Distributed Computing

What is distributed computing?

- So far, we've discussed programs that can run on a single computer.
- Most interesting computer applications involve many computers interacting with each other.
- **Distributed computing** is using a bunch of computers that communicate and coordinate with each other to accomplish a common goal.

Big Data

- "Big Data" processing is a common example of distributed computing
- **Big Data** refers to *extremely* large datasets that can be analyzed to provide useful information.
- Examples of Big Data:
  - Facebook's daily logs: 60 Terabytes (60,000 Gigabytes)
  - Google's web index: 10+ Petabytes (10,000,000 Gigabytes)
- These datasets take a long time to read!
  - Reading 1 Terabyte of data from disk takes ~3 hours

Google Query Logs

How do computers communicate?

- **IP (Internet Protocol)** provides an addressing system for computers
  - Every computer has a unique name and IP address

- **DNS (Domain Name System)** is the yellowbook of the internet and provides a mapping of domain names to IP addresses.
  - Example:
    - Domain name: 'www.cs61a.org'
    - IP Address: '104.199.121.146'

How does the internet work?

Domain Name System

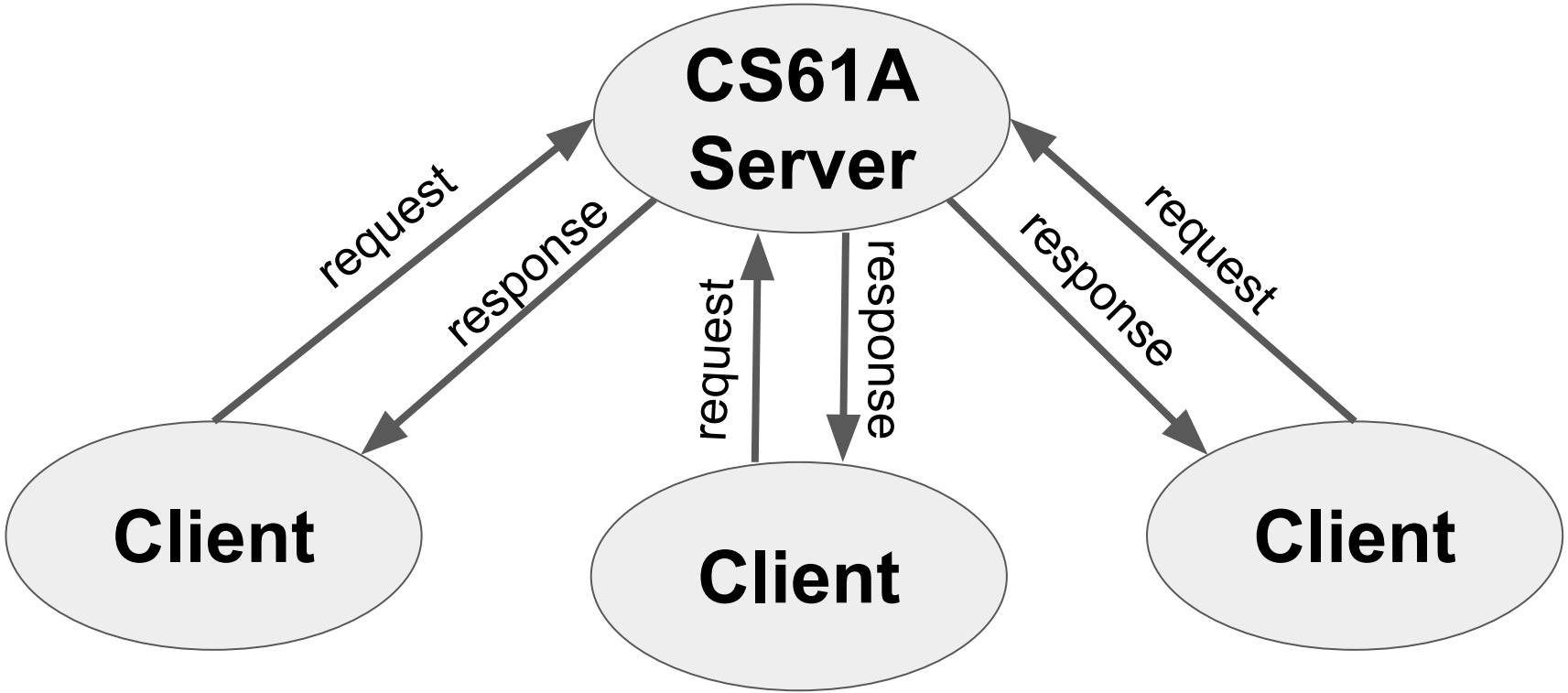
Given a domain name, I can tell you the IP address!

DNS in action! (demo)

Client / Server Architecture

- Client / Server Architecture is a way to disperse a service from a central source.
- A server provides a service, and multiple clients communicate with the server to consume that service.
- A **server**'s role is to respond to requests from clients.
- A **client's** role is to issue requests and make use of the server's response in order to perform their task of interest.

Client / Server Architecture



Client / Server in action! (demo)

Client / Server in Python! (demo)

Drawbacks of Client / Server Architecture

- Single Point of Failure
  - If the server goes down, then the entire system is broken

- Clients don't contribute any processing power
  - Computing resources become scarce if there are too many clients.
  - Clients increase demand on the system without contributing any computing resources.

Peer-to-Peer Systems

- **Peer-to-Peer (P2P)** describes distributed systems where the labor is divided among all the computers in the system.
- All computers that participate also contribute some processing power and memory.
- As a P2P system increases in size, its capacity and computational resources increases too!
- An identifying characteristic of a P2P system is division of labor among all participants.
    - This means the peers need to be able to communicate with each other reliably.

Tor (The "Dark Web")

- Goal: Keep your internet activities hidden from advertisers, governments, and everyone else.
- Normally, the server that you are sending requests to can see your IP address. Tor allows users to hide this information from everyone, even the server.

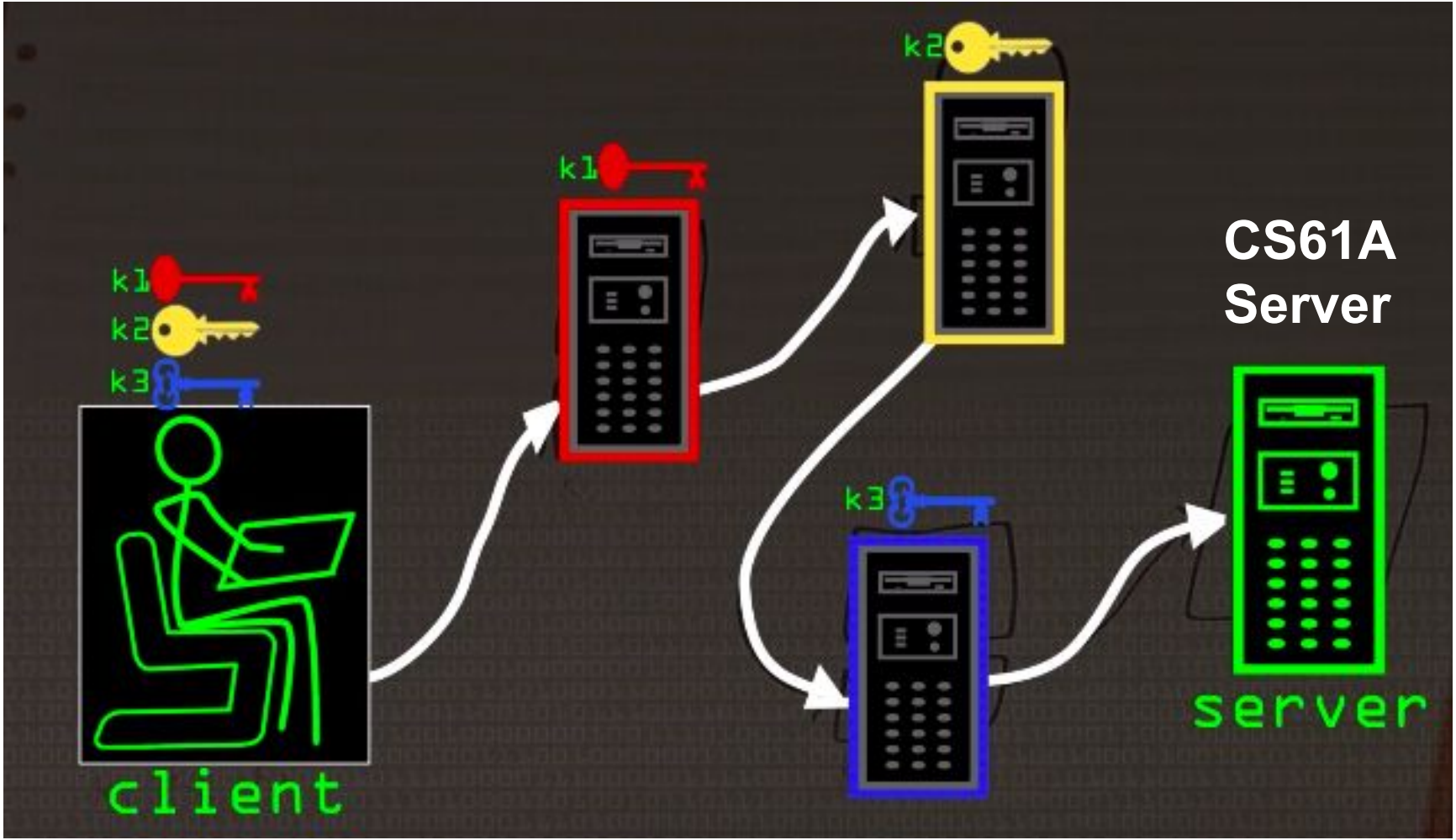
A Relevant Tangent on Encryption

- Encrypting data means encoding data such that only trusted parties can decode and view the original data.
  - Usually, "decoding" involves using a "key" that only the trusted parties have.
- Caesar Cipher

Original	Encrypted
"Sunset tomorrow is at five oclock."	"Vxqvhw wrpruurz lv dw ilyh rforfn."
"We will attack at sunset tomorrow "	"Zh zloo dwwdfn dw vxqvhw wrpruurz "

Tor (The Onion Router)

- When using Tor, every request you make is encased in multiple layers of encryption.
- Then, your message is sent through a number of "intermediate nodes" that peel back those layers one by one.
  - "Intermediate nodes" = other computers who are in the Tor network
- No single node can remove all the layers.
- Each node only decrypts enough information to know where to send your request next. None of the nodes know your identity or the website that you are trying to access.



Source: Computerphile, Youtube

Source: Computerphile, Youtube

Tor (The Onion Router)

- Tor as a Peer-to-Peer system
  - Often, users of Tor also serve as "intermediate nodes"
  - As more people use Tor, there is also more processing power for decrypting and passing along messages throughout the network
  - We are not reliant on one server to do all of that work!
- Tor is not pure Peer-to-Peer
  - Tor relies on "directory servers" where you can look up which keys to encrypt your messages with.
- Tor does not guaranteed anonymity
  - It doesn't prevent you from providing personal information.
  - It only hides your IP address and the website you are accessing from prying eyes.

Pros & Cons of Peer-to-Peer Systems

- Benefits:
  - No single point of failure!
  - As the network grows, so does the number of computers contributing their computing resources.
- Drawbacks:
  - Data Access
    - Absence of centralized server may make it difficult to efficiently access or scan data, because our data is located on different computers
  - Vulnerability
    - If someone controls too many nodes in the network, security can be compromised.

Parallel Computing

- In practice, big companies do not use Peer-to-Peer, because they usually do not want to rely on untrusted clients to power their network.

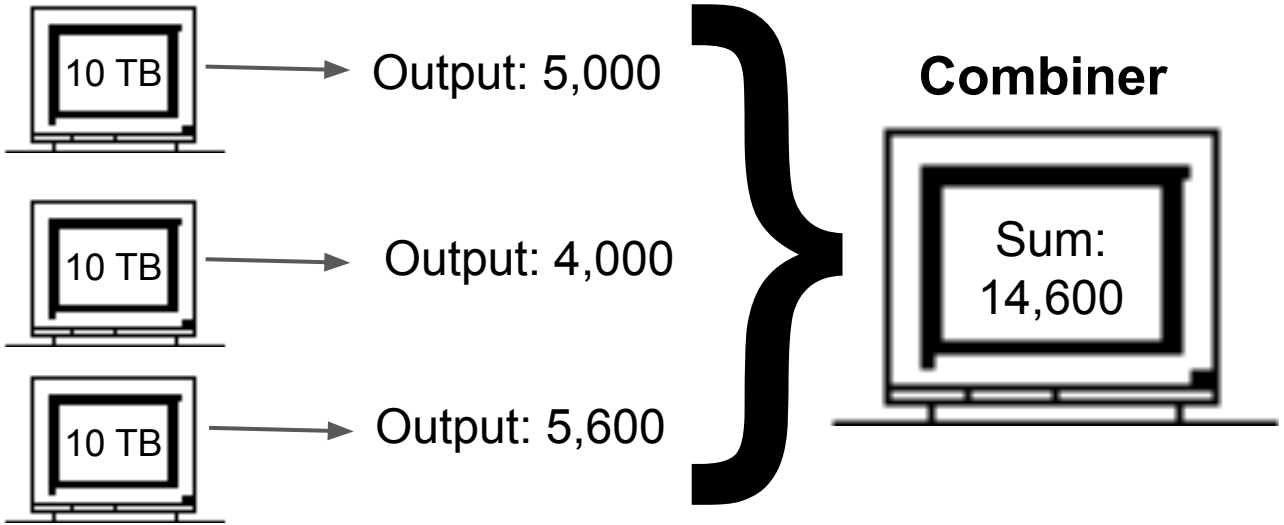
- Example of Parallel Computing: Big Data processing!
  - Recall how reading 1 TB of data would take ~3 hours.
  - What if we divided the work among 3 separate machines, which were all processing 1/3 of the data?
  - How long would it take to read 1 TB?

Parallel Computing

- **Parallel Computing** allows us to divide up the work of a big problem on a bunch of different computers **at the same time**, and therefore speed up our computation.

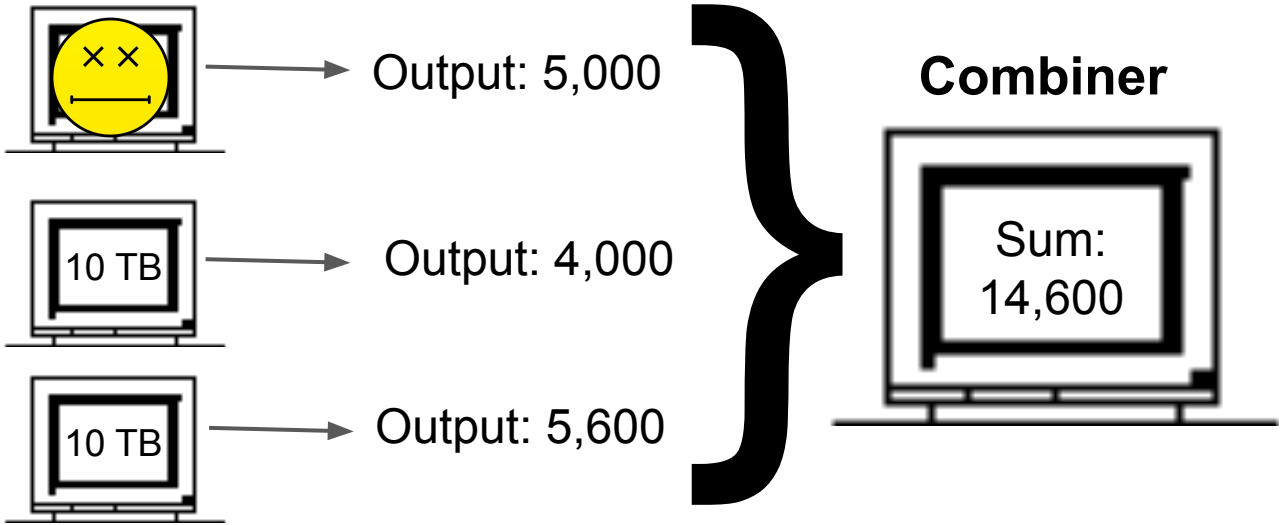
Parallel Computing

- Let's say I want to count all of today's Facebook posts that contain the word "CS61A" and Facebook's post dataset is 30 TB.
- Instead of storing all of the data on one machine, I divide it up among 3 machines.

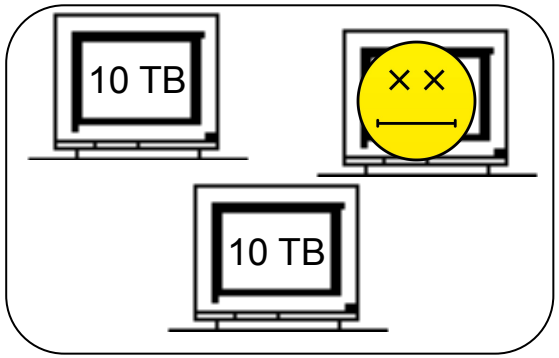


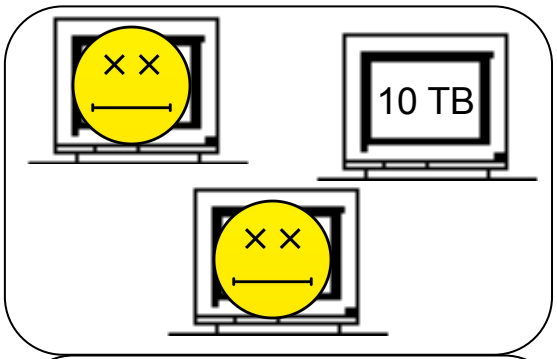
Fault Tolerance

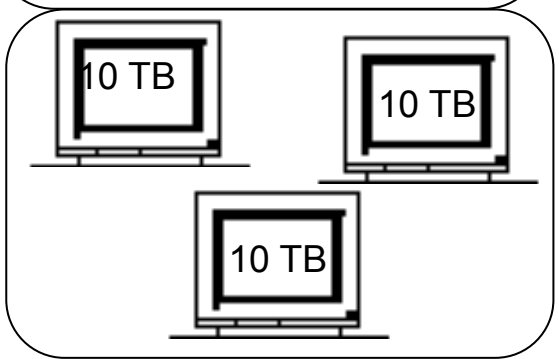
- What if one of the computers breaks?
- What if one of the computers takes a long time?



Cost vs. Reliability





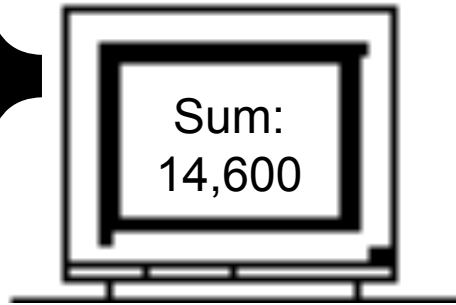


Output: 5,000

Output: 4,000

Output: 5,600

Combiner



Distributed Computing - It's all at your fingertips!

- Client / Server Architecture
  - Host your own webpage on Github
- Peer-to-Peer Systems
  - Mine bitcoin
- Parallel Computing
  - Apache Spark

#notspon