Have you used distributed system?

Have you used distributed system?

Have you used distributed system?
What is a distributed system?

- A system of multiple computers (nodes) communicating over a network

Some following questions:
- What is a decentralized system?
- What is a cloud system?
- What is a centralized distributed system?
Network Basics

- We connect computers via point-to-point links:
  - Local area network, DNS and ISP routers
  - Communications are unreliable
  - No global control of the network

Example: HTTP Layer Encapsulation

End Hosts vs. Routers
Finding Nodes

- Each interface on a host has a unique MAC address:
  - My machine 48-bit ethernet address = 32:00:19:ac:b1:40
Network Basics

- Each interface on a host has a unique MAC address:
  - My machine 48-bit ethernet address = 32:00:19:ac:b1:40

  Why we need a physical address?

Network Basics

- Each interface on a host has a unique MAC address:
  - My machine 48-bit ethernet address = 32:00:19:ac:b1:40

  Why we need a physical address?

  Which layer in OSI model it belongs to?

Network Basics

- Each interface on a host has a unique MAC address:
  - My machine 48-bit ethernet address = 32:00:19:ac:b1:40

  This is not too interesting to us as programmers
  - We usually do not communicate at the data link layer

Network Basics

- Addressing applications:
  - IP address (32-bit for IPv4) and port number (16-bit)
  - Well-known port numbers (0-1023), e.g., ftp, ssh and http
Network Basics

- Addressing applications:
  - IP address (32-bit for IPv4) and port number (16-bit)
  - Well-known port numbers (0-1023), e.g., ftp, ssh and http

- We have two transport-layer protocols
  - TCP (SSH and FTP) and UDP (Streaming and local broadcast)
  - What is the difference?
Today's Cluster

PC

Server

Cluster

Today's Cluster

Today's Cluster
Today's Cluster

Network switches (connects nodes with each other and with other racks)

Many nodes/blades (often identical)

Storage device(s)
Today's Cluster

- What if cluster is too big to fit into machine room?

Datacenter

- What if cluster is too big to fit into machine room?
  - Build a separate building for the cluster
  - Building can have lots of cooling and power

Datacenter

- What if cluster is too big to fit into machine room?
  - Build a separate building for the cluster
  - Building can have lots of cooling and power
  - Result: Data center

Google Datacenter in Oregon
Data centers (size of a football field)

Google Datacenter in Oregon

A warehouse-sized computer
- A single data center can easily contain 10,000 racks with 100 cores in each rack (1,000,000 cores total)

Google Datacenters in the US

Google Datacenters in this World
End Hosts vs. Routers

Network APIs

- Programmers need to access the network
- A network application programming interface (API)
  - Socket programming
  - Remote procedure calls

Socket (TCP)

```python
import socket
s = socket.socket(AF_INET,
                  SOCK_STREAM)
s.bind(host, port)
s.listen(5)
while 1:
    conn, addr = s.accept()
    msg = conn.recv()
    conn.close
s.close
```
import socket
s = socket.socket(AF_INET, SOCK_STREAM)
s.bind(host, port)
s.listen(5)
while 1:
    conn, addr = s.accept()
    msg = conn.recv()
    conn.close
s.close
Socket (TCP)

```python
import socket

s = socket.socket(AF_INET, SOCK_STREAM)
s.bind(host, port)
s.listen(5)
while 1:
    conn, addr = s.accept()
    msg = conn.recv()
    conn.close
s.close
```

Socket (TCP)

```python
import socket

s = socket.socket(AF_INET, SOCK_STREAM)
a = socket.gethostbyname(host)
s.connect(a, port)
s.sendall(msg)
s.close
```

Socket (UDP)

```python
Create a socket
Name the socket (assign local address, port)
Set the socket for listening
Wait for and accept a connection; get a socket for the connection
read / write byte streams
close the socket
```

```
import socket

s = socket.socket(AF_INET, SOCK_DGRAM)
a = socket.gethostbyname(host)
s.sendto(msg, a)
s.recvfrom()"
What's the Cloud Computing

Cloud computing is a business model for enabling convenient network access to a shared pool of configurable resources which can be rapidly provisioned and released with minimal management effort or service provider interaction.

--- according to NIST (National Institute of Standards and Technology)

Have You Used the Cloud?
Have You Used the Cloud?

Why We Like It?

- Why users like it?
  - Do not care where it is, it is “just there”
  - Access from “any” platform
Why We Like It?

• Why users like it?
  - Do not care where it is, it is “just there”
  - Access from “any” platform

| Cloud Services v.s. Traditional Distributed Systems |

Why We Like It?

• Why users like it?
  - Do not care where it is, it is “just there”
  - Access from “any” platform

• Why CS researchers like it?
  - High-performance computation with less money
  - Lots of hard and interesting new challenges

Building Blocks

• What techniques are used to support cloud?
  - Internet
  - Smart and cheap personal devices
  - Robust and scalable software systems
  - Virtualization
  - ... ...
Types of Cloud Services

- Three types of services:
  - Software as a Service (SaaS)
    - Analogy: Restaurant. Prepares & serves entire meal, does the dishes, etc.
  - Platform as a Service (PaaS)
    - Analogy: Take-out food. Prepares meal but does not serve it.
  - Infrastructure as a Service (IaaS)
    - Analogy: Grocery store. Provides raw ingredients.
Software as a Service (SaaS)

- SaaS provider offers an entire application
  - Word processor, spreadsheet, CRM software, etc.

- Customer pays cloud provider and uses the service
  - Google Apps, Salesforce.com, etc.
Software as a Service (SaaS)

- SaaS provider offers an entire application
  - Word processor, spreadsheet, CRM software, etc.
  - Customer pays cloud provider and uses the service

SaaS Example: Gmail

Gmail Provider
• Outsourcing your e-mail software:
  - Distributed, replicated message store in BigTable
Outsourcing your e-mail software:
- Distributed, replicated message store in BigTable
- Weak consistency model for some operations (e.g., msg read)
- Stronger consistency for others (e.g., send msg)
Platform as a Service (PaaS)

Cloud Provider (i.e., PaaS Provider)
- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)

Customer pays SaaS provider for the service
SaaS provider pays the cloud for the platform
Example: Windows Azure, Google App Engine, etc.

App Provider
- App provider pays the cloud for the platform
  - For example, Microsoft Common Language Runtime (CLR)
Platform as a Service (PaaS)

- Cloud provides middleware/infrastructure
  - For example, Microsoft Common Language Runtime (CLR)
  - App provider pays the cloud for the platform
  - Customer pays app provider for the service
  - Example: Windows Azure, Google App Engine, etc.
• Facebook offers PaaS capabilities to App provider

Facebook offers PaaS capabilities to App provider.
Facebook offers PaaS capabilities to App provider
- Facebook APIs allow access to social network properties
- App providers adopt their services (e.g., game) onto Facebook

Facebook itself also uses PaaS provided by its company, e.g., log analysis for recommendations
Facebook offers PaaS capabilities to App provider
- Facebook APIs allow access to social network properties
- App providers adopt their services (e.g., game) onto Facebook
- Facebook itself also uses PaaS provided by its company, e.g., log analysis for recommendations

Cloud provides raw computing resources
- Cloud Provider (i.e., IaaS Provider)
Cloud provides raw computing resources
- Virtual machines, blade servers, hard disk, etc.
- App provider pays the cloud for the resources
- Customer pays App provider for the service

Example: Amazon Web Services, Rackspace Cloud, etc.
Infrastructure as a Service (IaaS)

- Cloud provides raw computing resources
  - Virtual machines, blade servers, hard disk, etc.
  - App provider pays the cloud for the resources
  - Customer pays App provider for the service

IaaS Example: EC2 and S3
(Elastic Compute Cloud & Simple Storage Service)

Amazon

EC2
S3
Netflix (app) heavily depends on Amazon AWS:
- Media files are stored in S3
- Transcoding to target devices (e.g., iPad) using EC2
- Analysis of streaming sessions based on Elastic MapReduce
Netflix (app) heavily depends on Amazon AWS:
- Media files are stored in S3
- Transcoding to target devices (e.g., iPad) using EC2

Types of Cloud Services
- Three types of services:
  - Software as a Service (SaaS)
    - Analogy: Restaurant. Prepares&serves entire meal, does the dishes, etc
  - Platform as a Service (PaaS)
    - Analogy: Take-out food. Prepares meal but does not serve it.
  - Infrastructure as a Service (IaaS)
    - Analogy: Grocery store. Provides raw ingredients.

The Major Cloud Providers
- Amazon is the big player:
  - Infrastructure as a service (e.g., EC2)
  - Storage as a service (e.g., S3)
The Major Cloud Providers

- **Amazon** is the big player:
  - Infrastructure as a service (e.g., EC2)
  - Storage as a service (e.g., S3)
- But there are many others:
  - **Microsoft Azure**: It has similar services to Amazon, with an emphasis on .Net programming model
  - **Google App Engine**: It offers programming interface, Hadoop, also software as a service, e.g., Gmail and Google Docs
  - **IBM, HP, Yahoo!**: They seem to focus on enterprise scale cloud apps

Challenges?

In the cloud, we have much more data and users than before

Data centers (size of a football field)

- **A warehouse-sized computer**
  - A single data center can easily contain 10,000 racks with 100 cores in each rack (1,000,000 cores total)
Google’s Datacenter Locations

Challenges?

- How to manage a huge group of data?
- How to store the data?
- How to process and extract something from the data?
- How to handle multiple availability and consistency?
- How to preserve the data privacy?

Example: Google

- How to manage a huge group of data?
  - How to store the data?
  - How to process and extract something from the data?
  - How to handle multiple availability and consistency?
  - How to preserve the data privacy?

BitCoin ≠ Blockchain
The Blockchain

A: $10
B: $2
D: $7
C: $5

A->B: $3
A: 10 - 3 = 7
B: 2 + 3 = 5
Blockchain is used to decentralize the log:

- Decentralization
- Public accountability
- Efficiency
The Blockchain

• Each block contains multiple transactions
• Each user locally maintains a ledger
• All ledgers should have the same data

Will disk space become a burden?

• Transactions are hashed in a Merkle Tree.
• If we suppose blocks are generated every 10 minutes, then 4.2MB per year.

• Each hash identifies the entire prefix of the log
Transactions in the Blockchain

A: $7
B: $5

D: $7
C: $5

A->B : 4

Transactions in the Blockchain

A: $3
B: $9

D: $7
C: $5

A->B : 4

Transactions in the Blockchain

A: $3
B: $9

D: $7
C: $5

B->D : 1

Transactions in the Blockchain

A: $3
B: $9

D: $7
C: $5

A->B : 4
Transactions in the Blockchain

A: $3
B: $8
C: $5
D: $8

A -> B: 4
B -> D: 1

Consensus

I am the leader

A: $3
B: $8
C: $5
D: $8

A -> B: 4
B -> D: 1

New Block Generation

A: $3
B: $8
C: $5
D: $8

New Block

A -> B: 4
B -> D: 1

New Block Generation

A: $3
B: $8
C: $5
D: $8

New Block

A -> B: 4
B -> D: 1
The Blockchain

- Blockchain can be used to decentralize any centralized service:
  - Making them decentralized (without single-point-fault)
  - Public accountability

- We still have two problems:
  - How to achieve consensus?
  - How to preserve the privacy?

How to decentralize app via blockchain?

Log (or Ledger)

- What data we want to put as “transaction”
- The data is what we want to audit

Smart Contract

Contract: If xx then yy
**Smart Contract**

A→B : 3

**Example**

- You are planning to ship a laptop to your friend Bob
  - You trust Bob, but you do not trust trucker Tom
  - Tom will carry your laptop
  - Tom does not trust since maybe you will not pay him

**Example**

- You are planning to ship a laptop to your friend Bob
  - You trust Bob, but you do not trust trucker Tom
  - Tom will carry your laptop
  - Tom does not trust since maybe you will not pay him

**Example**

- We can use smart contract:
  - You and Tom define all the rules in code
  - You make a payment for shipment to smart contract on a day of loading.
  - It holds payment till shipment delivery is confirmed by Bob.
  - Smart contract releases the payment and money is transferred to Tom automatically.

You and Tom have to sign a contract.
Another Example

Doctor informs patient that they need to exercise

Patient agrees to exercise regime

A "HealthCare" is placed - a smart contract is placed in the patient's wallet (with debit/credit)

A ledger records all changes

Shared Ledger

Another Example

Doctor informs patient that they need to exercise

Patient agrees to exercise regime

A "HealthCare" is placed - a smart contract is placed in the patient's wallet (with debit/credit)

As an individual performs agreed actions, health costs change (either go up or down - tracked by servicable)
Another Example

The Blockchain

- Blockchain can be used to decentralize any centralized service:
  - Making them decentralized (without single-point-fault)
  - Public accountability

- We still have two problems:
  - How to achieve consensus?
  - How to preserve the privacy?

Deployment of BitCoin Nodes

- Blockchain is used for a decentralized bank:
  - Each user has several wallets (public keys)
  - They sign the money transaction using the private key

How to compute BitCoin?

If B’s initial value is 0, then B is 4-1+1-2+1=3
How to compute BitCoin?

<table>
<thead>
<tr>
<th>Previous output index</th>
<th>Amount</th>
<th>From address</th>
<th>Type</th>
<th>Scriptsig</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Inputs

<table>
<thead>
<tr>
<th>Index</th>
<th>Reward at input</th>
<th>Amount</th>
<th>To address</th>
<th>Type</th>
<th>Scriptsig</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>139.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>139.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outputs

<table>
<thead>
<tr>
<th>Index</th>
<th>Reward at output</th>
<th>Amount</th>
<th>To address</th>
<th>Type</th>
<th>Scriptsig</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Who should generate a new block to include these two transactions?
They need to compete, and the winner can earn money

How to compute BitCoin?

\[ X = \text{SHA256}(H + \text{salt}) \]
\[ X \text{ should be '0000....'} \]

SHA256("The quick brown fox jumps over the lazy dog")
0x d7a8fbb307d780946eca9acbcb0012e4f8d56514ed3cd762d02df37cde592
SHA256("The quick brown fox jumps over the lazy dog.")
0x ef53725c95b9a782526529a9b63d97aa631564d5d79c2765448cb635b6c
How to compute BitCoin?

SHA256("The quick brown fox jumps over the lazy dog")
0x d7a8fbb307d7899469ca9abc0082e4f8d551e4ed3c1db762d2d029f7c7e592
SHA256("The quick brown fox jumps over the lazy dog.")
0xe953725a9b9a782a2629d97d05e3196f5d708ea775448c80b65f80c

X = SHA256(H + salt)
X should be '0000....'

Salt=8

Proof of Work

- BitCoin uses the proof of work to achieve many goals:
  - Generating additional money
  - Achieving consensus while tolerating malicious users
  - A great incentive mechanism
Proof of Work

- BitCoin uses the proof of work to achieve many goals:
  - Generating additional money
  - Achieving consensus while tolerating malicious users
  - A great incentive mechanism

Proof of Work

- Occasionally, more than one block will be solved at the same time, leading to several possible branches.

Proof of Work

- We should build on top of the first one you received.
- Others may have received the blocks in a different order, and will be building on the first block they received.

Example

[Diagram showing blockchain and connections across the world]
Proof of Work

• We do not need to worry about the branch problem:
  - You always immediately switch to the longest branch
  - The math makes it rare for blocks to be solved at the same time, and even more rare for this to happen multiple times
  - The end result is the block chain quickly stabilizes

• ~10 minutes to generate a new block
• Your transactions are confirmed after 6 blocks

Proof of Work

• We do not need to worry about the branch problem:
  - You always immediately switch to the longest branch
  - The math makes it rare for blocks to be solved at the same time, and even more rare for this to happen multiple times
  - The end result is the block chain quickly stabilizes

Miners in BitCoin can earn a lot of money!