

南京大学本科生实验报告

课程名称：计算机网络 任课教师：田臣/李文中 助教：方毓楚、郑浩、陈伟等（排名不分先后）

学院	地理与海洋科学学院	专业	地理信息科学
学号	191830173	姓名	徐嵩
Email	song.xv@outlook.com	开始/完成日期	2022/3/24

1. 实验名称

Lab 7: Content Delivery Network

2. 实验目的

通过 OpenNetLab 构建部署一个内容分发网络（CDN）。

3. 实验内容

- A. 构建 DNS Server，包括 Remote DNS Server 返回 CNAME 记录，以及 CDN DNS Server 返回基于地理位置的最佳服务器 IP
- B. 生成 ICMP 错误信息

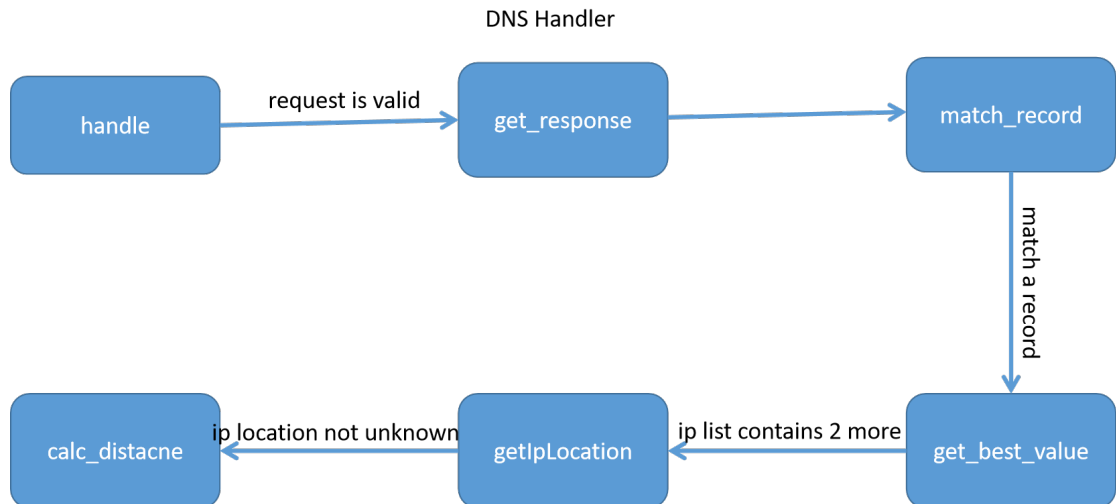
4. 实验结果

Show how you implement the features of DNS server.

在读取 DNS table 时统一在 domainName 末尾处加“.”便于处理，如下图所示：

```
def parse_dns_file(self, dns_file):
    # -----
    # TODO: your codes here. Parse the dns_table.txt file
    # and load the data into self._dns_table.
    # -----
    with open(dns_file, 'r') as f:
        lines = f.readlines()
    for line in lines:
        domainName = line.strip().split()[0]
        recordType = line.strip().split()[1]
        recordValues = line.strip().split()[2:]
        if domainName[len(domainName) - 1] != '.':
            domainName = domainName + '.'
        record = [domainName, recordType, recordValues]
        self._dns_table.append(record)
```

response 实现逻辑如下图所示：



Show how you implement the features of caching server.

当 http request 到达时，caching server 调用 handler 的 do_GET()方法，方法如下图所示：

```

@trace
def do_GET(self):
    ''' Logic when receive a HTTP GET.
    Notice that the URL is automatically parsed and the path is stored in self.path.
    '''
    # TODO: implement the logic to response a GET.
    # Remember to leverage the methods in CachingServer.
    Item = self.server.touchItem(self.path)
    if Item is not None:
        self.sendHeaders(Item[0])
        self.sendBody(Item[1])
    else:
        self.send_error(HTTPStatus.NOT_FOUND, "File not found")
  
```

Handler 调用 cachingServer 的 touchItem 方法，返回 None 表示资源不存在，否则返回 headers 和 body 元组，当资源不存在 cache 中时，调用 requestMainServer 方法从主服务器获取 response，如下图所示：

```

def touchItem(self, path: str):
    ''' Touch the item of path.
    This method, called by HttpHandler, serves as a bridge of server and
    handler.
    If the target doesn't exist or expires, fetch from main server.
    Write the headers to local cache and return the body.
    ...
    # TODO: implement the logic described in doc-string
    if path not in self.cacheTable or self.cacheTable.expired(path):
        response = self.requestMainServer(path)
        if response is None:
            return None
        headers = response.getheaders()
        body = response.read()
        self.cacheTable.setHeaders(path, headers)
        self.cacheTable.appendBody(path, body)
        return [headers, body]
    else:
        headers = self.cacheTable.getHeaders(path)
        body = self.cacheTable.getBody(path)
        return [headers, body]

    return None

```

Show how much the CDN cache shortens the request time. Write the procedure and analysis in your report with screenshots.

在第一次请求时，Cache Server 的 cache 未命中，于是向 main Server 发送 request，main server 返回后缓存在 cache 中，并向 client 发送 response，耗时 461ms。

第二次请求相同的资源 '/doc/success.jpg' 直接从 cache server 返回，用时 2ms。

第三次请求 cache 不存在的资源，main Server 返回 None，主服务器中也不存在，向 client 发送 404 response，耗时 459ms。测试结果如下：

Client

[Request time] 461.13 ms

[Request time] 2.60 ms

[Request time] 459.96 ms

Cache Server

2022/05/22-11:50:58| [Info] Fetched '/doc/success.jpg' from main server '20.188.122.123:8888'

2022/05/22-11:50:58| [From 10.0.0.24:38076] "GET /doc/success.jpg HTTP/1.1" 200 -

2022/05/22-11:50:59| [From 10.0.0.24:38078] "GET /doc/success.jpg HTTP/1.1" 200 -

2022/05/22-11:51:00| [Error] File not found on main server '20.188.122.123:8888'

2022/05/22-11:51:00| [From 10.0.0.24:38080] code 404, message 'File not found'

2022/05/22-11:51:00| [From 10.0.0.24:38080] "GET /noneexist HTTP/1.1" 404 -

```
DNS Server
```

```
[Info] Receiving DNS request from '10.0.0.24' asking for 'stfw.localhost.computer.'
```

```
[Info] Receiving DNS request from '10.0.0.24' asking for 'stfw.localhost.computer.'
```

```
[Info] Receiving DNS request from '10.0.0.24' asking for 'stfw.localhost.computer.'
```

5. 核心代码

6. 总结与感想

通过实验 7, 初步体会了分布式服务网络的原理和工作方式。尝试了实现 Stream Forwarding, 大概知道应该在 http header 添加 Content-Range 字段, 但是是一些 python 语法细节不清楚, 实现遇到了问题, 最后放弃。