

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

**Cloud Computing –
 How did we get here? – part III,
 Introduction to Cloud Computing**

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



1

OBJECTIVES – 10/10

Questions from 10/5

- Tutorial 0, Tutorial 1, Tutorial 2
- Term Project Proposal
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units
- Speed-up, Amdahl's Law, Scaled Speedup
- Properties of distributed systems
- Modularity
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture

October 10, 2023 TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.2

2

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (47 respondents):
- 1-mostly review, 5-equal new/review, 10-mostly new
- Average – 6.80** (↓ - previous 6.83)

- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- Average – 5.64** (↓ - previous 6.26)

Response rates:

- TCSS 462: 42/45 – 93.33%
- TCSS 562: 22/25 – 88.00%

October 10, 2023 TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.3

3

FEEDBACK FROM 10/5

- I don't understand why bit-level and instruction-level parallelism are implicit*
- My understanding for implicit is that everything is working on (the) inside and there has (to be) an automatic detection by (the) compiler, so (the) programmer doesn't need to focus on the task (of making things run in parallel).*

Follow-on:

- Wasn't entirely clear on what made something implicitly parallel vs explicitly parallel*

October 10, 2023 TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.4

4

FEEDBACK - 2

- Pipeline processing – I don't understand it completely*

- Specific details regarding processor implementations of pipeline progressing are beyond the scope of the class
- The main idea we are trying to convey is that of **SLIDE 2.42** where stages of instruction execution are broken into separate compute units on the CPU (i.e. fetch, decode, execute, write-back)
- Execution of an instruction must go through all stages
- Units can work on executing different instructions in parallel

October 10, 2023 TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.5

5

AWS CLOUD CREDITS UPDATE

- Our AWS cloud credit request has been received after UW census day (last Friday) finalized the registration of the course
- Amazon is processing our request for cloud credits
- We hope that cloud credits will be made available soon for TCSS 462/562

October 10, 2023 TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.6

6

OBJECTIVES – 10/10

- Questions from 10/5
- Tutorial 0, Tutorial 1, Tutorial 2**
- Term Project Proposal
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units
- Speed-up, Amdahl's Law, Scaled Speedup
- Properties of distributed systems
- Modularity
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.7
------------------	--	------

7

TUTORIAL 1

- Introduction to Linux & the Command Line**
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_tutorial_1.pdf
- Tutorial Sections:**
 - The Command Line
 - Basic Navigation
 - More About Files
 - Manual Pages
 - File Manipulation
 - VI – Text Editor
 - Wildcards
 - Permissions
 - Filters
 - Grep and regular expressions
 - Piping and Redirection
 - Process Management

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.8
------------------	--	------

8

TUTORIAL 2

- Introduction to Bash Scripting**
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_tutorial_2.pdf
- Review tutorial sections:
- Create a BASH webservice client
 - What is a BASH script?
 - Variables
 - Input
 - Arithmetic
 - If Statements
 - Loops
 - Functions
 - User Interface
- Call service to obtain IP address & lat/long of computer
- Call service to obtain weather forecast for lat/long

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.9
------------------	--	------

9

OBJECTIVES – 10/10

- Questions from 10/5
- Tutorial 0, Tutorial 1, Tutorial 2
- Term Project Proposal**
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units
- Speed-up, Amdahl's Law, Scaled Speedup
- Properties of distributed systems
- Modularity
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.10
------------------	--	-------

10

OBJECTIVES – 10/10

- Questions from 10/5
- Tutorial 0, Tutorial 1, Tutorial 2
- Term Project Proposal
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units**
- Speed-up, Amdahl's Law, Scaled Speedup
- Properties of distributed systems
- Modularity
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.11
------------------	--	-------

11

GRAPHICAL PROCESSING UNITS (GPUs)

- GPU provides multiple SIMD processors
- Typically 7 to 15 SIMD processors each
- 32,768 total registers, divided into 16 lanes (2048 registers each)
- GPU programming model: single instruction, multiple thread
- Programmed using CUDA- C like programming language by NVIDIA for GPUs
- CUDA threads – single thread associated with each data element (e.g. vector or matrix)
- Thousands of threads run concurrently

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.12
------------------	--	-------

12

OBJECTIVES – 10/10

- Questions from 10/5
- Tutorial 0, Tutorial 1, Tutorial 2
- Term Project Proposal
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units
- **Speed-up, Amdahl's Law, Scaled Speedup**
- Properties of distributed systems
- Modularity
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.13

13

PARALLEL COMPUTING

- Parallel hardware and software systems allow:
 - Solve problems demanding resources not available on single system.
 - Reduce time required to obtain solution
- The *speed-up* (S) measures effectiveness of parallelization:

$$S(N) = T(1) / T(N)$$

T(1) → execution time of total sequential computation
 T(N) → execution time for performing N parallel computations in parallel

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.14

14

SPEED-UP EXAMPLE

- Consider embarrassingly parallel image processing
- Eight images (multiple data)
- Apply image transformation (greyscale) in parallel
- 8-core CPU, 16 hyperthreads
- Sequential processing: perform transformations one at a time using a single program thread
 - 8 images, 3 seconds each: T(1) = 24 seconds
- Parallel processing
 - 8 images, 3 seconds each: T(N) = 3 seconds
- Speedup: S(N) = 24 / 3 = 8x speedup
- Called "**perfect scaling**"
- Must consider data transfer and computation setup time

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.15

15

AMDAHL'S LAW

- Amdahl's law is used to estimate the speed-up of a job using parallel computing
- 1. Divide job into two parts
- 2. Part A that will still be sequential
- 3. Part B that will be sped-up with parallel computing
- Portion of computation which cannot be parallelized will determine (i.e. limit) the overall speedup
- Amdahl's law assumes jobs are of a fixed size
- Also, Amdahl's assumes no overhead for distributing the work, and a perfectly even work distribution

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.16

16

AMDAHL'S LAW

Speed-up formula →
$$S = \frac{1}{(1-f) + \frac{f}{N}}$$

- S = theoretical speedup of the whole task
- f = fraction of work that is parallel (ex. 25% or 0.25)
- N = proposed speed up of the parallel part (ex. 5 times speedup)
- % improvement of task execution = 100 * (1 - (1/S))
- **Using Amdahl's law, we can find the maximum possible speed-up (S) for a given scenario (e.g. -8x)...**

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.17

17

AMDAHL'S LAW EXAMPLE

- Program with two independent parts:
 - Part A is 75% of the execution time
 - Part B is 25% of the execution time
- Part B is made 5 times faster with parallel computing
- Estimate the percent improvement of task execution
- Original Part A is 3 seconds, Part B is 1 second
- N=5 (speedup of part B)
- f=.25 (only 25% of the whole job (A+B) will be sped-up)
- S=1 / ((1-f) + f/S)
- S=1 / ((.75) + .25/5)
- S=1.25 (speed up is 1.25x faster)
- % improvement = 100 * (1 - 1/1.25) = **20%**

Two independent parts A B

Original process: [Red bar 75%] [Blue bar 25%]

Make B 5x faster: [Red bar 75%] [Blue bar 5%]

Make A 2x faster: [Red bar 37.5%] [Blue bar 6.25%]

from Wikipedia

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.18

18

GUSTAFSON'S LAW

- Calculates the **scaled speed-up** using "N" processors

$$S(N) = N + (1 - N) \alpha$$

N: Number of processors
 α: fraction of program run time which can't be parallelized (e.g. must run sequentially)

- Can be used to estimate runtime of parallel portion of program

October 10, 2023
TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L4.19

19

GUSTAFSON'S LAW

- Calculates the **scaled speed-up** using "N" processors

$$S(N) = N + (1 - N) \alpha$$

N: Number of processors
 α: fraction of program run time which can't be parallelized (e.g. must run sequentially)

- Can be used to estimate runtime of parallel portion of program
- Where $\alpha = \sigma / (\pi + \sigma)$
- Where σ = sequential time, π = parallel time
- Our Amdahl's example: $\sigma = 3s$, $\pi = 1s$, $\alpha = .75$

October 10, 2023
TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L4.20

20

GUSTAFSON'S LAW

- Calculates the **scaled speed-up** using "N" processors

$$S(N) = N + (1 - N) \alpha$$

N: Number of processors
 α: fraction of program run time which can't be parallelized (e.g. must run sequentially)

- Example:**
 Consider a program that is embarrassingly parallel, but 75% cannot be parallelized. $\alpha = .75$
QUESTION: If deploying the job on a 2-core CPU, what scaled speedup is possible assuming the use of two processes that run in parallel?

October 10, 2023
TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L4.21

21

GUSTAFSON'S EXAMPLE

- QUESTION:**
 What is the maximum theoretical speed-up on a **2-core CPU**?
 $S(N) = N + (1 - N) \alpha$
 $N=2$, $\alpha=.75$
 $S(N) = 2 + (1 - 2) .75$
 $S(N) = ?$
- What is the maximum theoretical speed-up on a **16-core CPU**?
 $S(N) = N + (1 - N) \alpha$
 $N=16$, $\alpha=.75$
 $S(N) = 16 + (1 - 16) .75$
 $S(N) = ?$

October 10, 2023
TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L4.22

22

GUSTAFSON'S EXAMPLE

- QUESTION:**
 What is the maximum theoretical speed-up on a **2-core CPU**?
 $S(N) = N + (1 - N) \alpha$
 $N=2$, $\alpha=.75$
 $S(N) = 2 + (1 - 2) .75$
 $S(N) = ?$
 For 2 CPUs, speed up is 1.25x
- What is the maximum theoretical speed-up on a **16-core CPU**?
 $S(N) = N + (1 - N) \alpha$
 $N=16$, $\alpha=.75$
 $S(N) = 16 + (1 - 16) .75$
 $S(N) = ?$
 For 16 CPUs, speed up is 4.75x

October 10, 2023
TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L4.23

23

MOORE'S LAW

- Transistors on a chip doubles approximately every 1.5 years
- CPUs now have 100s of cores
- Power dissipation is a major concern
 What kind of processor are modern Intel CPUs?
 Transistors are packed so closely that heat removal is a challenge
- Symmetric core processor** – multi-core CPU, all cores have the same computational resources and speed
- Asymmetric core processor** – on a multi-core CPU, some cores have more resources and speed
- Dynamic core processor** – processing resources and speed can be dynamically configured among cores
- Observation: asymmetric processors offer a higher speedup**

October 10, 2023
TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L4.24

24

OBJECTIVES – 10/10

- Questions from 10/5
- Tutorial 0, Tutorial 1, Tutorial 2
- Term Project Proposal
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units
- Speed-up, Amdahl's Law, Scaled Speedup
- **Properties of distributed systems**
- Modularity
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.25
------------------	--	-------

25

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 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.26
------------------	--	-------

26

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 understand
- 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 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.27
------------------	--	-------

27

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 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.28
------------------	--	-------

28

OBJECTIVES – 10/10

- Questions from 10/5
- Tutorial 0, Tutorial 1, Tutorial 2
- Term Project Proposal
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units
- Speed-up, Amdahl's Law, Scaled Speedup
- Properties of distributed systems
- **Modularity**
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture

October 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.29
------------------	--	-------

29

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 10, 2023	TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L4.30
------------------	--	-------

30

CLOUD COMPUTING – HOW DID WE GET HERE? - PART III
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 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.31

31

CLOUD COMPUTING – HOW DID WE GET HERE? - PART III
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
- GPU:** ideal for programs with high arithmetic intensity
 - SIMD and Vector processing supported by many large registers

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.32

32

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

- Speed-up (S)**
 $S(N) = T(1) / T(N)$
- Amdahl's law:**
 $S = 1 / \alpha$
 α = percent of program that must be sequential
- Scaled speedup with N processes:**
 $S(N) = N - \alpha(N-1)$
- 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 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.33

33


OBJECTIVES – 10/10

- Questions from 10/5
- Tutorial 0, Tutorial 1, Tutorial 2
- Term Project Proposal
- Cloud Computing – How did we get here? - part III (Marinescu Ch. 2 - 1st edition, Ch. 4 - 2nd edition)
- Graphics processing units
- Speed-up, Amdahl's Law, Scaled Speedup
- Properties of distributed systems
- Modularity
- Introduction to Cloud Computing – loosely based on book #1: Cloud Computing Concepts, Technology & Architecture**

October 10, 2023 TCSS462/562: Software Engineering for Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L4.34

34

INTRODUCTION TO CLOUD COMPUTING



October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma L4.35

35

OBJECTIVES – 10/10

- Introduction to Cloud Computing
 - Why study cloud computing?**
 - History of cloud computing
 - Business drivers
 - Cloud enabling technologies
 - Terminology
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma L4.36

36

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 10, 2023
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma
L4.37

37

WHY STUDY CLOUD COMPUTING? - 2

- Computerworld Magazine

October 10, 2023
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma
L4.38

38

OBJECTIVES – 10/10

- Introduction to Cloud Computing
 - Why study cloud computing?
 - **History of cloud computing**
 - Business drivers
 - Cloud enabling technologies
 - Terminology
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 10, 2023
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma
L4.39

39

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 10, 2023
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma
L4.40

40

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 10, 2023
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma
L4.41

41

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 10, 2023
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma
L4.42

42

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 10, 2023 | TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma | L4.43

43

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 10, 2023 | TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma | L4.44

44

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

October 10, 2023 | TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma | L4.45

45

OBJECTIVES - 10/10

- Introduction to Cloud Computing
 - Why study cloud computing?
 - History of cloud computing
 - Business drivers**
 - Cloud enabling technologies
 - Terminology
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 10, 2023 | TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma | L4.46

46

BUSINESS DRIVERS FOR CLOUD COMPUTING

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

October 10, 2023 | TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma | L4.47

47

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 10, 2023 | TCSS562: Software Engineering for Cloud Computing [Fall 2019] School of Engineering and Technology, University of Washington - Tacoma | L4.48

48

DIYLOL.COM
 Dwight, The Office TV sitcom

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.49

49

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.50

50

CAPACITY PLANNING

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.51

51

CAPACITY PLANNING - 2

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.52

52

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.53

53

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.54

54

OBJECTIVES - 10/10

- Introduction to Cloud Computing
 - Why study cloud computing?
 - History of cloud computing
 - Business drivers
 - Cloud enabling technologies
 - Terminology
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.55

55


TECHNOLOGY INNOVATIONS LEADING TO CLOUD

- Cluster computing
- Grid computing
- Virtualization
- Others

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.56

56

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.57

57

GRID COMPUTING



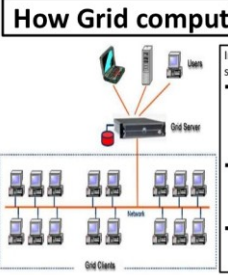
- On going research area since early 1990s
- Distributed heterogeneous computing resources organized into logical pools of loosely coupled resources
- For example: heterogeneous servers connected by the internet
- Resources are heterogeneous and geographically dispersed
- Grids use middleware software 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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.58

58

GRID COMPUTING - 2

How Grid computing works ?




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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.59

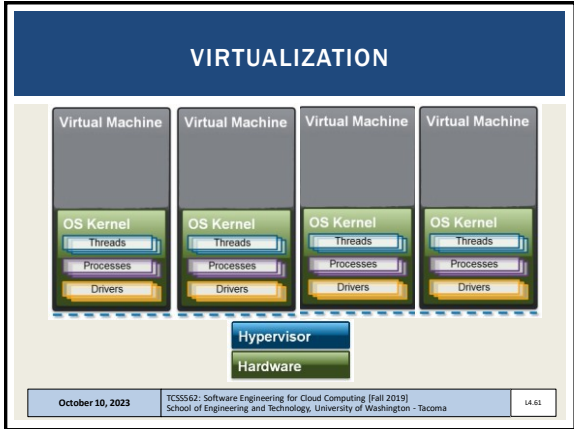
59

VIRTUALIZATION



October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.60

60



61

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.62

62

OBJECTIVES - 10/10

- Introduction to Cloud Computing
 - Why study cloud computing?
 - History of cloud computing
 - Business drivers
 - Cloud enabling technologies
 - **Terminology**
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.63

63

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.64

64

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 shows a vertical scaling process. A VM with 2 CPUs (labeled A) is upgraded to a VM with 4 CPUs (labeled B). The upgrade is indicated by an upward arrow labeled 'vertical scaling'.

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.65

65

HORIZONTAL SCALING

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

The diagram illustrates horizontal scaling. At the top, 'pooled physical servers' are shown. Below them, 'virtual servers' are shown as individual units. Arrows labeled 'demand' point from the physical servers to the virtual servers. The virtual servers are labeled A, B, and C, and are arranged in a row. A horizontal arrow at the bottom is labeled 'horizontal scaling'.

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.66

66

HORIZONTAL VS VERTICAL SCALING

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

October 10, 2023
L4.67
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma

67

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 10, 2023
L4.68
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma

68

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 10, 2023
L4.69
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma

69

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 10, 2023
L4.70
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma

70

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 10, 2023
L4.71
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma

71

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 10, 2023
L4.72
TCSS562: Software Engineering for Cloud Computing [Fall 2019]
School of Engineering and Technology, University of Washington - Tacoma

72

OBJECTIVES - 10/10

- Introduction to Cloud Computing
 - Why study cloud computing?
 - History of cloud computing
 - Business drivers
 - Cloud enabling technologies
 - Terminology
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.73

73

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.74

74

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.75

75


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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.76

76

OH YOU NEED MORE SERVERS?



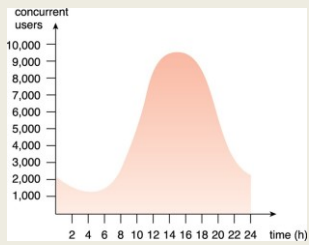
INTERESTING... I HAVE SOMETHING TO SHOW YOU...

Gene Wilder, Charlie and the Chocolate Factory

77

CLOUD BENEFITS

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



October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.78

78

OBJECTIVES - 10/10

- Introduction to Cloud Computing
 - Why study cloud computing?
 - History of cloud computing
 - Business drivers
 - Cloud enabling technologies
 - Terminology
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.79

79

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.80

80

NETWORK LATENCY COSTS

The diagram illustrates the organizational boundaries of a cloud consumer (Organization A) and a cloud provider (Cloud A). Both have reliable internal networks. However, the connection between them is an 'unreliable network connection', which is highlighted with a lightning bolt icon, indicating network latency costs.

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.81

81

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 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.82

82

CLOUD: VENDOR LOCK-IN

The diagram shows a cloud consumer interacting with two cloud providers: Cloud A (Cloud Provider X) and Cloud B (Cloud Provider Y). Cloud A supports message encryption and digital signatures, while Cloud B only supports message encryption. A lightning bolt icon indicates a barrier or lock-in preventing the consumer from easily switching to Cloud B.

October 10, 2023 TCSS562: Software Engineering for Cloud Computing [Fall 2019]
 School of Engineering and Technology, University of Washington - Tacoma L4.83

83

QUESTIONS

October 10, 2023 TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2020]
 School of Engineering and Technology, University of Washington - Tacoma L4.84

84