UNIVERSITY of VIRGINIA

CS4740
CLOUD COMPUTING

Midterm Review

Prof. Chang Lou, UVA CS, Fall 2025

MIDTERM

- **Fri 10/24 12:00-12:50pm (we will start right on time)**
- 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

MIDTERM

CS4740, Fall 2025– Midterm Exam
Cloud Computing

Instructor: Chang Lou

2025/10/24

Name: _____
UVA Computing ID: _____

Please write your name on this cover sheet (and other sheets if you tear out any).

This exam contains 8 pages (including this cover page) and 5 questions. You have 50 minutes to finish all questions. Some questions may present challenges. Should you encounter difficulties, prioritize moving on to subsequent questions. Try to give detailed and precise answers. In instances of uncertainty, attempt to articulate any thoughts or partial answers you may have.

You are allowed to bring one piece of US letter paper (or in equivalent size) as your notes. You cannot use any additional materials during the exam. Please do not re-distribute this exam paper to others.

Good luck!

Distribution of Marks

Question	Points	Score
1	12	
2	12	
3	12	
4	12	
5	16	
Total:	64	

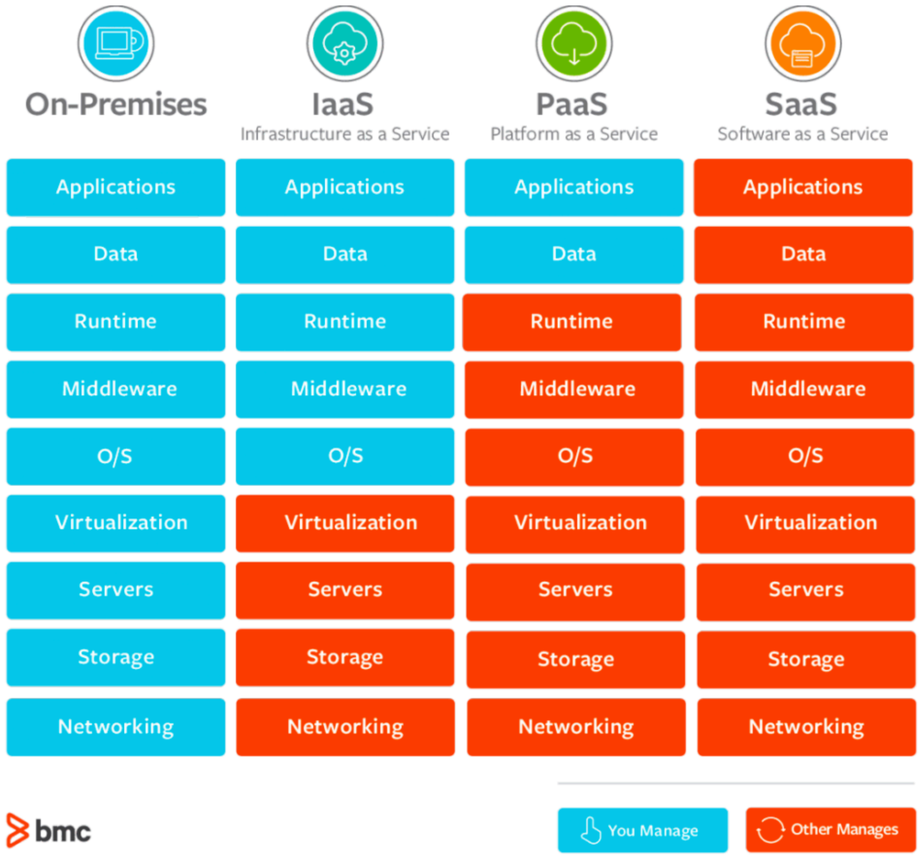
INTRODUCTION TO CLOUD

- 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

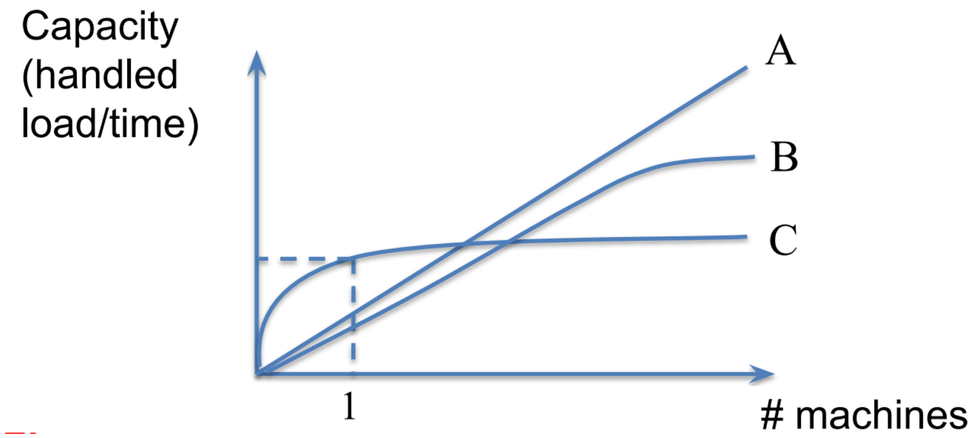


DISTRIBUTED SYSTEM FOUNDATION

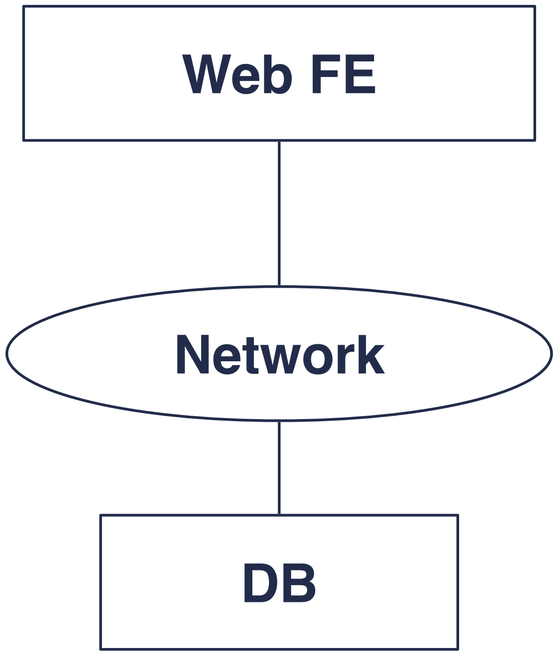
— What are basic goals and challenges of DS?

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

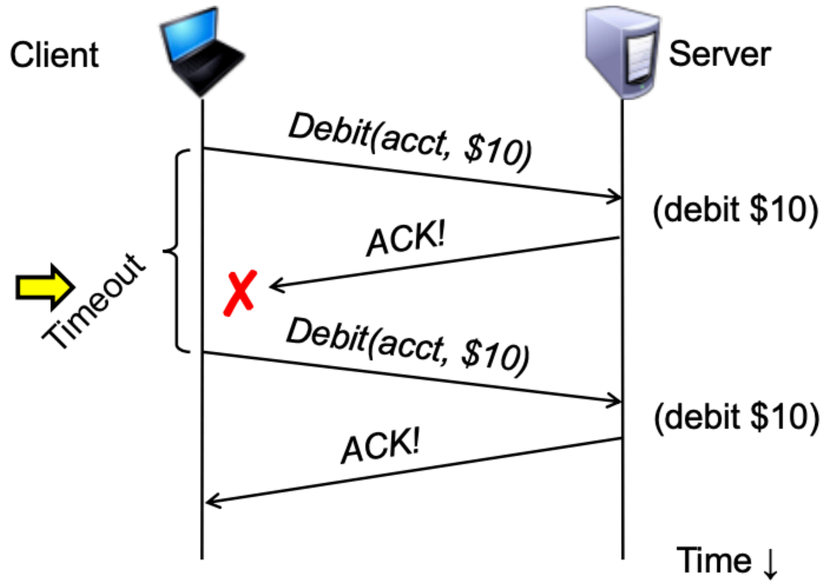
- How MapReduce deal with different design challenges?
  - How to improve performance?
  - How to deal with failures?
  - How to deal with slow nodes?

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?



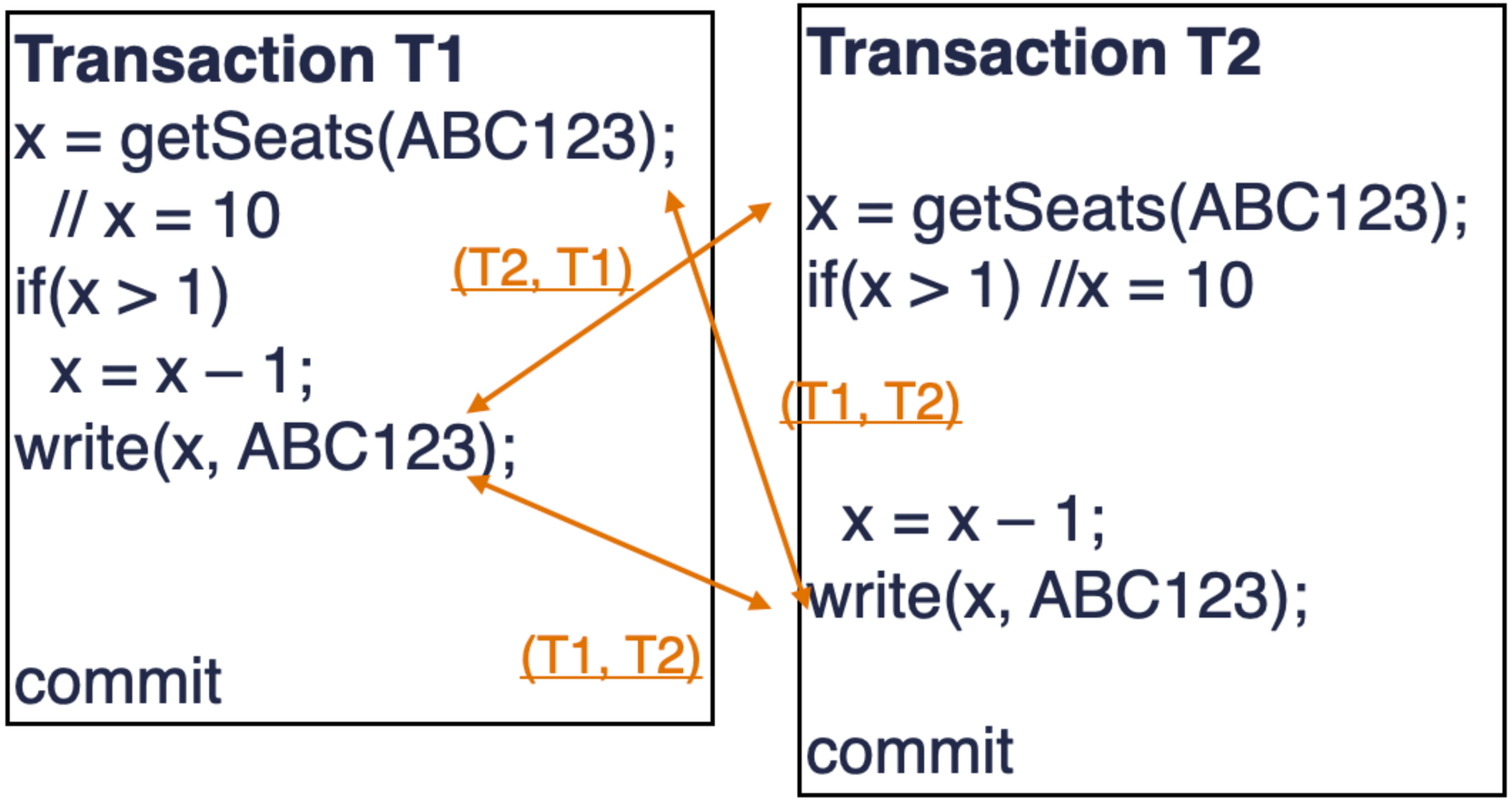
TRANSACTION

- 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.)

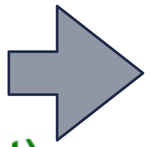
- How to check serial equivalence?
- How does 2PL work?
  - can you show how 2PL ensures serial equivalence by blocking other transactions?
  - how to resolve deadlock?
- How does timestamp ordering work?

TRANSACTION (CONTD.)



TRANSACTION (CONTD.)

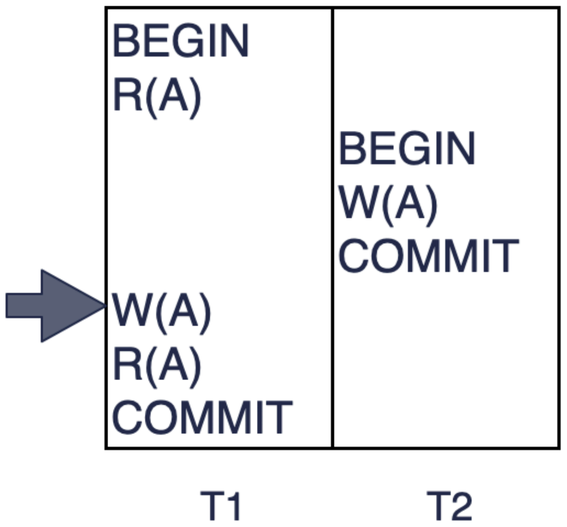
```
TRANSFER(src, dst, x)
00  txID = Begin()                  ← lock(src)
01  src_bal = Read(txID, src)
02  if (src_bal > x):
03    src_bal -= x
04    Write(txID, src_bal, src)
05    dst_bal = Read(txID, dst)     ← lock(dst)
06    dst_bal += x
07    Write(txID, dst_bal, dst)
09    return Commit(txID)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



?

TRANSACTION (CONTD.)

SCHEDULE



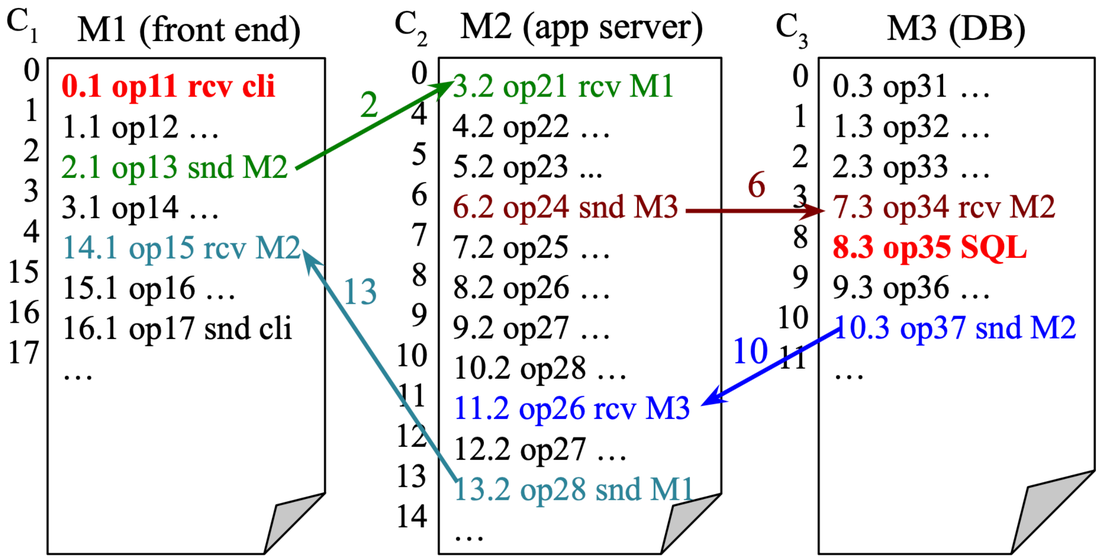
DATABASE

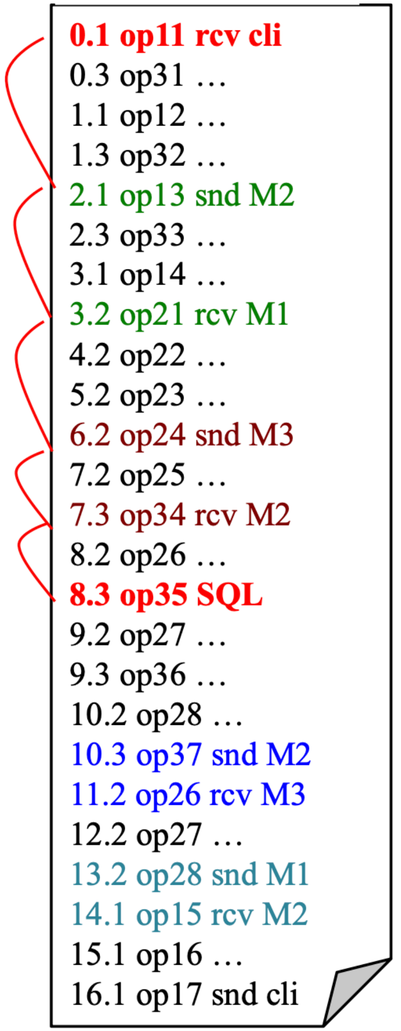
OBJECT	R-TS	W-TS
A	1	2
B	0	0

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?





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.

TAKEAWAYS

- Next class: **midterm exam**