TCSS 562:
SOFTWARE ENGINEERING
FOR CLOUD COMPUTING

Cloud Computing Concepts and Models

Wes J. Lloyd
Institute of Technology
University of Washington - Tacoma

OBJECTIVES

- Cloud Computing Concepts and Models
  - Roles and boundaries
  - Cloud characteristics
  - Cloud delivery models
  - Cloud deployment models

- Term project proposal

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.2

SELECT FEEDBACK FROM 3/26

- Is vertical scaling, increasing the size of the CPU like VMs?

- And is horizontal scaling increasing the number of VMs?

- Why is horizontal scaling in the cloud cost effective?

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.3

FEEDBACK FROM 3/28

- What is a "sunk cost"?
  - From economics…

- What is elasticity (a cloud characteristic)?

- What is hyper-threading?

- What is resilience?

- From the IaaS performance comparison, what is the "m-bound" vs. "d-bound" application?

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.4

FEEDBACK - 2

- What are the implications of over-provisioning?
  - From the cloud provider's perspective
  - From the cloud consumer's perspective

- What is the border line between PaaS and IaaS?

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.5

FEEDBACK - 3

- **What is a Native Cloud Application?**
- Reference: https://thenewstack.io/cloud-native-architecture-one-stack-many-options/
- The architecture is microservice-based
  - Loosely coupled systems are easier to build, deploy and update
- It is automated
  - Continuous integration/continuous delivery (CICD), automated configuration management — everything is automated.
- DevOps drives it
  - The people who build an application also run that application. No more throwing applications over the wall.

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.6

CLOUD NATIVE APPLICATION - 2

- **Cloud native implies container native**

- **Fast:** Containers launch much faster than VMs because containers on the same host share an OS (and VM).
  - Containers don't wait for an OS to boot...

- **Lightweight:** Because containers are lightweight, more can be hosted on a single host (VM)
  - Linux does a good job of providing resource isolation.

- **Consistent:** Because containers package together dependencies, it is easier to run containerized applications consistently across different host environments.

April 2, 2018 · TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma · L3.7

FEEDBACK - 4

- With all of the tight coupling between the items (e.g. AWS Lambda, Dynamo DB, etc.) are cloud application build with these technologies still considered as "Native Cloud"?

April 2, 2018 · TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma · L3.8

OBJECTIVES

- Cloud Computing Concepts and Models
  - **Roles and boundaries**
  - Cloud characteristics
  - Cloud delivery models
  - Cloud deployment models

- Term project proposal

April 2, 2018 · TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma · L3.9

ROLES REVIEW

- Cloud provider
- Cloud consumer
- Cloud service owner
- Cloud resource administrator
- Cloud auditor
- Cloud brokers
- Cloud carriers

April 2, 2018 · TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma · L3.10

OBJECTIVES

- Cloud Computing Concepts and Models
  - Roles and boundaries
  - **Cloud characteristics**
  - Cloud delivery models
  - Cloud deployment models

- Term project proposal

April 2, 2018 · TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma · L3.11

CLOUD CHARACTERISTICS

- On-demand usage
- Ubiquitous access
- Multitenancy (resource pooling)
- Elasticity
- Measured usage
- Resiliency

- Assessing these features helps measure the value offered by a given cloud service or platform

April 2, 2018 · TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma · L3.12

OBJECTIVES

- Cloud Computing Concepts and Models
  - Roles and boundaries
  - Cloud characteristics
  - Cloud delivery models
  - Cloud deployment models

- Term project proposal

EXAMPLE CLOUD SERVICES

SERVERLESS COMPUTING

- New cloud platform for hosting application code

- Every cloud vendor provides their own:
  - AWS Lambda, Azure Functions, Google Cloud Functions, IBM OpenWhisk

- Similar to platform-as-a-service

- Replace opensource web container (e.g. Apache Tomcat) with abstracted vendor-provided **black-box** environment
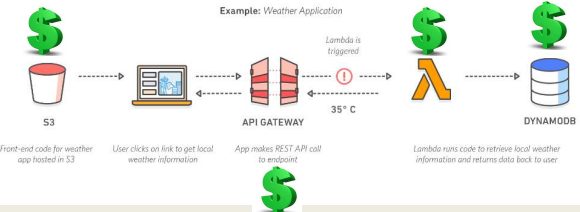
SERVERLESS COMPUTING - 2

- Many challenging features of distributed systems are provided automatically

- *Built into the platform:*

- Highly availability (24/7)

- Scalability

- Fault tolerance

SERVERLESS SOFTWARE ARCHITECTURE

- Every service with a different pricing model

SERVERLESS COMPUTING BILLING MODELS

- **AWS Lambda Pricing**

- **FREE TIER:**
  - first 1,000,000 function calls/month → FREE
  - first 400 GB-sec/month → FREE

- Afterwards: *obfuscated pricing (AWS Lambda):*
  - $0.0000002 per request
  - $0.000000208 to rent 128MB / 100-ms

PRICING EXAMPLE

- Consider a continuously running webservice
- Each service call requires 1-second of 100% CPU utilization, and 4GB of memory
- Renting an m4.large 2-vCPU Virtual Machine this workload can be hosted for $72/month
- m4.large costs 10¢ an hour, 24 hours/day,
  30 days/month
- How much would costing this workload on AWS Lambda cost?
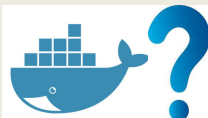
April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.19

PRICING EXAMPLE

- Consider a continuously running webservice
- Each service call requires 1-second of 100% CPU utilization, and
- Ren... ...d can be
- m4...

Worst-case scenario = 4.8x !
AWS EC2: $72.00
AWS Lambda: $346.51

- How... ...ost?

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.20
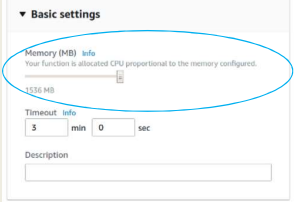

AWS LAMBDA IMPLEMENTATION

- Microservice code deployed to individual container-like environment
- Service requests routed to "containers" by the Lambda platform
- AWS API Gateway provides a REST URL, and routing
- Containers run on virtual machines, presumably EC2 instances (c4 series?)

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.21

MICROSERVICE MEMORY RESERVATION QUESTION...

- Memory must be reserved for functions
- UI provides a "slider bar" to set the function's memory allocation

  ▼ Basic settings
  Memory (MB) Info
  Your function is allocated CPU proportional to the memory configured.
  1536 MB
  Timeout Info
  3 min 0 sec
  Description

- CPU power also coupled to the slider bar: every **doubling** of memory, **doubles** CPU...
- **But how much memory does code require?**

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.22

FAAS CODE DISAGGREGATION

- How should legacy application code be decomposed into microservices?
- Lambda function limits:
- All source files and libraries must fit into:
- 64MB compressed, 256MB uncompressed
- What are the implications for cost based on how we disaggregate code into individual functions?
- How does this impact # of invocations and memory utilization?

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.23

LAMBDA@EDGE

Introducing Lambda@Edge

- Lambda@Edge is an extension of AWS Lambda that allows you to run Node.js code at AWS global edge locations.
- Bring your own code to the edge and customize your content very close to your users, improving the end-user experience.

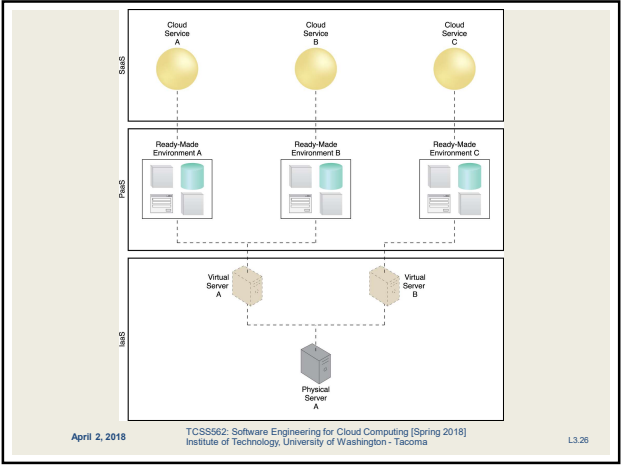No servers to manage   Continuous scaling   Globally distributed   Never pay for idle – no cold servers

April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018] Institute of Technology, University of Washington - Tacoma | L3.24

SOFTWARE-AS-A-SERVICE

- Software applications as shared cloud service
- Nearly all server infrastructure management is abstracted away from the user
- Software is generally configurable
- SaaS can be a complete GUI/UI based environment
- Or UI-free (database-as-a-service)

- SaaS offerings
  - Google Docs
  - Office 365
  - Cloud9 Integrated Development Environment
  - Salesforce

| April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L3.25 |

| April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L3.26 |

OTHER CLOUD SERVICE MODELS

- IaaS
  - Storage-as-a-Service
- PaaS
  - Integration-as-a-Service
- SaaS
  - Database-as-a-Service
  - Testing-as-a-Service
  - Model-as-a-Service
- ?
  - Security-as-a-Service
  - Integration-as-a-Service

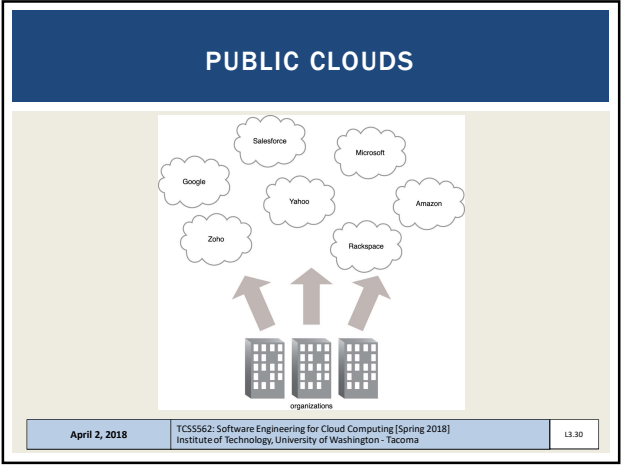| April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L10.27 |

OBJECTIVES

- Cloud Computing Concepts and Models
  - Roles and boundaries
  - Cloud characteristics
  - Cloud delivery models
  - Cloud deployment models

- Term project proposal

| April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L3.28 |

CLOUD DEPLOYMENT MODELS

- Distinguished by ownership, size, access

- Four common models
  - Public cloud
  - Community cloud
  - Hybrid cloud
  - Private cloud

| April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L3.29 |

PUBLIC CLOUDS

| April 2, 2018 | TCSS562: Software Engineering for Cloud Computing [Spring 2018]
Institute of Technology, University of Washington - Tacoma | L3.30 |

COMMUNITY CLOUD

- Specialized cloud built and shared by a particular community

- Leverage economies of scale within a community

- Research oriented clouds

- Examples:
  - Bionimbus - bioinformatics
  - Chameleon
  - CloudLab

PRIVATE CLOUD

- Compute clusters configured as IaaS cloud

- Open source software

- Eucalyptus
- Openstack
- Apache Cloudstack
- Nimbus

- Virtualization: XEN, KVM, ...

HYBRID CLOUD

- Extend private cloud typically with public or community cloud resources

- Cloud bursting: Scale beyond one cloud when resource requirements exceed local limitations

- Some resources can remain local for security reasons

OTHER CLOUDS

- Federated cloud
  - Simply means to aggregate two or more clouds together
  - Hybrid is typically private-public
  - Federated can be public-public, private-private, etc.
  - Also called inter-cloud

- Virtual private cloud
  - Google and Microsoft simply call these virtual networks
  - Ability to interconnect multiple independent subnets of cloud resources together
  - Resources allocated private IPs from individual network subnets can communicate with each other (10.0.1.0/24) and (10.0.2.0/24)
  - Subnets can span multiple availability zones within an AWS region

SIMPLE VPC

- Recommended when using Amazon EC2

VPC SPANNING AVAILABILITY ZONES

OBJECTIVES

- Cloud Computing Concepts and Models
  - Roles and boundaries
  - Cloud characteristics
  - Cloud delivery models
  - **Cloud deployment models**

- Term project proposal

TERM PROJECT PROPOSAL

- Overview

QUESTIONS