

*Welcome back!*

# CS 744: SINAN

Shivaram Venkataraman

Spring 2025

# ADMINISTRIVIA

Grading updates → tonight (gradescope)  
→ tomorrow check ins

Midterm 2, April 24<sup>th</sup>  
-- Check Piazza

Poster session: May 1st  
— More details soon ] → by Monday 1 pm  
CS lobby  
Template  
Where to print etc.

ML for improving system

X

# MICROSERVICES

software design & development

are not stateful → scale them up/down

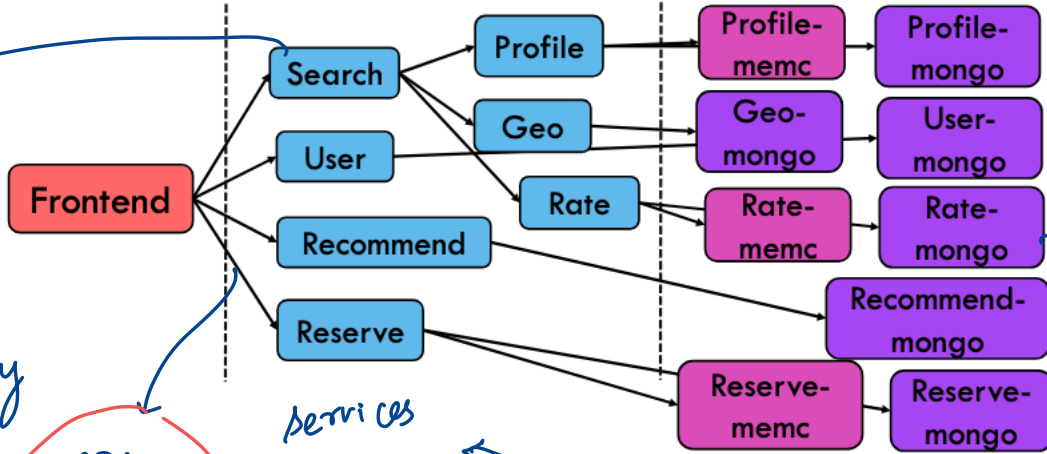
Frontend

Business logic

Caching & DB

impl of Search

↳ deployment could be on 1 or many



social networking

mongo DB

RPC calls

services  
lots of boxes!

# RESOURCE MANAGEMENT CHALLENGES

decide how many CPUs each microservice gets

## Dependencies among "tiers"

↳ fixed amount of resources  
end to end latency - depends on all tiers

## "Delayed" queuing

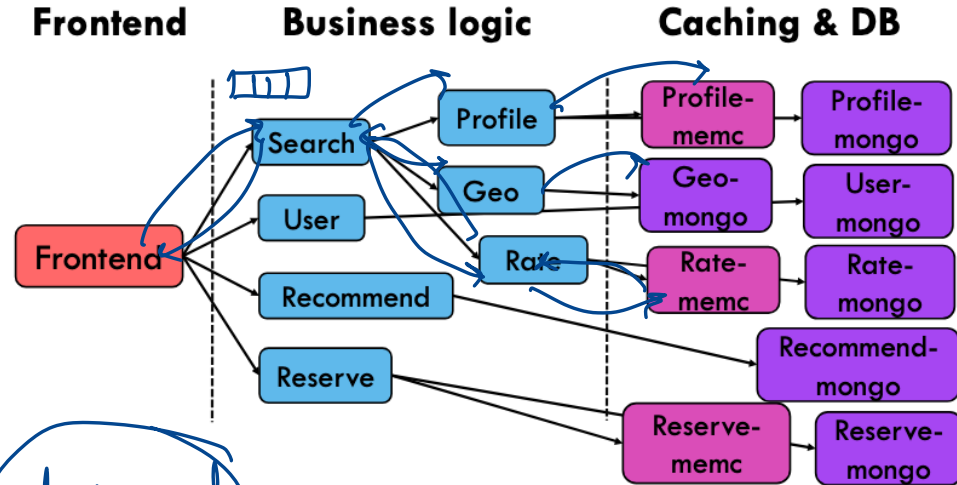
Scale up "Search" service at each tier + processing time  
↳ includes queuing delay

## Resource allocation space

↳ large number of possible allocations → latency

end to end latency

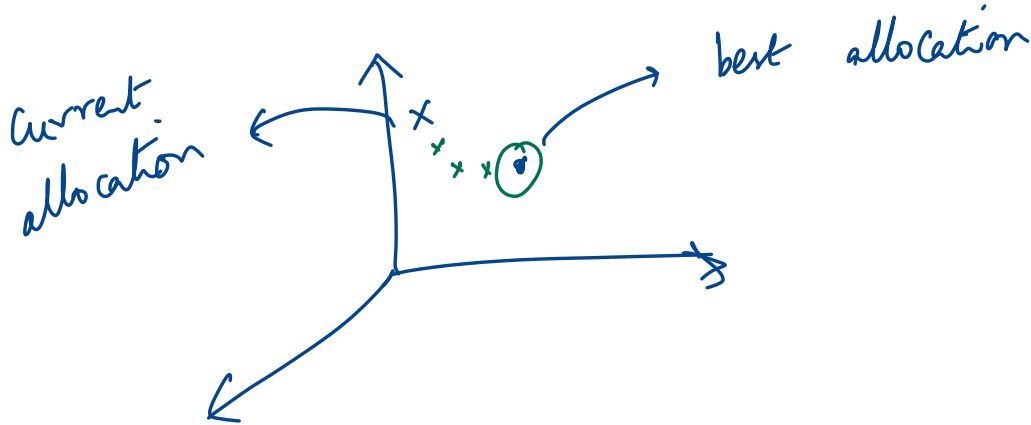
→ traverse through microservice graph



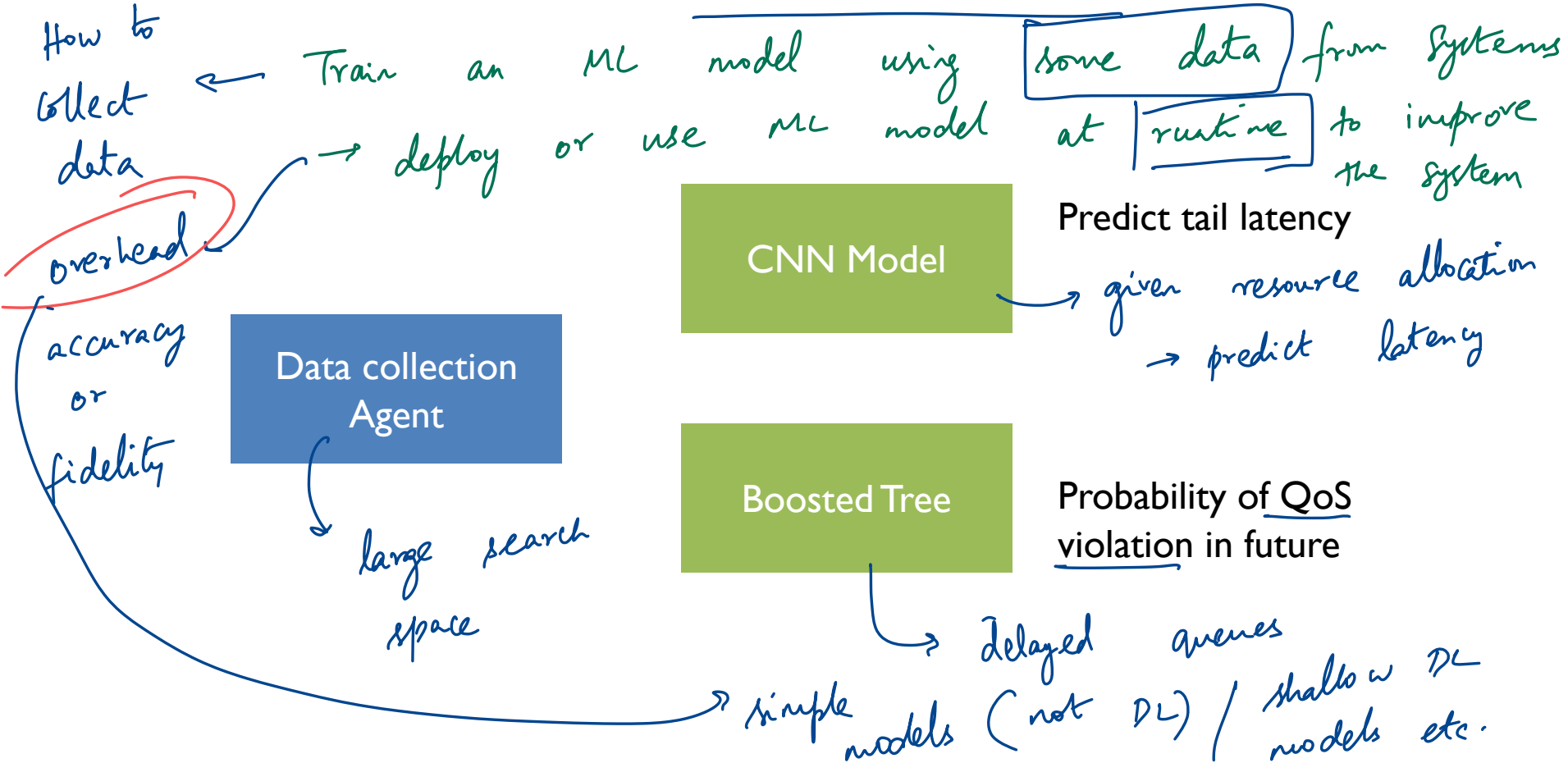
# SINAN APPROACH

Narrow down the search space of possible allocations.

Given an allocation use ML to predict performance



# APPLYING ML TO SYSTEMS



# MODELING CHALLENGES

Goal: Predict latency given a resource allocation

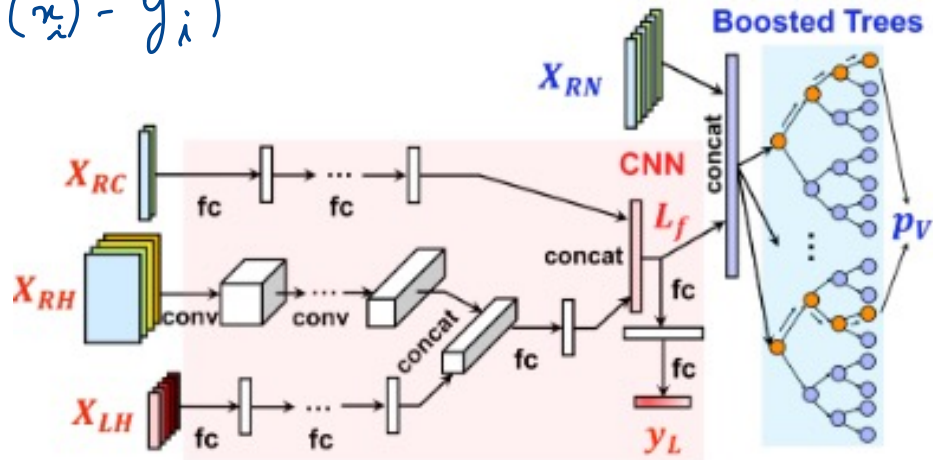
Latency at what time point?

QoS violation: Classification vs. regression

real valued number  
regression - style problems

$$\min \sum_i (f(x_i) - y_i)^2$$

binary value  
classification  
1 or 0



# TWO STAGE MODEL: CNNs

given an allocation  
 ↳ predict instantaneous latency

[ ← last 5 mins → ]

3D tensor ("image") resource allocation over time

Dimensions include: microservice tier, resource (CPU, mem), time

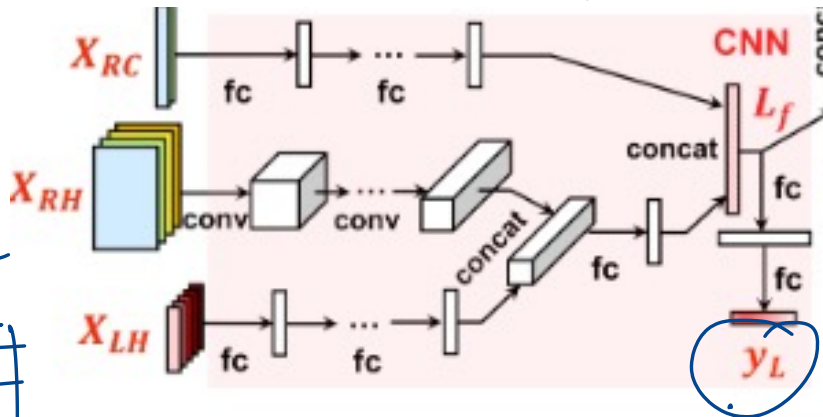
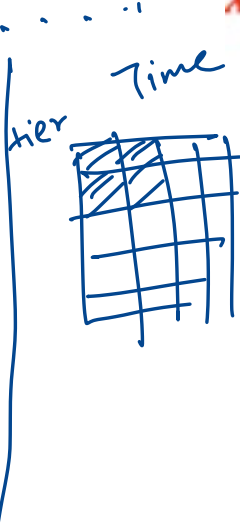
[ 1, 5, 3 ], [ 2, 4, 5 ]

End-to-end latency distribution within time window ( $X_{LH}$ )

[ 25.5, 24.3, 17 ] ...

Resource configuration for next step ( $X_{RC}$ )

[ 4, 2, 3 ]



$$\mathcal{L}(X, \hat{y}, W) = \sum_i^n (\hat{y}_i - f_W(x_i))^2$$

Training loss function

# TWO STAGE MODEL: VIOLATION PREDICTION

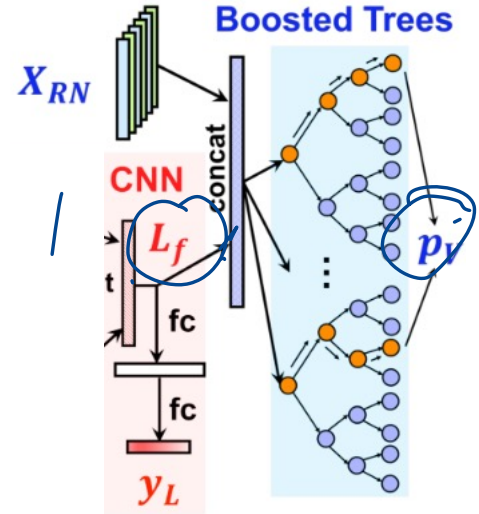
Boosted Trees *XGBoost*

Classification problem: QoS violation or not

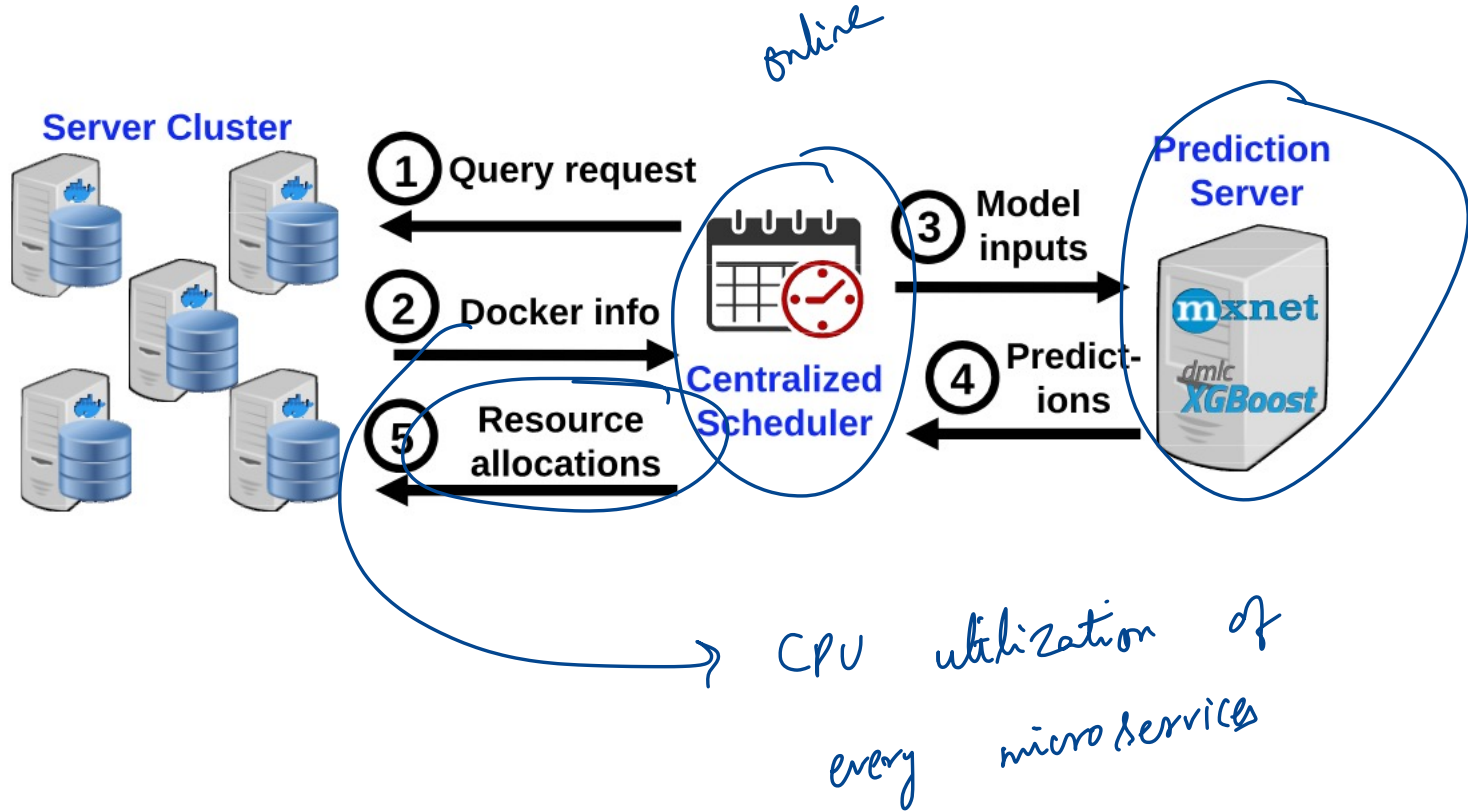
Input: last layer of CNN (smaller in dim than inputs)  
Resource allocation

Output: probability of violation

$p_v > 50\%$  then  
don't do this etc.



# SYSTEM ARCHITECTURE



# HOW TO COLLECT TRAINING DATA

Needs to cover a spectrum of application behaviors

cover entire space of allocations

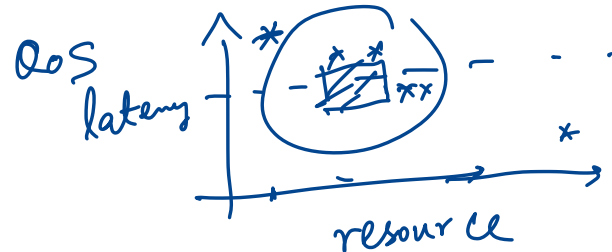
" " Boundary of the resource allocation space. Why?

min resources use to meet QoS

Multi-arm bandit based approach

- treat each tier as an arm.
- each time select one tier to explore (max information gain)

algorithm for choosing data that give you most "information"



# TRAINING DATA ACTION SPACE

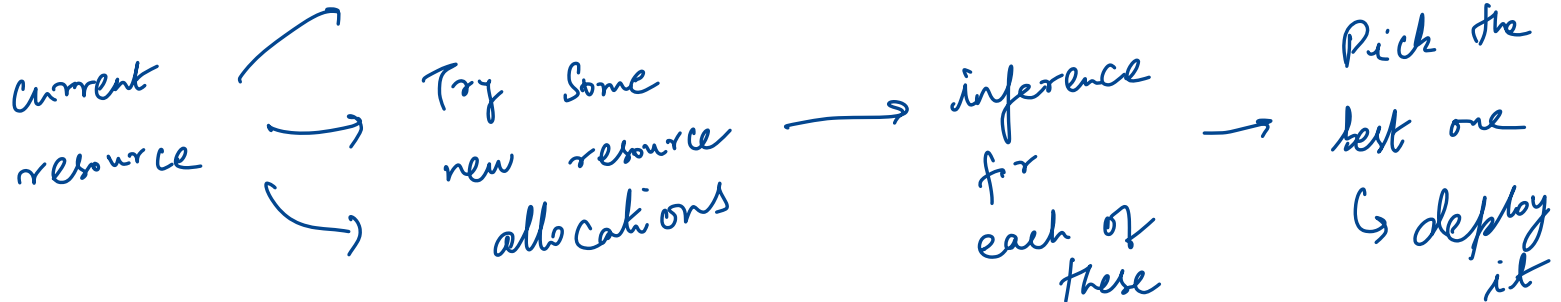
Only allowed a predefined set of operations

reducing or increasing the CPU allocation by 0.2 up to 1.0 CPU

increasing or reducing the total CPU allocation by 10% or 30

upper limit on CPU utilization at each tier

Training data should see some samples which are above QoS



# ONLINE SCHEDULER

Exclude operations whose predicted tail latency is high

Use the predicted violation probability to filter out risky operations

Category	Actions
Scale Down	Reduce CPU limit of 1 tier
Scale Down Batch	Reduce CPU limit of $k$ least utilized tiers, ( $1 < k \leq N$ )
Hold	Keep current resource allocation
Scale Up	Increase CPU limit of 1 tier
Scale Up All	Increase CPU limit of all tiers
Scale Up Victim	Increase CPU limit of recent victim tiers, that are scaled down in previous $t$ cycles

undo previous action?

# SUMMARY

Applying ML to Systems → new direction

Sinan: Apply ML to microservice scheduling

Open questions

- Model arch – 2 level with CNN, Boosted trees

- Training data – multi-arm bandit based approach

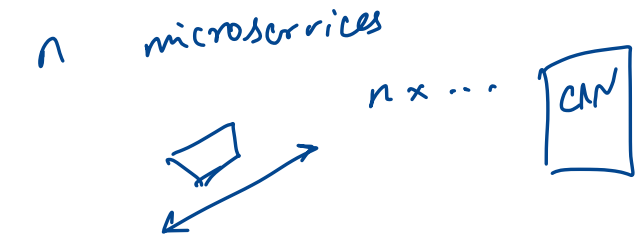
- Model inference – online scheduler with smaller search space



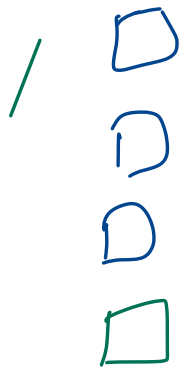
# DISCUSSION

<https://forms.gle/xSyIBkPWGedDKYAZ6>

If you add a new service to your graph or make changes to complexity of a service, how will you make sure Sinan handles it?



$n+1$  microservices  
web rendering



③ Train a new model!!

① Projection matrix

$n+1$  to  $n$

"fine tune" CNN

②

Pre-group so you know where it will slot in.

Assume new one similar to this group

How should the scheduler handle cases where the ML model returns incorrect predictions?

- ① Have a second model  
or  
human? → Picks a config /  
overrides
- ② Roll back to state that  
worked in the past
- ③ Just over provision

# NEXT STEPS

Next week schedule

Tue: Luminix (Inference scheduler)

Thu: Midterm 2