



# TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

## Introduction to Cloud Computing

Wes J. Lloyd  
School of Engineering and Technology  
University of Washington – Tacoma  
TR 5:00-7:00 PM



1

### OBJECTIVES – 10/12

■ Questions from 10/7

■ 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

■ 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021

TCSS562:Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington – Tacoma

L4.2

2

# ONLINE DAILY FEEDBACK SURVEY

■ Daily Feedback Quiz in Canvas – Take After Each Class

■ Extra Credit for completing

Announcements

Assignments

Discussions

Zoom

Grades

People

Pages

Files

Quizzes

Collaborations

UW Libraries

UW Resources

▼ Upcoming Assignments

Class Activity 1 – Implicit vs. Explicit Parallelism

Available until Oct 11 at 11:59pm | Due Oct 7 at 7:50pm | -/10 pts

Tutorial 1 - Linux

Available until Oct 19 at 11:59pm | Due Oct 15 at 11:59pm | -/20 pts

▼ Past Assignments

TCSS 562 - Online Daily Feedback Survey - 10/5

Available until Dec 18 at 11:59pm | Due Oct 6 at 8:59pm | -/1 pts

TCSS 562 - Online Daily Feedback Survey - 9/30

Available until Dec 18 at 11:59pm | Due Oct 4 at 8:59pm | -/1 pts

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.3

3

## TCSS 562 - Online Daily Feedback Survey - 10/5

Started: Oct 7 at 1:13am

### Quiz Instructions

Question 1

0.5 pts

On a scale of 1 to 10, please classify your perspective on material covered in today's class:

12345678910

Mostly Review To MeEqual New and ReviewMostly New to Me

Question 2

0.5 pts

Please rate the pace of today's class:

12345678910

SlowJust RightFast

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.4

4

MATERIAL / PACE

- Please classify your perspective on material covered in today’s class (29 respondents):
  - 1-mostly review, 5-equal new/review, 10-mostly new
  - **Average – 7.31 (↑ - previous 6.58)**
- Please rate the pace of today’s class:
  - 1-slow, 5-just right, 10-fast
  - **Average – 5.52 (↓ - previous 5.65)**

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.5
------------------	--	------

5

FEEDBACK FROM 10/7

- ***I’m not sure what a pipeline is?***
- A pipeline in the context of CPUs, refers to processor architectures which feature independent stages of execution, enabling multiple instructions to be executed at the same time in the same CPU core’s “pipeline”
- A pipeline in the context of data processing or application control flow, refers to a sequential sequence of functions that execute iteratively where the output of one function provides the input to the next, etc.
- ***I sense this class is going to shift gear significantly after the initial history + nomenclature phase. Is that accurate?***
- Generally, yes, we will focus more on cloud computing. Things will also become fairly hands-on with the tutorials.

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.6
------------------	--	------

6

OBJECTIVES FROM – 10/7 CONT'D

- Questions from 10/5
- Tutorial 2 – Introduction to Bash Scripting
- Class Activity 1 – Implicit vs Explicit Parallelism
- SIMD architectures, vector processing, multimedia extensions
- Graphics processing units
- Speed-up, Amdahl's Law, Scaled Speedup
- ➡▪ Properties of distributed systems
- Modularity
- Introduction to Cloud Computing –based on book #1: Cloud Computing Concepts, Technology & Architecture

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.7
------------------	--	------

7

DISTRIBUTED SYSTEMS

- Collection of autonomous computers, connected through a network with distribution software called “middleware” that enables coordination of activities and sharing of resources
- Key characteristics:
- Users perceive system as a single, integrated computing facility.
- Compute nodes are autonomous
- Scheduling, resource management, and security implemented by every node
- Multiple points of control and failure
- Nodes may not be accessible at all times
- System can be scaled by adding additional nodes
- Availability at low levels of HW/software/network reliability

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.8
------------------	--	------

8

DISTRIBUTED SYSTEMS - 2

- Key non-functional attributes
  - Known as “ilities” in software engineering
- Availability – 24/7 access?
- Reliability - Fault tolerance
- Accessibility – reachable?
- Usability – user friendly
- Understandability – can under
- Scalability – responds to variable demand
- Extensibility – can be easily modified, extended
- Maintainability – can be easily fixed
- Consistency – data is replicated correctly in timely manner

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.9
------------------	--	------

9

TRANSPARENCY PROPERTIES OF  
DISTRIBUTED SYSTEMS

- **Access transparency:** local and remote objects accessed using identical operations
- **Location transparency:** objects accessed w/o knowledge of their location.
- **Concurrency transparency:** several processes run concurrently using shared objects w/o interference among them
- **Replication transparency:** multiple instances of objects are used to increase reliability
  - *users are unaware if and how the system is replicated*
- **Failure transparency:** concealment of faults
- **Migration transparency:** objects are moved w/o affecting operations performed on them
- **Performance transparency:** system can be reconfigured based on load and quality of service requirements
- **Scaling transparency:** system and applications can scale w/o change in system structure and w/o affecting applications

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.10
------------------	--	-------

10

OBJECTIVES – 10/12

- Questions from 10/7
  - 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
  - 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.11
------------------	--	-------

11

TYPES OF MODULARITY

- Soft modularity: TRADITIONAL
  - Divide a program into modules (classes) that call each other and communicate with shared-memory
  - A procedure calling convention is used (or method invocation)
- Enforced modularity: CLOUD COMPUTING
  - Program is divided into modules that communicate only through message passing
  - The ubiquitous client-server paradigm
  - Clients and servers are independent decoupled modules
  - System is more robust if servers are stateless
  - May be scaled and deployed separately
  - May also FAIL separately!

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.12
------------------	--	-------

12

CLOUD COMPUTING – HOW DID WE GET HERE?  
SUMMARY OF KEY POINTS

- Multi-core CPU technology and hyper-threading
- What is a
  - Heterogeneous system?
  - Homogeneous system?
  - Autonomous or self-organizing system?
- **Fine grained vs. coarse grained parallelism**
- Parallel message passing code is easier to debug than shared memory (e.g. p-threads)
- Know your application's max/avg **Thread Level Parallelism (TLP)**
- **Data-level parallelism:** Map-Reduce, (SIMD) Single Instruction Multiple Data, Vector processing & GPUs

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.13
------------------	--	-------

13

CLOUD COMPUTING – HOW DID WE GET HERE?  
SUMMARY OF KEY POINTS - 2

- **Bit-level parallelism**
- **Instruction-level parallelism** (CPU pipelining)
- **Flynn's taxonomy:** computer system architecture classification
  - **SISD** – Single Instruction, Single Data (modern core of a CPU)
  - **SIMD** – Single Instruction, Multiple Data (Data parallelism)
  - **MIMD** – Multiple Instruction, Multiple Data
  - MISD is RARE; application for fault tolerance...
- **Arithmetic Intensity:** ratio of calculations vs memory RW
- **Roofline model:**  
Memory bottleneck with low arithmetic intensity
- **GPUs:** ideal for programs with high arithmetic intensity
  - SIMD and Vector processing supported by many large registers

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.14
------------------	--	-------

14



CLOUD COMPUTING – HOW DID WE GET HERE?  
SUMMARY OF KEY POINTS - 3

- **Speed-up (S)**  
 $S(N) = T(1) / T(N)$
- **Amdahl's law:**  
 $S = 1 / ((1-f) + f/N)$   
f= fraction of work that is parallel (e.g. 0.25)  
N= proposed speed up of the parallel part (e.g. 5x)
- **Gustafson's Scaled speedup with N processes:**  
 $S(N) = N + (1 - N) \alpha$   
N: Number of processors  
 $\alpha$ : fraction of program run time which can't be parallelized
- Moore's Law
- Symmetric core, Asymmetric core, Dynamic core CPU
- Distributed Systems Non-function quality attributes
- Distributed Systems – Types of Transparency
- Types of modularity- Soft, Enforced

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.15
------------------	--	-------

15

INTRODUCTION TO  
CLOUD COMPUTING



October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.16
------------------	--	-------

16



OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562:Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.17
------------------	---	-------

17

OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562:Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.18
------------------	---	-------

18

WHY STUDY CLOUD COMPUTING?

- **LINKEDIN - TOP IT Skills** from job app data
  - #1 Cloud and Distributed Computing
  - <https://learning.linkedin.com/week-of-learning/top-skills>
  - #2 Statistical Analysis and Data Mining
- **FORBES Survey – 6 Tech Skills That’ll Help You Earn More**
  - #1 Data Science
  - #2 Cloud and Distributed Computing
  - <http://www.forbes.com/sites/laurencebradford/2016/12/19/6-tech-skills-thatll-help-you-earn-more-in-2017/>

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.19

19

WHY STUDY CLOUD COMPUTING? - 2

- **Computerworld Magazine**

**TECH FORECAST 2017 SPECIAL REPORT**

## Hot Skills

**Top 10 skills respondents plan to hire for in the next 12 months:**

Source: Computerworld's Forecast 2017 survey of 196 IT managers, directors and executives.

Base: 57 respondents who expect to increase IT head count in the next 12 months.

Programming/application development	35%
Help desk/tech support	35%
Security/compliance/governance	26%
<b>Cloud/SaaS</b>	<b>26%</b>
Business intelligence/analytics	26%
Web development	26%
Database administration	25%
Project management	25%
Big data	25%
Mobile applications and device management	21%

© COMPUTERWORLD

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.20

20

OBJECTIVES – 10/12


- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.21
------------------	--	-------

21

A BRIEF HISTORY OF CLOUD COMPUTING

- John McCarthy, 1961
  - Turing award winner for contributions to AI
- “If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry...”



October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.22
------------------	--	-------

22

CLOUD HISTORY - 2

- Internet based computer utilities
  - Since the mid-1990s
  - Search engines: Yahoo!, Google, Bing
  - Email: Hotmail, Gmail
- 2000s
  - Social networking platforms: MySpace, Facebook, LinkedIn
  - Social media: Twitter, YouTube
- Popularized core concepts
- Formed basis of cloud computing

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.23
------------------	--	-------

23

CLOUD HISTORY: SERVICES - 1

- Late 1990s – Early Software-as-a-Service (SaaS)
  - Salesforce: Remotely provisioned services for the enterprise
- 2002 -
  - Amazon Web Services (AWS) platform: Enterprise oriented services for remotely provisioned storage, computing resources, and business functionality
- 2006 – Infrastructure-as-a-Service (IaaS)
  - Amazon launches Elastic Compute Cloud (EC2) service
  - Organization can “lease” computing capacity and processing power to host enterprise applications
  - Infrastructure

October 12, 2021	TCCS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.24
------------------	--	-------

24

CLoud HISTORY: SERVICES - 2

- 2006 – **Software-as-a-Service (SaaS)**
  - Google: Offers Google DOCS, “MS Office” like fully-web based application for online documentation creation and collaboration
- 2009 – **Platform-as-a-Service (PaaS)**
  - Google: Offers Google App Engine, publicly hosted platform for hosting scalable web applications on google-hosted datacenters

October 12, 2021


TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.25

25

CLoud COMPUTING  
NIST GENERAL DEFINITION

“Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications and services) that can be rapidly provisioned and reused with minimal management effort or service provider interaction”...



October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.26

26

## MORE CONCISE DEFINITION

“Cloud computing is a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources.”

From Cloud Computing Concepts, Technology, and Architecture  
Z. Mahmood, R. Puttini, Prentice Hall, 5<sup>th</sup> printing, 2015

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.27

27

## OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.28

28

BUSINESS DRIVERS  
FOR CLOUD COMPUTING

- Capacity planning
- Cost reduction
- Operational overhead
- Organizational agility

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.29
------------------	--	-------

29

BUSINESS DRIVERS  
FOR CLOUD COMPUTING

- Capacity planning
  - Process of determining and fulfilling future demand for IT resources
  - Capacity vs. demand
    - Discrepancy between capacity of IT resources and actual demand
    - Over-provisioning: resource capacity exceeds demand
    - Under-provisioning: demand exceeds resource capacity
  - Capacity planning aims to minimize the discrepancy of available resources vs. demand

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.30
------------------	--	-------

30



Dwight, The Office TV sitcom

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.31

31

BUSINESS DRIVERS FOR CLOUD - 2

- Capacity planning
  - Over-provisioning: is costly due to too much infrastructure
  - Under-provisioning: is costly due to potential for business loss from poor quality of service
- Capacity planning strategies
  - Lead strategy: add capacity in anticipation of demand (pre-provisioning)
  - Lag strategy: add capacity when capacity is fully leveraged
  - Match strategy: add capacity in small increments as demand increases
- Load prediction
  - Capacity planning helps anticipate demand fluctuations

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.32

32



CAPACITY PLANNING

Capacity vs. Usage  
(Traditional Data Center)

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.33

33

CAPACITY PLANNING - 2

■ Ca

Predictions Cost Money...

Capacity

Compute Storage ...

Large Capital Expenditure

Opportunity Cost

You just lost customers

Source: Amazon Web Services

Time

Capacity-Cost Performance

Predicted Demand

Traditional Hardware

Actual Demand

Automated Cloud capacity

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.34

34

BUSINESS DRIVERS FOR CLOUD - 3

- Cost reduction
  - IT Infrastructure acquisition
  - IT Infrastructure maintenance
- Operational overhead
  - Technical personnel to maintain physical IT infrastructure
  - System upgrades, patches that add testing to deployment cycles
  - Utility bills, capital investments for power and cooling
  - Security and access control measures for server rooms
  - Admin and accounting staff to track licenses, support agreements, purchases

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.35
------------------	--	-------

35

BUSINESS DRIVERS FOR CLOUD - 4

- Organizational agility
  - Ability to adapt and evolve infrastructure to face change from internal and external business factors
  - Funding constraints can lead to insufficient on premise IT
  - Cloud computing enables IT resources to scale with a lower financial commitment

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.36
------------------	--	-------

36

OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.37
------------------	--	-------

37

TECHNOLOGY INNOVATIONS  
LEADING TO CLOUD

- Cluster computing
- Grid computing
- Virtualization
- Others

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.38
------------------	--	-------

38

## CLUSTER COMPUTING



- **Cluster computing (clustering)**
  - Cluster is a group of independent IT resources interconnected as a single system
  - Servers configured with homogeneous hardware and software
    - Identical or similar RAM, CPU, HDDs
  - Design emphasizes redundancy as server components are easily interchanged to keep overall system running
    - Example: if a RAID card fails on a key server, the card can be swapped from another redundant server
  - Enables warm replica servers
    - Duplication of key infrastructure servers to provide HW failover to ensure high availability (HA)

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.39

39

## GRID COMPUTING



- On going research area since early 1990s
- Distributed heterogeneous computing resources organized into logical pools of loosely coupled resources
- For example: heterogeneous servers connected by the internet
- Resources are heterogeneous and geographically dispersed
- Grids use middleware software layer to support workload distribution and coordination functions
- Aspects: load balancing, failover control, autonomic configuration management
- Grids have influenced clouds contributing common features: networked access to machines, resource pooling, scalability, and resiliency

October 12, 2021

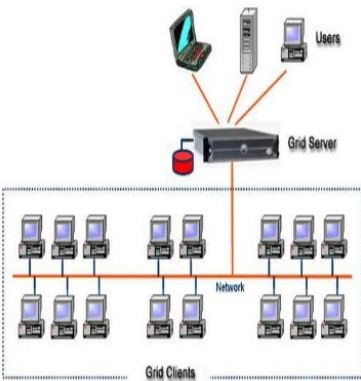
TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.40

40

GRID COMPUTING - 2

### How Grid computing works ?



The diagram illustrates the architecture of a grid computing system. At the top, three mobile devices (laptop, PDA, and smartphone) are labeled 'Users'. Arrows point from these users to a central 'Grid Server' represented by a server rack icon. Below the server, a horizontal line represents the 'Network'. Underneath the network, there are multiple groups of computer icons, each group labeled 'Grid Clients'. Arrows indicate communication between the Grid Server and the Grid Clients through the Network.

In general, a grid computing system requires:

- At least one computer, usually a server, which handles all the administrative duties for the System
- A network of computers running special grid computing network software.
- A collection of computer software called middleware

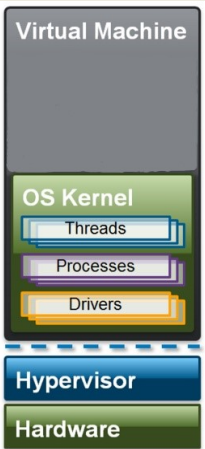
October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.41

41

VIRTUALIZATION



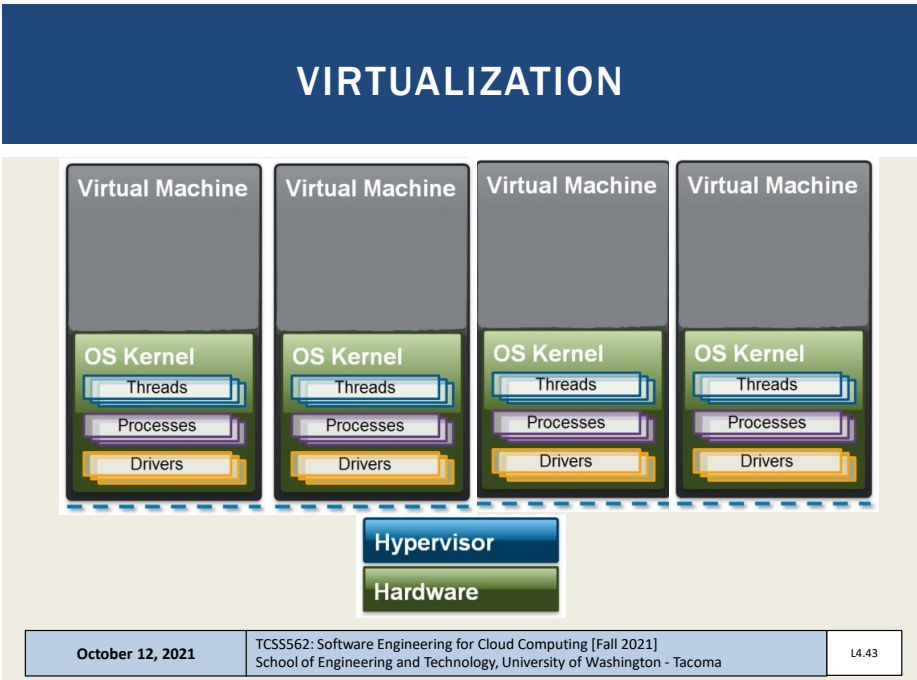
The diagram shows a vertical stack of components representing the virtualization layer. At the top is a box labeled 'Virtual Machine'. Below it is a green box labeled 'OS Kernel', which contains three sub-components: 'Threads' (blue), 'Processes' (purple), and 'Drivers' (yellow). Below the OS Kernel is a blue box labeled 'Hypervisor'. At the bottom is a green box labeled 'Hardware'. A dashed line separates the OS Kernel/Hypervisor layer from the Hardware layer.

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.42

42



43

# 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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.44
------------------	--	-------

44

OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562:Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.45
------------------	---	-------

45

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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.46
------------------	--	-------

46

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

The diagram illustrates vertical scaling. It shows a vertical axis labeled 'vertical scaling'. At the bottom, a dashed box labeled 'A' represents a virtual machine with '2 CPUs'. An upward arrow points to a solid box labeled 'B' at the top, representing a virtual machine with '4 CPUs'. Both VMs are shown within a larger solid box representing a physical host machine.

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.47

47

HORIZONTAL SCALING

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

The diagram illustrates horizontal scaling. At the top, two solid boxes represent 'pooled physical servers'. Arrows point from these servers to a row of dashed boxes representing 'virtual servers'. The sequence of virtual servers is: one box labeled 'A', followed by an arrow labeled 'demand' pointing to two boxes labeled 'A' and 'B', followed by an arrow labeled 'demand' pointing to three boxes labeled 'A', 'B', and 'C'. A long arrow at the bottom labeled 'horizontal scaling' points to the right, indicating the direction of scaling out.

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.48

48



HORIZONTAL VS VERTICAL SCALING

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

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.49

49

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

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.50

50

## 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

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.51

51

## 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

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.52

52

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

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.53

53

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

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.54

54

OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562:Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.55
------------------	---	-------

55

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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.56
------------------	--	-------

56

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...



October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.57
------------------	--	-------

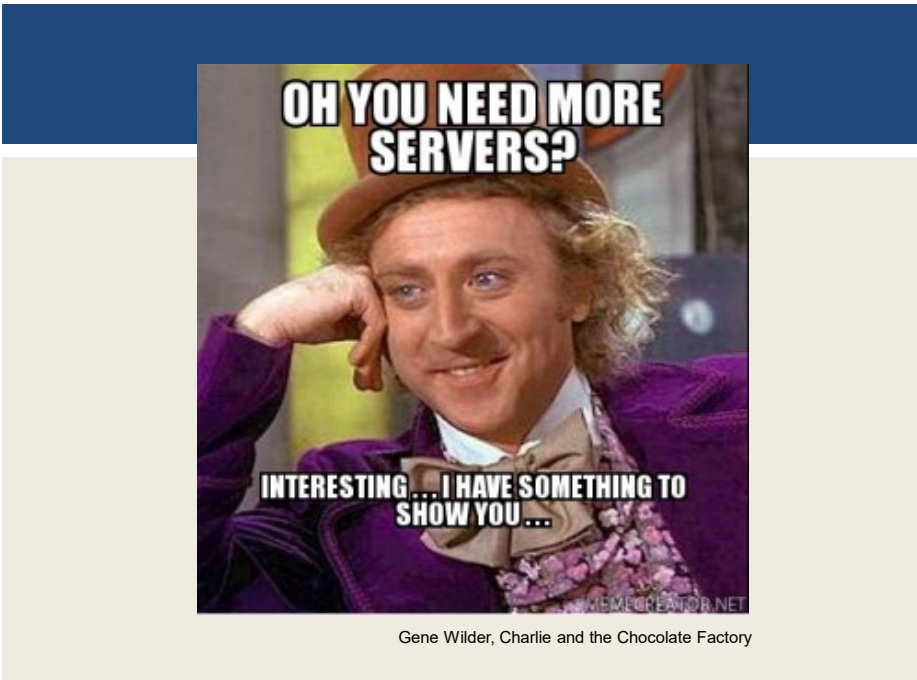
57

CLOUD BENEFITS - 3

- Example: Using 100 servers for 1 hour costs the same as using 1 server for 100 hours
- Rosetta Protein Folding: Working with a UW-Tacoma graduate student, we recently 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)

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.58
------------------	--	-------

58



59

CLOUD BENEFITS

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

The graph illustrates the demand for a service over a 24-hour period. The y-axis represents the number of concurrent users, ranging from 1,000 to 10,000 in increments of 1,000. The x-axis represents time in hours, ranging from 2 to 24. The demand starts at approximately 2,000 users at 2 hours, dips to a minimum of about 1,500 users at 6 hours, then rises sharply to a peak of approximately 9,500 users at 16 hours, before declining back to about 2,000 users by 24 hours.

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.60

60

OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.61
------------------	--	-------

61

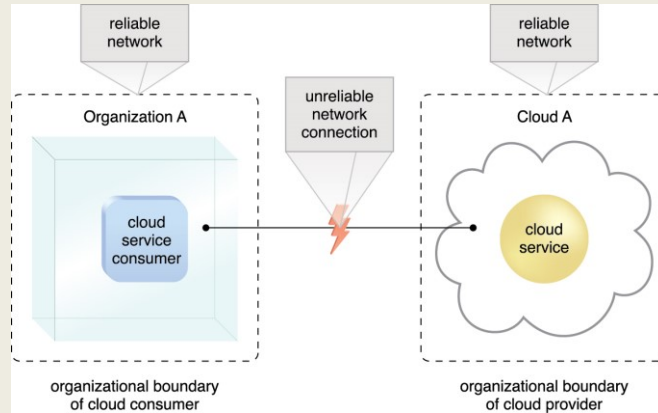
CLOUD ADOPTION RISKS

- 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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.62
------------------	--	-------

62

## NETWORK LATENCY COSTS



October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.63

63

## 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

October 12, 2021

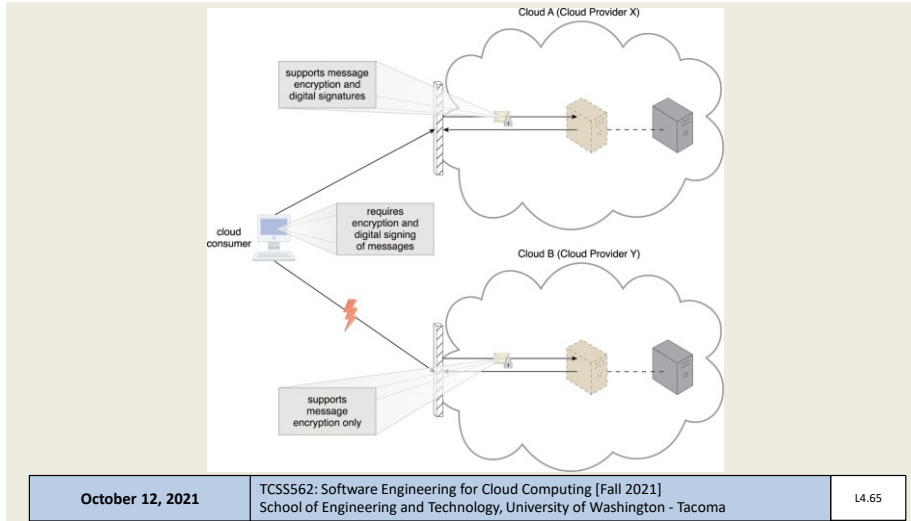
TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.64

64



## CLOUD: VENDOR LOCK-IN



65

## OBJECTIVES – 10/12

- Questions from 10/7
- 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
- 2<sup>nd</sup> hour: TCSS 562 Term Project

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma



L4.66

66

TCSS 562  
TERM PROJECT

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma



L4.67

67

TCSS 562 TERM PROJECT

- Build a serverless cloud native application
- Application provides case study to investigate architecture/design trade-offs
  - Application provides a vehicle to compare and contrast one or more trade-offs
- Alternate 1: Cloud Computing Related Research Project
- Alternate 2: Literature Survey/Gap Analysis
  - \*- as an individual project*

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.68

68

DESIGN TRADE-OFFS

- Service composition
  - Switchboard architecture:
    - compose services in single package
    - Address COLD Starts
    - Infrastructure Freeze/Thaw cycle of AWS Lambda (FaaS)
  - Full service isolation (each service is deployed separately)
- Application flow control
  - client-side, step functions, server-side controller, asynchronous hand-off
- Programming Languages
- Alternate FaaS Platforms

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.69
------------------	--	-------

69

DESIGN TRADE-OFFS - 2

- Alternate Cloud Services (e.g. databases, queues, etc.)
  - Compare alternate data backends for data processing pipeline
- Performance variability (by hour, day, week, and host location)
  - Deployments (to different zones, regions)
- Service abstraction
  - Abstract one or more services with cloud abstraction middleware: Apache libcloud, apache jcloud; make code cross-cloud; measure overhead

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.70
------------------	--	-------

70

OTHER PROJECT IDEAS

- Elastic File System (EFS)  
Performance & Scalability Evaluation
- Docker container image integration with AWS Lambda – performance & scalability
- Resource contention study using CpuSteal metric
  - Investigate the degree of CpuSteal on FaaS platforms
    - What is the extent? Min, max, average
    - When does it occur?
    - Does it correlate with performance outcomes?
    - Is contention self-inflicted?
- & others

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.71

71

SERVERLESS APPLICATIONS

- Extract Transform Load Data Processing Pipeline
  - \* >>>This is the STANDARD project<<< \*
  - Batch-oriented data
  - Stream-oriented data
- Image Processing Pipeline
  - Apply series of filters to images
- Stream Processing Pipeline
  - Data conversion, filtering, aggregation, archival storage
  - What throughput (records/sec) can Lambda ingest directly?
  - Comparison with AWS Kinesis Data Streams and DB backend:
    - <https://aws.amazon.com/getting-started/hands-on/build-serverless-real-time-data-processing-app-lambda-kinesis-s3-dynamodb-cognito-athena/>
  - Kinesis data streams claims multiple GB/sec throughput
    - What is the cost difference?

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.72

72

SERVERLESS APPLICATIONS - 2

- **Map-Reduce Style Application**
  - Function 1: split data into chunks, usually sequentially
  - Function 2: process individual chunks concurrently (in parallel)
    - Data process is considered to be Embarrassingly Parallel
  - Function 3: aggregate and summarize results
- **Image Classification Pipeline**
  - Deploy pretrained image classifiers in a multi-stage pipeline
- **Machine Learning**
  - Multi-stage inferencing pipelines
  - Natural Language Processing (NLP) pipelines
  - Training (?)

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.73
------------------	--	-------

73

AWS LAMBDA PLATFORM LIMITATIONS

- Maximum 10 GB memory per function instance
- Maximum 15-minutes execution per function instance
- Access to 500 MB of temporary disk space for local I/O
- Access up to 6 vCPUs depending on memory reservation size
- 1,000 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
- See: <https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html>

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.74
------------------	--	-------

74

EXTRACT TRANSFORM LOAD  
DATA PIPELINE

- Service 1: **TRANSFORM**
  - Read CSV file, perform some transformations
  - Write out new CSV file
- Service 2: **LOAD**
  - Read CSV file, load data into relational database
  - Cloud DB (AWS Aurora), or local DB (Derby/SQLite)
    - Derby DB and/or SQLite code examples to be provided in Java

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.75
------------------	--	-------

75

EXTRACT TRANSFORM LOAD  
DATA PIPELINE - 2

- Service 3: **QUERY**
  - Using relational database, apply filter(s) and/or functions to aggregate data to produce sums, totals, averages
  - Output aggregations as JSON

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.76
------------------	--	-------

76

SERVICE COMPOSITION

Remote Client

API Gateway

Fine grained services

A	B	C	3 services Full Service Isolation
A B C			2 services
A	B C		2 services
A B C			1 service Full Service Aggregation

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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.77
------------------	--	-------

77

SWITCH-BOARD ARCHITECTURE

Remote Client

API Gateway

Switchboard

1 service

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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.78
------------------	--	-------

78

APPLICATION FLOW CONTROL

- **Serverless Computing:**
  - AWS Lambda (FAAS: Function-as-a-Service)
  - Provides HTTP/REST like web services
  - Client/Server paradigm
- **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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.79
------------------	--	-------

79

APPLICATION FLOW CONTROL - 2

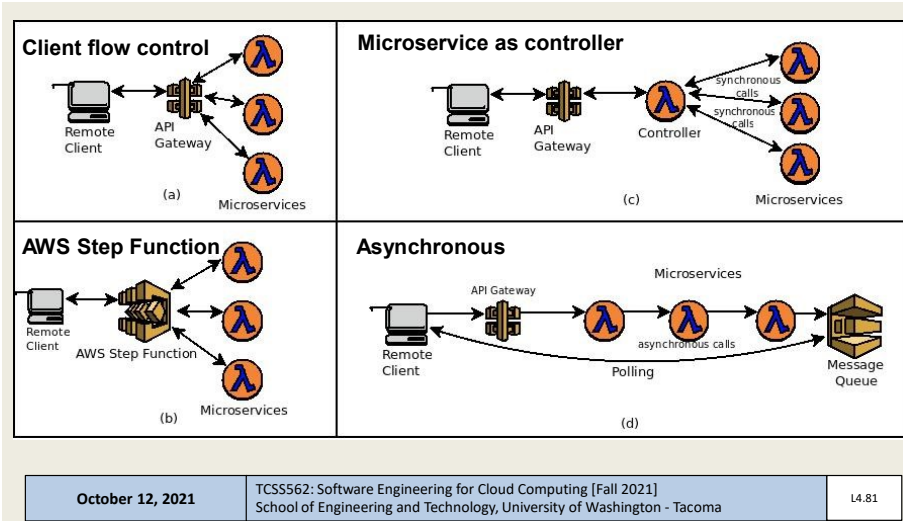
- **Asynchronous web service**
  - Client calls service
  - Server responds to client with OK message
  - Client closes connection
  - Server performs the work associated with the service
  - Server posts service result in an external data store
    - AWS: S3, SQS (queueing service), SNS (notification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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.80
------------------	--	-------

80



## APPLICATION FLOW CONTROL - 3



81

## 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

October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.82

82

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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.83
------------------	--	-------

83

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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.84
------------------	--	-------

84

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...

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.85
------------------	--	-------

85


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

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.86
------------------	--	-------

86

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*:
  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 8<sup>th</sup> column returned
  - Metric can be read using SAAF in tutorial #4

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.87
------------------	--	-------

87


CPUSTEAL 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?

October 12, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L4.88
------------------	--	-------

88

QUESTIONS



October 12, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L4.89