Cloud Programming Simplified: A Berkeley View on Serverless Computing

Outline

- **Introduction**
- Emergence of Serverless Computing
- Limitation of Today's Serverless Computing Platforms
- What Serverless Computing Should Become

Introduction

Cloud computing advantages in 2009

- The infinite computing resources on demand
- The elimination of an up-front commitment
- The ability to pay for use of computing resources
- Economies of scale
- Simplifying operation via resource virtualization
- Higher hardware utilization by multiplexing workloads

Introduction

The two competing approaches in 2009

- Amazon EC2 (O)
  - Which looks like physical hardware, and can be controlled nearly the entire software stack
- Google App Engine (X)
  - Which enforces an application structure of clean separation between a stateless computation tier and a stateful storage tier

Introduction

The downside of low-level virtual machine

- The developers had to manage virtual machines themselves
  - By becoming system administrators
  - Or by working with them to set up environments
- Billed based on allocation

Introduction

The serverless computing

- The cloud functions represent the core of serverless computing
  - FaaS (Function as a Service)
- The cloud platforms provide specialized serverless framework
  - BaaS (Backend as a Service)
- Serverless computing = FaaS + BaaS

Introduction

The serverless computing must

- Scale automatically without **explicit provision**

- Be billed based on **usage**

Outline

- Introduction
- **Emergence of Serverless Computing**
- Limitation of Today's Serverless Computing Platforms
- What Serverless Computing Should Become

Emergence of Serverless Computing

	Characteristic	*AWS Serverless Cloud*	*AWS Serverful Cloud*
PROGRAMMER	When the program is run	On event selected by Cloud user	Continuously until explicitly stopped
	Programming Language	JavaScript, Python, Java, Go, C#, etc.[4]	Any
	Program State	Kept in storage (stateless)	Anywhere (stateful or stateless)
	Maximum Memory Size	0.125 - 3 GiB (Cloud user selects)	0.5 - 1952 GiB (Cloud user selects)
	Maximum Local Storage	0.5 GiB	0 - 3600 GiB (Cloud user selects)
	Maximum Run Time	900 seconds	None
	Minimum Accounting Unit	0.1 seconds	60 seconds
	Price per Accounting Unit	$0.0000002 (assuming 0.125 GiB)	$0.0000867 - $0.4080000
	Operating System & Libraries	Cloud provider selects[5]	Cloud user selects
SYSADMIN	Server Instance	Cloud provider selects	Cloud user selects
	Scaling[6]	Cloud provider responsible	Cloud user responsible
	Deployment	Cloud provider responsible	Cloud user responsible
	Fault Tolerance	Cloud provider responsible	Cloud user responsible
	Monitoring	Cloud provider responsible	Cloud user responsible
	Logging	Cloud provider responsible	Cloud user responsible

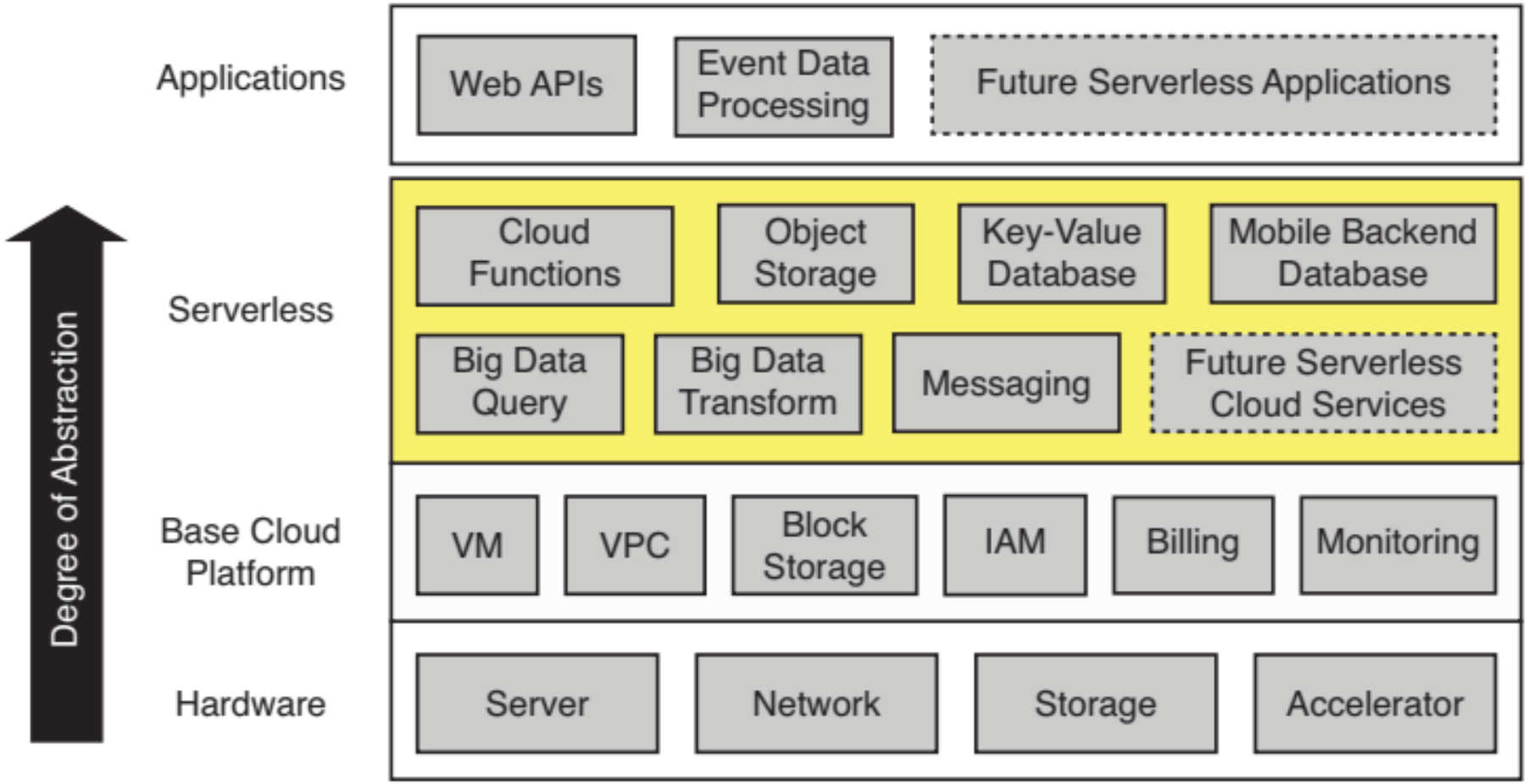
Characteristics of serverless cloud function *vs*. serverful cloud VMs

Emergence of Serverless Computing

The critical distinctions

- Decoupled computation and storage
- Execution code without managing resource allocation
- Paying in proportion to resources used instead of for resource allocated

Emergence of Serverless Computing



The Architecture of Serverless Computing

Emergence of Serverless Computing

The innovation over PaaS and others

- Better autoscaling
- Strong isolation
- Platform flexibility
- Service ecosystem support

Outline

- Introduction
- Emergence of Serverless Computing
- **Limitation of Today's Serverless Computing Platforms**
- What Serverless Computing Should Become

Limitation of Today's Serverless Computing Platforms

Inadequate storage for fine-grained operations

- Object storage service are highly scalable and inexpensive, but exhibit high access cost and high access latencies

- Key-value databases provide high IOPS, but are expensive and can take a long time to scale up

Limitation of Today's Serverless Computing Platforms

Lack of fine-grained coordination

- None of the existing cloud storage services come with notification capabilities

- Applications are left with no choice but to either
  - Manage a VM-based system that provides notifications
  - Or implement their own notification mechanism

Limitation of Today's Serverless Computing Platforms

The code start latency

- The time it takes to start a cloud function
- The time it takes to initialize the software environment of the function
- Application-specific initialization in user code

Outline

- Introduction
- Emergence of Serverless Computing
- Limitation of Today's Serverless Computing Platforms
- **What Serverless Computing Should Become**

What Serverless Computing Should Become

Abstraction Challenges

- Resource requirements
  - Enable developers to specify requirements
  - The cloud provider infer requirements from static code analysis

- Data dependencies
  - The cloud provider to expose an API that allows an application to specify its computation graph

What Serverless Computing Should Become

System Challenges

- High-performance, affordable transparently provisioned storage
  - Ephemeral storage
  - Durable storage
- Coordination/signal service
- Minimize startup time
  - Scheduling and starting resources -> by developing new lightweight isolation mechanisms
  - Downloading the application software environment -> by leveraging unikernels (libOS)

What Serverless Computing Should Become

Security Challenges

- Scheduling randomization and physical isolation
  - Lower the risk of co-locating the attacker and the victim
- Fine-grained security contexts
- Oblivious serverless computing
  - The access patterns and timing information can be protected

Thanks