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

Cloud Computing – *How did we get here? – part IV,*

Introduction to Cloud Computing

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

TR 5:50-7:50 PM

1

JOB RECRUITER IN CLASS – TODAY (FIRST 10 MINUTES)

- Tuesday October 14, 3:40pm
- Fast Enterprises, LLC
- Hires consultants that work with state and local governments, and also some federal agencies on an international scale.
- Positions open to domestic and international graduates
  - They do not sponsor H1B visas
  - Professor has asked whether they can hire an OPT student on F-1

2

OBJECTIVES – 10/14

- Questions from 10/9
- 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

3

OFFICE HOURS – FALL 2025

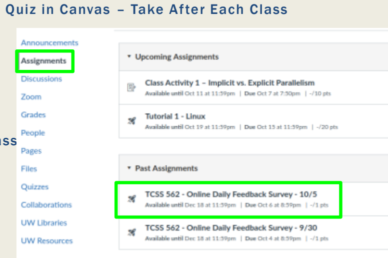
- **Thursdays:**
  - 6:00 to 7:00 pm - CP 229 and Zoom
- **Fridays**
  - 11:00 am to 12:00 pm – ONLINE via Zoom*
- Or email for appointment

> *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.*

4

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)



5

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

6

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:

1 2 3 4 5 6 7 8 9 10

Mostly Review To Me — Equal New and Review — Mostly New to Me

Question 2 — 0.5 pts

Please rate the pace of today's class:

1 2 3 4 5 6 7 8 9 10

Slow — Just Right — Fast

7

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (48 respondents, 35 in-person, 13 online):
- 1-mostly review, 5-equal new/review, 10-mostly new
- **Average – 7.33 (↑ - *previous 7.14*)**

- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- **Average – 4.96 (↓ - previous 5.09)**

8

FEEDBACK FROM 10/9

- ***>Term project themes: if choosing LLM comparison or programming language comparison, just need to compare performance?***
  - Yes, but multiple metrics should be used to quantify performance:
  - Average turnaround time, average runtime, average throughput, cost
- ***>If building the standard or an alternative application, need to select which design trade-offs to compare and to compare performance?***
  - Two course themes for "**design trade-off**" comparison, choose:
    1. LLM comparison: implement same app, with 2 or more LLMs
    2. Prog Lang comparison: implement same app, in 2 or more LLMs
  - Groups are free to compare other design trade-offs, but others are not the "standard" course themes for Fall 2025

9

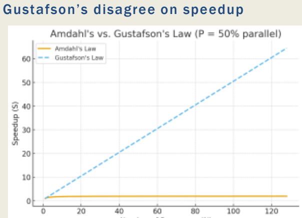
FEEDBACK - 2

- ***>What makes Amdahl's Law different from Gustafson's Law?***
- Amdahl's law was conceived to consider the speed-up for a fixed problem size
  - The amount of work is fixed, and some part is parallel, the rest sequential
- Gustafson's Law considers how much larger of a problem can be solved in the same amount of time when having access to more processors
  - Considers data parallelism, and the ability to process more data in less time with more compute resources
  - The speed-up can grow linearly with the number of processors because the parallel portion (work) grows with the workload and dominates the run time
- ***KEY: Use Amdahl's for estimating speedup of a fixed amount of work, Gustafson for speedup of scalable amount of work***

10

FEEDBACK - 3

- Consequently, if computing the speed-up for a job where 50% is parallelizable, Amdahl's and Gustafson's disagree on speedup
- **Amdahl's Law** (solid line) levels off near a max speedup of about 2×, showing diminishing returns once ½ the workload remains sequential.
- **Gustafson's Law** (dashed line) scales almost linearly up to about 64× with 128 processors, reflecting that if the workload grows, the parallel half can dominate the runtime.



11

FEEDBACK - 4

- ***>What is the grading rubric for the term project?***
- The full body of work produced is considered.
- The final presentation or term paper is graded.
  - Code and other artifacts can be considered
- Points are assigned to each of the components:
  - Description of the case study
  - Description of the application
  - Description of the experiments
  - (*) Project results presentation including graphs, tables, statistics,
  - Results analysis and discussion in paper/presentation
  - Results conclusions in paper/presentation
  - Formatting
- 4.0 = Projects that produce high quality reports that clearly communicate the implications and key outcomes of design trade-offs of their respective case studies

12

FEEDBACK - 5

- *>How does vCPU differ from CPU ?*

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.13

13

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.14

14

TUTORIAL 1 – DUE OCT 16

- 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

October 11, 2022 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L4.15

15

TUTORIAL 2 – DUE 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 11, 2022 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L4.16

16

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.17

17

OBJECTIVES – 10/14

- **Questions from 10/9**
- **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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.18

18

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

19

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

20

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

21

WE WILL RETURN AT 5:02PM

22

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

23

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!

24

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

25

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

26

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) α
  N: Number of processors
  α: 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

27

INTRODUCTION TO CLOUD COMPUTING

28

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

29

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

30

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.31

31

WHY STUDY CLOUD COMPUTING? - 2

- Computerworld Magazine

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.32

32

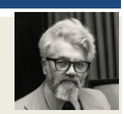
OBJECTIVES – 10/14

- **Questions from 10/9**
- 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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.33

33

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.34

34

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.35

35

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 – <u>Infrastructure-as-a-Service (IaaS)</u>
  - Amazon launches Elastic Compute Cloud (EC2) service
  - Organization can "lease" computing capacity and processing power to host enterprise applications
  - Infrastructure

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.36

36

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

37

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"…

38

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, 5th printing, 2015

39

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

40

BUSINESS DRIVERS FOR CLOUD COMPUTING

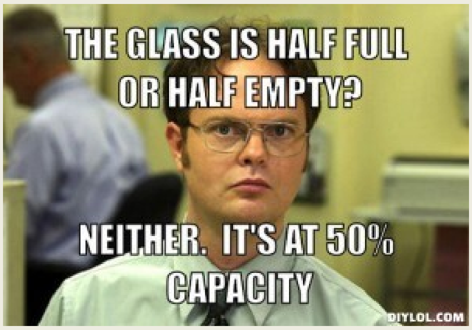
- Capacity planning
- Cost reduction
- Operational overhead
- Organizational agility

41

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

42



Dwight, The Office TV sitcom

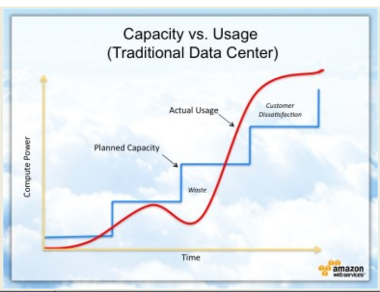
43

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 flucations

44

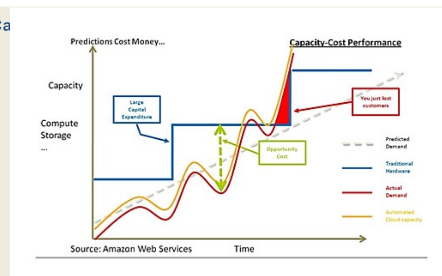
CAPACITY PLANNING



45

CAPACITY PLANNING - 2

- Ca



46

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

47

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

48

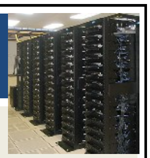
OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

TECHNOLOGY INNOVATIONS LEADING TO CLOUD

- Cluster computing
- Grid computing
- Virtualization
- Others

CLUSTER COMPUTING

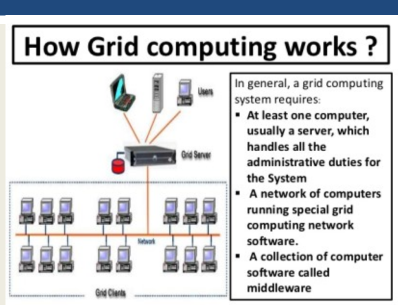


- Cluster computing (clustering)
  - Cluster is a group of independent IT resources interconnected as a single system
  - Servers configured with homogeneous hardware and software
    - Identical or similar RAM, CPU, HDDs
  - Design emphasizes redundancy as server components are easily interchanged to 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)

GRID COMPUTING

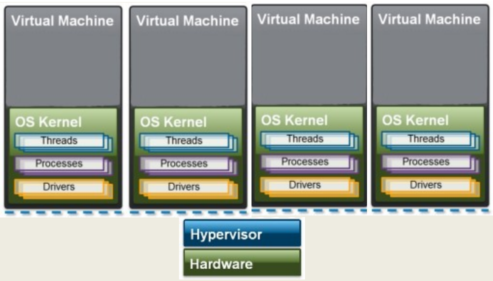
- On going research area since early 1990s
- Distributed heterogeneous computing resources organized into logical pools of loosely coupled resources
- For example: heterogeneous servers connected by the internet
- Resources are heterogeneous and geographically dispersed
- Grids use middleware software 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

GRID COMPUTING - 2



VIRTUALIZATION

VIRTUALIZATION



55

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

56

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

57

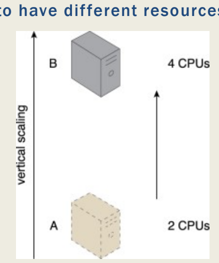
KEY TERMINOLOGY

- **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

58

VERTICAL SCALING

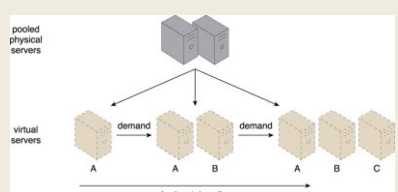
- Reconfigure virtual machine to have different resources:
  - CPU cores
  - RAM
  - HDD/SDD capacity
- May require VM migration if physical host machine resources are exceeded



59

HORIZONTAL SCALING

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



60

HORIZONTAL VS VERTICAL SCALING

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

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.61

61

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.62

62

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.63

63

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.64

64

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.65

65

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 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.66

66

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

67

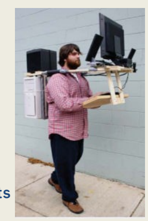
GOALS AND BENEFITS

- 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

68

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...



69

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)

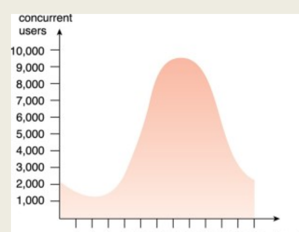
70

Gene Wilder, Charlie and the Chocolate Factory

71

CLOUD BENEFITS

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



72

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

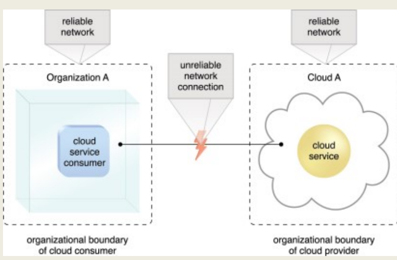
73

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

74

NETWORK LATENCY COSTS



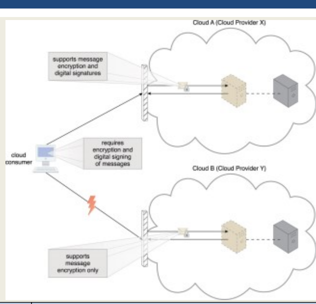
75

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

76

CLOUD: VENDOR LOCK-IN



77

OBJECTIVES – 10/14

- **Questions from 10/9**
- 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

78

TCSS 462/562 TERM PROJECT

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.79

79

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

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.80

80

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

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.81

81

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

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.82

82

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)

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.83

83

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

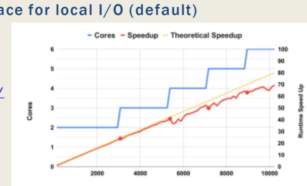


Figure 1: AWS Lambda Performance Speedup for Sysbench Prime Number Generation vs. Function Memory

October 14, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L5.84

84

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

85

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

86

EFFECTS OF SCALING FUNCTION MEMORY ON CPU TIME SHARE

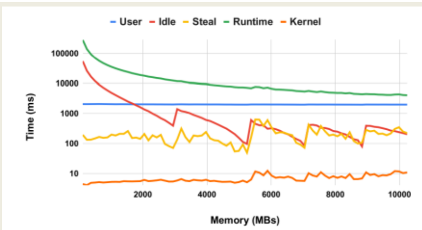


Figure 2: Linux CPU Utilization (log scale) vs. Function Memory for Sysbench Prime Number Generation

87

EFFECTS OF SCALING FUNCTION MEMORY ON CPU TIME SHARE

<u>Key observations:</u>
- 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

88

FUNCTION INSTANCE LIFE CYCLES

- Function states:
- <u>**COLD**</u>: 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)
- <u>**WARM**</u>: 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

89

WARM VS COLD FUNCTION INSTANCES

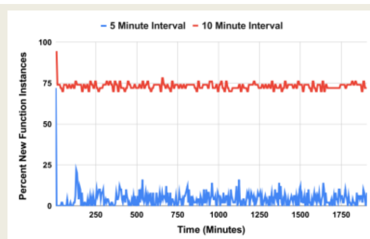


Figure 3: AWS Lambda Function Instance Replacement vs. Function Call Interval over 24-hours

90

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

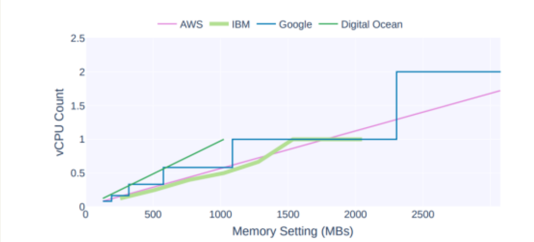
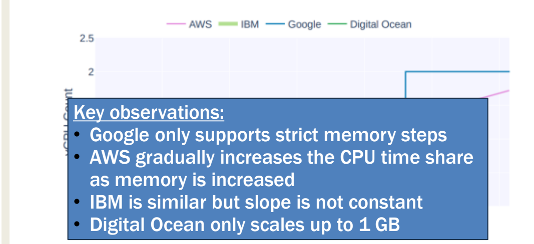


Fig. 2. Allocated vCPUs available at each memory setting on each platform.

91

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.

92

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

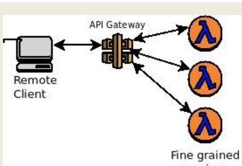
93

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

94

SERVICE COMPOSITION

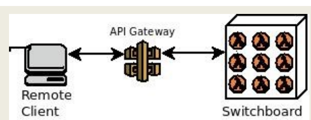


A	B	C	*3 services* **Full Service Isolation**
A B		C	*2 services*
A	B C		*2 services*
A B C			*1 service* **Full Service Aggregation**

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

95

SWITCH-BOARD ARCHITECTURE



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

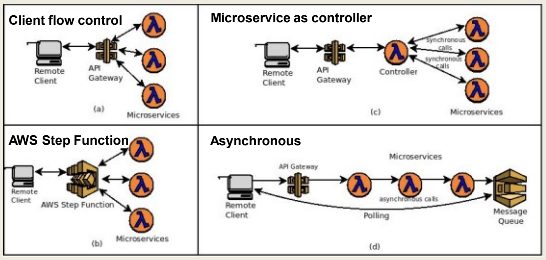
Entry method contains "switchboard" logic
Case statement that route calls to proper service

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

Goal: reduce # of COLD starts to improve performance

96

APPLICATION FLOW CONTROL - 3



97

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

98

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)*

99

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)
- <u>SQL / Relational:</u>
- Amazon Aurora (serverless cloud DB), Amazon RDS (cloud DB), DB on a VM (MySQL), DB inside Lambda function (SQLite, Derby)
- <u>NO SQL / Key/Value Store:</u>
- Dynamo DB, MongoDB, S3

100

PERFORMANCE VARIABILITY

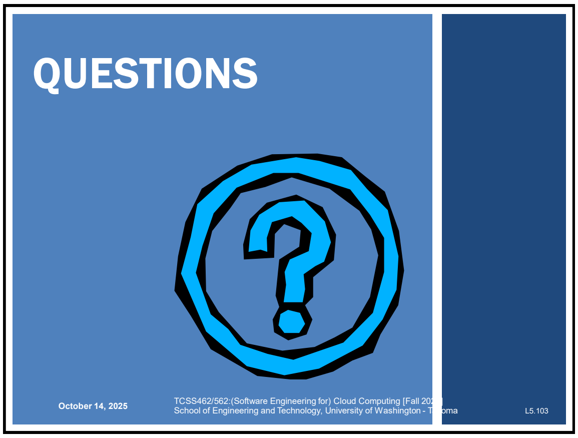
- Cloud platforms exhibit performance variability which varies over time
- Goal of this case study is to measure performance variability (i.e. extent) for AWS Lambda services by hour, day, week to look for common patterns
- Can also examine performance variability by availability zone and region
  - 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…

101

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?

102


QUESTIONS
October 14, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 20
School of Engineering and Technology, University of Washington - T
oma
L5.103

103