

《计算机网络第六次作业》

班级：信安 2302 班

学号：202308060227

姓名：石云博

目录

一. 问题描述.....	2
二. 问题分析.....	3
三. 实验过程及代码.....	5
四. 结论.....	19
参考文献.....	20

一.问题描述

- 1.找到尽可能多的 DNS 服务器(至少 10,000 个), 提供它们的 IP 和网络延迟的列表, 绘制一张反映它们分布的全球地图(将它们分类为递归、授权、顶级域和根), 使用现代工具如 echarts、d3 等。
- 2.查询它们对于著名网站如亚马逊、AWS、谷歌、百度、抖音等的响应, 分析它们的回应并告诉我们你的发现, 它们在地理上如何分布?它们是否共享相似的 IP?
- 3.尝试找到著名 CDNs 的 IP 范围, 从全球视角看。
- 4.尝试找到为 HNU 的学生服务的至少 10 个著名网站的服务器 IP, 它们中有多少个有本地服务器, 有多少个没有。

二.问题分析

首先第一个问题是为什么要有 DNS?

互联网中，一台计算机与其他计算机通信时，通过 IP 地址唯一的标志自己。此时的 IP 地址就类似于我们日常生活中的电话号码。但是，这种纯数字的标识是比较难记忆的，而且数量也比较庞大。例如，每个 IPv4 地址是一个 32 位长的二进制数字，或者采用点分十进制展示成 192.168.1.1 这种格式，有接近 43 亿个的 IPv4 地址。DNS 的作用就是将人类可读的名称转换为机器识别的 IP 地址，供计算机相互连接。DNS 的工作原理和电话簿相似，都是管理名称和数字之间的映射关系。就像我们日常打电话，一般使用人名查找，很少直接输入电话号码一样。当我们上网打开某个网页、视频时，也很少直接使用 IP 地址，而是在浏览器里输入的 URL 地址，例如：<https://www.huawei.com>，这其实使用的就是计算机的名字，一般称为域名。

再然后，我们来看看域名服务器的分类

(1) 根域名服务器

根域名服务器是最高层次的域名服务器，所有的根域名服务器都知道所有的顶级域名服务器的 IP 地址。

根域名服务器也是最重要的域名服务器，不管是哪个本地域名服务器，若要对因特网上任何一个域名进行解析，只要自己无法解析，就首先要求助于根域名服务器。

因特网上有 13 个根域名服务器，尽管我们将这 13 个根域名服务器中的每个都视为单个服务器，但每个“服务器”实际上是冗余服务器的集群，以提供安全性和可靠性。

需要注意的是，根域名服务器用来管辖顶级域(如.com)，通常它并不直接把待查询的域名直接转换成 IP 地址，而是告诉本地域名服务器下一步应当找哪个顶级域名服务器进行查询。

(2) 顶级域名服务器

这些域名服务器负责管理在该顶级域名服务器注册的所有二级域名。

收到 DNS 查询请求时,就给出相应的回答(可能是最后的结果,也可能是下一步应当查找的域名服务器的 IP 地址)。

(3) 授权域名服务器(权限域名服务器)

每台主机都必须在授权域名服务器处登记。为了更加可靠地工作，一台主机最好至少有两个授权域名服务器。

实际上，许多域名服务器都同时充当本地域名服务器和授权域名服务器。

授权域名服务器总能将其管辖的主机名转换为该主机的 IP 地址。

(4) 本地域名服务器

本地域名服务器对域名系统非常重要。

每个因特网服务提供者(ISP)，或一所大学，甚至一所大学中的各个系，都可以拥有一个本地域名服务器。

当一台主机发出 DNS 查询请求时，这个查询请求报文就发送给该主机的本地域名服务器。

事实上，我们在 Windows 系统中配置“本地连接”时，就需要填写 DNS 地址，这个地址就是本地 DNS (域名服务器)的地址。

三.实验过程及代码

第一个任务：找到尽可能多的 DNS 服务器(至少 10,000 个)，提供它们的 IP 和网络延迟的列表，绘制一张反映它们分布的全球地图(将它们分类为递归、授权、顶级域和根)。我们首先需要找到 DNS 服务器，这里可以使用 python 编写的代码来进行扫描，然后将结果进行过滤之后存入数据库中。

```
import requests
from bs4 import BeautifulSoup
import sqlite3

def fetch_public_dns(url="https://public-dns.info/nameservers.csv"):
    # 假设数据以 CSV 格式提供 (也可能是 HTML, 需要按实际情况调整)
    response = requests.get(url)
    if response.status_code == 200:
        # 将 CSV 按行分割, 并提取 IP 地址 (假设第一列为 IP)
        dns_list = []
        lines = response.text.strip().split("\n")
        for line in lines[1:]: # 跳过标题行
            cols = line.split(',')
            ip = cols[0].strip()
            dns_list.append(ip)
        return list(set(dns_list))
    else:
        print("数据获取失败, 状态码: ", response.status_code)
        return []

def save_to_db(dns_list, db_file="dns_servers.db"):
    conn = sqlite3.connect(db_file)
    cursor = conn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS dns_servers (ip TEXT PRIMARY KEY)")
    for ip in dns_list:
        try:
            cursor.execute("INSERT INTO dns_servers (ip) VALUES (?)", (ip,))
        except sqlite3.IntegrityError:
            continue # 已存在的跳过
    conn.commit()
    conn.close()

if __name__ == "__main__":
    dns_ips = fetch_public_dns()
    print(f"获取到 {len(dns_ips)} 个 DNS IP")
    save_to_db(dns_ips)
```

用于扫描 DNS 的程序

运行该程序，将扫描结果存入文件 dns_servers.db 中。

```
(kali㉿kali)-[~/桌面]
$ python3 DNS.py
获取到 62790 个 DNS IP
```

	ip
	过滤
1	201.87.50.78
2	195.19.215.132
3	172.64.37.160
4	172.64.36.116
5	217.109.132.150
6	195.9.134.178
7	169.255.36.58
8	187.44.0.18
9	88.221.162.14
10	45.65.172.76
11	98.161.229.109
12	68.57.76.207
13	12.131.180.85
14	94.158.157.142
15	92.198.57.36

可以看到一共扫描到了 62790 个 DNS IP，我们将他们存放到数据库文件中，方便后续处理。

存好 IP 之后，我们就要对其进行测速了，我们这里选择了使用 masscan 来进行测速，使用指令“udo masscan "0.0.0.0/0" -p53 --rate=1000 --exclude 255.255.255.255 -oL masscan_dns.txt ”来测速并将结果存放到文件“ masscan_dns.txt ”之中。

```
(kali㉿kali)-[~/桌面]
$ sudo masscan "0.0.0.0/0" -p53 --rate=1000 --exclude 255.255.255.255 -oL masscan_dns.txt

[sudo] kali 的密码:
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2025-04-04 08:20:03 GMT
Initiating SYN Stealth Scan
Scanning 4294967295 hosts [1 port/host]
Rate: 0.99-kpps, 0.00% done,1219:32:41 remaining, found=18
```

```
无法测速  
Processing IP: 187.141.99.33  
  延迟: 235.28 ms, 位置: (17.0665, -96.7226)  
Processing IP: 95.84.186.240  
无法测速  
Processing IP: 4.17.81.105  
无法测速  
Processing IP: 5.21.5.130  
  延迟: 252.74 ms, 位置: (23.578, 58.4021)  
Processing IP: 212.231.194.131  
无法测速
```

```
Processing IP: 103.155.18.225  
无法测速  
Processing IP: 103.145.164.221  
  延迟: 356.75 ms, 位置: (23.7754, 90.3954)  
Processing IP: 50.192.13.172  
  延迟: 271.7 ms, 位置: (40.3404, -76.7882)  
Processing IP: 175.144.214.180  
  延迟: 221.75 ms, 位置: (3.0763, 101.5168)  
Processing IP: 92.175.225.206  
无法测速
```

我们测试时候发现，其中有部分的 IP 我们无法测速，查找到的原因可能

1 有非开放递归解析

部分 DNS 服务器并非用于公共递归解析，而是只提供权威服务。因此，当你对一个随机的域名（例如 example.com）发起查询时，它们可能拒绝提供服务或直接不响应。

2 防火墙或安全策略

为防止滥用，有些 DNS 服务器会在防火墙或安全设置中屏蔽来自未知源的 DNS 查询请求，从而导致测速失败。

3 超时时间设置较短

如果网络环境较差或者目标服务器响应较慢，2 秒的超时时间可能不足以等待到完整的响应。可以尝试延长超时时间，看是否能改善测速成功率。

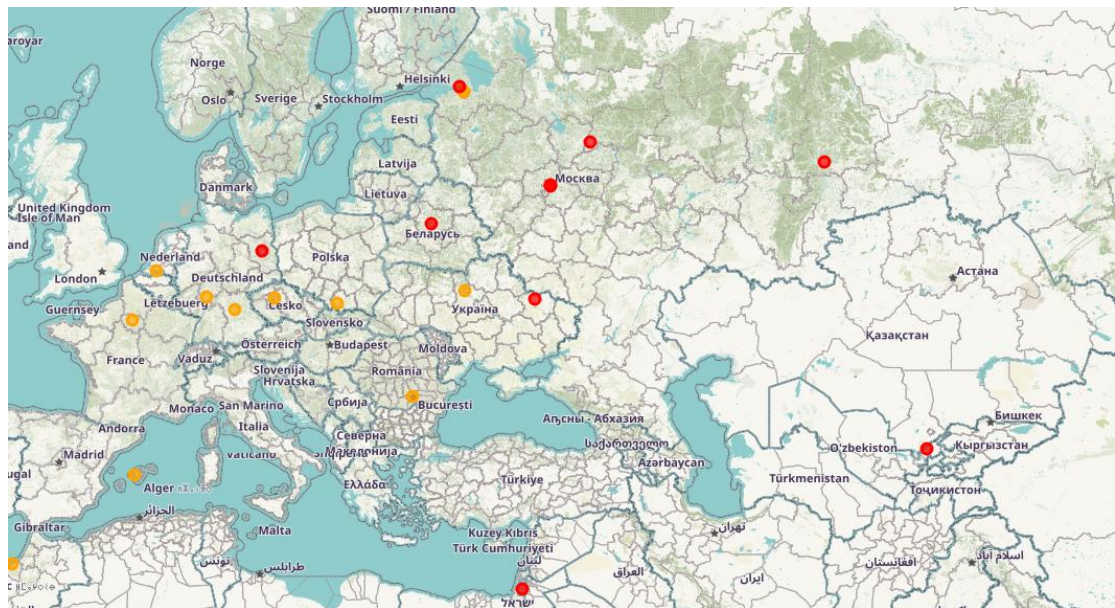
4 数据库中存在无效或不可达 IP

数据库中收集的 IP 地址可能有部分已经失效、配置错误或只是在特定网络环境下才能访问，导致测速失败。

5 使用的测试域名问题

使用同一个域名 (如 example.com) 进行测速, 并不一定适用于所有类型的 DNS 服务器。有些服务器可能仅对其管理的域名进行响应, 而对其他域名则拒绝解析。

扫描完成之后, 我们打开生成的地图文件:



可以看到根据不同的颜色来对节点进行了分类。

接下来是第二个任务: 查询它们对于著名网站如亚马逊、AWS、谷歌、百度、抖音等的响应, 分析它们的回应并告诉我们你的发现, 它们在地理上如何分布? 它们是否共享相似的 IP? 对此我们可以编写程序, 读取数据库中存储的 IP, 并且将其作为 DNS 服务器来找到网站对其的响应, 我们这里使用了百度和抖音两个网站进行测试。


```

def main():
    # 配置参数
    DB_PATH = "dns_servers.db"

    try:
        # 加载随机样本
        dns_servers = load_dns_servers(DB_PATH)
        print(f"成功加载{len(dns_servers)}个DNS服务器")

        # 执行批量查询
        results = {'baidu': [], 'douyin': []}
        with ThreadPoolExecutor(max_workers=20) as executor:
            for domain in ['baidu', 'douyin']:
                futures = {
                    executor.submit(dns_query, s, f"{domain}.com"): s
                    for s in dns_servers[:500] # 确保只取500个
                }
                for future in futures:
                    server = futures[future]
                    ips = future.result()
                    asn_info = {ip: get_asn_info(ip) for ip in ips}
                    results[domain].append({
                        'server': server,
                        'ips': ips,
                        'asn_info': asn_info
                    })

        # 打印结果
        print("\n分析结果:")
        print("百度TOP ASN分布:")
        for asn, count in report['asn_distribution']['baidu'].items():
            print(f"AS{asn}: {count}次")

        print("\n抖音TOP ASN分布:")
        for asn, count in report['asn_distribution']['douyin'].items():
            print(f"AS{asn}: {count}次")

        print("\n相似性分析:")
        print(f"共享ASN数量: {report['similarity']['common_asn_count']}")
        print(f"共享C类地址数量: {report['similarity']['common_c_class_count']}")
        if report['similarity']['common_asn_example']:
            print("示例共享ASN:", report['similarity']['common_asn_example'])

    except Exception as e:
        print(f"运行失败: {str(e)}")

```

我们运行该文件，可以得到结果：

```
===== 检测报告 =====
总测试 IP 数: 500
百度解析成功率: 4.20%
抖音解析成功率: 2.20%

地理分布 TOP5:
United States: 83个
Unknown: 76个
Russia: 54个
France: 21个
Thailand: 16个

常见 IP 段 TOP3:
45.90.28.0/24: 4个 IP
45.90.30.0/24: 3个 IP
103.168.90.0/24: 2个 IP
```

第三个任务：尝试找到著名 CDNs 的 IP 范围，从全球视角看。我们从网上搜集资料得知，我们能够有几种方式来解决该问题。

Cloudflare

Cloudflare 官方提供了两个页面列出其 IPv4 与 IPv6 地址范围，分别用于代理流量和其他服务。其官方链接为：<https://www.cloudflare.com/ips>

IP 地址范围

最后更新：2023 年 9 月 28 日

一些应用程序或主机提供商可能会发现了解 Cloudflare 的 IP 地址很有用。本页面旨在成为 Cloudflare 当前 IP 地址范围的权威来源。

您也可以使用 [Cloudflare API](#) 访问此列表

IPv4	IPv6
103.21.244.0/22	2400:cb00::/32
103.22.200.0/22	2606:4700::/32
103.31.4.0/22	2803:f800::/32
104.16.0.0/13	2405:b500::/32
104.24.0.0/14	2405:8100::/32
108.162.192.0/18	2a06:98c0::/29
131.0.72.0/22	2c0f:f248::/32

Amazon CloudFront

Amazon 将所有服务的 IP 地址范围统一发布在一个 JSON 文件中。

我们可以在下载地址 <https://ip-ranges.amazonaws.com/ip-ranges.json> 中得到该

文件。

```

{
  "syncToken": "1743803002",
  "createDate": "2025-04-04-21-43-22",
  "prefixes": [
    {
      "ip_prefix": "3.4.12.4/32",
      "region": "eu-west-1",
      "service": "AMAZON",
      "network_border_group": "eu-west-1"
    },
    {
      "ip_prefix": "3.5.140.0/22",
      "region": "ap-northeast-2",
      "service": "AMAZON",
      "network_border_group": "ap-northeast-2"
    },
    {
      "ip_prefix": "15.190.244.0/22",
      "region": "ap-east-2",
      "service": "AMAZON",
      "network_border_group": "ap-east-2"
    },
    {
      "ip_prefix": "15.230.15.29/32",
      "region": "eu-central-1",
      "service": "AMAZON",
      "network_border_group": "eu-central-1"
    },
    {
      "ip_prefix": "15.230.15.76/31",
      "region": "eu-central-1",
      "service": "AMAZON",
      "network_border_group": "eu-central-1"
    },
    {
      "ip_prefix": "15.230.221.0/24",
      "region": "us-east-1",
      "service": "AMAZON",
      "network_border_group": "us-east-1"
    },
    {
      "ip_prefix": "35.180.0.0/16",
      "region": "eu-west-3",
      "service": "AMAZON",
      "network_border_group": "eu-west-3"
    },
  ],

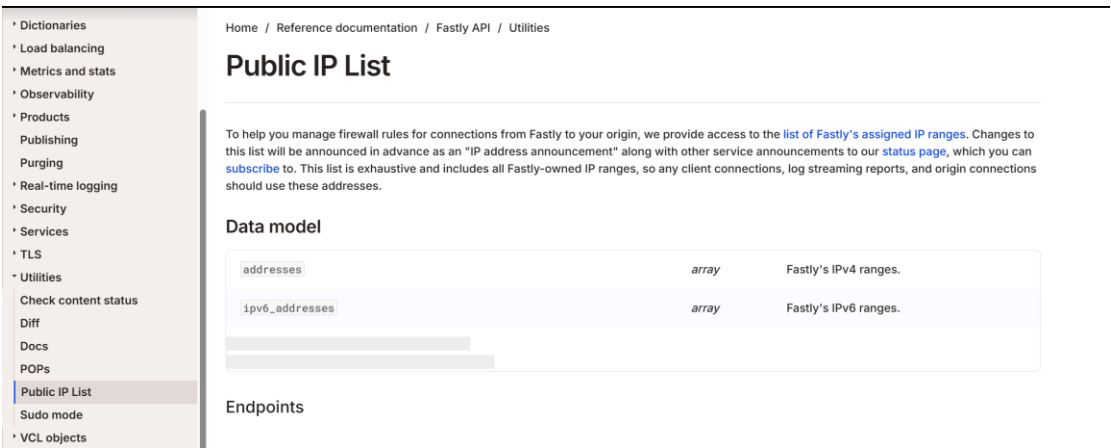
```

在该文件中，筛选 "service": "CLOUDFRONT" 的部分即可获得 CloudFront 全球边缘节点的 IP 范围。

Fastly

Fastly 提供专门的文档页面来查看其全球边缘 IP 地址范围。参考链接为：

<https://www.fastly.com/documentation/reference/api/utls/public-ip-list>。













此外，GitHub 上也有第三方项目整理了 Fastly 的 IP 列表，可供进一步参考。

Akamai

Akamai 的 IP 范围较为分散，并且并没有一个统一的官方文件。可参考第三方站点例如 networksdb.io 来获取 Akamai 全球公开网络的 CIDR 列表。

Top Organizations

Top 10 largest companies according to the number of IPv4 addresses they own or operate. ([see more](#))

	DoD Network Information Center	(222, 217, 216)
	AT&T Corp	(99, 153, 376)
	Amazon.com, Inc	(95, 518, 326)
	Verizon Business	(91, 379, 693)
	Comcast Cable Communications, Inc	(88, 754, 308)
	Microsoft Corporation	(81, 051, 677)
	Amazon Technologies Inc	(77, 227, 561)
	China Unicom	(68, 342, 038)
	Charter Communications Inc	(63, 635, 366)
	Deutsche Telekom A.G.	(56, 104, 505)

也可以利用 RADB 或 RIPE 等公共数据库，通过 ASN（如 AS12222、AS16702 等）查询其 IP 范围。

Google Cloud CDN

Google 的 CDN 服务通常嵌入在其云服务中，Google 会发布供客户使用的外部 IP 范围。相关信息可在 Google Cloud 文档中找到，或通过查询 TXT 记录（例如 `_cloud-netblocks.googleusercontent.com`）获得最新数据。

第四个任务：尝试找到为 HNU 的学生服务的至少 10 个著名网站的服务器 IP。

我们在校园网环境下，使用 `nslookup` 工具来对网站进行测试，这里我们可以选择谷歌，百度，抖音，b 站，等等网站来进行测试：

```
C:\Users\lian ton>nslookup www.google.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      www.google.com
Addresses:  2001::1
            31.13.68.169
```

“非权威应答”意味着返回的数据并非直接来自 `www.google.com` 的权威 DNS 服务器，而是由中间的缓存服务器提供的答案。返回的“名称: `www.google.com`”表示查询目标正确。“Addresses: `2001::1`”和“`31.13.68.169`”分别是该域名解析出的一个 IPv6 地址和一个 IPv4 地址。

```
C:\Users\lian ton>nslookup www.baidu.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      www.a.shifen.com
Addresses:  2409:8c54:870:310:0:ff:b0ed:40ac
            2409:8c54:870:187:0:ff:b0d9:bb1c
            183.240.99.169
            183.240.99.58
Aliases:   www.baidu.com
```

对于百度：输出中没有显示任何属于本地回环地址（如 `127.0.0.1`）或常见私有地

址（如 192.168.x.x、10.x.x.x）的记录，所有返回的 IP 均为百度在国内常见的公共 IP。因此，从这个结果看，并没有迹象表明有额外的本地服务器在参与解析，而只是使用了网络中默认 DNS 服务器。

```
C:\Users\lian ton>nslookup www.douyin.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:     hcdndyn.zjv6.c.cdnhwc2.com
Addresses: 2409:8c50:800:13:1800::36
           2409:8087:5050:10::36
           2409:8087:5050:10::38
           2409:8c50:800:13:1800::34
           2409:8087:50a0:10::54
           2409:8087:50a0:10::55
           2409:8087:50e0:10::2035
           2409:8087:50a0:10::52
           2409:8087:5050:10::37
           2409:8087:50d0:10::1033
           2409:8087:50e0:10::2034
           2409:8087:50d0:10::1034
           39.136.191.59
           39.136.189.35
           111.23.4.53
           39.136.191.61
           39.136.186.33
           111.23.4.52
           39.136.189.31
           39.136.191.63
           39.137.71.63
           39.137.71.61
           39.136.186.31
           39.137.71.62
```

对于抖音：查询 www.douyin.com 得到的“名称”为 hcdndyn.zjv6.c.cdnhwc2.com，这说明 www.douyin.com 实际上使用了 CNAME 重定向，指向了另一个域名，该域名可能代表了 Douyin 的 CDN 节点。

列出的“Aliases”中还包含 www.douyin.com.bytedns1.com 以及 www.douyin.com.51296c5e.c.cdnhwc2.com，进一步说明 DNS 解析过程中存在

多级别的别名重定向, 常用于负载均衡和流量调度。这也符合大多数大型网站(如 Douyin) 使用全球 CDN 的常规做法, DNS 解析返回的是离用户最近的全球节点, 而非某个局部内网服务器。

```
C:\Users\lian ton>nslookup www.bilibili.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      a.w.bilicdn1.com
Addresses:  2409:8c28:202:8::203
            2409:8c38:c40:100::2
            2409:8c4c:c00:9::23
            2409:8c38:c40:100::3
            2409:8c4c:c00:9::22
            2409:8c38:c40:100::241
            2409:8c38:c40:100::242
            2409:8c38:c40:100::243
            111.48.57.46
            112.13.92.195
            111.48.57.45
            223.111.252.67
            223.111.249.190
            112.13.92.199
            112.13.92.203
            117.169.96.199
            223.111.250.53
            36.158.237.14
            36.158.237.15
            117.169.96.198
            112.13.92.202
            36.158.237.17
            117.169.98.21
```

结果中列出了多个 IPv6 地址 (以 2409 开头) 和多个 IPv4 地址 (例如 111.48.x.x、112.13.x.x、117.169.x.x、36.158.237.x、223.111.x.x 等), 这些都是 bilibili 用于内容分发的公共 IP。多个 IP 地址的返回意味着 bilibili 使用了全球分布的 CDN 节点, 通过 Anycast 或负载均衡技术, 将用户请求引导到最近的节点, 从而提高访问速度和可靠性。


```
C:\Users\lian ton>nslookup www.taobao.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      www.taobao.com.danuoyi.tbcache.com
Addresses:  2409:8c50:a00:201f:3::7d5
            2409:8c50:a00:201f:3::7d4
            120.226.78.123
            120.226.78.122
Aliases:   www.taobao.com
```

类似上面的内容，这里的 IP 地址都是属于阿里云 CDN 的公网节点，并非位于本地网络（如 192.168.x.x、10.x.x.x、127.0.0.1 这类内网地址）。这说明访问 www.taobao.com 的内容时，会由最近的阿里 CDN 边缘节点提供服务，但仍然是公网的一部分。

```
C:\Users\lian ton>nslookup www.jd.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      wwwv6.jcloudimg.com
Addresses:  2409:8c50:a00:2212:8000::193
            120.226.8.193
Aliases:   www.jd.com
            www.jd.com.gslb.qianxun.com
            www.jd.com.s.galileo.jcloud-cdn.com
```

类似上面，这些地址都是公网地址，并且属于京东的 CDN 网络，并不是你本地网络内部的服务器。

```
C:\Users\lian ton>nslookup www.pinduoduo.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      www.pinduoduo.com
Addresses:  2402:4e00:1400:4600:0:8f53:529:b7ae
            150.158.224.195
```

依旧和上面的内容差不多。

```
C:\Users\lian ton>nslookup www.qq.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      ins-r23tsuuf.ias.tencent-cloud.net
Addresses:  2409:8c54:871:2::34
            2409:8c54:871:1::73
            112.53.42.114
            112.53.42.52
Aliases:   www.qq.com
```

综合来看, 该 nslookup 结果表明 www.qq.com 通过 CNAME 指向了腾讯云的节点, 并返回了多个公共 IPv4/IPv6 地址, 用于全球访问。这是一种常见的 CDN 或负载均衡策略, 并未显示有本地服务器参与解析。

```
C:\Users\lian ton>nslookup www.weixin.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      sznewcomm.weixin.qq.com
Addresses:  240e:97c:2f:1::3c
            240e:97c:2f:1::7
            120.233.21.30
            120.233.18.43
Aliases:   www.weixin.com
            newcomm.weixin.qq.com
```

返回的名称为 sznewcomm.weixin.qq.com。这表明 www.weixin.com 实际上是一个 CNAME 别名, 其真实服务名称为 newcomm.weixin.qq.com (别名列表中同时显示了 newcomm.weixin.qq.com), 这是微信用于负载均衡或流量调度的常见做法。这些都是公共 IP 地址, 表明微信服务采用的是全球 (或区域) 分布的公共服务器, 而非本地或私有网络中的服务器。

```
C:\Users\lian ton>nslookup www.weibo.com
服务器:  UnKnown
Address:  202.197.96.100

非权威应答:
名称:      weibo.com
Addresses:  2409:8c00:7821:2303::56
            2409:8c00:8421:1303::54
            2409:8c00:8421:1303::55
            2409:8c00:7821:2303::57
            111.13.134.130
            111.13.134.131
            111.13.225.83
            111.13.225.75
Aliases:   www.weibo.com
```

输出中仅显示了公共的 IPv6/IPv4 地址，没有局域网地址（如 127.0.0.1、192.168.x.x 等），因此这说明解析结果指向的是微博的公共服务节点，而非本地或内部网络的服务器。

就我们测试的这几个网站的结果而言，大部分是没有本地服务器的，返回的基本都是其公共服务节点。

四.结论

本次实验首先进行了问题分析，根据题目分别对“递归、授权、顶级域和根服务器”、“CDN”、和“本地服务器”这三个题目中提到的名词进行了大致的解释根据查阅的资料给出了它们的大致含义。

然后，在实验过程中，首先选择通过公开的 DNS 服务器列表的网站找到了 10000 多个 DNS 服务器的 IP 地址，然后编写脚本收集 DNS 服务器的相关信息，包括 DNS 服务器的网络延迟和其 IP 地址的经纬度等。随后再使用 Python 脚本

将 DNS 服务器分类, 并使用 echarts 绘制一张反映 DNS 服务器分布的全球地图保存在 HTML 文件里。

接着, 使用 nslookup 命令来查询 DNS 服务器对于著名网站如百度、抖音的响应, 并分析了响应在地理上如何分布以及它们是否共享相似的 IP。分析的结果是响应在地理上通常会指向最近的 CDN 节点, 有些响应确实会共享相似的 IP 地址, 但这些 IP 地址背后的服务器可能分布在不同的地理位置。

此外, 通过查阅官方资料等操作, 找到了著名 CDNs 如:Cloudflare、Amazon CloudFront、Fastly、Microsoft Azure CN 的 IP 地址范围。

最后, 选择谷歌、百度、抖音、b 站、淘宝、京东、拼多多、QQ、微信、微博这十个为 HNU 学生服务的著名网站。连接校园网, 使用 nslookup 命令查询得到了它们的服务器 IP 地址。通过测试发现大部分都是返回的公共 IP 地址。

参考文献

1. <https://www.zhihu.com/question/64037117>. 本地服务器是什么? . qcbxy 2023.11.25.

[2] <https://info.support.huawei.com/info-finder/encyclopedia/zh/DNS.html> 什么是 DNS? . 豆翠翠. 2024.11.14.

[3] https://blog.csdn.net/weixin_43914604/article/details/105583806. 6.2 详解 DNS 域名解析系统 (域名、域名服务器[根、顶级、授权/权限、本地]、域名解析过程[递归与迭代]) . BitHachi.2021.01.24.

[4]<https://www.fastly.com/documentation/reference/api/utils/public-ip-list/>. Public IP List.2023.8.15.

[5] <https://blog.csdn.net/glmichelin/article/details/113186020>. nslookup 命令详解和实战例子（全）. 老鸟诗人.2022.3.24

[6] <https://docs.pingcode.com/ask/318225.html>. Edit2. 什么是 CND，有哪些作用. 2024.9.16.

[7] https://blog.csdn.net/python2021_/article/details/123652555. Python 绘制地图神器 folium 介绍及安装使用教程. python2021_. 2022.8.15.