Through The Clouds:
An Introduction to Cloud Computing

Day 4: Projects & Wrap Up!

Instructors: Anurag Khandelwal, Ramla Ijaz, Garrett Sager
TA: Joaquin Soto

Yale

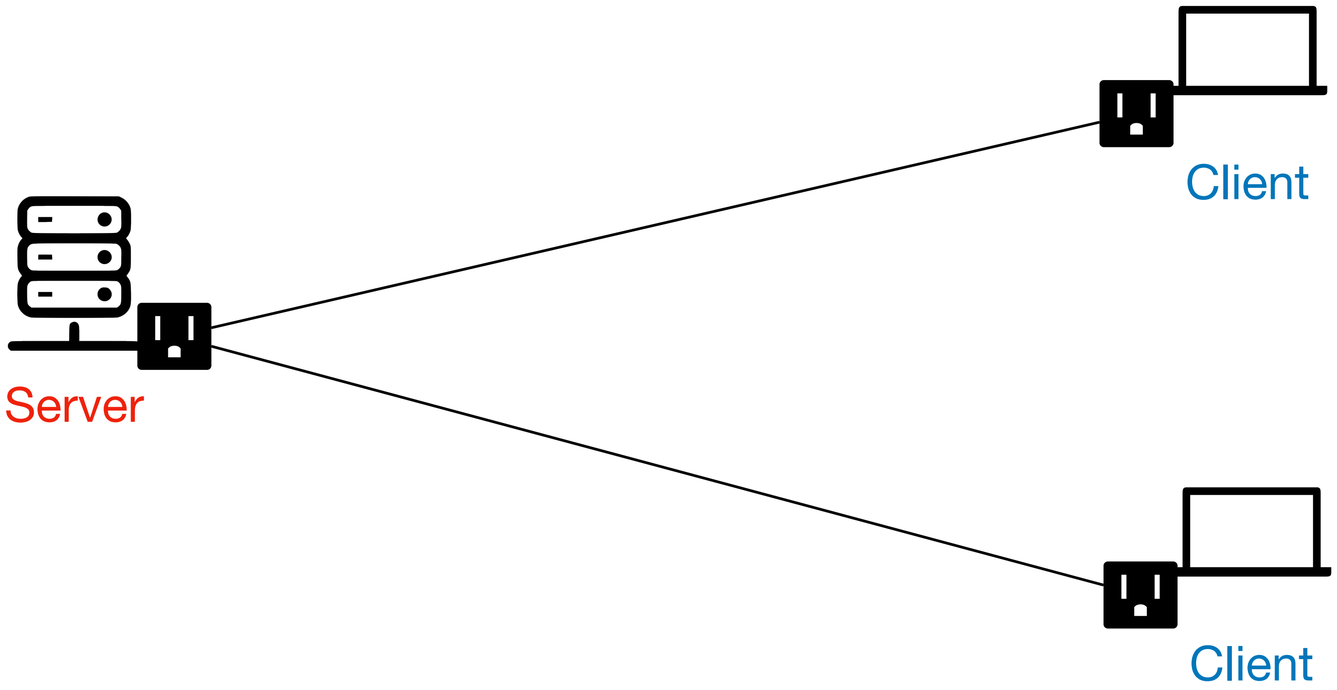
Today’s Agenda

Today’s Agenda

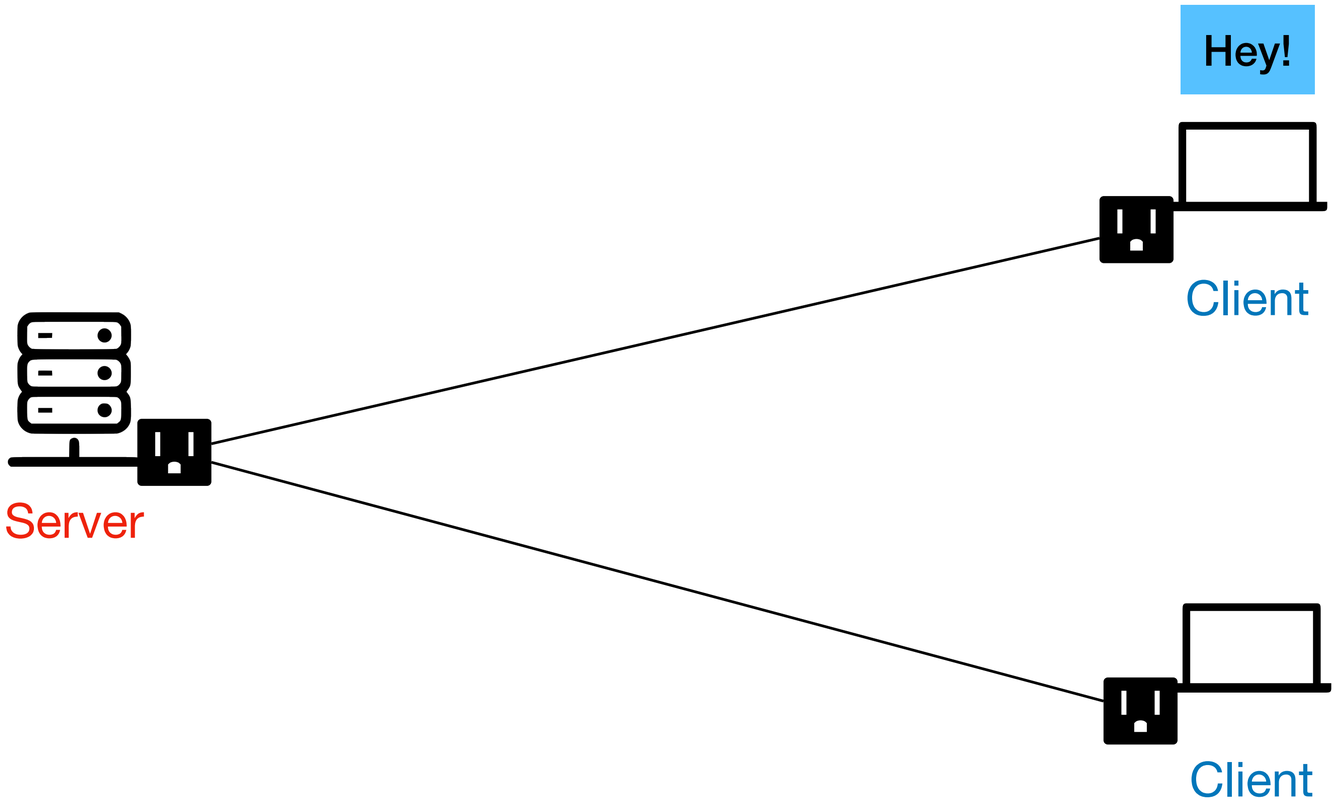
- Wrap up the chat application!
  - Client Registry
  - Instant Message
- What more do cloud applications have to worry about?

Demo & Coding

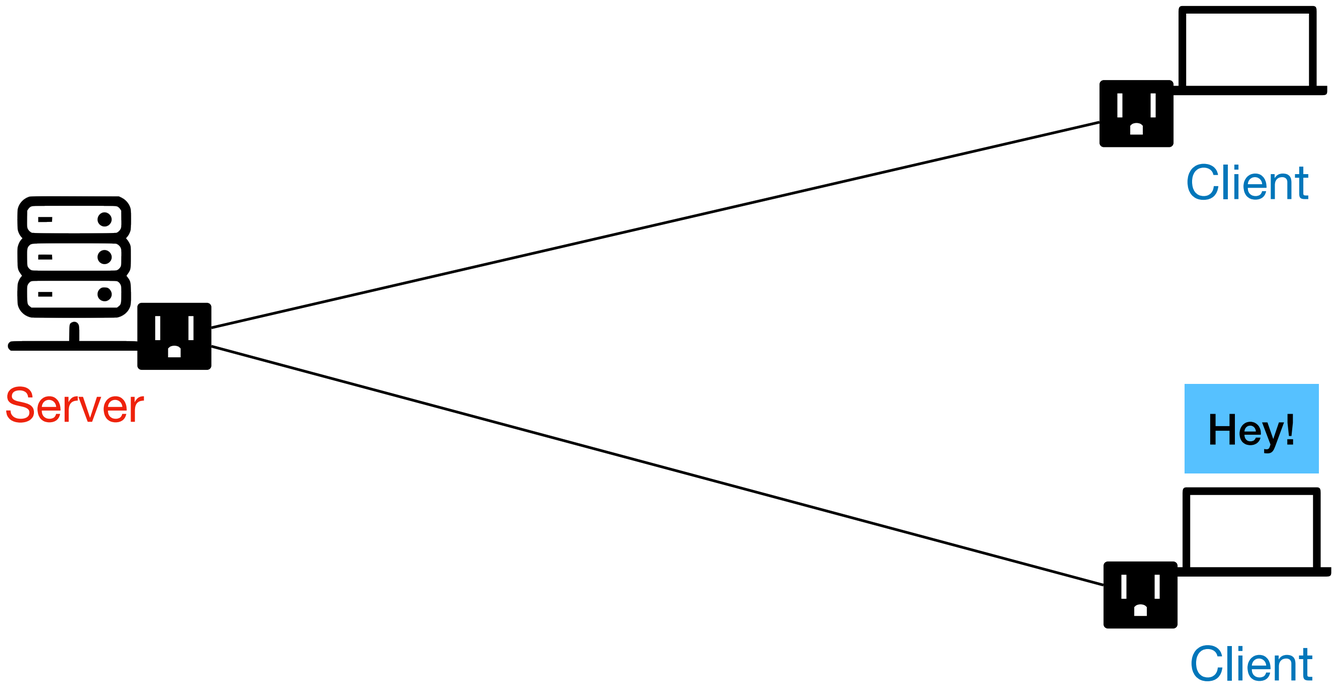
Final Step: Instant message!



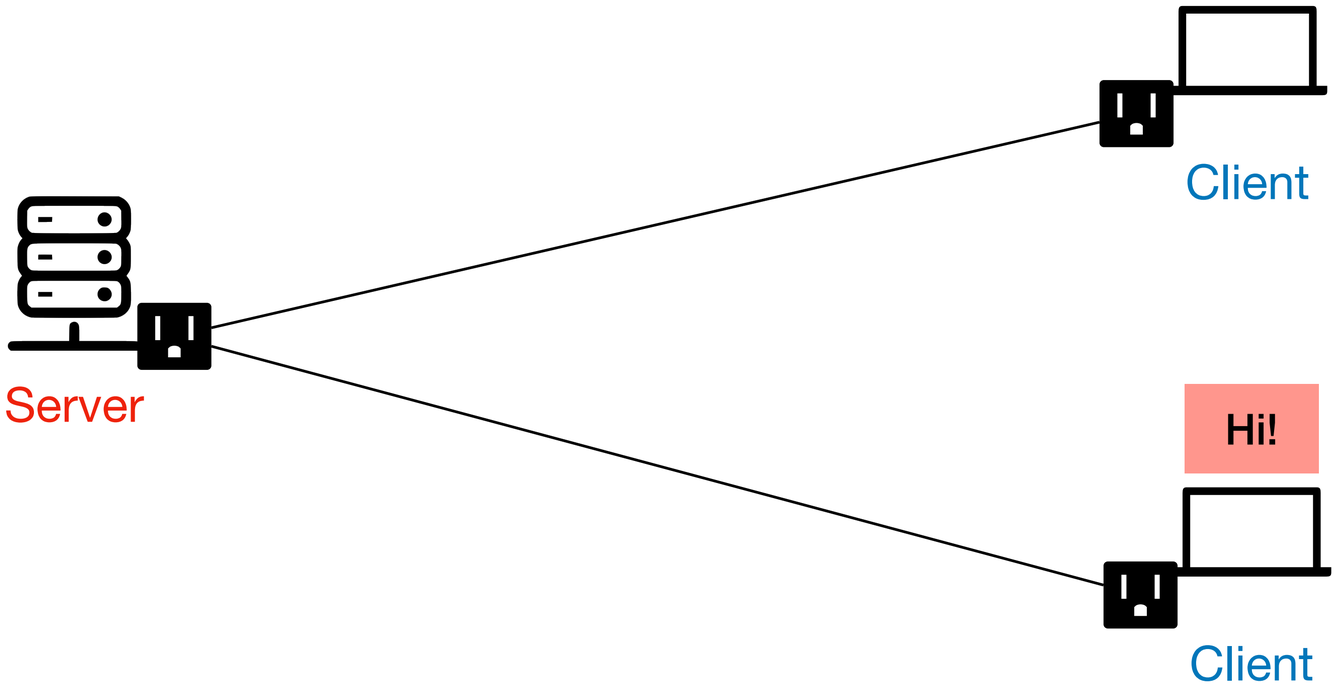
Final Step: Instant message!



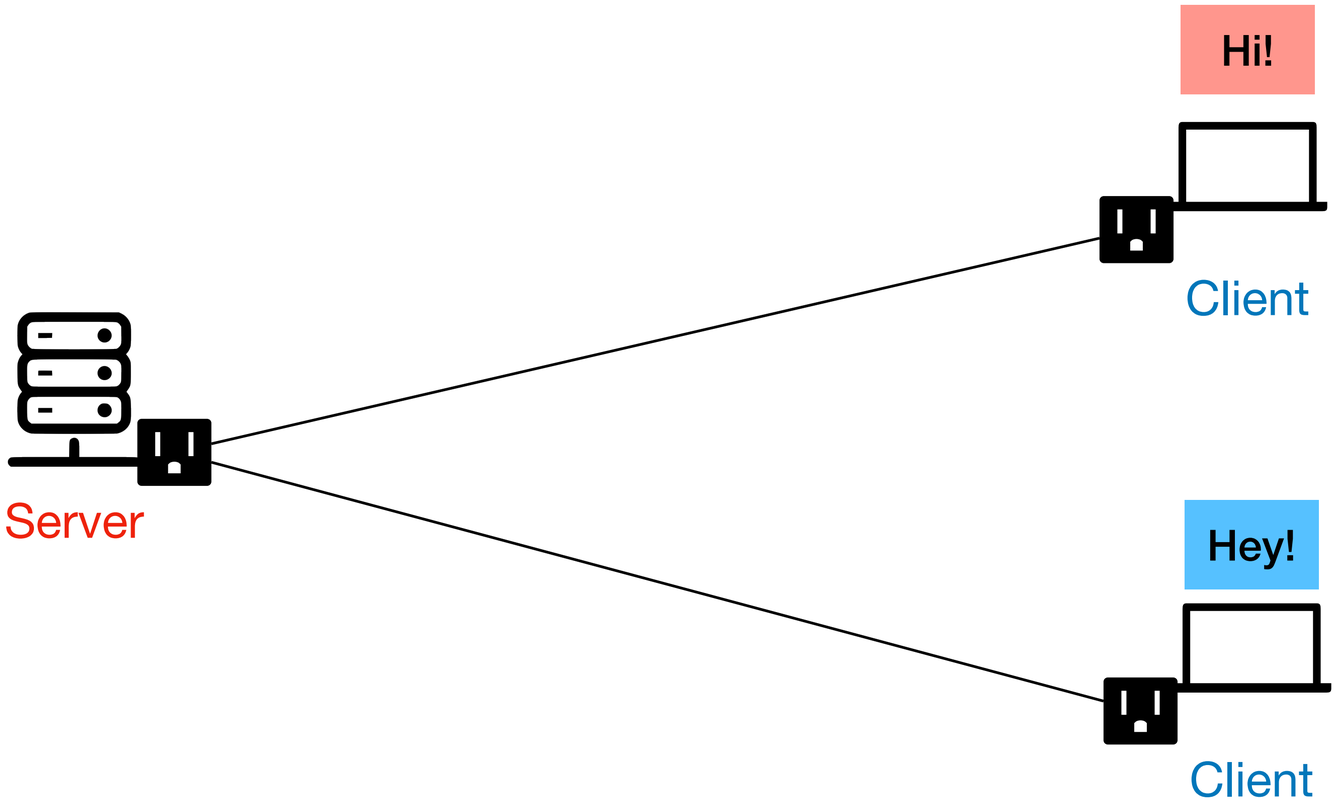
Final Step: Instant message!



Final Step: Instant message!



Final Step: Instant message!



How do you do it?

How do you do it?

- The functions you have to edit:
- In server_logic.py:

```
async def forward_message_to_client(server, receiver_username, msg):
  # your code here
```

- Helper functions:
  - `server.logged_in()` returns true if the sending client has already logged in
  - `server.user_is_logged_in(username)` returns true if a client with `username` is logged in

How do you test it?

How do you test it?

- Open **three** terminals
- Go to the directory where the code resides (`client_registry`) on all terminals

  `cd /path/to/client_registry` (on Linux & MacOS)
  `cd \path\to\client_registry` (on Windows)

- On one terminal, run the server:

  ```
  python3 server.py
  ```

- On the other two terminals, run two clients:

  ```
  python3 client.py
  ```

How do you test it?

How do you test it?

- Open **three** terminals
- Go to the directory where the code resides (`pingpong2`) on all terminals

  `cd /path/to/pingpong2` (on Linux & MacOS)
  `cd \path\to\pingpong2` (on Windows)

- On one terminal, run the server:

  ```
  python3 server.py
  ```

- On the other two terminals, run the clients:

  ```
  python3 client1.py
  python3 client2.py
  ```

Demo & Coding

You’re Done!

Any Questions?

If you’re still looking for a challenge

If you’re still looking for a challenge

- Try implementing the harder client_registry2.0
  - Server-side logic to handle the set of users that are registered/logged in

If you’re still looking for a challenge

- Try implementing the harder client_registry2.0
  - Server-side logic to handle the set of users that are registered/logged in
- Think about how to implement a group chat feature
  - How do you add users to a group chat?
  - How do you forward a message from one user to all?
  - Any other challenges?
  - Try implementing it! (Use instant_message as a starting point)

What haven’t we considered?

Problem#1: Scale

Problem#1: Scale

- We’ve only tested our code with very few users

Problem#1: Scale

- We’ve only tested our code with very few users
  - Real-world applications need to cater to millions to billions of users!

Problem#1: Scale

- We’ve only tested our code with very few users
  - Real-world applications need to cater to millions to billions of users!
- What breaks at those many clients?

Problem#1: Scale

- We’ve only tested our code with very few users
  - Real-world applications need to cater to millions to billions of users!
- What breaks at those many clients?
- A single server isn’t enough…

Problem#1: Scale

- We’ve only tested our code with very few users
  - Real-world applications need to cater to millions to billions of users!
- What breaks at those many clients?
- A single server isn’t enough…
  - Even if we store 1 Megabyte data for a single user, the total amount of data will be 1 Petabyte, much more than a single server (computer) can store!

Problem#1: Scale

- We’ve only tested our code with very few users
  - Real-world applications need to cater to millions to billions of users!
- What breaks at those many clients?
- A single server isn’t enough…
  - Even if we store 1 Megabyte data for a single user, the total amount of data will be 1 Petabyte, much more than a single server (computer) can store!
  - How do you implement server logic that runs on hundreds or thousands of computers?

Problem#1: Scale

- We’ve only tested our code with very few users
  - Real-world applications need to cater to millions to billions of users!
- What breaks at those many clients?
- A single server isn’t enough…
  - Even if we store 1 Megabyte data for a single user, the total amount of data will be 1 Petabyte, much more than a single server (computer) can store!
  - How do you implement server logic that runs on hundreds or thousands of computers?
  - Distributed computing!

Problem#2: Reliability

Problem#2: Reliability

- Suppose each user’s data is stored in a single computer

Problem#2: Reliability

- Suppose each user’s data is stored in a single computer
  - What happens if that computer crashes?

Problem#2: Reliability

- Suppose each user’s data is stored in a single computer
  - What happens if that computer crashes?
- How do we prevent data loss?

Problem#2: Reliability

- Suppose each user’s data is stored in a single computer
  - What happens if that computer crashes?
- How do we prevent data loss?
- Replication

Problem#2: Reliability

- Suppose each user’s data is stored in a single computer
  - What happens if that computer crashes?
- How do we prevent data loss?
- Replication
  - Create multiple copies of data across multiple computer

Problem#3: Security

Problem#3: Security

- In your chat application, anyone can claim to be anyone else!

Problem#3: Security

- In your chat application, anyone can claim to be anyone else!
- Anyone who can snoop into your network can look at what messages you are exchanging!

Problem#3: Security

- In your chat application, anyone can claim to be anyone else!
- Anyone who can snoop into your network can look at what messages you are exchanging!
  - Not safe or secure!

Problem#3: Security

- In your chat application, anyone can claim to be anyone else!
- Anyone who can snoop into your network can look at what messages you are exchanging!
  - Not safe or secure!
- Need mechanisms to ensure access to data is *authenticated,* i.e., can access only if I have the password

Problem#3: Security

- In your chat application, anyone can claim to be anyone else!
- Anyone who can snoop into your network can look at what messages you are exchanging!
  - Not safe or secure!
- Need mechanisms to ensure access to data is *authenticated,* i.e., can access only if I have the password
- Need mechanisms to ensure data communications are *secure,* i.e., no one else can see what my network traffic contains (e.g., using *encryption*)

And Many More!

And Many More!

- How do you minimize the cost?

And Many More!

- How do you minimize the cost?
  - More computers = more $$!

And Many More!

- How do you minimize the cost?
  - More computers = more $$!
- How do you ensure performance?

And Many More!

- How do you minimize the cost?
  - More computers = more $$!
- How do you ensure performance?
  - Users want their chat messages to get across in <100ms!

And Many More!

- How do you minimize the cost?
  - More computers = more $$!
- How do you ensure performance?
  - Users want their chat messages to get across in <100ms!
- How do you handle increase or decrease in load?

And Many More!

- How do you minimize the cost?
  - More computers = more $$!
- How do you ensure performance?
  - Users want their chat messages to get across in <100ms!
- How do you handle increase or decrease in load?
  - E.g., if suddenly a lot of users start using your chat

And Many More!

- How do you minimize the cost?
  - More computers = more $$!
- How do you ensure performance?
  - Users want their chat messages to get across in <100ms!
- How do you handle increase or decrease in load?
  - E.g., if suddenly a lot of users start using your chat
- etc., etc., etc…

Cloud Computing is a very active field!

Cloud Computing is a very active field!

- A lot of big industry players

Cloud Computing is a very active field!

- A lot of big industry players
  - Amazon, Google, Microsoft, IBM, Alibaba, …

Cloud Computing is a very active field!

- A lot of big industry players
  - Amazon, Google, Microsoft, IBM, Alibaba, …
  - $150 billion industry!

Cloud Computing is a very active field!

- A lot of big industry players
  - Amazon, Google, Microsoft, IBM, Alibaba, …
  - $150 billion industry!
  - Always looking for smart people to solve their problems! :)

Cloud Computing is a very active field!

- A lot of big industry players
  - Amazon, Google, Microsoft, IBM, Alibaba, …
  - $150 billion industry!
  - Always looking for smart people to solve their problems! :)
- A lot of active research

Cloud Computing is a very active field!

- A lot of big industry players
  - Amazon, Google, Microsoft, IBM, Alibaba, …
  - $150 billion industry!
  - Always looking for smart people to solve their problems! :)
- A lot of active research
  - Many conferences, with researchers actively exploring solutions to problems discussed here, and more!

Cloud Computing is a very active field!

- A lot of big industry players
  - Amazon, Google, Microsoft, IBM, Alibaba, …
  - $150 billion industry!
  - Always looking for smart people to solve their problems! :)
- A lot of active research
  - Many conferences, with researchers actively exploring solutions to problems discussed here, and more!
  - Also always looking for smart people to solve their problems! :)

Any Questions?

Thank you!

I hope you will consider a career in Cloud Computing! :)