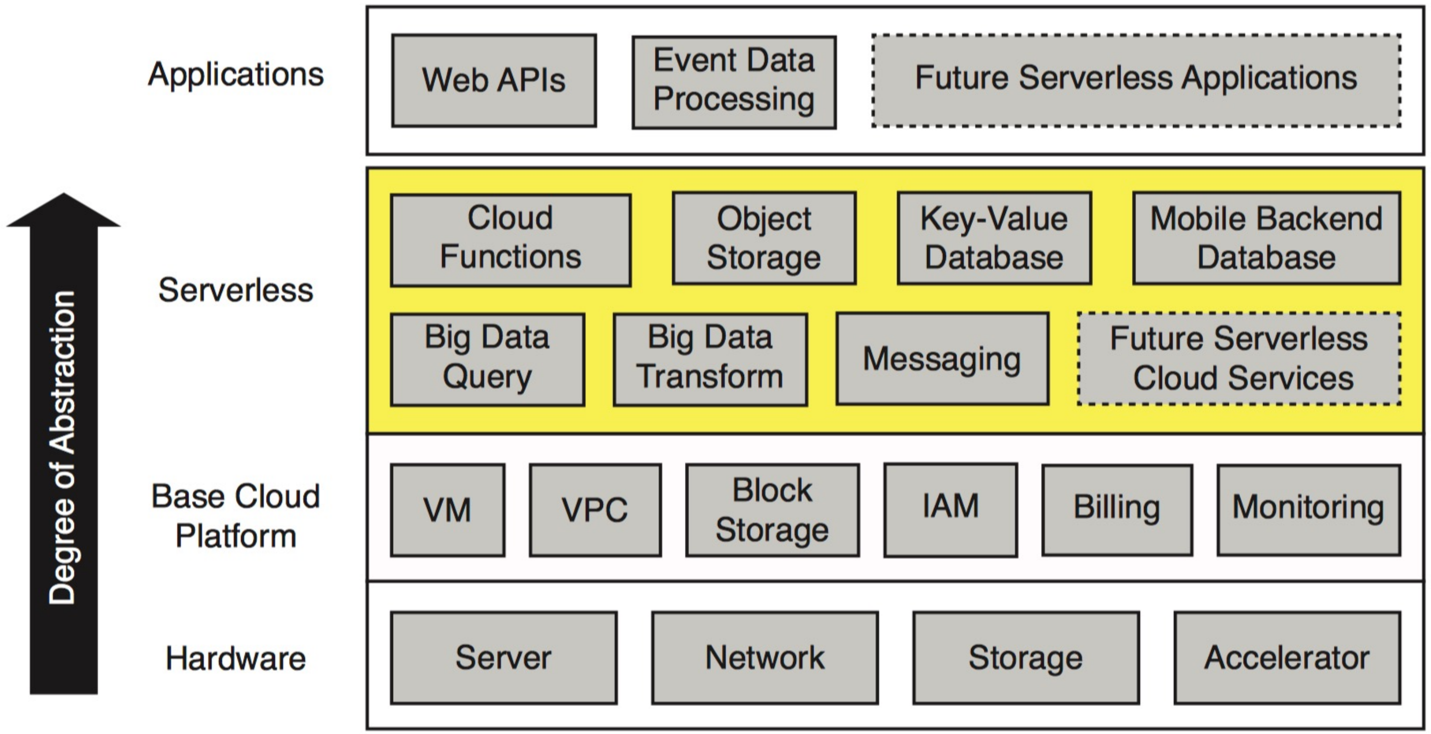
Serverless computing makes the cloud easier to use!

What is serverless computing?

- A new paradigm of cloud computing
  - No provisioning, reserving and configuring
  - Automatic and fast scaling from 0 (and back)
  - Pay per use, fine-grained billing

Cloud users are agnostic to servers, liberating them from managing resources!

Serverless in the cloud



Source: E. Jonas et al. “Cloud Programming Simplified: A Berkeley View on Serverless Computing”

Serverless offerings

- Backend-as-a-Service (BaaS)
  - Domain-specific

e.g., AWS S3 (object storage) and DynamoDB (key-value database)

Function-as-a-Service (FaaS)

- Bounded-time “functions” w/o persistent state across invocations

In most cases, Serverless = FaaS + BaaS

Compute Storage

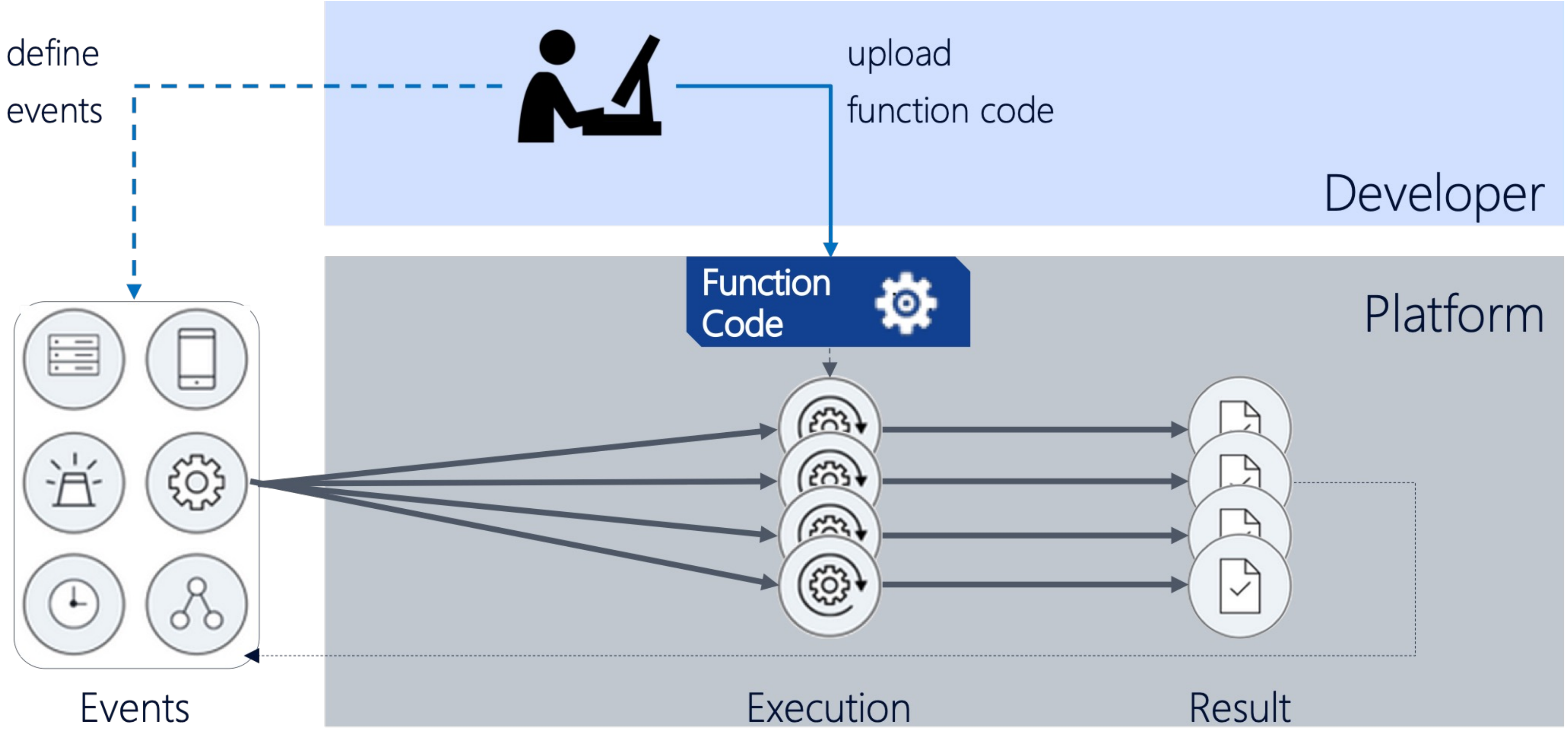
Function-as-a-Service (FaaS)

- Using cloud functions
  - Users upload function code and get an endpoint from the serverless provider
  - Functions get executed when “triggered” (e.g., HTTP request, timer, or other event sources)

“cloud user simply writes the code and leaves all the server provisioning and administration tasks to the cloud provider”[1]

[1] Eric Jonas et al. “Cloud Programming Simplified: A Berkeley View on Serverless Computing”

FaaS Overview



Source: KNIX@Nokia Bell Lab

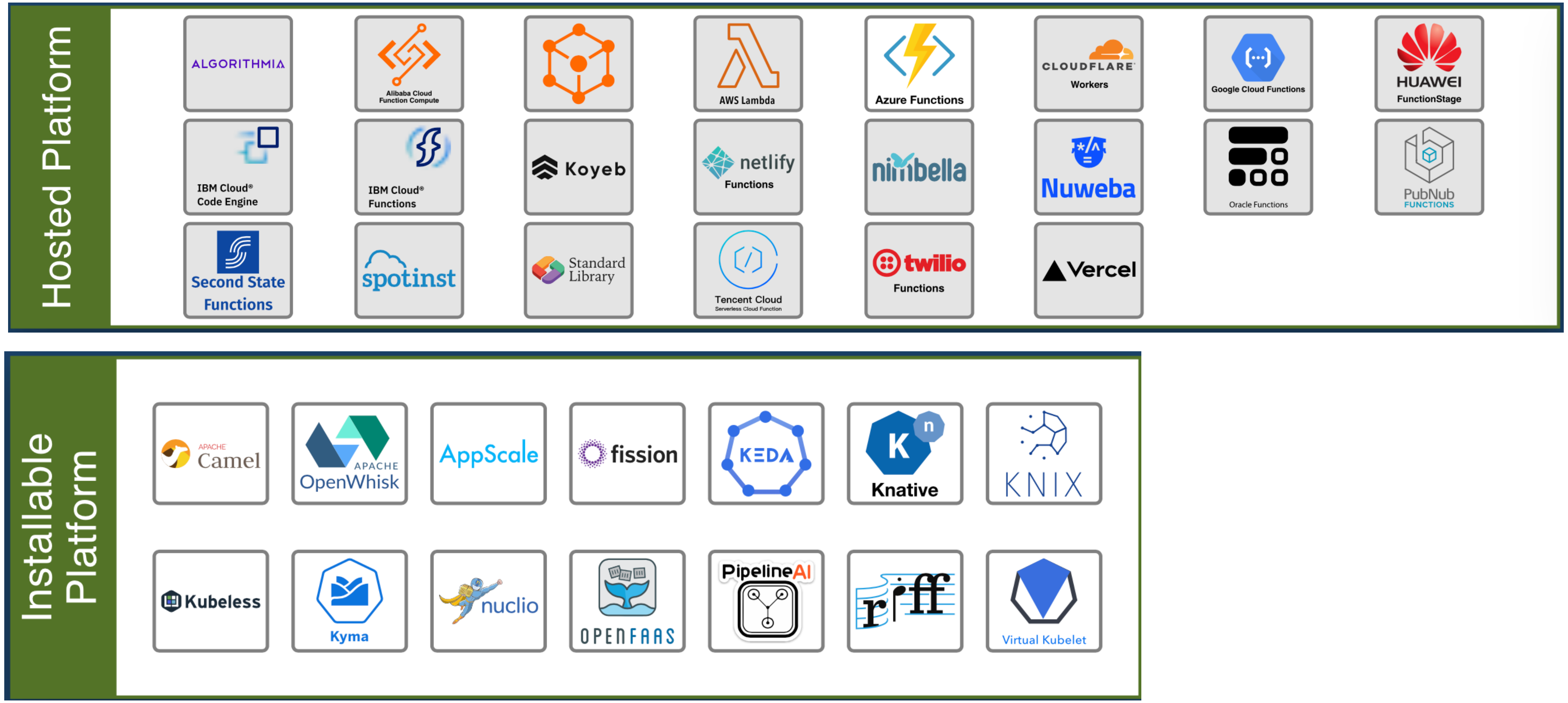
Serverless vs. Serverful

Characteristic	**Serverful (IaaS)**	**Serverless (FaaS)**
Server Instance	Cloud users select	Cloud providers select
Operating System & Libraries	Cloud users select	Cloud providers select
Deployment	Cloud users handle	Cloud providers handle
Fault Tolerant	Cloud users handle	Cloud providers handle
Monitoring	Cloud users handle	Cloud providers handle
Scaling	Cloud users handle	Cloud providers handle
Billing	Per allocation	Per use
Charging Basis	Per second to per hour	Per millisecond

Source: J. Schleier-Smith et al. "The Next Phase of Cloud Computing," in Commun. ACM, 64(5), 2021.

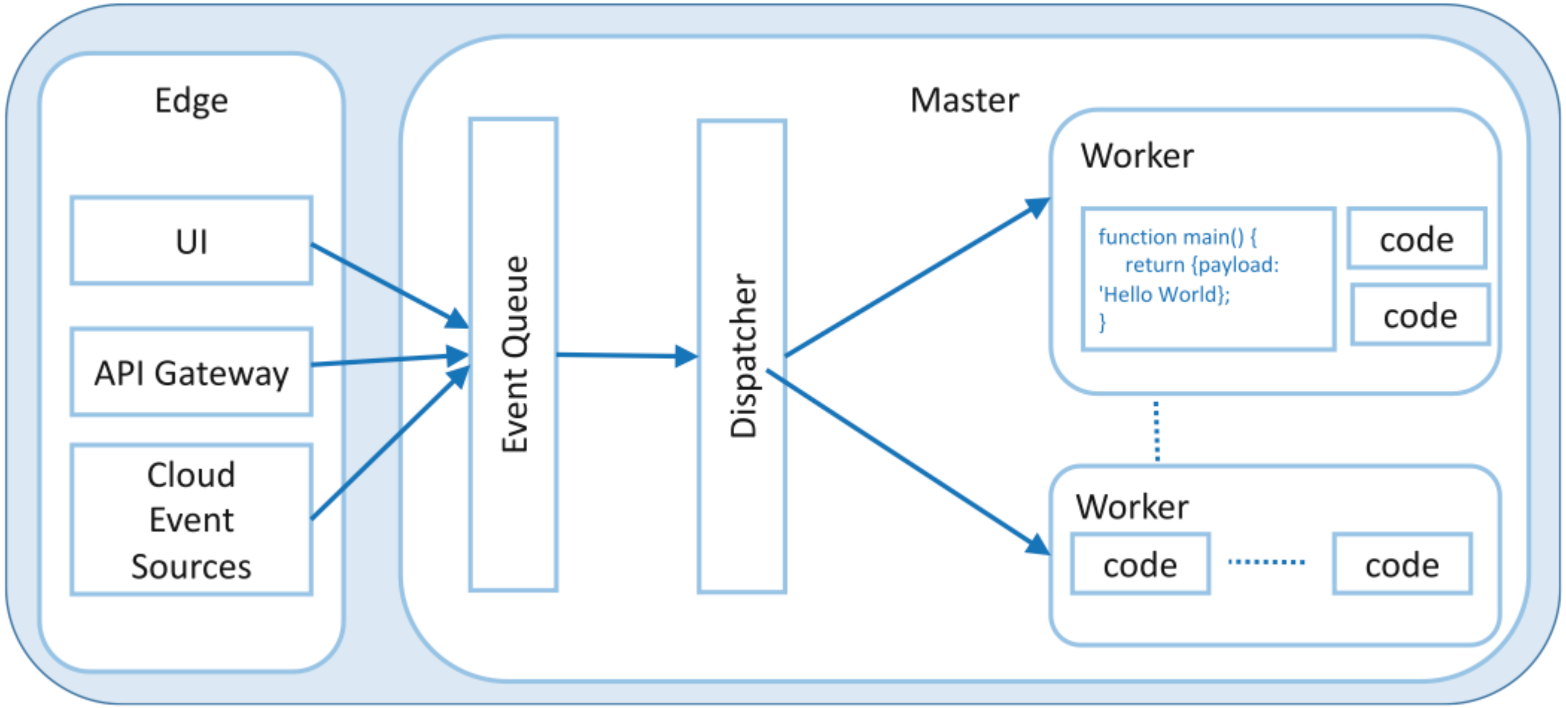
Serverless platforms

Cloud Native Computing Foundation (CNCF) serverless landscape



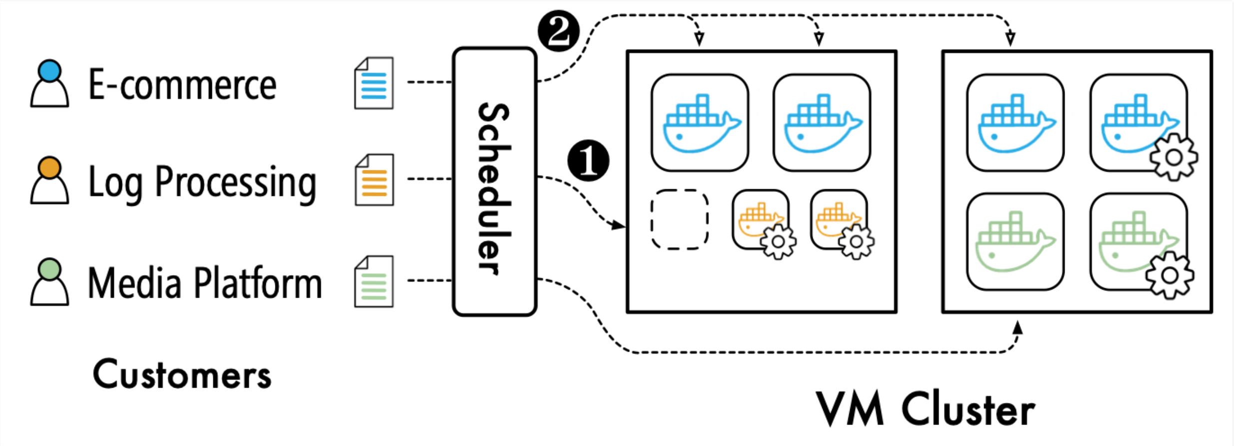
Source: https://landscape.cncf.io/serverless

Basic serverless architecture



Source: Ioana Baldini et al. “Serverless Computing: Current Trends and Open Problems”

Function placement and request scheduling



1. **Placement**: Launch a new instance in a worker

2. **Routing**: route the requests to some existing instances for processing

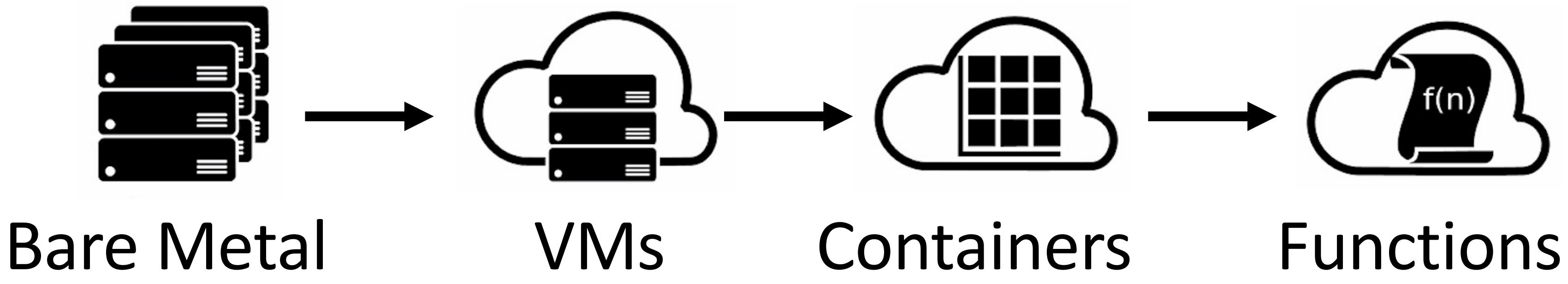
Core functionality

- Manage a set of user-defined functions
- Take an event received from an event source (e.g., HTTP)
- Determine which function(s) to which to dispatch the event
- Find an existing function instance or create a new instance
- Send the event to the function instance
- Wait for a response
- Gather execution logs
- Make the response available to the user
- Stop the function when it is no longer needed

Serverless platforms

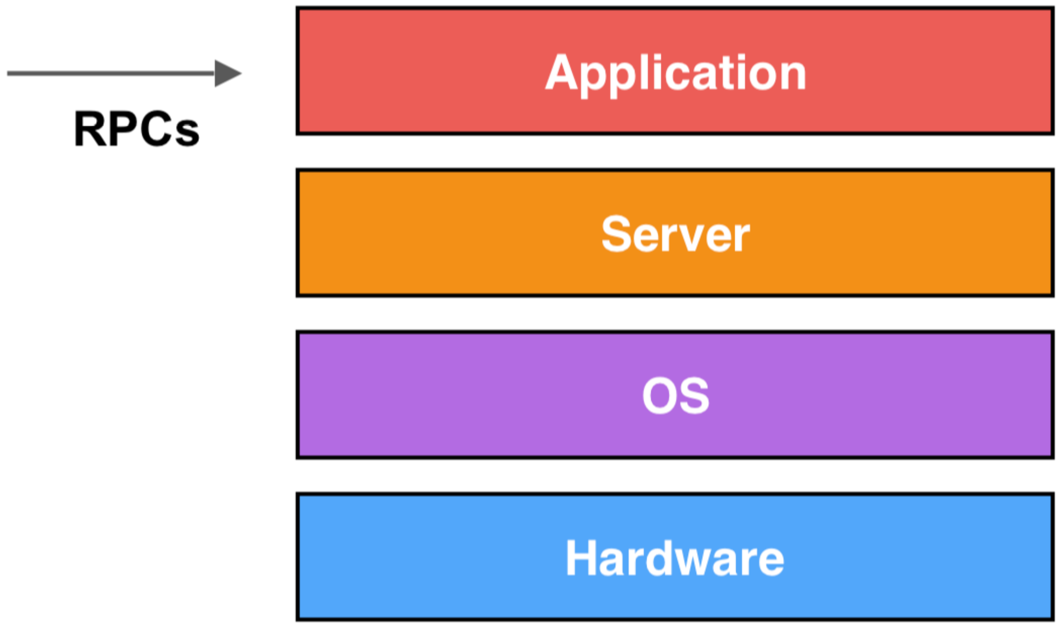
- Different platforms can have different implementations. Some metrics need to be considered in implementation, including scalability, cost and fault tolerance. For example,
  - Quickly schedule and start the execution of functions
  - Efficiently manage the resources
  - Carefully handle failures

Evolution of virtualization



	Bare Metal	VMs (IaaS)	Containers	Functions (FaaS)
Unit of scale	Server	VM	Application/Pod	Function
Provisioning	Ops	DevOps	DevOps	Serverless provider
Initialization time	Days	~1 min	Few seconds	Few seconds -> 10-100 ms
Scaling	Buy new hardware	Allocate new VMs	1 to many, auto	0 to many, auto
Typical lifetime	Years	Hours	Minutes	O(100ms) to O(10s)
Payment	Per server	Per allocation	Per allocation	Per use
State	Anywhere	Anywhere	Anywhere	Elsewhere

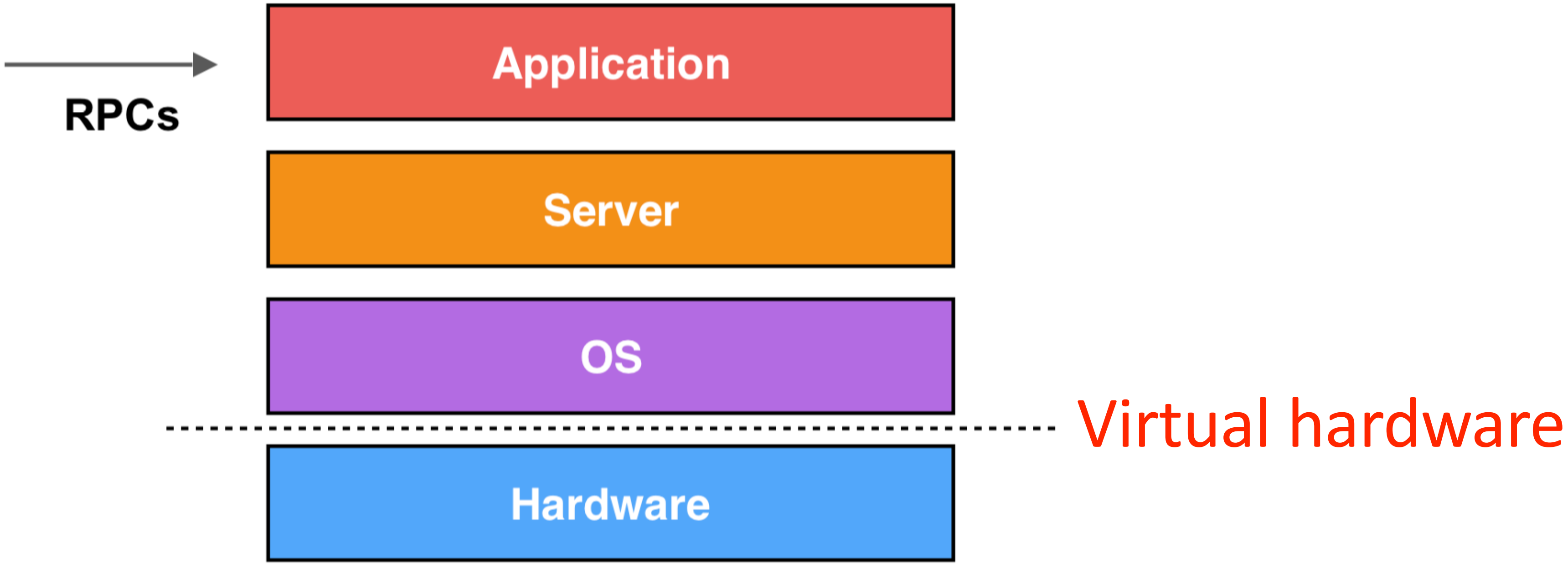
Example: classic web stack



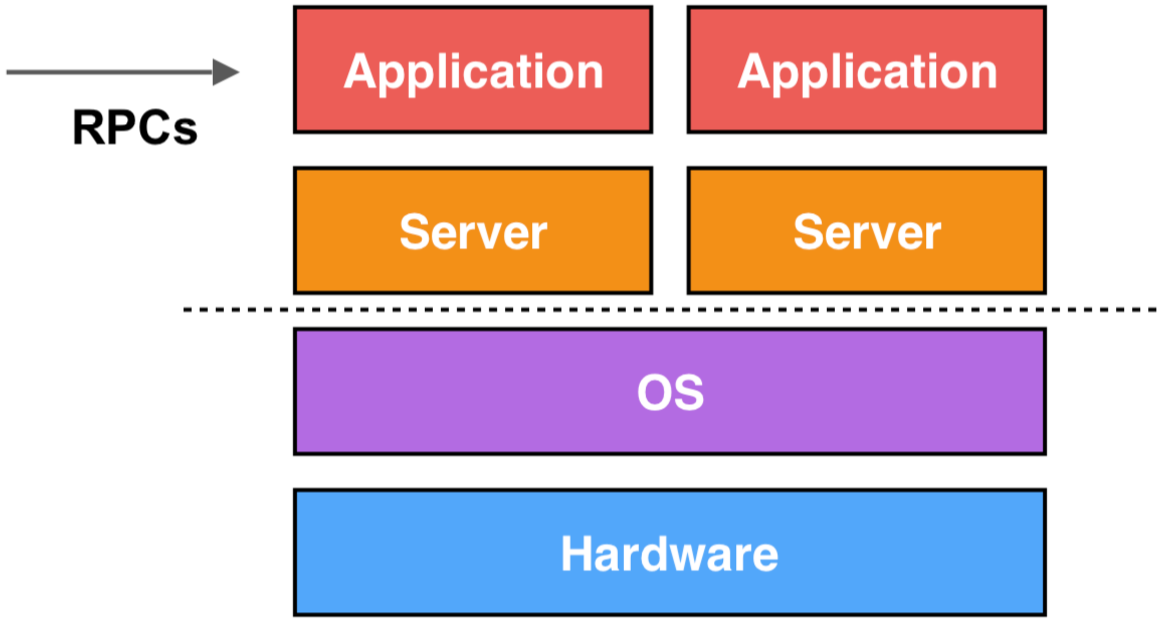
Bare metal

Source: "Serverless Computation with OpenLambda" https://www.usenix.org/node/196323

1st Generation: Virtual Machines

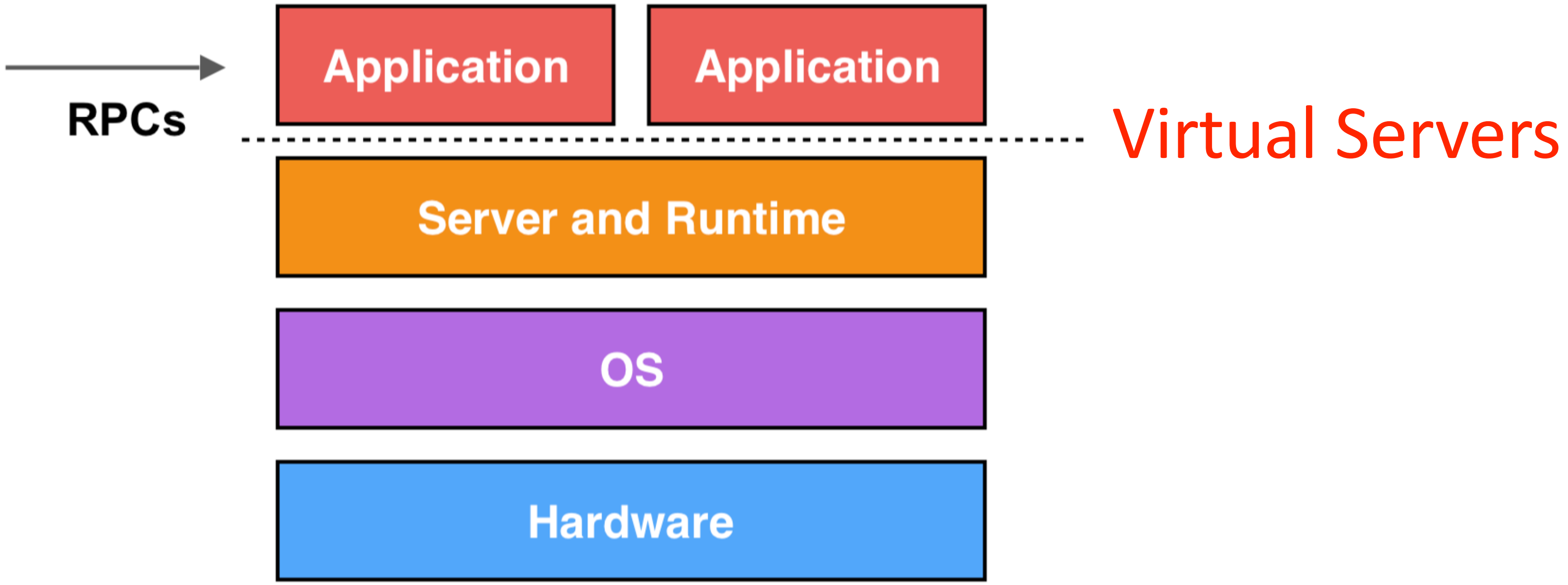


2nd Generation: Containers



Virtual OS

3rd Generation: Functions



Tradeoffs

Functions Containers VMs

Isolation

low

Flexibility

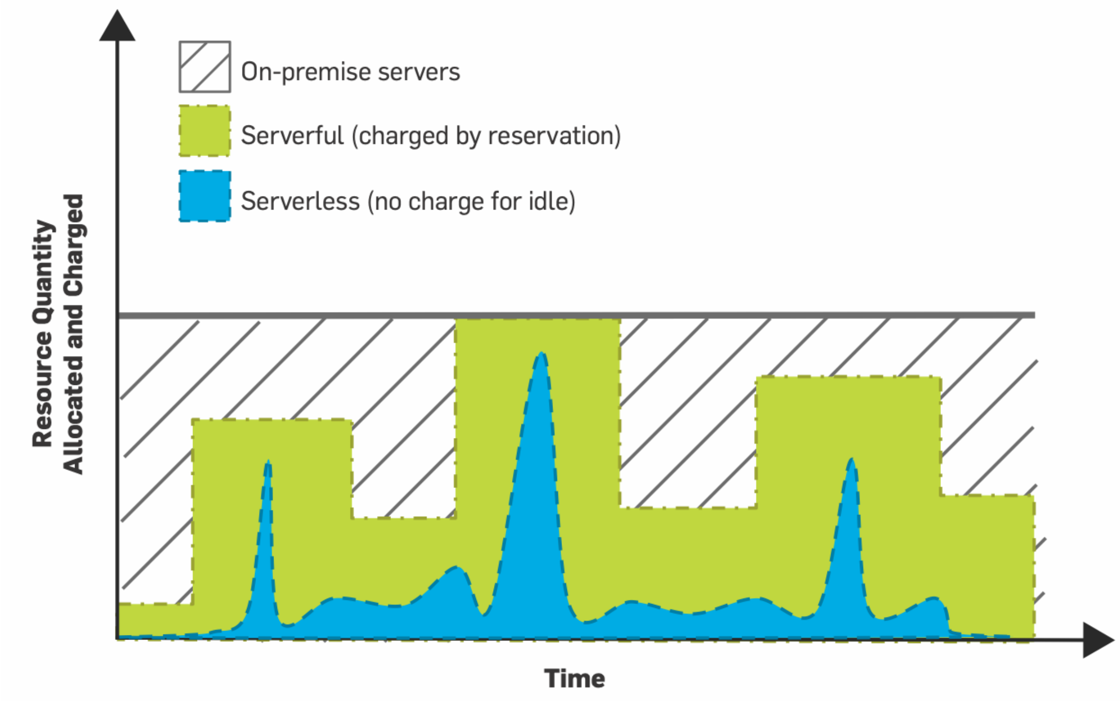
high

Overhead

e.g., resource footprint, startup latency

Evolution of virtualization

- Serverless users pay only for resources consumed, not for idle reserved capacity



Source: J. Schleier-Smith et al. “The Next Phase of Cloud Computing,” in Commun. ACM, 64(5), 2021.

Serverless is really hot

Serverless press

"...more than 20 percent of global enterprises will have deployed serverless computing technologies by 2020."
Gartner, Dec 2018

7 Reasons
Why Serverless is the Future
claudiobernasconi.ch

A CLOUD GURU
Serverless — the future of software architecture?
The future is transitioning from 3-tiered architectures to thick-client apps connected to cloud-based microservice functions
Sam Kroonenburg Follow
Oct 21, 2015 · 7 min read

psc TRANSITION TECHNOLOGIES
13 September 2019 ★★★★★ 5 (4)
Why serverless is the future of software and apps

Survey Shows More than 75% Use or Plan to Use Serverless in Next 18 Months
Source: The New Stack Serverless Survey 2018. Q. Is your organization using a serverless architecture? n=608.
© 2018 THENEWSTACK

Case Study

Case study

Serving ML models using AWS Lambda

- AWS Lambda is one of the most popular commercial serverless platforms
- ML serving is performed in real time with dynamic requests. Therefore, cloud functions are well-suited for hosting ML models due to the high scalability and fine-grained billing.

AWS Lambda

AWS Lambda

- In Lambda, function code runs on lightweight virtual machines (microVMs), called Firecracker, which achieves isolation at low overhead[1]
  - Consistent performance
  - Fast function initialization (e.g., 100ms)

[1] Alexandru Agache et al. “Firecracker: Lightweight Virtualization for Serverless Applications”, NSDI’20

AWS Lambda

- Using AWS Lambda
  - Write or upload a cloud function. Currently, AWS Lambda supports functions in Java, Go, PowerShell, Node.js, C#, Python, and Ruby. e.g., a Python function.

```python
def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

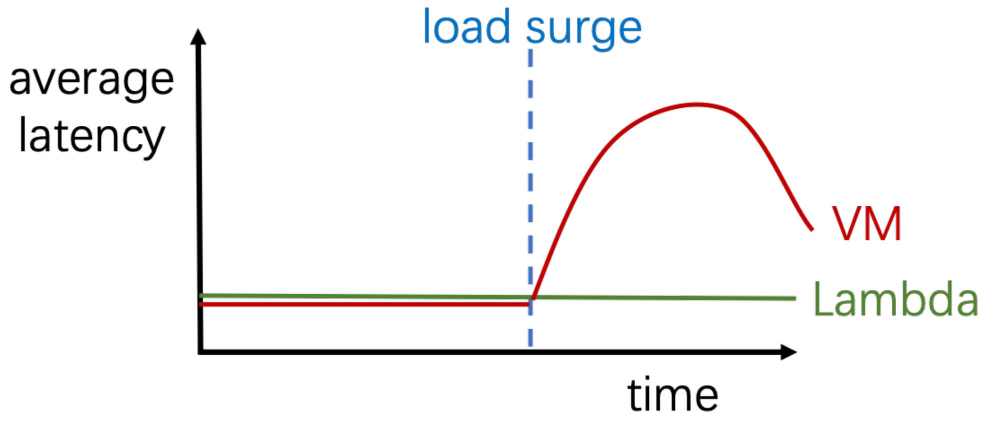
  - Configure resource and timeout of a function instance (optional)
  - Add a trigger that defines an event to invoke the function (e.g., HTTP request)

AWS Lambda

- AWS Lambda charges users based on the number of requests and the duration of function execution
  - $0.2 per 1M requests and 1M free requests per month
  - $0.0000000166667 for every GB-millisecond

Lambda vs. VM

- Compared with VM-based model serving, AWS Lambda is more scalable to dynamic inference requests.
  - It takes about 1 minute to provision a VM
  - Lambda has much shorter startup latency—thousands of function instances can be started in less than one second, leading to low latency without the cost of over-provisioning.



Serving ML models

FaaSwap: SLO-Aware, GPU-Efficient Serverless Inference via Model Swapping

Minchen Yu[†‡] Ao Wang[§] Dong Chen[†] Haoxuan Yu[†] Xiaonan Luo[†] Zhuohao Li[†]
Wei Wang[†] Ruichuan Chen[*] Dapeng Nie[§] Haoran Yang[§]
[†]Hong Kong University of Science and Technology [‡]Chinese University of Hong Kong, Shenzhen
[§]Alibaba Group [*]Nokia Bell Labs

Gillis: Serving Large Neural Networks in Serverless Functions with Automatic Model Partitioning

Minchen Yu[*], Zhifeng Jiang[*], Hok Chun Ng[*], Wei Wang[*], Ruichuan Chen[†], Bo Li[*]
[*]Hong Kong University of Science and Technology
{myuaj, zjiangaj, hcngac, weiwa, bli}@cse.ust.hk
[†]Nokia Bell Labs
ruichuan.chen@nokia-bell-labs.com

MArk: Exploiting Cloud Services for Cost-Effective, SLO-Aware Machine Learning Inference Serving

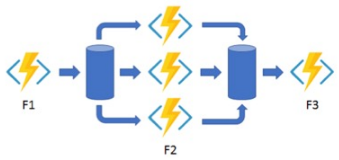
Chengliang Zhang Minchen Yu Wei Wang
HKUST
{czhangbn, myuaj, weiwa}@cse.ust.hk

Feng Yan
University of Nevada, Reno
fyan@unr.edu

Function orchestration

- In practice, a serverless application can rely on multiple functions, e.g., microservices. These functions need to interact with each other at runtime.
- Function orchestration systems are designed for coordinating and synchronizing multiple functions in an application.

AWS Step Functions

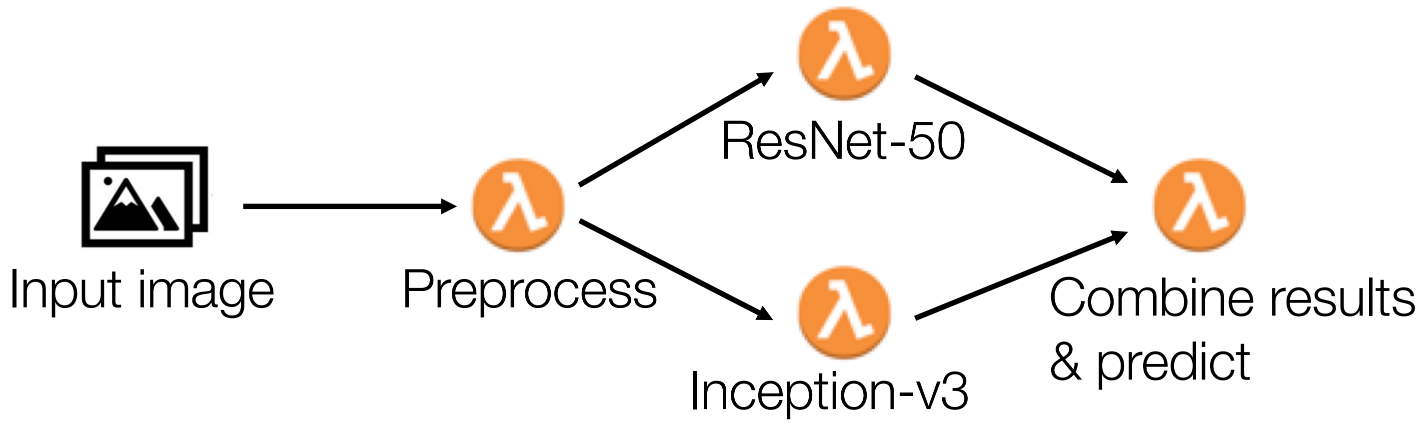


Azure Durable Functions

IBM Function Composer

Function orchestration

- As a common practice, a serverless application can be represented as a workflow of cloud functions.
  - e.g., a ML serving pipeline

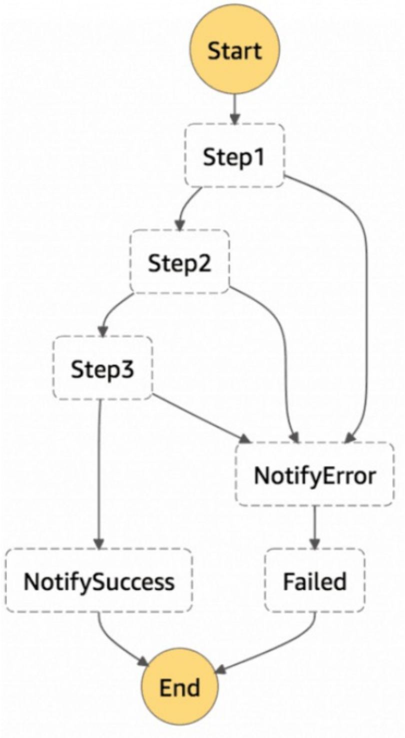


How it works

- Function orchestration systems manage runtime states and control the execution of workflows. There can be different representations of workflows.
    - State machine, e.g., AWS Step Functions
    - High-level programming language, e.g., Azure Durable Functions

Examples

- AWS Step Functions



- Azure Durable Functions

```python
import azure.functions as func
import azure.durable_functions as df


def orchestrator_function(context: df.DurableOrchestrationContext):
    x = yield context.call_activity("F1", None)
    y = yield context.call_activity("F2", x)
    z = yield context.call_activity("F3", y)
    result = yield context.call_activity("F4", z)
    return result
```

For more details:
https://docs.aws.amazon.com/step-functions/index.html
https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview

Recent research on function orchestration

- Improve the applicability and usability of function orchestration
    - Easily and effectively applied to various applications
- Improve the performance of function interactions
    - Fast data sharing

Following the Data, Not the Function:
Rethinking Function Orchestration in Serverless Computing

Minchen Yu[†] Tingjia Cao[‡*] Wei Wang[†] Ruichuan Chen[§]
[†]Hong Kong University of Science and Technology
[‡]University of Wisconsin-Madison [§]Nokia Bell Labs

Limitations

Limitations

- Despite the success of current serverless cloud, there still exists problems. For example,
  - millisecond-scale function startup overhead
  - lack of efficient state management
  - lack of supports for heterogenous hardware

  ……

Future of serverless cloud

- These problems make serverless cloud ill-suited for many applications, including latency-sensitive applications and data-intensive workloads.
- On the other hand, this motivates us to build next-generation serverless platforms that are more efficient, general-purpose, and ease-of-use.

BY JOHANN SCHLEIER-SMITH, VIKRAM SREEKANTI, ANURAG KHANDELWAL, JOAO CARREIRA, NEERAJA J. YADWADKAR, RALUCA ADA POPA, JOSEPH E. GONZALEZ, ION STOICA, AND DAVID A. PATTERSON

What Serverless Computing Is and Should Become:

The Next Phase of Cloud Computing