Putting it all together – Final Review

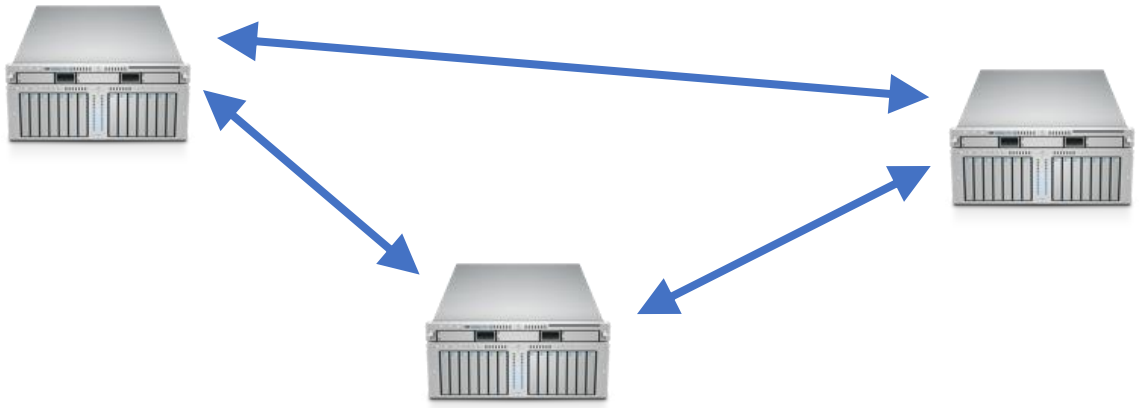
CS 4740: Cloud Computing

Fall 2024

Yue Cheng

Back to Lec 1

Cloud systems: What?



- Multiple cooperating computers – distributed systems
  - Connected by a network
  - Doing something together
- Storage for big websites, MapReduce, etc.
- Cloud infrastructures are distributed

Cloud systems: Why?

- Or, why not 1 computer to rule them all?
- To organize physically separate entities
- To tolerate faults and failures
- To scale up/out throughput

Goals of cloud systems

- Service with higher-level abstractions/interface
  - E.g., file system, database, key-value store, programming model, …
- High complexity
  - Scalable (scale-out)
  - Reliable (fault-tolerant)
  - Well-defined semantics (consistent)
- Do “heavy lifting” or “messy plumbing” so app developers don’t need to

(AAA) Themes

- Abstractions
  - Process of simplifying complex systems by exposing essential features and hiding irrelevant (impl) details
- Algorithms
  - Procedures / rules to coordinate tasks among distributed processes
- (Advanced) Systems
  - Platforms, frameworks, services

Themes

- **Abstractions**
- Algorithms
- Systems

Abstractions

- Remote procedure calls (RPCs)

Abstractions

- Remote procedure calls (RPCs)

- MapReduce programming abstraction

Abstractions

- Remote procedure calls (RPCs)
- MapReduce programming abstraction
- Strong consistency
  - **Linearizability**

Themes

- Abstractions
- **Algorithms**
- Systems

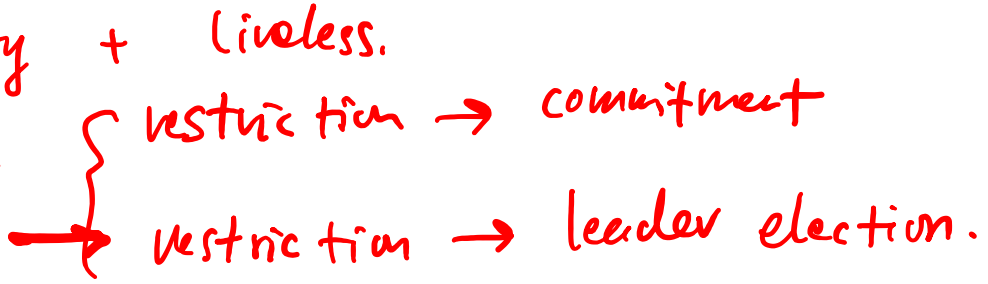
Algorithms

- Time and clocks
  - **Vector clocks**

Algorithms

- Time and clocks
  - **Vector clocks**

- Consensus algorithms
  - **Paxos**
  - **Raft**



Algorithms

- Time and clocks
  - **Vector clocks**

- Consensus algorithms
  - **Paxos**
  - **Raft**

trade offs.

- RAID and (Reed-Solomon) **Erasure Coding**

Algorithms

- Time and clocks
  - **Vector clocks**

- Consensus algorithms
  - **Paxos**
  - **Raft**

- RAID and (Reed-Solomon) **Erasure Coding**

Decentralized.

- **Consistent hashing** VN

Themes

- Abstractions
- Algorithms
- **Systems**

Systems

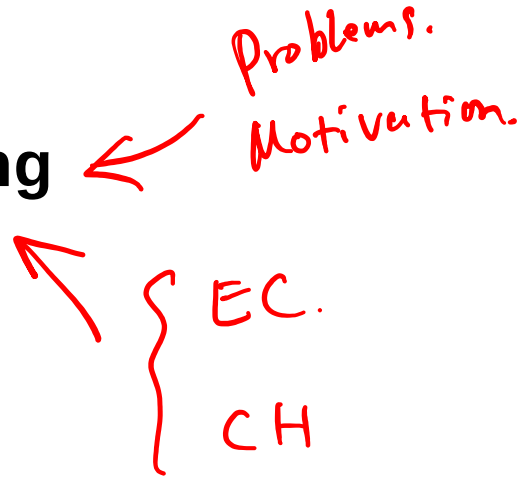
- Virtualization
  - **Virtual machine monitors (VMMs)**
  - **Containers (e.g., Docker)**

x86 { para v. / full v. / hardware (VMX)

Systems

- Virtualization
  - **Virtual machine monitors (VMMs)**
  - **Containers (e.g., Docker)**

- Serverless computing
  - **AWS Lambda**
  - **Serverless parallel computing**
  - **Serverless function storage**



Final exam

- Friday, Dec 13, 2 pm – 4 pm
  - 120 minutes
  - Open-book, open-notes (you may use class notes, papers, and lab materials)

- Covering topics from lec-1 to lec-14
  - **26%** before midterm **74%** after midterm

- Question types
  - Multi-choice and multi-answer questions
  - High-level design questions

Logistics

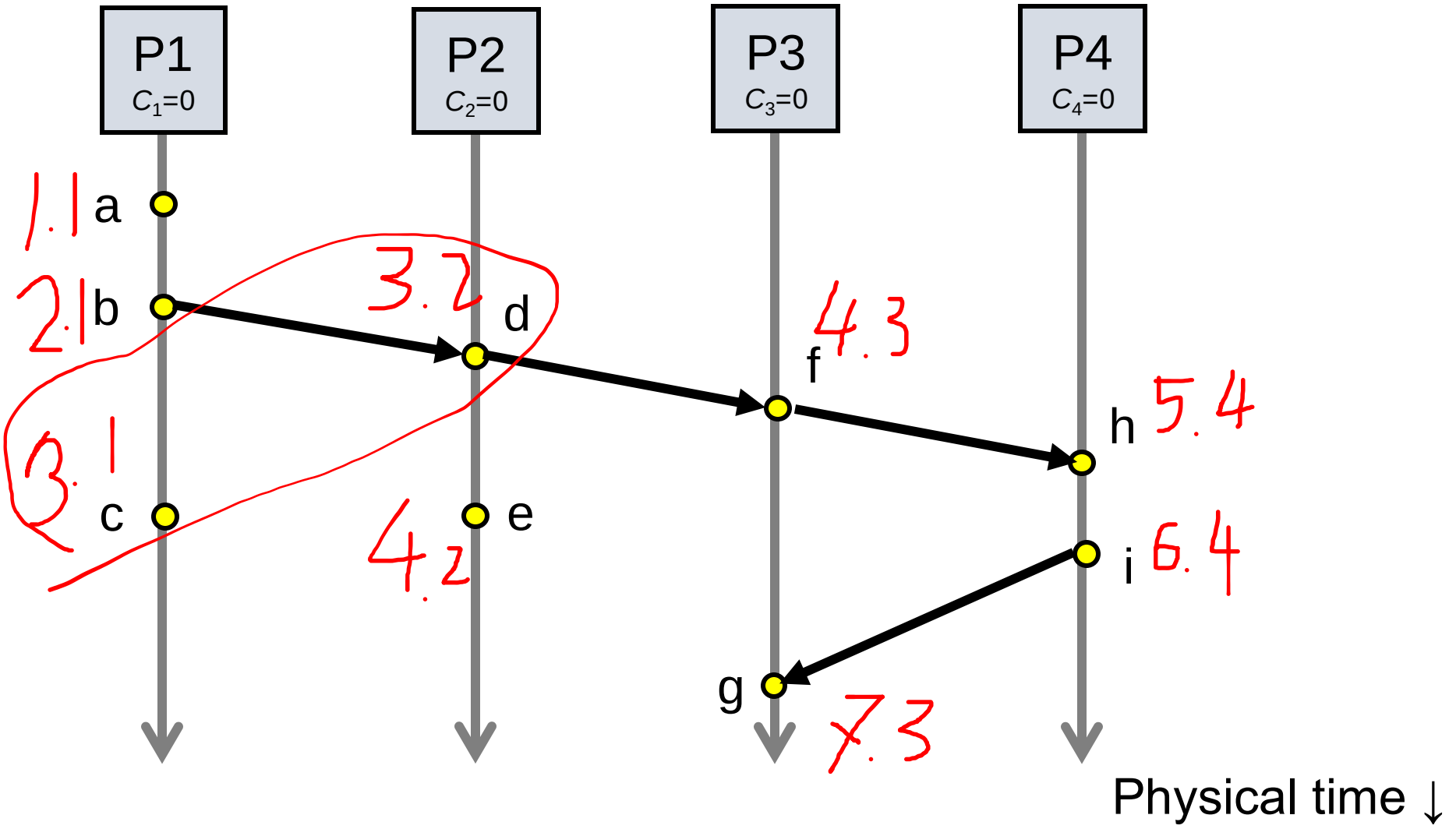
- The exam will be remote
- The exam sheet will be available on `gradescope` at 2 pm
- You should work directly on `gradescope`
- Submission closes at 4 pm

Topics

1. Vector clocks (10%)
2. Consistency and linearizability (16%)
3. Paxos (8%)
4. Raft (26%)

   Virtualization
5. ~~Cloud~~ and serverless computing (40%)

Quiz 1: Order all these events



Quiz 2: Valid sequence (causal)?

P1:	W(x)a		W(x)c		
P2:		R(x)a	W(x)b		
P3:		R(x)a		R(x)c	R(x)b
P4:		R(x)a		R(x)b	R(x)c

- **Valid** under causal consistency
- **Why?** *W(x)b* and *W(x)c* are concurrent
  - So all processes don't (need to) see them in same order
- P3 and P4 read the values 'a' and 'b' in order as potentially causally related. No 'causality' for 'c'.

Quiz 2: Valid sequence (sequential)?

P1:	W(x)a			W(x)c	
P2:		R(x)a	W(x)b		
P3:		R(x)a		R(x)c	R(x)b
P4:		R(x)a		R(x)b	R(x)c

- **Invalid** under sequential consistency
- **Why?** P3 and P4 see b and c in different order
- But fine for causal consistency
  - b and c are not causally dependent

Quiz 3: Paxos

Q: Why must a proposer receive a prepare response from a majority of servers before moving to the accept phase?

A: The majority ensures that a new proposer is guaranteed to see any value that might already have been agreed on (i.e., accepted by a majority).

Quiz 4: Virtualization

- Q1: Does one need to run a VMM in order to run a container on ~~Linux~~?
  - No

Docker:

Mac OS. Yes.

- Q2: Does one need to modify the host OS for OS-level virtualization?
  - No

Quiz 5: RAID and consistent hashing

- Q1: What's the primary tradeoff of using RAID 5 instead of RAID 1

- ○ RAID 1 has better read and write performance
- → ☑ RAID 1 requires more storage capacity for redundancy
- ○ RAID 1 is more complex to implement than RAID 5
- ○ RAID 1 is more reliable (can tolerate more disk failures) than RAID 5

Quiz 5: RAID and consistent hashing (cont.)

- Q2: If there are N nodes and K keys, what is the (approximately) average number of keys that need to be remapped when a node joins or leaves (assuming both N and K are large enough)?

  - ○ K * (N-1) / N
  - ○ K / N^2
  - ○ K / N^3
  - ☑ K / N

Midterm key is on Canvas

Don’t forget to fill out the Student Experiences of Teaching form

Thank you all! Good luck! ☺