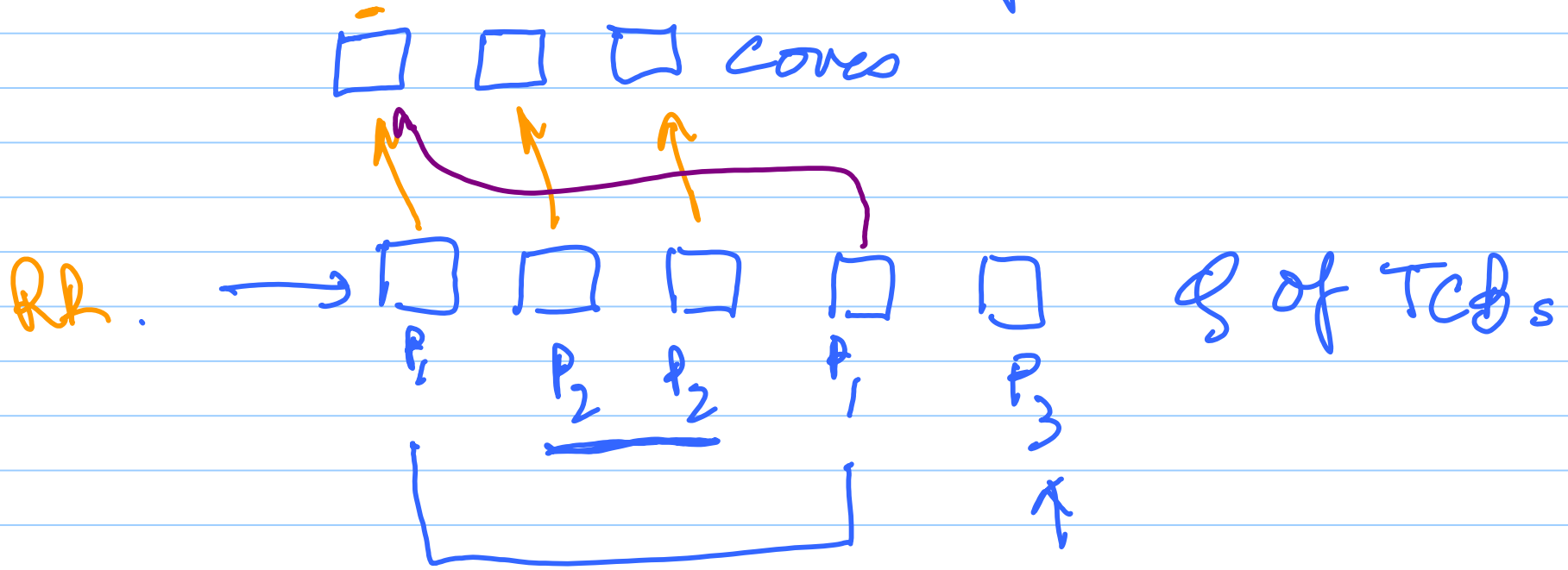
CSE 531: Distributed and Multiprocessor Operating Systems

Class: 15 Date: 10/14

Fall 2020

Note Title

Multiprocessor Scheduling



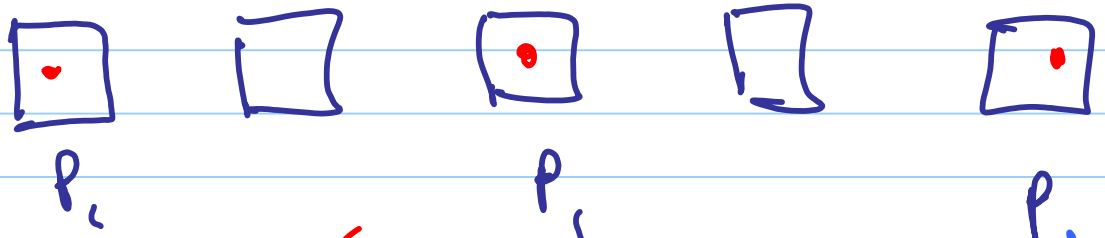
Coscheduling

↓

All (or almost all) threads of the same process should be scheduled at the same time

Multicore

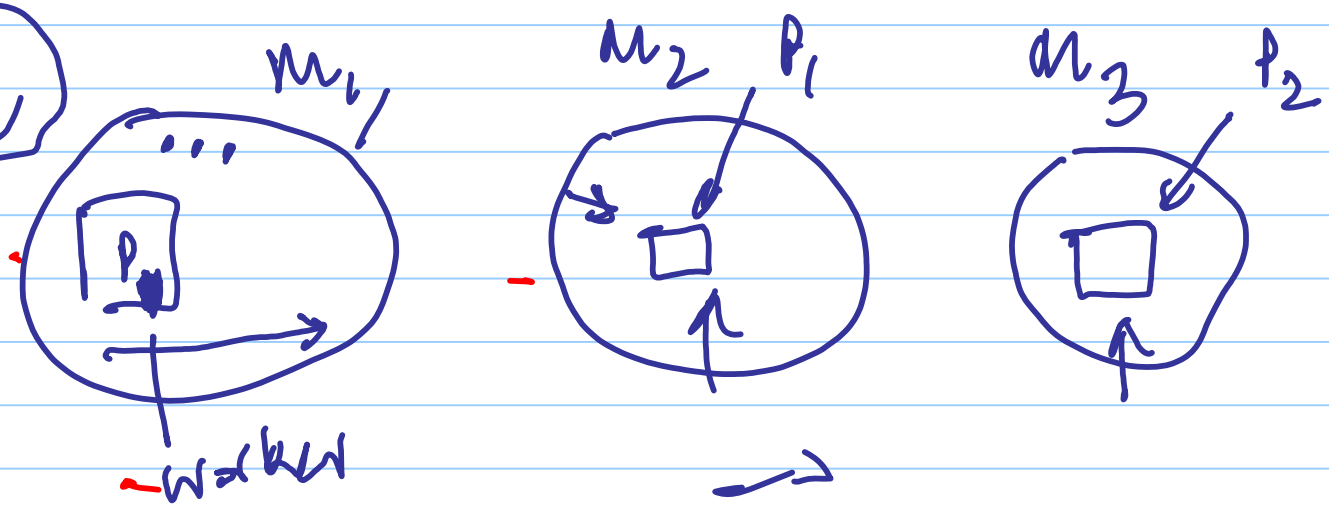
TCB



RR ✓ ✓ ✓ ✓

Cosched ✓ ✓

Cluster



- Gang Scheduling
  → ALL threads @ same time

- Matrix Coscheduling
- Process Aware RR

- Dynamic Partition —
- Hand off sched —
- Affinity scheduling → T_1 was on core 1 later T_1 should be on core 1

→ Dist operating system

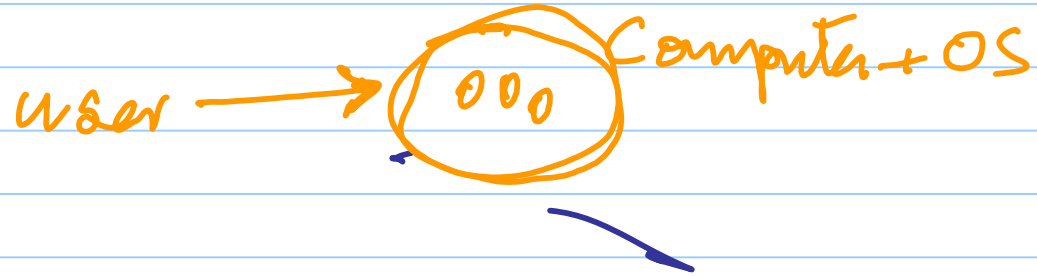
- Cloud
- Cluster
- Grid
- Fog
- workstations
- servers
- RPC / API

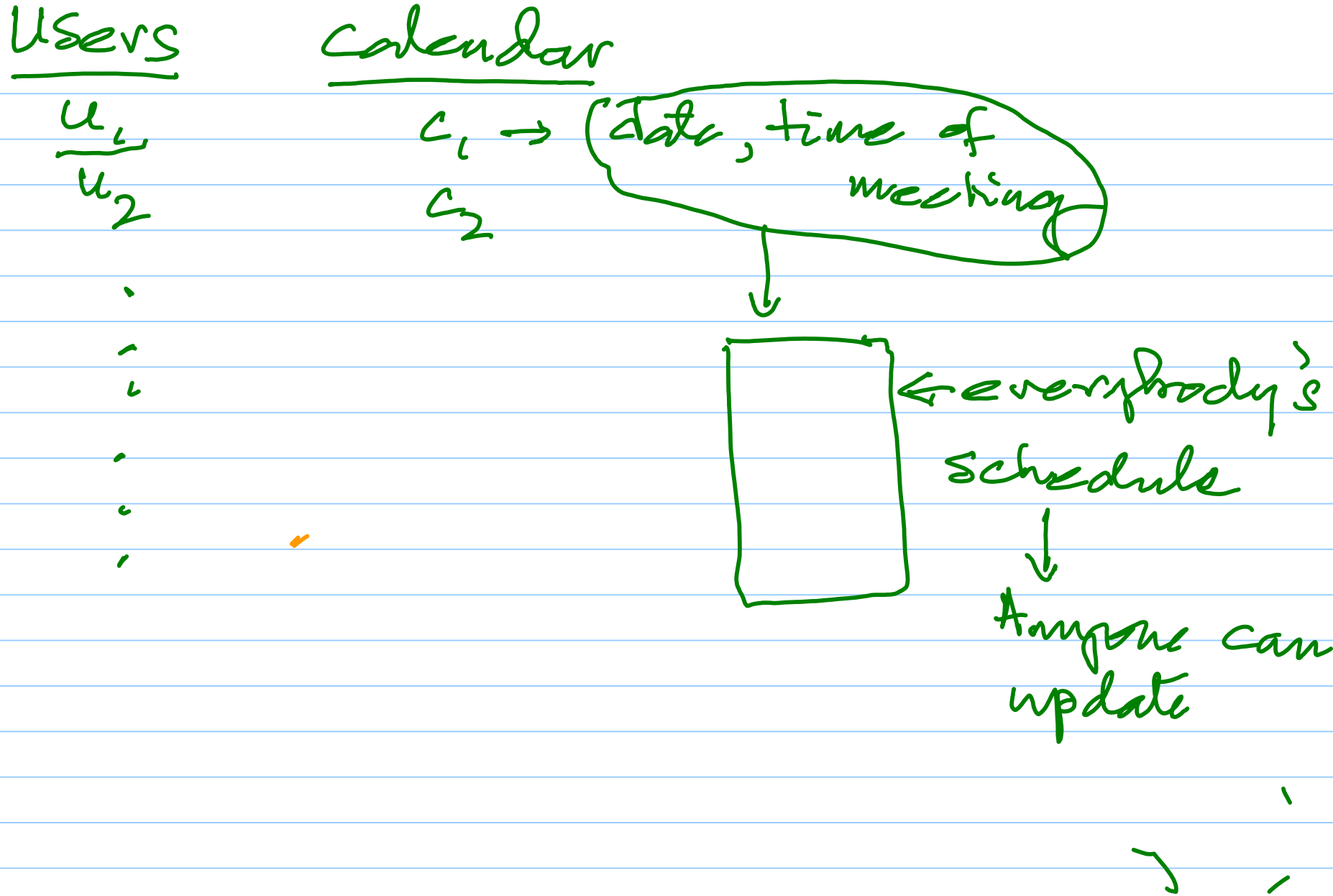
Dist OS

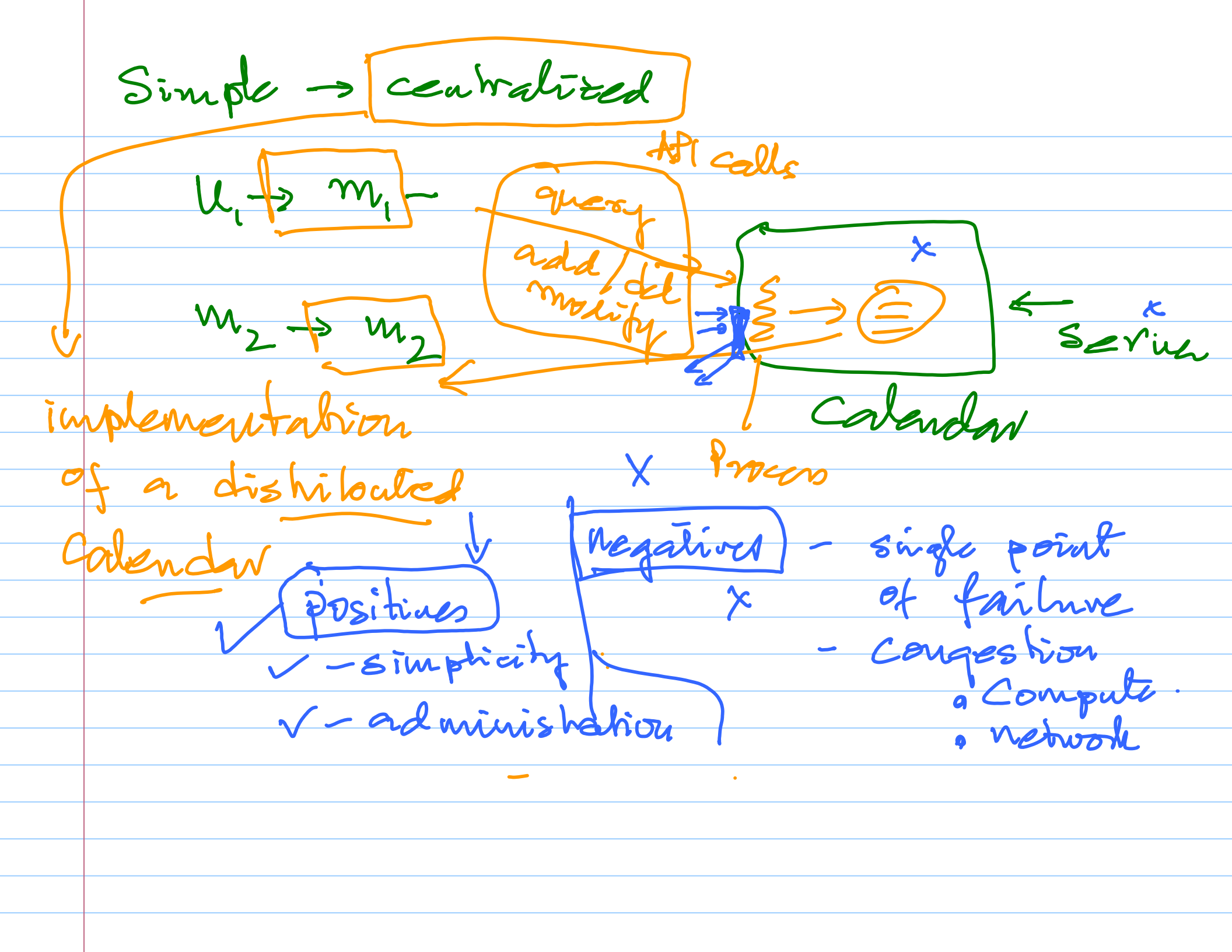
Tannenbaum

→ A distributed operating system is a collection of autonomous computing elements that appear to its users as a single coherent (or integrated) system.

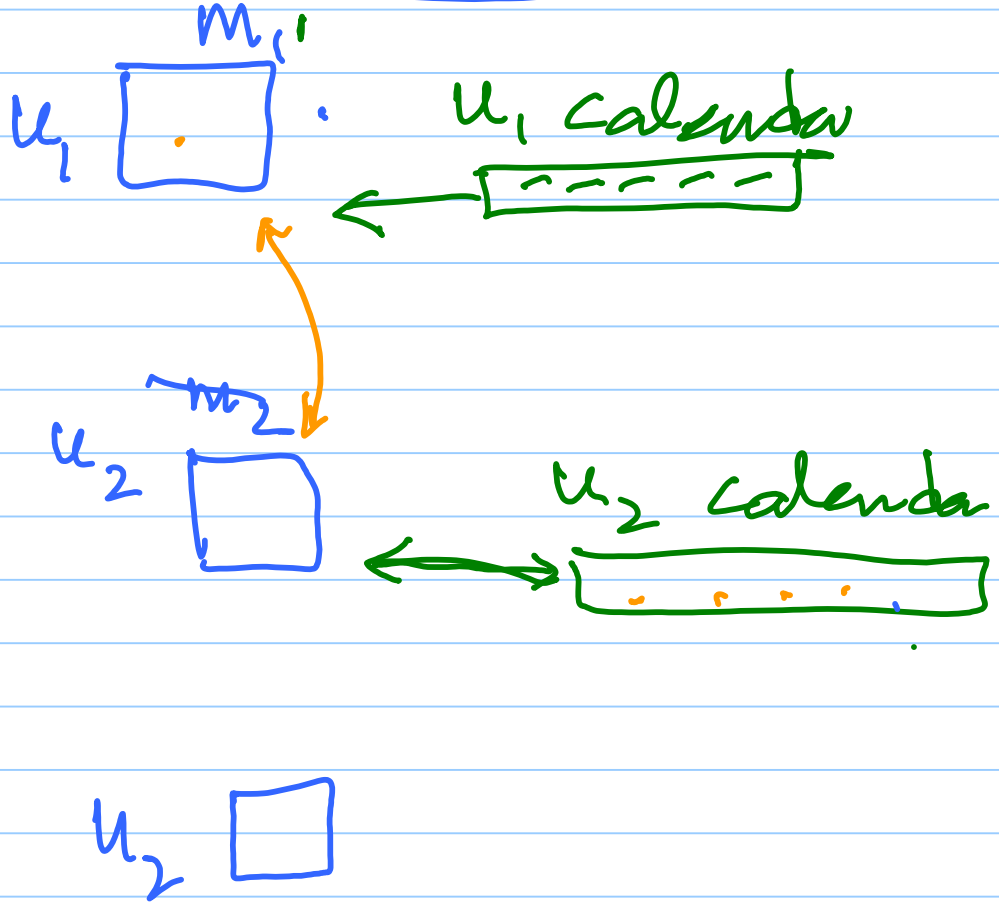
Cooperation + opposites




Users
Calendar
u_1
u_2
c_1 → Date, time of meeting
c_2
everybody's schedule
Anyone can update


Simple → centralized
API calls
query
add / del
modify
x
Server
Calendar
implementation
of a distributed
Calendar
Process
Positives
Negatives
- single point of failure
- congestion
• Compute
• network
✓ - simplicity
✓ - administration
-

fully distributed (but per user)



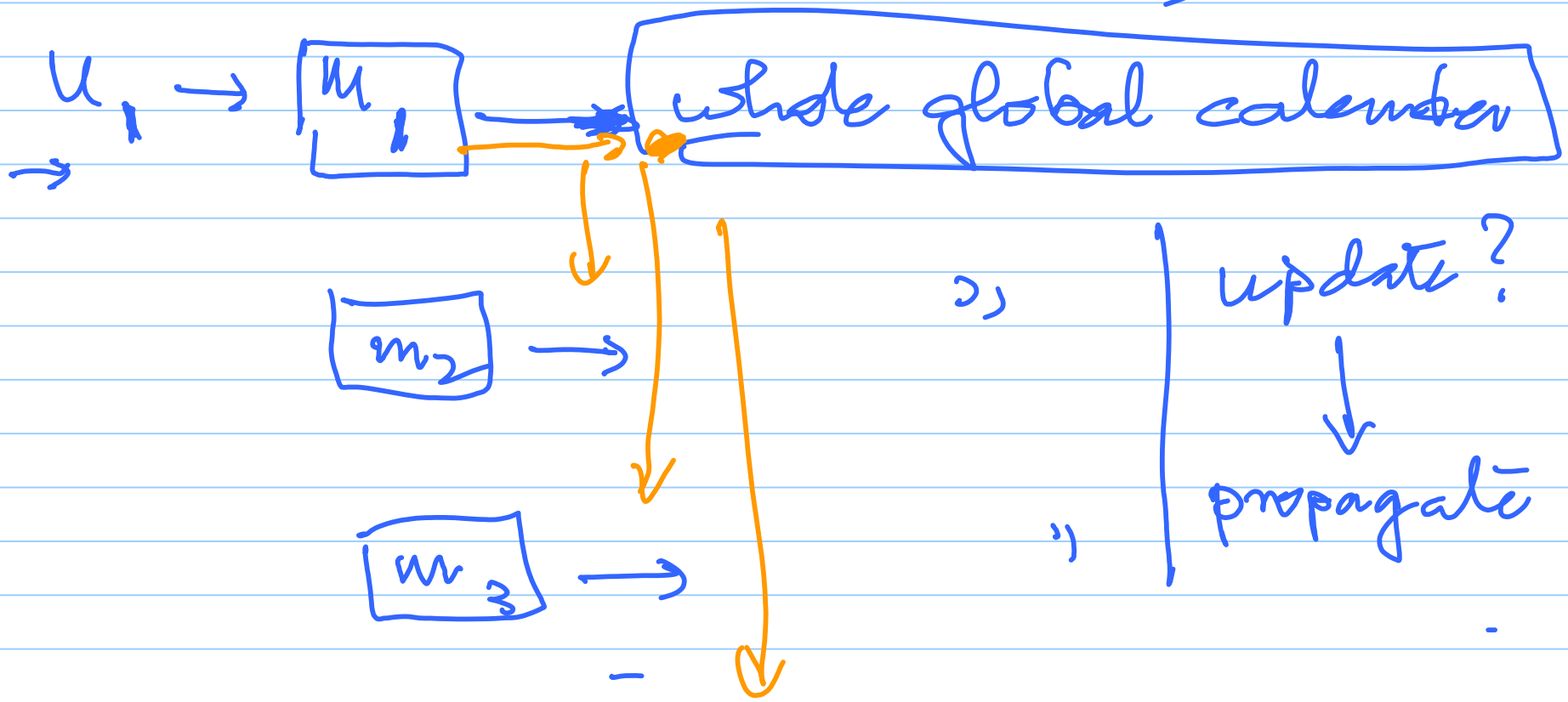
No server

- no single pt of failure
- no congestion

- global queries
- partial

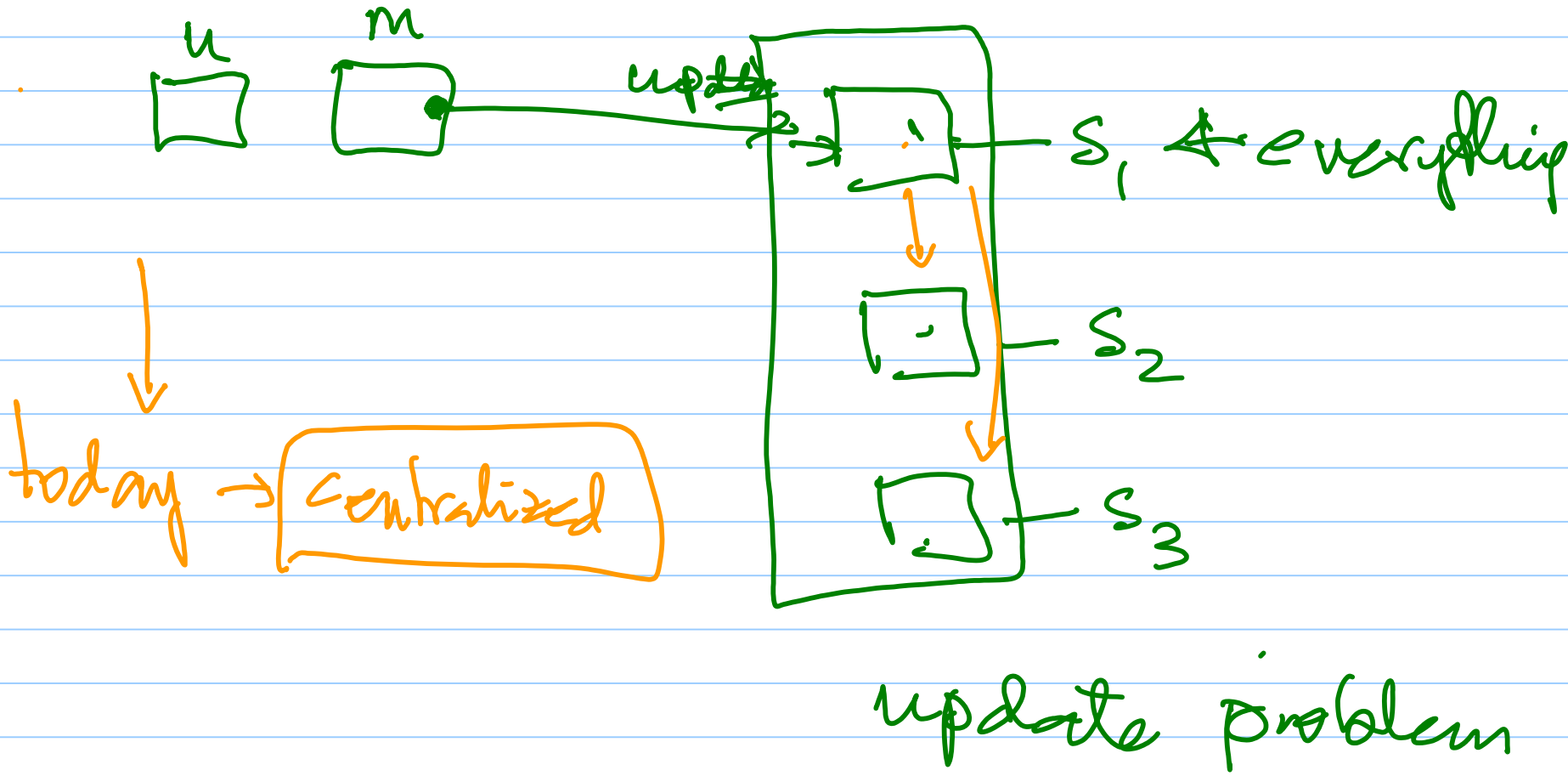
fully distributed - replicated

(everyone has everything)



Partitioned

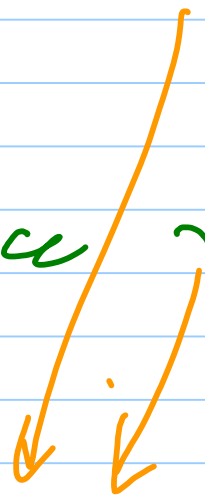
Replicated Servers



update problem

Goals

- reduce server computation
  - client side scripting
- reduce network traffic
  - client caching
- reduce overhead (?)
  - network, compute] load + latency



Network - speed

antenna

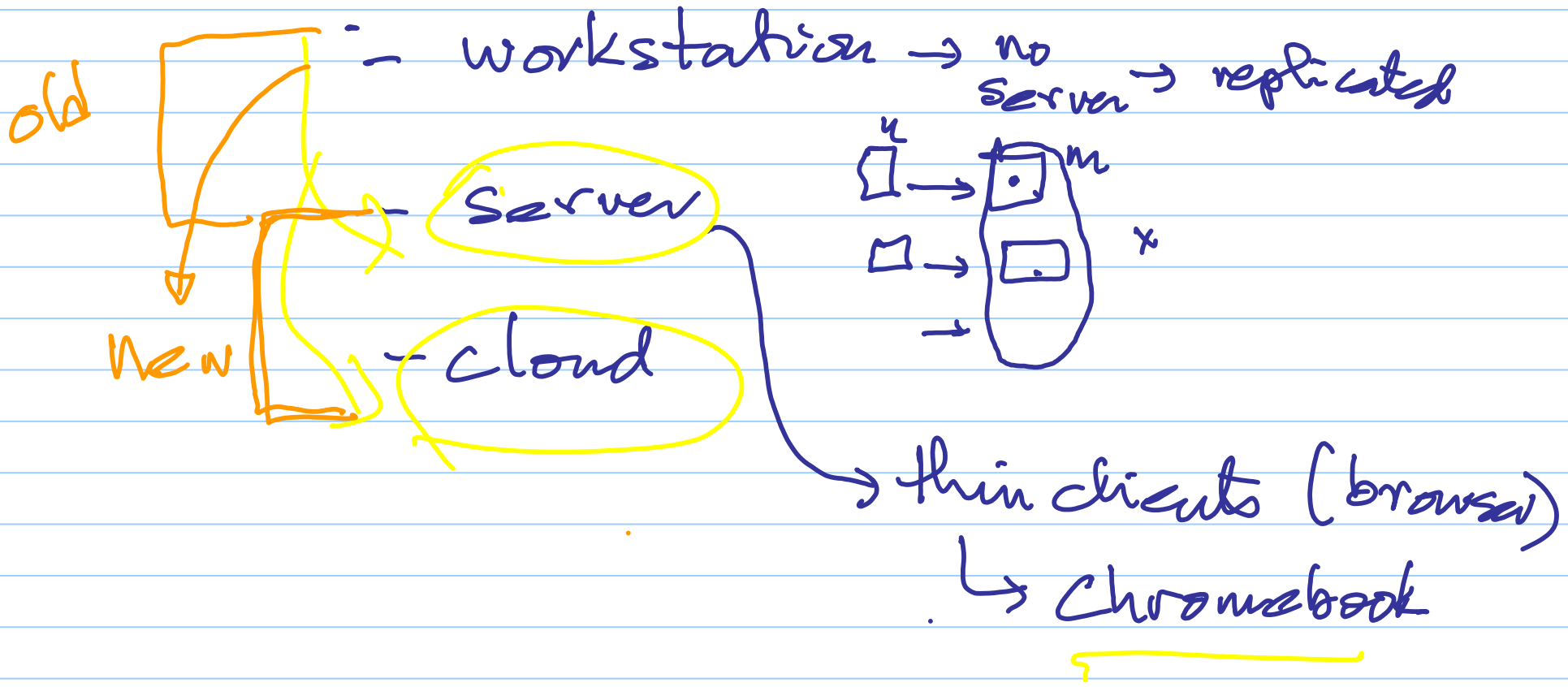
bytes / sec

fat pipe

ping time

distance

Dist OS model



Properties

- cooperation
- autonomy
- Efficiency
- Availability
- Reliability
- Scalability