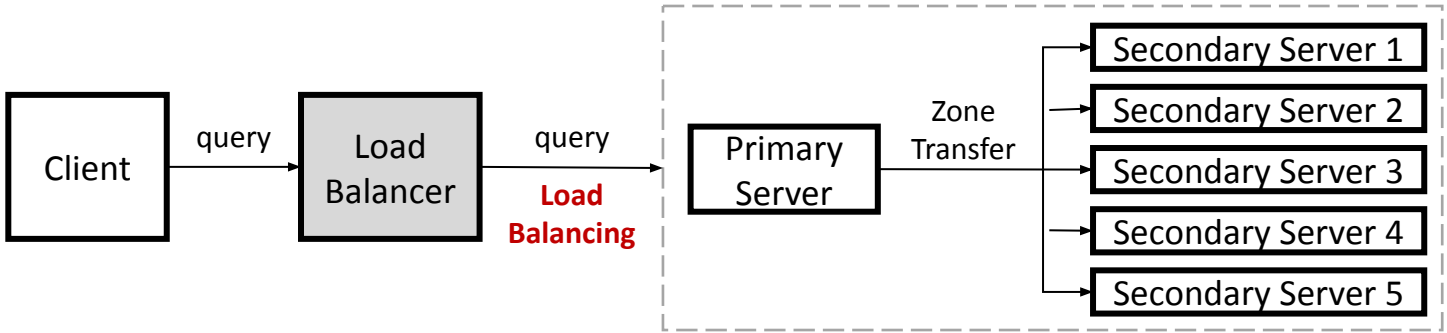
The DNS Singularity Uptime Beyond the Black Hole

Erjie Zhang erjzhang@ucdavis.edu
Hezhi Xie hezxie@ucdavis.edu
Muhammad Hassnain mhassnain@ucdavis.edu
Zeerak Babar zebabar@ucdavis.edu

Goals of the Project

- *Develop a DNS system against DoS Attack*
  - *Replicate data among different servers*
  - *Balance load among different servers for internet queries*
- *Test our system against simulated attacks*



Our System Design

Results

- *Successfully built a DNS system with multiple name servers*
  - *Multiple name servers built on multiple virtual machines*
- *Successfully replicate data among different servers*
  - *Primary-secondary zone transfer*
  - *Replicate automatically & periodically*
- *Successfully implement load balancing among different servers*
  - *Least Connection algorithm*
  - *Proactive load balancing by configuring proxy response limits and timeout duration*
- *Successfully test and compare the results with simulated attack*
  - *High volumes of queries sent at one time*
  - *100% response rate comparing to 99.1% response rate of single server*

Pictures and/or Diagrams

We simulated an attack with a high volume of queries on our system, and the experiment result shows the robustness of our system in this scenario.

- Scenario 1: Single server;
- Scenario 2: Two servers with load balancing;
- Scenarios 3: Six servers with load balancing.

Scenario	Number of Requests	Number of Answers	Respond Rate
1	50000	49551	99.1%
2	50000	49968	99.9%
3	50000	50000	100.0%

Response Rate Under Different Scenarios

APPENDIX

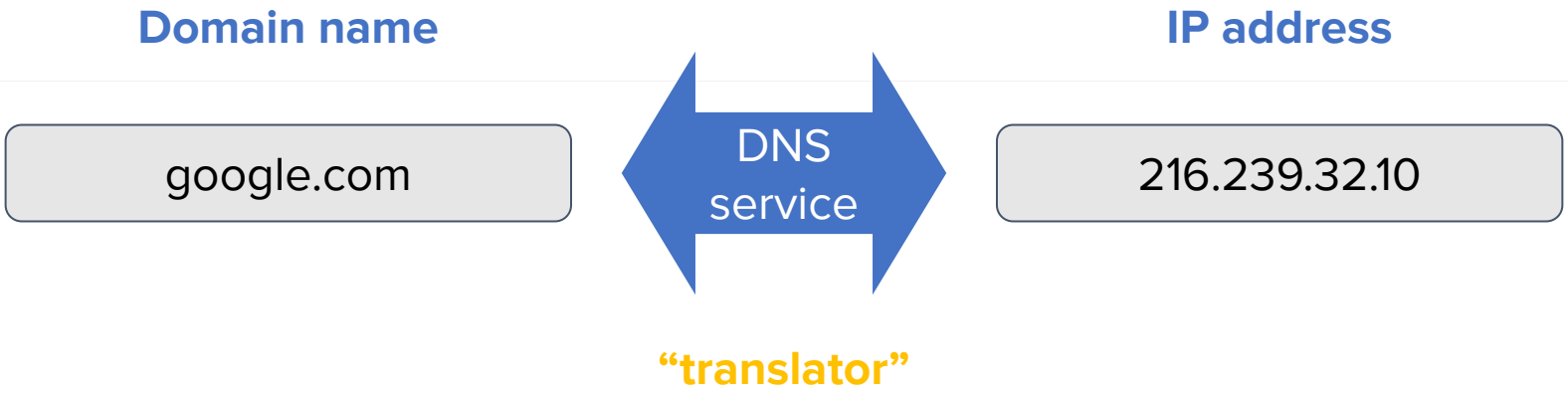
1. Introduction
2. Proposed Solutions and Challenges
3. Our Solution
4. Implementation
5. Conclusion

Part 01

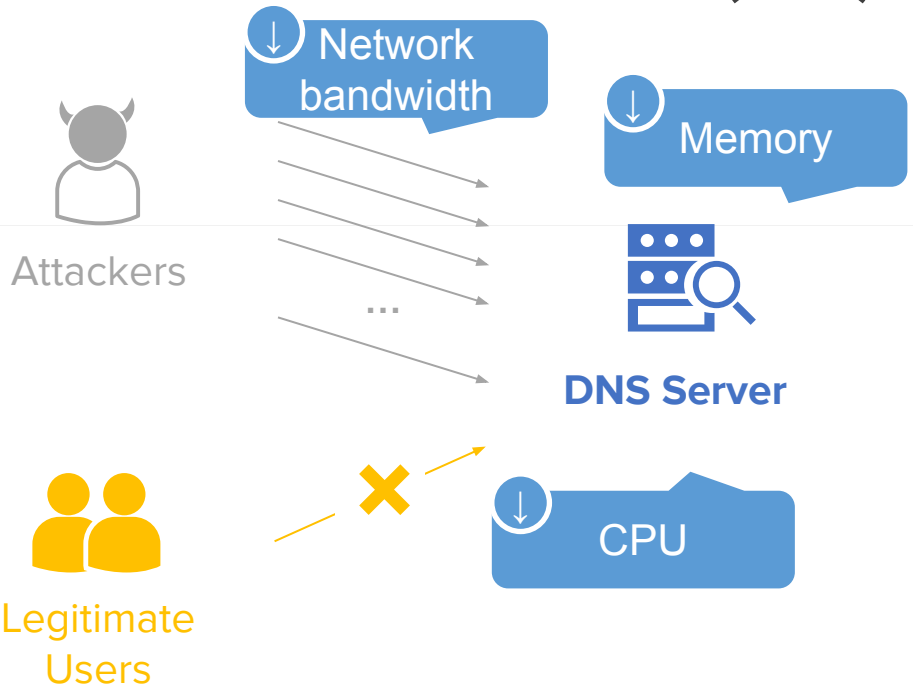
Introduction

UCDAVIS

What is Domain Name System (DNS)?



What is Denial of Service (DoS) Attack?



Three Main Types of DoS Attack

Flood attack

Flood the DNS server with a large volume of queries

Amplification attack

Flood the DNS server with short requests which require long responses

Reflection attack

Manipulate open DNS servers to overwhelm the target victim

What are Primary Defences against DoS Attack?

General

- Disable unused services;
- Install latest security patches;
- Locate DNS servers separately; …

Filtering

- Detect and block malicious traffic;
- Statistically analyze queries information such as IP addresses, protocol type, port number, …

Tolerance-based

Bandwidth capacity

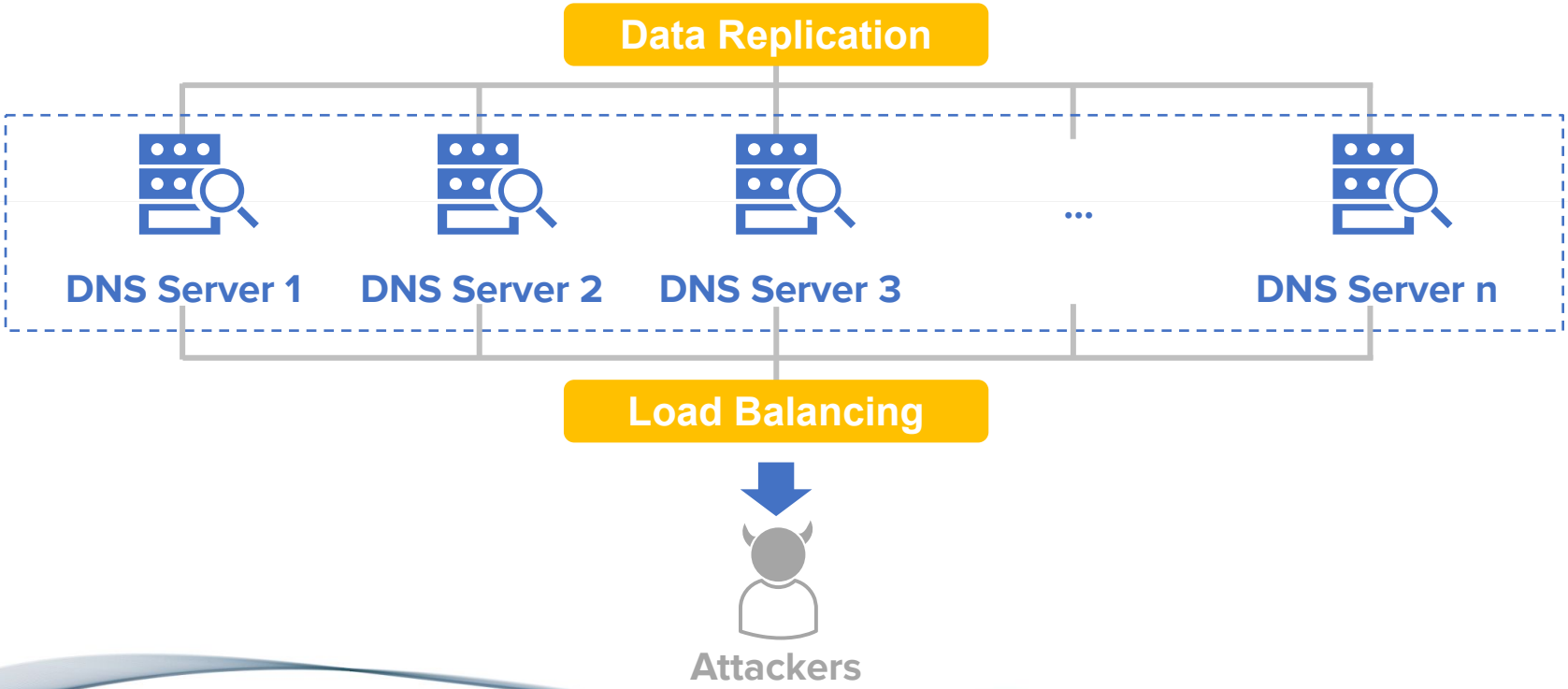
- Content Distribution Network
- Smart DNS resolution

Server capacity

- **Utilize multiple DNS servers (Data replication, load balancing, …)**
- Enhance the ability of DNS servers

UCDAVIS

What Are We Going To Do?



* The scope of our project does not include security during the replication process.

Part 02

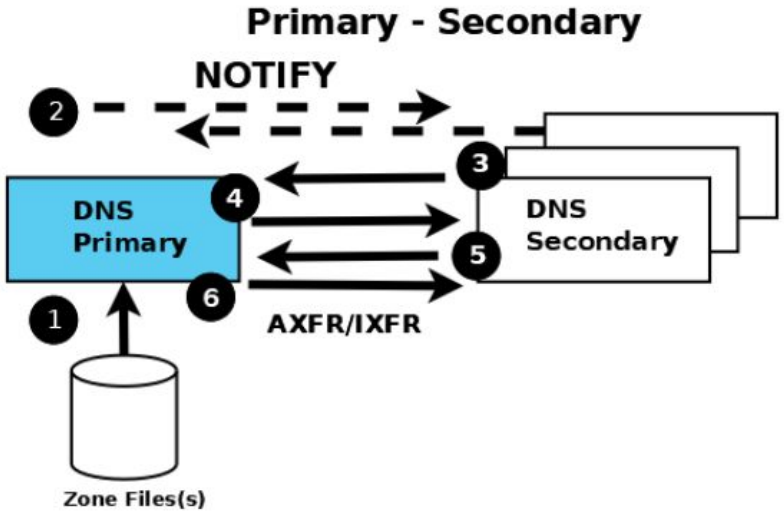
Proposed Solutions and Challenges

UCDAVIS

Proposed solutions

Data replication

- The basic data replication method for DNS is called Zone Transfer



Step1: The primary server loads zone data from designated zone files;

Step2: The secondary server loads zone data through a zone transfer operation (AXFR or IXFR);

Step3: Upon any modifications in the primary server's zone file, the primary server will notify the secondary server through a process named DNS NOTIFY, prompting it to initiate a zone transfer.

Proposed solutions

Load Balancing

- Anycast DNS
  - DNS servers at diverse locations are assigned the same IP address
  - Redirect DNS queries to the nearest server
  - Need the help of ISP, which is difficult to implement

- Load Balancer
  - DNS servers built on multiple virtual machines and are assigned with different IP addresses
  - A load balancer acts as a gateway for client queries, and distribute traffic to different servers
  - It is feasible for us to implement

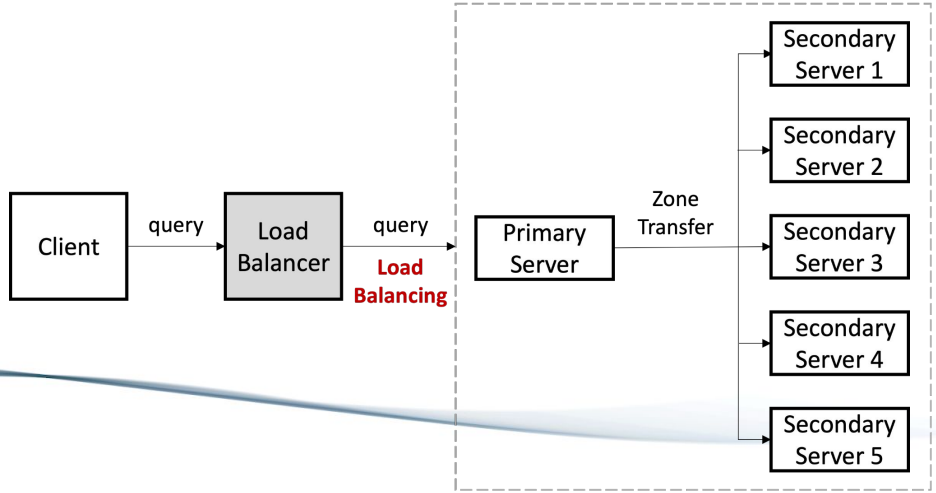
Part 03

Our Solution

UCDAVIS

Our Solution

- Deploy Virtual Machines (VMs) to create a network of DNS servers, including a primary server and multiple secondary servers;
- Each secondary server maintains identical zone files as the primary server through Zone Transfer;
- Deploy a load balancer across these DNS servers, which acts as a gateway for client queries, distributing traffic across the servers.



Part 04

Implementation

UCDAVIS

Step 1 Build DNS Servers

- Environment: Ubuntu 22
- DNS Tool: Berkeley Internet Name Domain (BIND 9)

No.	Role	IP
1	Client	192.168.64.100
2	Load balancer	192.168.64.2
3	Primary name server	192.168.64.10
4	Secondary name server 1	192.168.64.11
5	Secondary name server 2	192.168.64.12
6	Secondary name server 3	192.168.64.13
7	Secondary name server 4	192.168.64.14
8	Secondary name server 5	192.168.64.15

Table 1: IP Addresses of VMs

UCDAVIS

Step 1 Build DNS Servers

- Environment: Ubuntu 22
- DNS Tool: Berkeley Internet Name Domain (BIND 9)

```
hezhi@client:/etc$ dig www.example.com

; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33822
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: a44ac98d6c0ab1a8010000006576bfd81cd299ab5b419b29 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       10.0.0.7

;; Query time: 0 msec
;; SERVER: 192.168.64.10#53(192.168.64.10) (UDP)
;; WHEN: Mon Dec 11 07:52:56 UTC 2023
```

```
hezhi@client:/etc$ dig www.google.com

; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51874
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: c1e5f270907d26bf010000006576cdde93fbb966f4edacbe (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         300     IN      A       142.250.189.228

;; Query time: 212 msec
;; SERVER: 192.168.64.10#53(192.168.64.10) (UDP)
;; WHEN: Mon Dec 11 08:52:46 UTC 2023
;; MSG SIZE  rcvd: 87
```

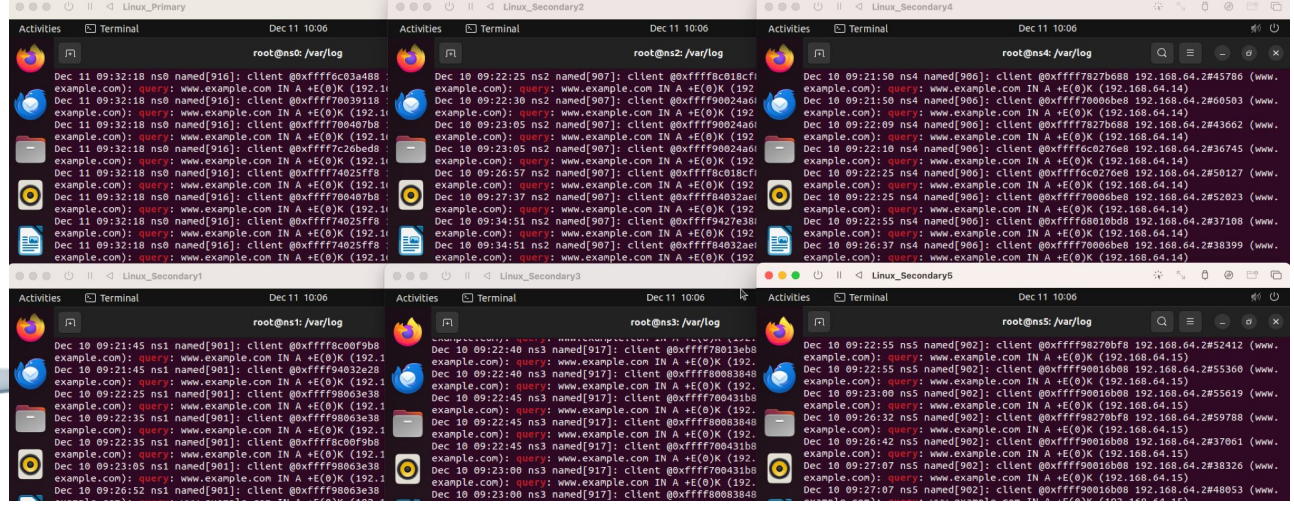
Step 2 Data Replication

- Method: Primary-secondary zone transfer

```
Dec 10 10:41:44 ns2 named[909]: transfer of 'example.com/IN' from 192.168.64.10#53: connected using 192.168.64.10#53
Dec 10 10:41:44 ns2 named[909]: transfer of '0.0.10.in_addr.arpa/IN' from 192.168.64.10#53: connected using 192.168.64.10#53
Dec 10 10:41:44 ns2 named[909]: zone example.com/IN: transferred serial 2008111001
Dec 10 10:41:44 ns2 named[909]: transfer of 'example.com/IN' from 192.168.64.10#53: Transfer status: success
Dec 10 10:41:44 ns2 named[909]: transfer of 'example.com/IN' from 192.168.64.10#53: Transfer completed: 1 messages, 8 records, 215 byt
es, 0.003 secs (71666 bytes/sec) (serial 2008111001)
Dec 10 10:41:44 ns2 named[909]: zone 0.0.10.in_addr.arpa/IN: transferred serial 2008111001
Dec 10 10:41:44 ns2 named[909]: transfer of '0.0.10.in_addr.arpa/IN' from 192.168.64.10#53: Transfer status: success
Dec 10 10:41:44 ns2 named[909]: transfer of '0.0.10.in_addr.arpa/IN' from 192.168.64.10#53: Transfer completed: 1 messages, 6 records,
 201 bytes, 0.003 secs (67000 bytes/sec) (serial 2008111001)
```

Step 3 Load balancing

- Tool: NGINX (an open-source software for load balancing)
- Load Balancing Algorithm: Least Connection Algorithm (LC)
- Parameters: proxy response limit=1; timeout duration=1s (If one server fails to respond within the specified time or query limit, the load balancer automatically redirects the request to the next available server)



UCDAVIS

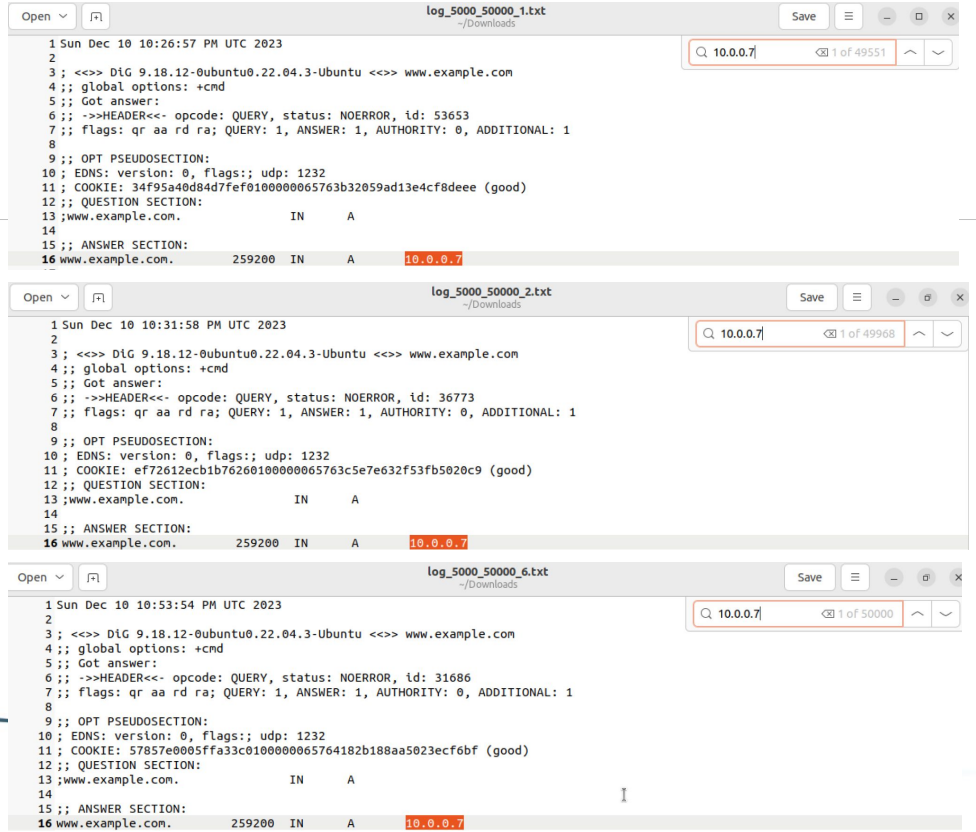
Step 4 Evaluation

- Limitation: Due to our limited resources, genuine DoS attack is infeasible
- Alternative Solution: Send high volume of queries at one time for testing
- Parameters: Number of processes=5000; Number of total request=50,000
  - Scenario 1: Single server;
  - Scenario 2: Two servers with load balancing;
  - Scenarios 3: Six servers with load balancing.

Scenario	Number of Requests	Number of Answers	Respond Rate
1	50000	49551	99.1%
2	50000	49968	99.9%
3	50000	50000	100.0%

Table 2: Respond Rate Under Different Scenarios

Step 4 Evaluation



log_5000_50000_1.txt
~/Downloads

```
1 Sun Dec 10 10:26:57 PM UTC 2023
2
3 ; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> www.example.com
4 ;; global options: +cmd
5 ;; Got answer:
6 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53653
7 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
8
9 ;; OPT PSEUDOSECTION:
10 ; EDNS: version: 0, flags:; udp: 1232
11 ; COOKIE: 34f95a40d84d7fef0100000065763b32059ad13e4cf8deee (good)
12 ;; QUESTION SECTION:
13 ;www.example.com.               IN      A
14
15 ;; ANSWER SECTION:
16 www.example.com.        259200  IN      A       10.0.0.7
```

log_5000_50000_2.txt
~/Downloads

```
1 Sun Dec 10 10:31:58 PM UTC 2023
2
3 ; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> www.example.com
4 ;; global options: +cmd
5 ;; Got answer:
6 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36773
7 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
8
9 ;; OPT PSEUDOSECTION:
10 ; EDNS: version: 0, flags:; udp: 1232
11 ; COOKIE: ef72612ecb1b76260100000065763c5e7e632f53fb5020c9 (good)
12 ;; QUESTION SECTION:
13 ;www.example.com.               IN      A
14
15 ;; ANSWER SECTION:
16 www.example.com.        259200  IN      A       10.0.0.7
```

log_5000_50000_6.txt
~/Downloads

```
1 Sun Dec 10 10:53:54 PM UTC 2023
2
3 ; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> www.example.com
4 ;; global options: +cmd
5 ;; Got answer:
6 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31686
7 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
8
9 ;; OPT PSEUDOSECTION:
10 ; EDNS: version: 0, flags:; udp: 1232
11 ; COOKIE: 57857e0005ffa33c0100000065764182b188aa5023ecf6bf (good)
12 ;; QUESTION SECTION:
13 ;www.example.com.               IN      A
14
15 ;; ANSWER SECTION:
16 www.example.com.        259200  IN      A       10.0.0.7
```

Part 05

Conclusion

UCDAVIS

Limitations and Future Work

Our work:

- We developed a DNS system on multiple virtual machines with fault tolerance;
- Our system includes features of automatic self-replication and an advanced mechanism of load balancing;
- In addition, we simulated an attack with a high volume of queries on our system, and the experiment result shows the robustness of our system in this scenario.

Future work:

- Optimize the load balancing algorithm and configuration to achieve better resilience;
- Test the results of the system more systematically given more resources;
- Consider other factors like data security.

UCDAVIS

Thank You