UNIVERSITY of VIRGINIA

CS4740
CLOUD COMPUTING

Midterm Review

Prof. Chang Lou, UVA CS, Fall 2026

MIDTERM

- **Wed 10/21 2:00-3:15pm (we will start right on time)**
  - Location: this classroom
- Based upon lecture materials and projects (lab 1)
- No laptops, cellphones or other electronic devices.
- You are allowed to bring one US letter or similar size double-sided note.
- Any violation of UVA academic integrity is prohibited.

COVERAGE

- Cover topics before 2PC
  - 2PC is **not** included

- How to use today's slides:
  - Use it as a basis to develop your cheatsheet
  - Use it to self-test (mock exam with friends, or AI)

MIDTERM

University of Virginia

CS 4740 – Cloud Computing

Midterm Examination

Fall 2026 · October 21, 2026

Instructor: Chang Lou

Student name: _____

UVA Computing ID: _____

Exam instructions

- **Time: 75 minutes. Total: 90 points.** Questions 1–4 are worth **60 points**; Question 5 is worth **30 points**.
- Write legibly and show reasoning where requested. Partial credit is available for clear, relevant work.
- You may use **one US-letter-sized sheet of notes**. No other reference materials or electronic devices are permitted.
- Write your name on any additional or detached pages. If you need more room, identify the question and continue on the scratch page.
- Do not distribute the exam to others.

Point distribution

Question	Points	Score
1	12	
2	18	
3	16	
4	14	
5	30	
Total:	90	

Good luck!

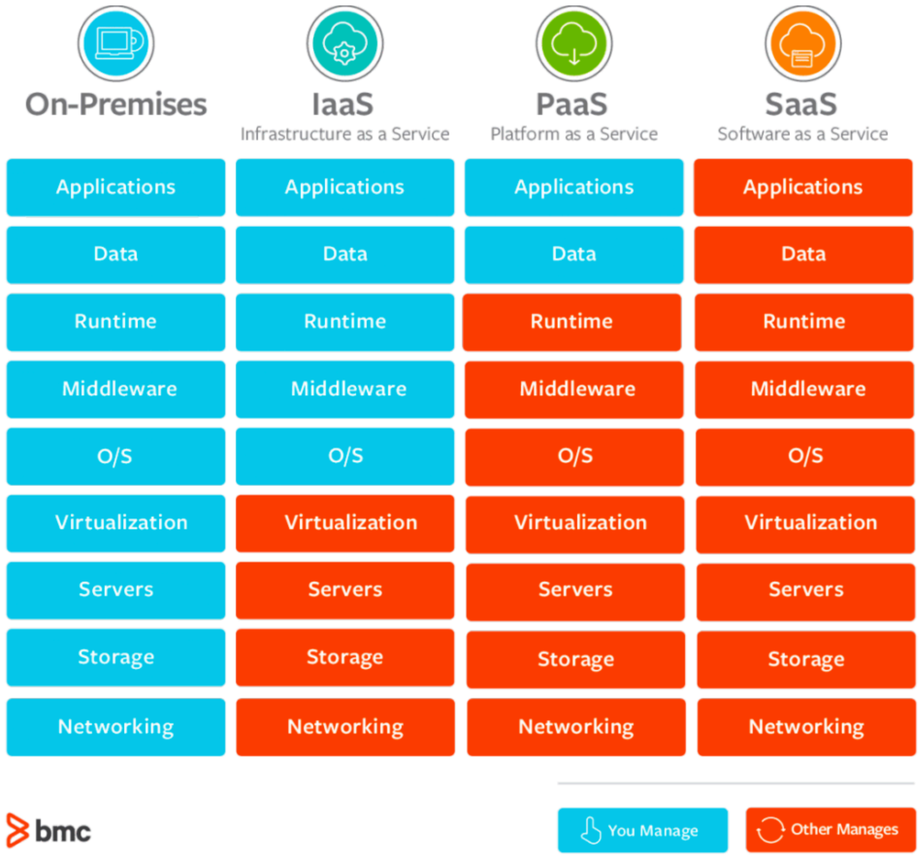
INTRODUCTION TO CLOUD

- Difference between public and private cloud
- The advantage and disadvantages of cloud computing
- What a single-site cloud (datacenter) consists of? distributed?
- Where does the idea of cloud computing come from?

INTRODUCTION TO CLOUD

- Difference between public and private cloud
- The advantage and disadvantages of cloud computing
- What a single-site cloud (datacenter) consists of? distributed?
- Where does the idea of cloud computing come from?
  - Timesharing, Grids, Clusters, P2P...
- What are the trends? (hardware, users)
- Elaborate on four new features in today's cloud
- Can you tell difference between *AAS?

INTRODUCTION TO CLOUD

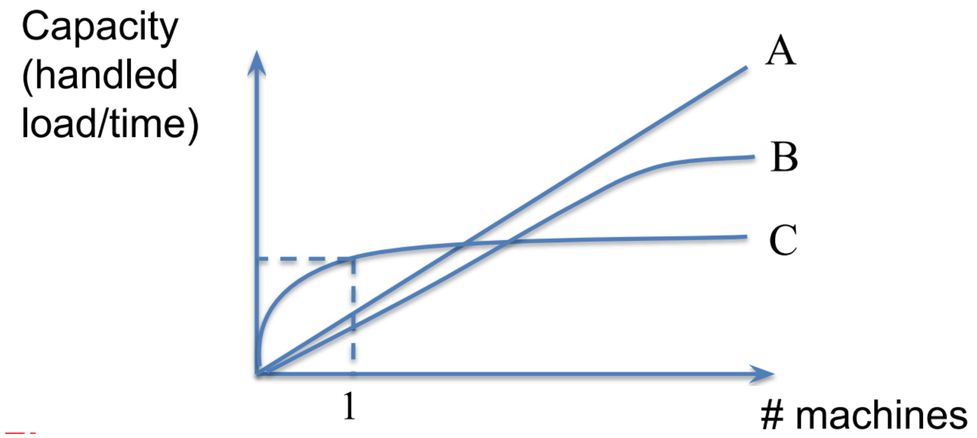


DISTRIBUTED SYSTEM FOUNDATION

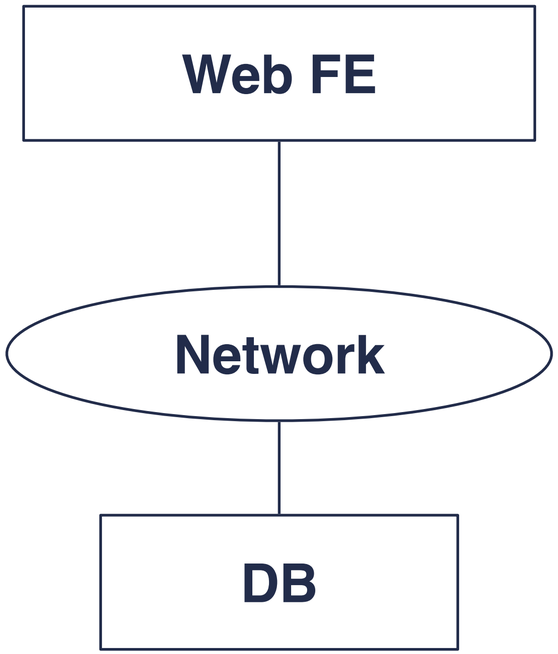
— What are basic goals and challenges of DS?

DISTRIBUTED SYSTEM FOUNDATION

- What are basic goals and challenges of DS?
  - Scalability, Fault-tolerance, Consistency (remember the Starbucks example?)
- What are "scale-out" and "scale-up"? How to understand perfect scalability?



DISTRIBUTED SYSTEM FOUNDATION



- How do we improve performance/ fault-tolerance/scalability for Web Service Architecture?
- How would system properties change after we add cache/replication/ sharding?

MAPREDUCE

- Why we need MapReduce?
- How MapReduce workflow works? Can you use examples to show?
- How to write Map() and Reduce() functions for different workloads?
  - and when to use chaining?

MAPREDUCE

– (Exercise) You are given a symmetric social network (like Facebook) where a is a friend of b implies that b is also a friend of a. The input is a dataset D (sharded) containing such pairs (a, b) – note that either a or b may be a lexicographically lower name. Pairs appear exactly once and are not repeated. Find the last names of those users who have at least 5 friends called "Mike" (with "Mike" as their first name).

MAPREDUCE

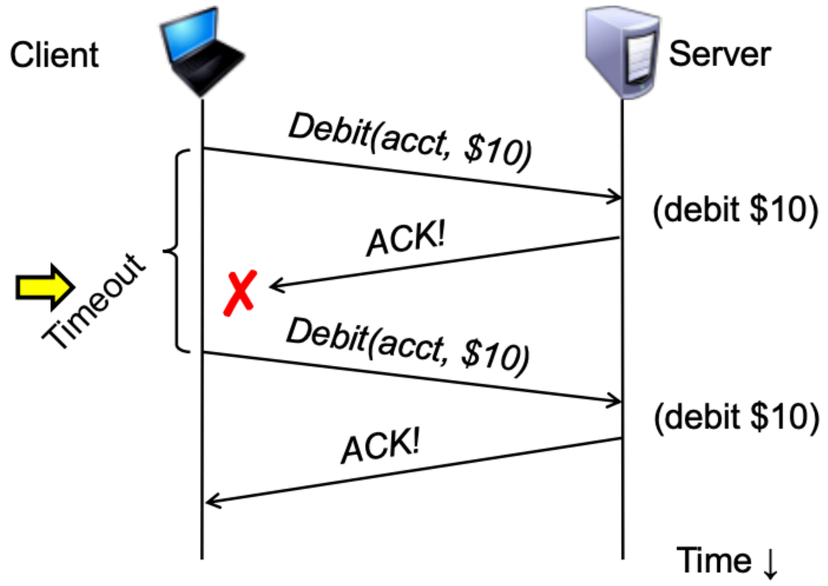
- How MapReduce deal with different design challenges?
  - How to improve performance?
  - How to deal with failures?
  - How to deal with slow nodes?

MAPREDUCE

- How MapReduce deal with different design challenges?
  - How to improve performance? => scheduling policy for data locality
  - How to deal with failures? => heartbeats + replication
  - How to deal with slow nodes? => redundant execution

RPC

- What are the downsides of socket programming?
- What is the key idea behind this abstraction?
- What are the steps of a RPC workflow?
- Comparison between LPC and RPC
- What are different semantics of RPC?
  - show failure scenarios for each semantics
  - how to implement them?



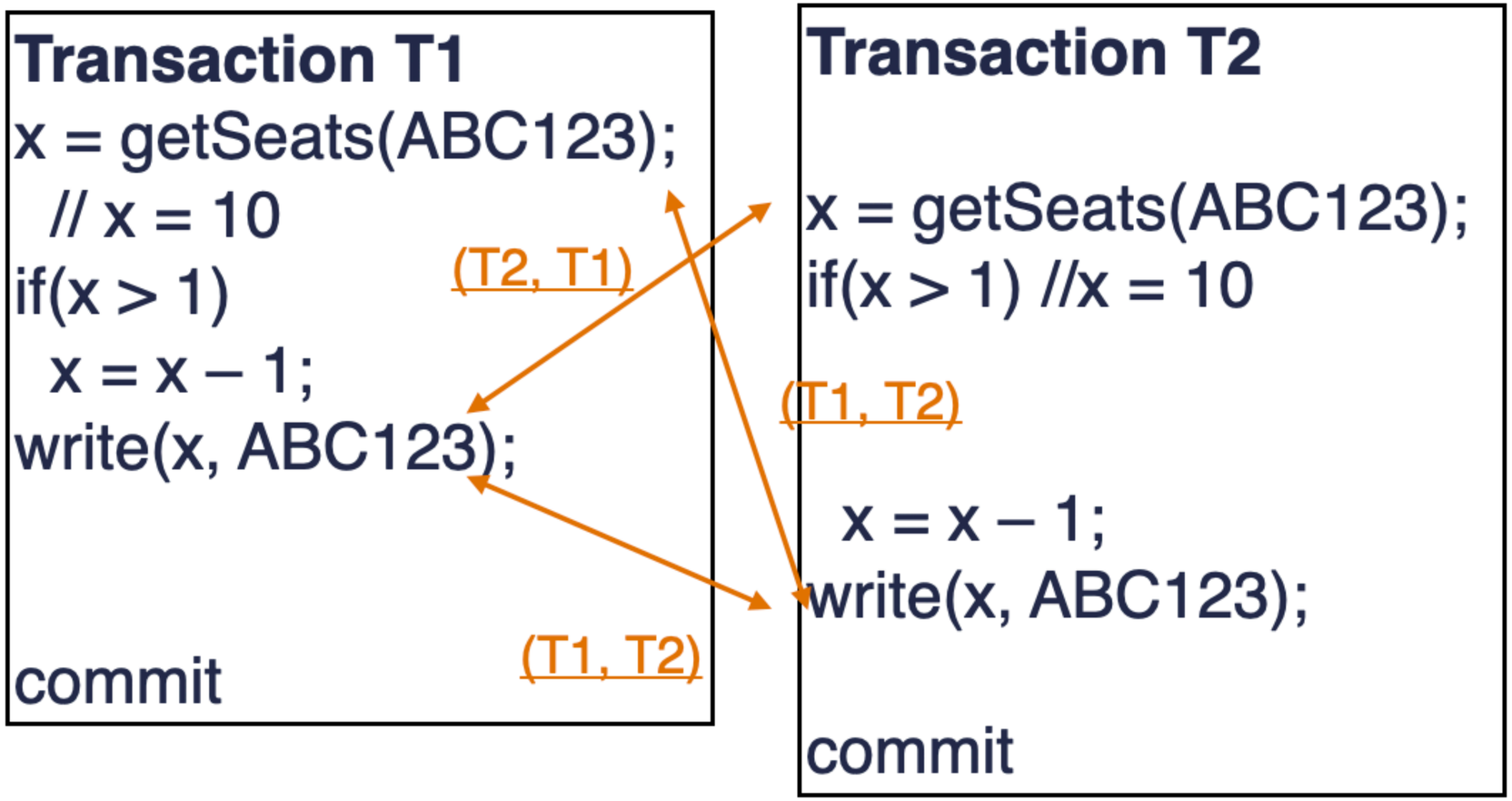
TRANSACTION

- What system challenges do transactions solve?
- What semantics do transactions provide?
- Given an example of client codes, can you give an example of possible failure/concurrency problem?
- How does WAL work?
  - why different logging schemes don't work?
  - show how WAL helps to recover crashes or handle aborts

TRANSACTION (CONTD.)

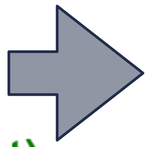
- How to check serial equivalence?
- How does 2PL work?
  - can you show how 2PL ensures serial equivalence by blocking other transactions?
  - how to resolve deadlock?
- How does timestamp ordering work?

TRANSACTION (CONTD.)



TRANSACTION (CONTD.)

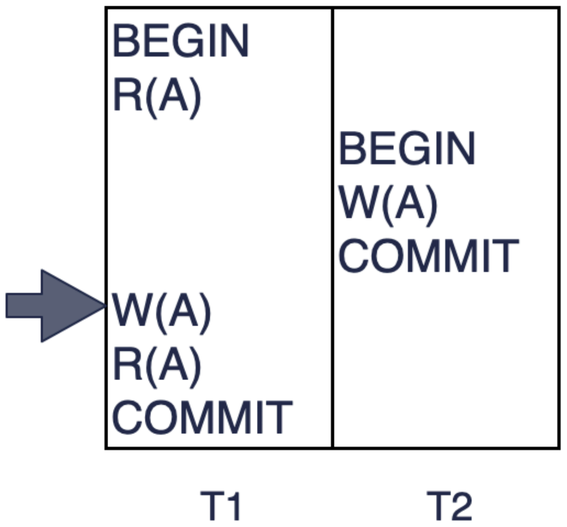
```
TRANSFER(src, dst, x)
00  txID = Begin()                  ← lock(src)
01  src_bal = Read(txID, src)
02  if (src_bal > x):
03    src_bal -= x
04    Write(txID, src_bal, src)
05    dst_bal = Read(txID, dst)     ← lock(dst)
06    dst_bal += x
07    Write(txID, dst_bal, dst)
09    return Commit(txID)           ← unlock(src,dst)
10 Abort(txID)                      ← unlock(src,dst)
11 return FALSE
```



?

TRANSACTION (CONTD.)

SCHEDULE



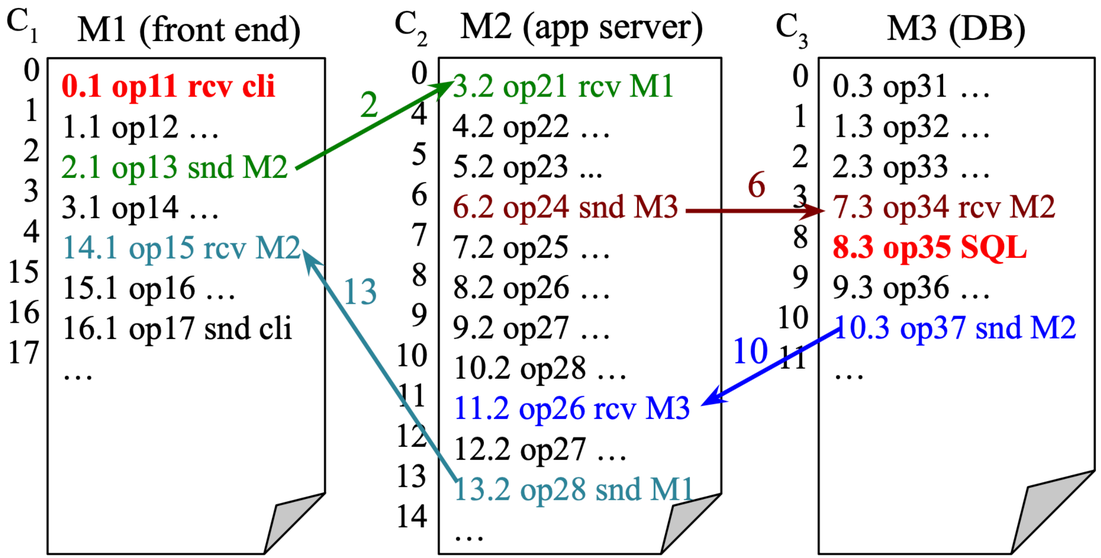
DATABASE

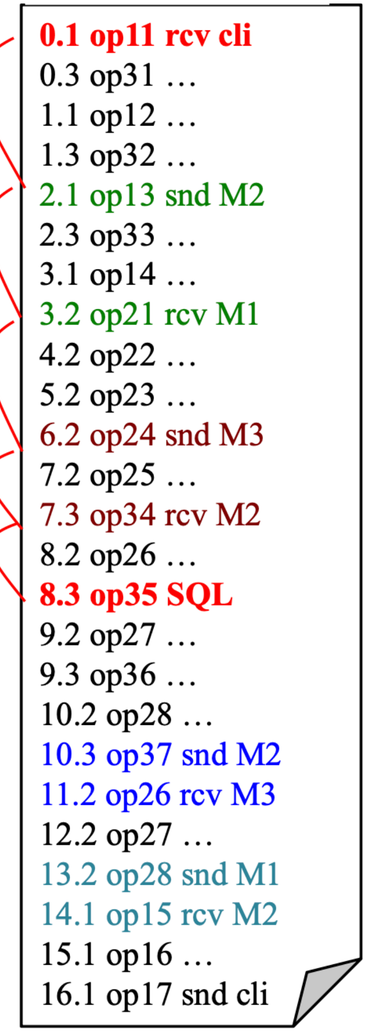
OBJECT	R-TS	W-TS
A	1	2
B	0	0

Violation:
TS(T1)<W-TS(A)

TIME AND SYNCHRONIZATION

- Why Physical clocks are hard to synchronize?
  - what protocols to use?
- How does Lamport Clock Protocol works?
  - how to create the total order?





AGREEMENT

- What is the difference between atomic commitment and consensus?
- Explain the insight of "Green cup, Red cup" game solution.
- Explain what challenge does two generals problem demonstrate.
- Explain CAP theorem.

LAB 1 PROJECT

- Review the overall architecture of your MapReduce implementation.
  - Understand how the coordinator manages task assignment, task states, and transitions between Map and Reduce phases.
  - Think about concurrency issues in the coordinator. What could go wrong when multiple workers interact with it simultaneously?
  - How does your implementation handle worker failures, timeouts, and task reassignment?
  - Reflect on your Lab 1 debugging experience: What bugs did you encounter, how did you diagnose them, and how did you fix them?

TAKEAWAYS

- Next class: **midterm exam**