TCSS 462/562: (SOFTWARE ENGINEERING FOR) CLOUD COMPUTING

Introduction to Cloud Computing - II

Wes J. Lloyd
School of Engineering and Technology
University of Washington – Tacoma

1

OBJECTIVES – 10/16

- Questions from 10/14
- Introduction to Cloud Computing II –From book #1 - Chapter 3: Understanding Cloud Computing Cloud Computing Concepts, Technology & Architecture
  - Benefits of cloud adoption
  - Risks of cloud adoption
- Background on AWS Lambda for the Term Project
- From Book #1: Chapter 4: Cloud Computing Concepts and Models
- At the end: Open Discussion on the Term Project
  - Discussion
  - Team Planning

2

OFFICE HOURS – FALL 2025

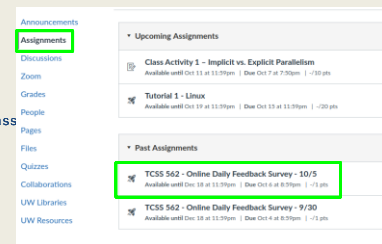
- **Thursdays:**
  - 6:00 to 7:00 pm - CP 229 & Zoom
- **Fridays**
  - 11:00 am to 12:00 pm – ONLINE via Zoom*
- Or email for appointment

> *Office Hours set based on Student Demographics survey feedback*
> ** - Friday office hours may be adjusted or canceled due meeting conflicts or other obligations. Adjustments will be announced via Canvas.*

3

ONLINE DAILY FEEDBACK SURVEY

- Daily Feedback Quiz in Canvas – Take After Each Class
- **1-point Extra Credit** for completing online
- **2-points Extra Credit** for completing in-person in class
- 36 points possible
- 2.5% added to final course grade for (36/36)



4

WARNING

- DO NOT SUBMIT BOTH A PAPER AND AN ONLINE SURVEY OR YOU WILL LOOSE POINTS
- CANVAS WILL AUTOMATICALLY REPLACE THE PAPER SURVEY SCORE (2 PTS) WITH THE ONLINE SURVEY (1 PT)
- *** COMPLETE ONLY ONE SURVEY FOR EACH CLASS SESSION ***
- WE WILL NOT BE ABLE TO DUPLICATE CHECK SURVEYS FOR EACH CLASS SESSION AND MAKE CORRECTIONS

5

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (44 respondents, 31 in-person, 13 online):
- 1-mostly review, 5-equal new/review, 10-mostly new
- **Average – 6.91 (↓ - *previous 7.33*)**
- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- **Average – 5.14 (↑ - previous 4.96)**

6

FEEDBACK FROM 10/14

- ***How long will it take to grade tutorial 2? if free trial ends and API expires before its graded, need to renew?***
- Tutorial 2 is due on Oct 21
- If the weather.bit free trail ends, the 16-day weather forecast reverts to 7-day
- Full credit for script regardless of license: 16 or 7-day forecast
- ***Is project more focused on writeup and presentation rather than implementation?***
  - The case study involves conducting performance experiments on at least 2 - alternate implementations.

7

FEEDBACK - 2

- ***Overall the end goal (of cloud computing?) is to make sure the user experience is the easiest to use for them ?***
- One goal of cloud applications can be ease-of-use (usability)
- Goals of cloud applications will vary based on customer and business needs
- Example: performance may be a higher priority than usability.
- Other potential cloud application key priorities (i.e. non-functional quality attributes):
  - Accessibility
  - 24/7 availability
  - Fault tolerance / resilience
  - Extensibility

8

FEEDBACK - 3

- **(Software Engineering Question)**
  What is the difference between a functional requirement and a non-function quality attribute of a computing application or system ?

9

FEEDBACK - 4

- ***What are the real-live examples that apply grid computing?***
- Worldwide LHC Computing Grid (WLCG)
- **Field:** High-energy physics
- **Purpose:** Processes data from CERN's Large Hadron Collider (LHC) experiments.
- **Scale:** Over 170 computing centers in 40+ countries.
- **Notable achievement:** Helped analyze the massive datasets that led to the discovery of the Higgs boson in 2012.
- The LHC is a giant particle accelerator that smashes subatomic particles, like protons, together at near-light speeds located near Geneva, Switzerland

10

SEARCH FOR EXTRATERRESTRIAL INTELLIGENCE (SETI@HOME)

- One of the most popular and influential applications of grid computing
- Combined public participation, distributed data analysis, and massive global computing power to analyze scientific data
- **Purpose:** Analyze radio telescope data; search for extraterrestrial signals.
- **How:** Volunteers downloaded screensaver to process small chunks of data when computers were idle.
- **Scale:** Millions of participants & hundreds of thousands of computers - one of the largest distributed computing efforts ever.
- **Impact**: Demonstrated power of using idle computing resources from ordinary users to perform large-scale scientific computation
- **Later at home computing projects**: Folding@home, Einstein@home, and BOINC (the Berkeley Open Infrastructure for Network Computing).

11

FEEDBACK - 5

- ***Why is coarse-grained parallelism preferred in cloud computing?***
- Embarrassingly parallel data parallel problems require minimum communication and coordination and are easy to distribute across nodes lead to massive speed-ups
- These workloads also help make use of the massive number of available CPUs on modern systems
- Easier to develop, deploy, and run from a programmer's perspective than fine-grained parallel programs which require more synchronization and potentially careful debugging

12

TUTORIAL 0

- Getting Started with AWS
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_0.pdf
- Create an AWS account
- Create account credentials for working with the CLI
- Install awsconfig package
- Setup awsconfig for working with the AWS CLI

October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.13

13

TUTORIAL 1 – SUBMIT BY TODAY

- Introduction to Linux & the Command Line
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_1.pdf
- Tutorial Sections:
  1. The Command Line
  2. Basic Navigation
  3. More About Files
  4. Manual Pages
  5. File Manipulation
  6. VI – Text Editor
  7. Wildcards
  8. Permissions
  9. Filters
  10. Grep and regular expressions
  11. Piping and Redirection
  12. Process Management

October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.14

14

TUTORIAL 2 – OCT 21

- Introduction to Bash Scripting
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_2.pdf
- Review tutorial sections:
- Create a BASH webservice client
  1. What is a BASH script?
  2. Variables
  3. Input
  4. Arithmetic
  5. If Statements
  6. Loops
  7. Functions
  8. User Interface
- Call service to obtain IP address & lat/long of computer
- Call weatherbit.io API to obtain weather forecast for lat/long

October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.15

15

TUTORIAL 3 – DUE OCT 30 (TEAMS OF 2)

- Best Practices for Working with Virtual Machines on Amazon EC2
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_3.pdf
- Creating a spot VM
- Creating an image from a running VM
- Persistent spot request
- Stopping (pausing) VMs
- EBS volume types
- Ephemeral disks (local disks)
- Mounting and formatting a disk
- Disk performance testing with Bonnie++
- Cost Saving Best Practices

October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.16

16

INTRODUCTION TO CLOUD COMPUTING


October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.17

17

CATCH UP – 10/10

- Questions from 10/10
- Properties of Distributed Systems, Modularity
- Introduction to Cloud Computing –based on book #1: Cloud Computing Concepts, Technology & Architecture
  - Why study cloud computing?
  - History of cloud computing
  - Business drivers
  - Cloud enabling technologies
  - Terminology
  - Benefits of cloud adoption
  - Risks of cloud adoption
- Background on AWS Lambda for the Term Project

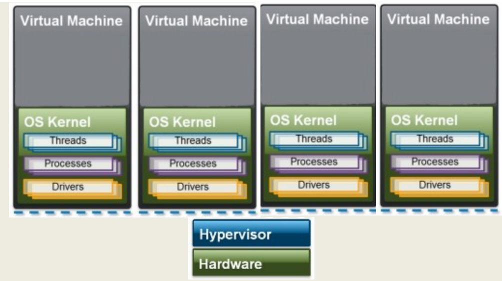
October 10, 2024 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.18

18

VIRTUALIZATION

19

VIRTUALIZATION



20

VIRTUALIZATION

- Simulate physical hardware resources via software
  - The virtual machine (virtual computer)
  - Virtual local area network (VLAN)
  - Virtual hard disk
  - Virtual network attached storage array (NAS)
- Early incarnations featured significant performance, reliability, and scalability challenges
- CPU and other HW enhancements have minimized performance GAPs

21

OBJECTIVES – 10/10

- **Questions from 10/10**
- Properties of Distributed Systems, Modularity
- Introduction to Cloud Computing –based on book #1: Cloud Computing Concepts, Technology & Architecture
  - Why study cloud computing?
  - History of cloud computing
  - Business drivers
  - Cloud enabling technologies
  - Terminology
  - Benefits of cloud adoption
  - Risks of cloud adoption
- Background on AWS Lambda for the Term Project

22

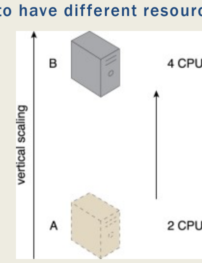
KEY TERMINOLOGY

- **On-Premise Infrastructure**
  - Local server infrastructure not configured as a cloud
- **Cloud Provider**
  - Corporation or private organization responsible for maintaining cloud
- **Cloud Consumer**
  - User of cloud services
- **Scaling**
  - **Vertical scaling**
    - Scale up: increase resources of a single virtual server
    - Scale down: decrease resources of a single virtual server
  - **Horizontal scaling**
    - Scale out: increase number of virtual servers
    - Scale in: decrease number of virtual servers

23

VERTICAL SCALING

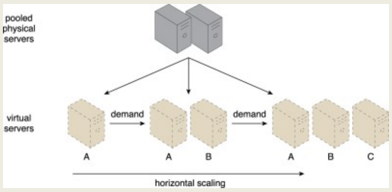
- Reconfigure virtual machine to have different resources:
  - CPU cores
  - RAM
  - HDD/SDD capacity
- May require VM migration if physical host machine resources are exceeded



24

HORIZONTAL SCALING

- Increase (scale-out) or decrease (scale-in) number of virtual servers based on demand



25

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers

26

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available

27

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available
Resource replication and automated scaling	Additional setup is normally needed

28

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available
Resource replication and automated scaling	Additional setup is normally needed
Additional servers required	No additional servers required

29

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available
Resource replication and automated scaling	Additional setup is normally needed
Additional servers required	No additional servers required
Not limited by individual server capacity	Limited by individual server capacity

30

KEY TERMINOLOGY - 2

- Cloud services
  - Broad array of resources accessible "as-a-service"
  - Categorized as Infrastructure (IaaS), Platform (PaaS), Software (SaaS)
- Service-level-agreements (SLAs):
  - Establish expectations for: uptime, security, availability, reliability, and performance

31

OBJECTIVES – 10/16

- Questions from 10/14
- Introduction to Cloud Computing II –From book #1 - Chapter 3: Understanding Cloud Computing
  Cloud Computing Concepts, Technology & Architecture
  - **Benefits of cloud adoption**
  - Risks of cloud adoption
- Background on AWS Lambda for the Term Project
- From Book #1:
  Chapter 4: Cloud Computing Concepts and Models
- At the end: Open Discussion on the Term Project

32

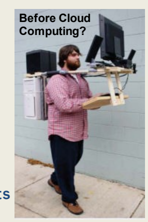
GOALS AND BENEFITS

- **Cloud providers**
  - Leverage economies of scale through mass-acquisition and management of large-scale IT resources
  - Locate datacenters to optimize costs where electricity is low
- **Cloud consumers**
  - Key business/accounting difference:
  - **Cloud computing enables anticipated capital expenditures to be replaced with operational expenditures**
  - Operational expenditures always scale with the business
  - Eliminates need to invest in server infrastructure based on anticipated business needs
  - Businesses become more agile and lower their financial risks by eliminating large capital investments in physical infrastructure

33

CLOUD BENEFITS - 2

- On demand access to pay-as-you-go resources on a short-term basis (less commitment)
- Ability to acquire "unlimited" computing resources on demand when required for business needs
- Ability to add/remove IT resources at a fine-grained level
- Abstraction of server infrastructure so applications deployments are not dependent on specific locations, hardware, etc.
  - The cloud has made our software deployments more agile...

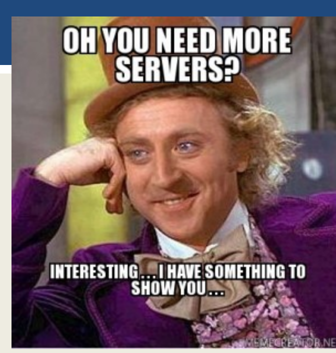


34

CLOUD BENEFITS - 3

- Example: Using 100 servers for 1 hour costs the same as using 1 server for 100 hours
- Rosetta Protein Folding Use Case: Working with a UW-Tacoma graduate student, we deployed this science model across 5,900 compute cores on Amazon for 2-days...
- ***What is the cost to purchase 5,900 compute cores?***
- Recent Dell Server purchase example:
  20 cores on 2 servers for $4,478...
- Using this ratio 5,900 cores costs $1.3 million (purchase only)

35

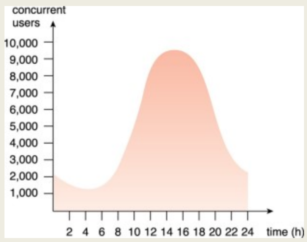


Gene Wilder, Charlie and the Chocolate Factory

36

CLOUD BENEFITS

- Increased scalability
  - Example demand over a 24-hour day →
- Increased availability
- Increased reliability



37

OBJECTIVES – 10/16

- **Questions from 10/14**
- Introduction to Cloud Computing II –From book #1 - Chapter 3: Understanding Cloud Computing Cloud Computing Concepts, Technology & Architecture
  - Benefits of cloud adoption
  - Risks of cloud adoption
- Background on AWS Lambda for the Term Project
- From Book #1: Chapter 4: Cloud Computing Concepts and Models
- At the end: Open Discussion on the Term Project

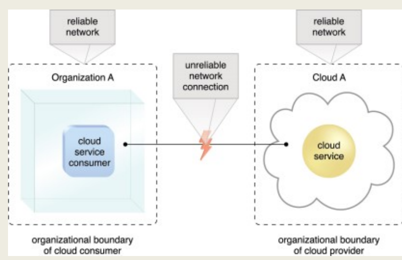
38

CLOUD ADOPTION RISKS

- **Increased security vulnerabilities**
  - Expansion of trust boundaries now include the external cloud
  - Security responsibility shared with cloud provider
- **Reduced operational governance / control**
  - Users have less control of physical hardware
  - Cloud user does not directly control resources to ensure quality-of-service
  - Infrastructure management is abstracted
  - Quality and stability of resources can vary
  - Network latency costs and variability

39

NETWORK LATENCY COSTS



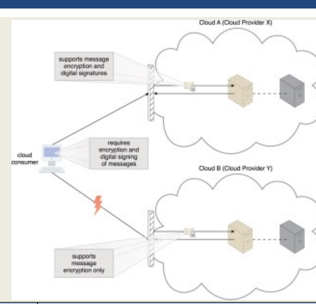
40

CLOUD RISKS - 2

- **Performance monitoring of cloud applications**
  - Cloud metrics (AWS cloudwatch) support monitoring cloud infrastructure (network load, CPU utilization, I/O)
  - Performance of cloud applications depends on the health of aggregated cloud resources working together
  - User must monitor this aggregate performance
- **Limited portability among clouds**
  - Early cloud systems have significant "vendor" lock-in
  - Common APIs and deployment models are slow to evolve
  - Operating system containers help make applications more portable, but containers still must be deployed
- Geographical issues
  - Abstraction of cloud location leads to legal challenges with respect to laws for data privacy and storage

41

CLOUD: VENDOR LOCK-IN



42

OBJECTIVES – 10/16

- Questions from 10/14
- Introduction to Cloud Computing II –From book #1 - Chapter 3: Understanding Cloud Computing Cloud Computing Concepts, Technology & Architecture
  - Benefits of cloud adoption
  - Risks of cloud adoption
- Background on AWS Lambda for the Term Project
- From Book #1: Chapter 4: Cloud Computing Concepts and Models
- At the end: Open Discussion on the Term Project

43

INTRO TO AWS LAMBDA

(IN SUPPORT OF THE TERM PROJECT)

44

SERVERLESS – KEY CONCEPTS

- **Function-as-a-Service (FaaS) platform**
  - A platform where developers deploy "functions" written in common languages (e.g. Java, Python, Go, Node.js) that run as microservices
  - AWS Lambda is a FaaS platform
    - We will discuss platform limitations
- **Function Instances**
  - This is an instantiation of a running function
  - A function instance is created when a client (user) calls the serverless function
  - Each concurrent (parallel) call to AWS Lambda to the same function will create a unique function instance to handle the request
  - The default maximum number of concurrently running function instances in your account is 10.
  - The default was originally 1,000 when the platform was introduced, and was dropped to 100, then 50, and is now just 10 in response to the growing popularity of AWS Lambda (they are running out of servers??)
  - You will want to request an increase in your AWS account's default concurrency. A minimum of 100 is recommended

45

AWS LAMBDA

- Lambda functions can be invoked by creating an HTTP REST endpoint that responds to HTTP POST requests
- A json object is provided as a request object to the function
- In the function code, the request object can be accessed to interpret how the user parameterized the function call
- The function generates a JSON response object
- AWS Lambda is introduced in detail in Tutorial 4

46

TYPES OF FUNCTION CALLS: SYNCHRONOUS

- **Serverless Computing:**
- AWS Lambda supports synchronous and asynchronous function calls
- Clients typically orchestrate synchronous calls and pipelines
- Asynchronous calls are often made via events
- **Synchronous web service:**
- Client calls service
- Client blocks (freezes) and waits for server to complete call
- Connection is maintained in the "OPEN" state
- Problematic if service runtime is long!
  - Connections are notoriously dropped
  - System timeouts reached
- Client can't do anything while waiting unless using threads

47

TYPES OF FUNCTION CALLS: ASYNCHRONOUS

- **Asynchronous web service**
- Client calls service
- Server responds to client with OK message
- Client closes connection
- Server performs the work associated with the service
- Server posts service result in an external data store
  - AWS: S3, SQS (queueing service), SNS (notification service)

48

AWS LAMBDA PLATFORM LIMITATIONS

- Maximum 10 GB memory per function instance
- Maximum 15-minutes execution per function instance
- 500 MB of /tmp disk space for local I/O (default)
- Up to 10 GB /tmp ephemeral storage (for additional charge)
  - https://aws.amazon.com/blogs/aws/aws-lambda-now-supports-up-to-10-gb-ephemeral-storage/
- Access up to 6 vCPUs depending on memory reservation size

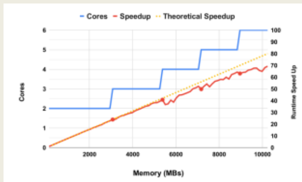


Figure 1: AWS Lambda Performance Speedup for Sysbench Prime Number Generation vs. Function Memory

49

AWS LAMBDA PLATFORM LIMITATIONS - 2

- 10 concurrent function executions inside account (default)
- Function payload: 6MB (synchronous), 256KB (asynchronous)
- Deployment package: 50MB (compressed), 250MB (unzipped)
- Container image size: 10 GB
- Processes/threads: 1024
- File descriptors: 1024
- Function instances run Amazon Linux 2
  - Pending upgrade to Amazon Linux 2023 ?
- See: https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html

50

CPUSTEAL

- *CpuSteal*: Metric that measures when a CPU core is ready to execute but the physical CPU core is busy and unavailable
- Symptom of over provisioning physical servers in the cloud
- Factors which cause *CpuSteal*: (x86 hyperthreading)
  1. Physical CPU is shared by too many busy VMs
  2. Hypervisor kernel is using the CPU
     - On AWS Lambda this would be the Firecracker MicroVM which is derived from the KVM hypervisor
  3. VM's CPU time share <100% for 1 or more cores, and 100% is needed for a CPU intensive workload.
- Man procfs – press "/" – type "proc/stat"
  - CpuSteal is the 8th column returned
  - Metric can be read using SAAF in tutorial #4

51

EFFECTS OF SCALING FUNCTION MEMORY ON CPU TIME SHARE

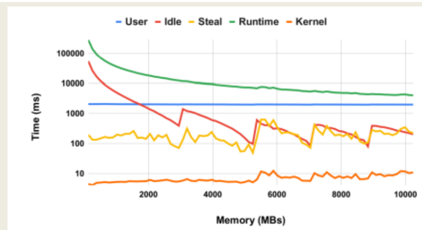
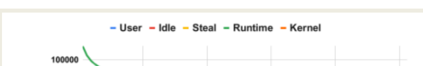


Figure 2: Linux CPU Utilization (log scale) vs. Function Memory for Sysbench Prime Number Generation

52

EFFECTS OF SCALING FUNCTION MEMORY ON CPU TIME SHARE



Key observations:
- Runtime decreases as vCPUs and CPU time share increase
- CPU user time remains constant for the prime number generation task – work doesn't change
- CPU idle time gradually decreases as memory and vCPUs increase (the idle time is becoming active time)
- When the 4th vCPU is added, cpuSteal tracks closely with cpuIdle time (hyperthreading effect)
- There is more cpu Kernel time after the 4th vCPU is added

53

FUNCTION INSTANCE LIFE CYCLES

- Function states:
- COLD: brand new function instance just initialized to run the request (more overhead)
  - Platform cold (first time ever run)
  - Host cold (function assets cached locally on servers)
- WARM: existing function instance that is reused
- All function instances persist for ~5 minutes before they begin to be "garbage collected" by the platform
  - 100% garbage collection may take up to ~30-40 minutes
- AWS Lambda appears to "recycle" infrastructure faster than other FaaS platforms
  - Presumably because of need, because the platform is busy

54

WARM VS COLD FUNCTION INSTANCES

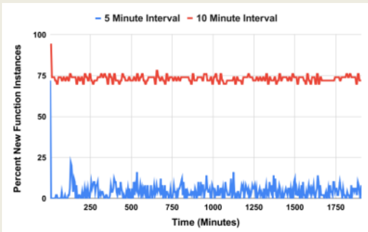


Figure 3: AWS Lambda Function Instance Replacement vs. Function Call Interval over 24-hours

CPU SCALING VS MEMORY: OTHER PLATFORMS: 0 TO 3 GB

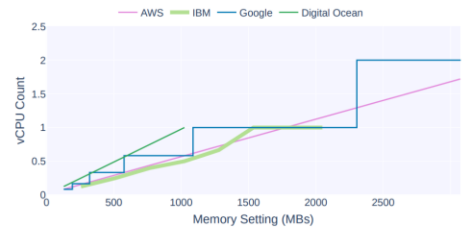
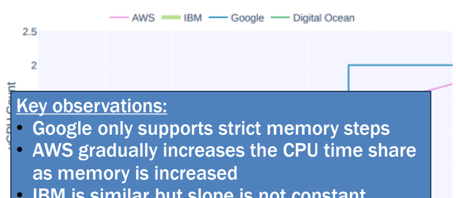


Fig. 2. Allocated vCPUs available at each memory setting on each platform.

CPU SCALING VS MEMORY: OTHER PLATFORMS: 0 TO 3 GB



Key observations:

- Google only supports strict memory steps
- AWS gradually increases the CPU time share as memory is increased
- IBM is similar but slope is not constant
- Digital Ocean only scales up to 1 GB

Fig. 2. Allocated vCPUs available at each memory setting on each platform.

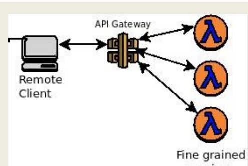
ELASTIC FILE SYSTEM (AWS EFS)

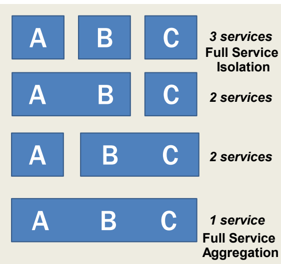
- Traditionally AWS Lambda functions have been limited to 500MB of storage space
- Recently the Elastic File System (EFS) has been extended to support AWS Lambda
- The Elastic File System supports the creation of a shared volume like a shared disk (or folder)
  - EFS is similar to NFS (network file share)
  - Multiple AWS Lambda functions and/or EC2 VMs can mount and share the same EFS volume
  - Provides a shared R/W disk
  - Breaks the 500MB capacity barrier on AWS Lambda
- *Downside: EFS is expensive: ~30 ¢/GB/month*
- **Project**: EFS performance & scalability evaluation on Lambda

SERVERLESS FILE STORAGE COMPARISON PROJECT

- Elastic File System (EFS): Performance, Cost, and Scalability Evaluation in the context of AWS Lambda / Serverless Computing
  - EFS provides a file system that can be shared with multiple Lambda function instances in parallel
- Using a common use case, compare performance and cost of extended storage options on AWS Lambda:
  - Docker container support (up to 10 GB) – read only
  - Emphemeral /tmp (up to 10 GB) – read/write
  - EFS (unlimited, but costly) – read/write
  - image integration with AWS Lambda – performance & scalability

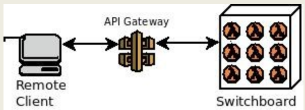
SERVICE COMPOSITION





Other possible compositions: group by library, functional cohesion, etc.

SWITCH-BOARD ARCHITECTURE



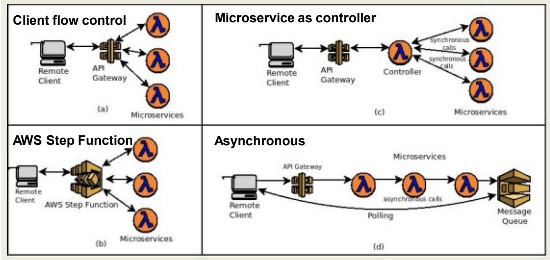
Single deployment package with consolidated codebase (Java: one JAR file)

Entry method contains "switchboard" logic
Case statement that route calls to proper service

Routing is based on data payload
Check if specific parameters exist, route call accordingly

Goal: reduce # of COLD starts to improve performance

APPLICATION FLOW CONTROL - 3



PROGRAMMING LANGUAGE COMPARISON

- FaaS platforms support hosting code in multiple languages
- AWS Lambda- common: Java, Node.js, Python
  - Plus others: Go, PowerShell, C#, and Ruby
- Also Runtime API ("BASH") which allows deployment of binary executables from any programming language

- August 2020 – Our group's paper:
- https://tinyurl.com/y46eq6np
- If wanting to perform a language study either:
  - Implement in C#, Ruby, or multiple versions of Java, Node.js, Python
  - OR implement different app than TLQ (ETL) data processing pipeline

FAAS PLATFORMS

- Many commercial and open source FaaS platforms exist
- TCSS562 projects can choose to compare performance and cost implications of alternate platforms.

- Supported by SAAF:
- AWS Lambda
- Google Cloud Functions
- Azure Functions
- IBM Cloud Functions
- Apache OpenWhisk *(open source, deploy your own FaaS)*
- Open FaaS *(open source, deploy your own FaaS)*

DATA PROVISIONING

- Consider performance and cost implications of the data-tier design for the serverless application
- Use different tools as the relational datastore to support service #2 (LOAD) and service #3 (EXTRACT)

- **SQL / Relational:**
- Amazon Aurora (serverless cloud DB), Amazon RDS (cloud DB), DB on a VM (MySQL), DB inside Lambda function (SQLite, Derby)

- **NO SQL / Key/Value Store:**
- Dynamo DB, MongoDB, S3

PERFORMANCE VARIABILITY

- Cloud platforms exhibit performance variability which varies over time
- Goal of this case study is to measure performance variability (i.e. extent) for AWS Lambda services by hour, day, week to look for common patterns
- Can also examine performance variability by availability zone and region
  - Do some regions provide more stable performance?
  - Can services be switched to different regions during different times to leverage better performance?
- Remember that performance = cost
- If we make it faster, we make it cheaper…

CPU STEAL CASE STUDY

- On AWS Lambda (or other FaaS platforms), when we run functions, how much CpuSteal do we observe?
- How does CpuSteal vary for different workloads? (e.g. functions that have different resource requirements)
- How does CpuSteal vary over time hour, day, week, location?
- How does CpuSteal relate to function performance?

67

CLOUD COMPUTING: CONCEPTS AND MODELS

68

OBJECTIVES – 10/16

- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Roles and boundaries
  - Cloud characteristics
- At the end: Open Discussion on the Term Project
  - Discussion
  - Team Planning

69

ROLES

- **Cloud provider**
  - Organization that provides cloud-based resources
  - Responsible for fulfilling SLAs for cloud services
  - Some cloud providers "resell" IT resources from other cloud providers
    - Example: Heroku sells PaaS services running atop of Amazon EC2
- **Cloud consumers**
  - Cloud users that consume cloud services
- **Cloud service owner**
  - Both cloud providers and cloud consumers can own cloud services
  - A cloud service owner may use a cloud provider to provide a cloud service (e.g. Heroku)

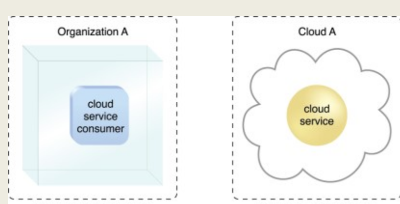
70

ROLES - 2

- **Cloud resource administrator**
  - Administrators provide and maintain cloud services
  - Both cloud providers and cloud consumers have administrators
- **Cloud auditor**
  - Third-party which conducts independent assessments of cloud environments to ensure security, privacy, and performance.
  - Provides unbiased assessments
- **Cloud brokers**
  - An intermediary between cloud consumers and cloud providers
  - Provides service aggregation
- **Cloud carriers**
  - Network and telecommunication providers which provide network connectivity between cloud consumers and providers

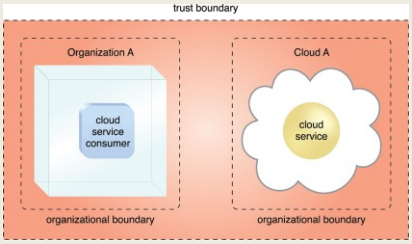
71

ORGANIZATION BOUNDARY



72

TRUST BOUNDARY



73

OBJECTIVES – 10/16

- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Roles and boundaries
  - **Cloud characteristics**
- At the end: Open Discussion on the Term Project
  - Discussion
  - Team Planning

74

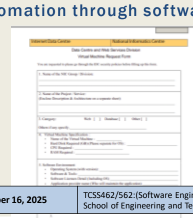
CLOUD CHARACTERISTICS

- 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

75

ON-DEMAND USAGE

- The freedom to self-provision IT resources
- Generally, with automated support
- Automated support requires no human involvement
- Automation through software services interface



76

UBIQUITOUS ACCESS

- Cloud services are widely accessible
- Public cloud: internet accessible
- Private cloud: throughout segments of a company's intranet
- 24/7 availability

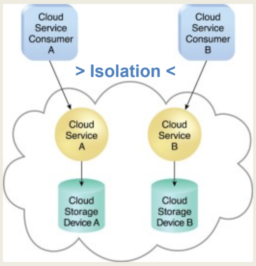
77

MULTITENANCY

- Cloud providers pool resources together to share them with many users
- Serve multiple cloud service consumers
- IT resources can be dynamically assigned, reassigned based on demand
- Multitenancy can lead to performance variation

78

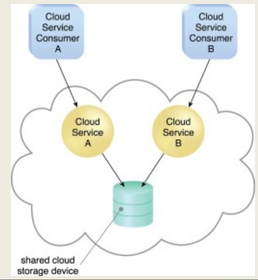
SINGLE TENANT MODEL



79

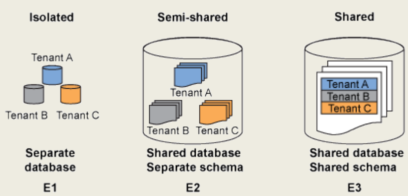
MULTITENANT MODEL

- Resource is "multiplexed" and share amongst multiple users
- Goal is to increase utilization
- Often server resources are underutilized
- There are many "sunk costs" whether usage is 0% or 100%
- Cloud computing tries to maximize "sunk cost" investments through **multi-tenancy**



80

MULTITENANT DATABASE

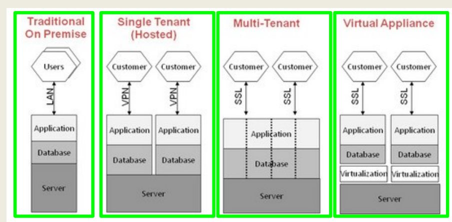


- Many users on a single database instance
- ***What issues may occur when sharing a single database instance?***

81

MULTITENANCY OF RESOURCES

- Where is the multitenancy?
  - >> What is shared? What is isolated?

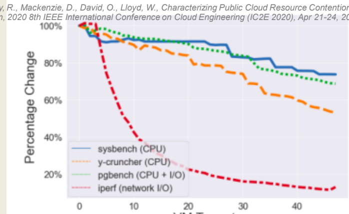


82

RESOURCE CONTENTION FROM MUTLI-TENANCY

- Despite best efforts at isolation, co-resident VMs on a single cloud server running identical benchmarks simultaneously do not perform equally.

From Han, X., Schooley, R., Mackenzie, D., David, O., Lloyd, W., Characterizing Public Cloud Resource Contention to Support Virtual Machine Co-residency Prediction, 2020 8th IEEE International Conference on Cloud Engineering (IC2E 2020), Apr 21-24, 2020.



Up to 48 VMs sharing same server !!

83

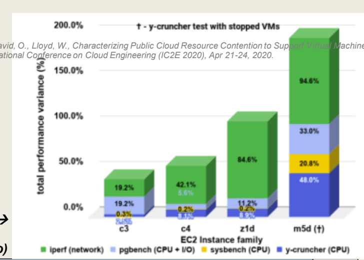
RESOURCE CONTENTION FROM MUTLI-TENANCY - 2

- Performance variation from multi-tenancy is increasing as cloud servers add more CPU cores

From Han, X., Schooley, R., Mackenzie, D., David, O., Lloyd, W., Characterizing Public Cloud Resource Contention to Support Virtual Machine Co-residency Prediction, 2020 8th IEEE International Conference on Cloud Engineering (IC2E 2020), Apr 21-24, 2020.

- Running many idle operating system instances can impose significant overhead for some workloads

Maximum potential resource contention (i.e. worst-case scenario) →



84

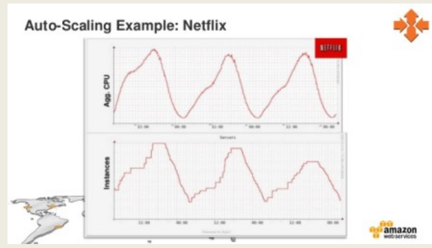
ELASTICITY

- Automated ability of cloud to transparently scale resources
- Scaling based on runtime conditions or pre-determined by cloud consumer or cloud provider
- Threshold based scaling
  - `CPU-utilization > threshold_A, Response_time > 100ms`
  - Application agnostic vs. application specific thresholds
  - Why might an application agnostic threshold be non-ideal?
- Load prediction
  - Historical models
  - Real-time trends

85

PREDICTABLE DEMAND

- AWS EC2 Scaling Example:



From: Kejariwal, A., 2013, March. Techniques for optimizing cloud footprint. In 2013 IEEE Int. Conf. on Cloud Engineering (IC2E), pp. 258-268.

86

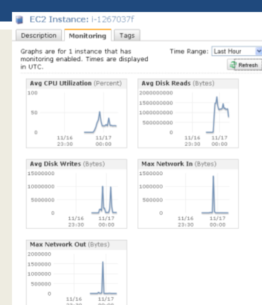
MEASURED USAGE

- Cloud platform tracks usage of IT resources
- For billing purposes
- Enables charging only for IT resources actually used
- Can be time-based (millisec, second, minute, hour, day)
  - Granularity is increasing…
- Can be throughput-based (data transfer: MB/sec, GB/sec)
- Can be resource/reservation based (vCPU/hr, GB/hr)

- Not all measurements are for billing
- Some measurements can support auto-scaling
- For example CPU utilization

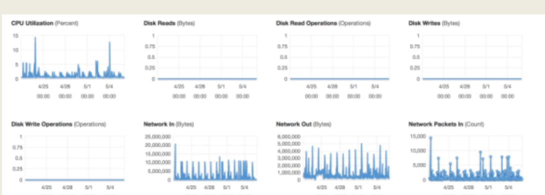
87

EC2 CLOUDWATCH METRICS



88

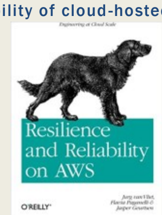
EC2 CLOUDWATCH METRICS



89

RESILIENCY

- Distributed redundancy across physical locations (regions on AWS)
- Used to improve reliability and availability of cloud-hosted applications
- Very much an engineering problem
- No “resiliency-as-a-service” for user deployed apps
- Unique characteristics of user applications make a one-size fits all service solution challenging



90

OBJECTIVES – 10/16

- Questions from 10/14
- Introduction to Cloud Computing II –From book #1 - Chapter 3: Understanding Cloud Computing Cloud Computing Concepts, Technology & Architecture
  - Benefits of cloud adoption
  - Risks of cloud adoption
- Background on AWS Lambda for the Term Project
- From Book #1: Chapter 4: Cloud Computing Concepts and Models
- At the end: Open Discussion on the Term Project
  - Discussion
  - Team Planning

October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.91

91

TCSS 462/562 TERM PROJECT

October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.92

92

QUESTIONS

October 16, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L6.93

93