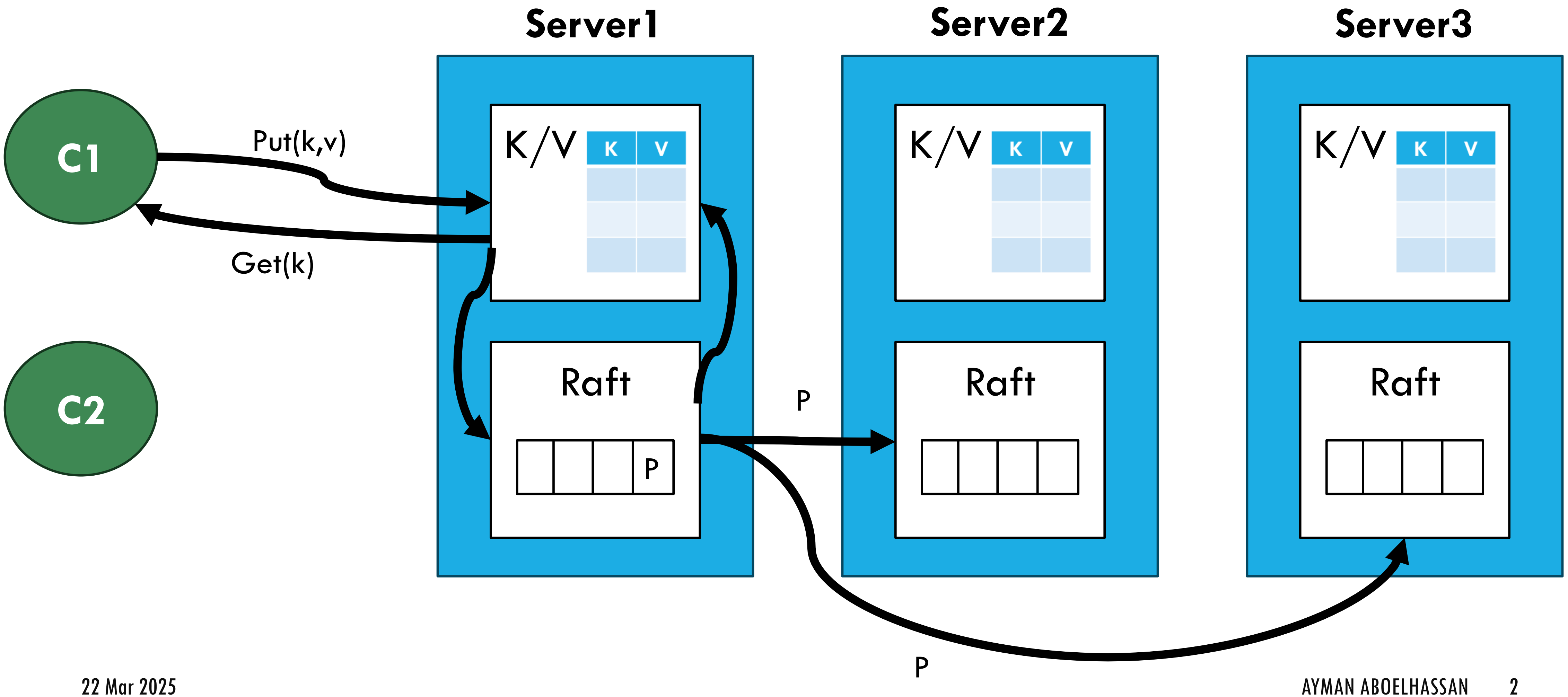
CMP636 Distributed Systems
Raft + Key/Value Server Lab

Ayman AboElHassan, PhD
Assistant Professor
ayman.abo.elmaaty@eng.cu.edu.eg

Basic Idea


Server1
Server2
Server3
C1
C2
Put(k,v)
Get(k)
K/V
K
V
K/V
K
V
K/V
K
V
Raft
P
Raft
Raft
P
P

Basic Idea

1. Client send request to KV server
2. KV server send operation to Raft
3. Raft reaches value consensus ➔ Total Broadcast to followers
4. Followers reply to leader
5. Once quorum Ack received ➔ Commit KV change

Requirement

Change last lab key/value server

1. Create 3 servers
2. Each server maintains its own in-memory map of key/value pairs (Local KV Map)
3. Clients sends read/write/append to 1 of the 3 servers
4. Servers reach consensus on KV operation through Raft

Installing Raft Library

We will install a Raft-driven library “PySyncObj”

- https://github.com/bakwc/PySyncObj

Python Implementation

Server

1. Create SyncObj
2. Create empty Replicated Dict
3. Create 1 RPC server stub
4. Wait for client requests
5. Perform client's request and print the operation input/result

Client

1. Create 5 client threads
2. Each thread
    1. Select a random request out of:
        - **Get** value of K1
        - **Put** value ClientNum in K1
        - **Append** value ClientNum to K1
    2. Send a request to Server
    3. Wait for response

Python Implementation

gRPC proto

- Service1: **Get**
- Message: **key**
- Behavior:
  return map[key]
- Response: **value**

- Service2: **Put**
- Message: **key, value**
- Behavior:
  map[key] = value
- Response: **success**

- Service3: **Append**
- Message: **key, args**
- Behavior:
  old_value = map[key]
  map[key] += args
- Response: **old_value**

Python Implementation

What happens when we increase the number of server threads?

Have some free time?

Implement your own Raft

Thank you