第十一届中国软件杯
大学生软件设计大赛
测试文档

队伍名称：泥头车队

队伍编号：62012196

参赛题目：A1-高性能民航旅客行程推荐系统

队伍队长：王凯

联系电话：18272023062

测试人员：王凯、范启航、叶承荣

测试时间：2022/8/10、2022/8/11

目录

一、测试用例与测试操作说明

测试版本为 8.10 最新优化后的作品（**主要优化是自研 HTTP 服务器**）。

本软件作品测试工作分为两个部分，分别是基本功能测试、性能测试。

（1）基本功能测试

主要根据赛题描述的基本功能要求，验证作品是否满足，主要是针对搜索功能和余座更新功能的正确性检验。

1）测试方法：

浏览器输入域名 hustairline.xyz，直接访问已部署的作品，调用搜索功能和余座更新功能。测试过程中，对于单航段输入，通过**人工方式**对数据文件（几个 txt 文件）使用查找功能逐一验证得到预期结果，再与网站中得到的结果比对。对于复杂的多航段输入，将之前单航段输入得到的结果组合排序，得到正确的票价序列，再与搜索结果比对，一致则进一步对信息进行比对。对于多乘客输入，先得到对应的单乘客输入的结果，在航班余座均足够且舱位变动不大的情况下，多乘客得到的行程推荐方案与单乘客相差不多，可以此为依据来加快验证。

2）测试用例：

以下测试用例中，记输入请求为 Request，乘客人数为 N，搜索结果为 Ans，No 为航班号。

1、单航段，单旅客，单代理，且最大返回结果数设为 1

```
Request 1:
                代理：DNH
                航段数：1
                N：1
                航段：AOG->BHY          2022-9-12
                Ans:
                                Time:4:00->5:15
                                No:GS8302
                                Price:960
Request 2:
                代理：WUX
                航段数：1
                N：1
                航段：BSD->PEK          2022-9-14
```

Ans:
Time:8:45->11:30
No:G44715
Price:1460

Request 3:
代理：NNG
航段数：1
N：1
航段：BHY->AOG 2022-9-19
Ans:
Time:10:15->13:30
No:HO1964
Price:1010

Request 4:
代理：SHS
航段数：1
N：1
航段：BFU->AVA 2022-10-07
Ans:
Time:16:00->18:00
No:XQ7345
Price:3690

Request 5:
代理：WUX
航段数：1
N：1
航段：CKG->CSX 2022-10-31
Ans:
Time:7:45->11:45
No:CA6352
Price:2150

2、单乘客，多航段多代理，且设置最大返回结果数

Request 1:
代理：全选
航段数：2
N：1
航段：
AQG->CGQ 2022-10-11
CGQ->BHY 2022-10-28
最大返回数：3

```
 Ans：AnsNum = 1
                方案 1：
                        flight 1:HO9269
                        flight 2:HO5830
                Price：6140

Request 2:
代理：全选
航段数：3
N：1
航段：
                  AEB->BSD 2022-12-01
                  AVA->AEB 2022-12-02
                  AVA->PEK 2022-12-03
最大返回数：3
Ans: AnsNum = 3
                方案 1：
                                flight 1:XQ3975
                                flight 2:HO9953
                                flight 3:3Q2239
                Price:11770

                方案 2：
                                flight 1: G52665
                                flight 2:HO9953
                                flight 3:3Q2239
                Price:11970

                方案 3：
                                flight 1:G52665
                                flight 2: HO9953
                                flight 3: 3Q2239
                Price:12090

Request 3:
代理：全选
航段数：2
N：1
航段：
                DLU->HNY 2022-12-29
                CGD->FUO 2023-01-03
最大返回数：4
Ans: AnsNum = 2
                方案 1：
```

```
                                flight 1: MF4584
                                flight 2:JD4337
                Price:2030

                方案 2：
                                flight 1:MF4584
                                flight 2:JD4337
                Price:2050

Request 4:
代理：全选
航段数：2
N：1
航段：
                BAV->AQG 2023-1-18
                CGQ->BHY 2023-3-14
最大返回数：4
Ans: AnsNum = 3
                方案 1：
                                flight 1: SC6286
                                flight 2:G44610
                Price:7000

                方案 2：
                                flight 1:SC6286
                                flight 2:G44610
                Price:7020

                方案 3：
                                flight 1:SC6286
                                flight 2:G44610
                Price:7080
```

3、多乘客，多代理人，多航段（乘客座位均为最优选择，结合余座数据方便验证，此处不展示）

```
Request 1:
代理：全选
航段数：2
N：3
航段：
        HMI->CZX 2023-2-8
        AVA->BSD 2023-2-10
```

最大返回数：4
Ans: AnsNum = 4
方案 1：
flight 1:HO1860
flight 2:XQ8871
Price:10040

方案 2：
flight 1:HO1860
flight 2:XQ8871
Price:10100

方案 3：
flight 1:HO1860
flight 2:X24835
Price:11330

方案 4：
flight 1:HO1860
flight 2:X24835
Price:11390
响应时间：79ms

Request2:
代理：全选
航段数：3
N：3
航段：
HMI->CZX 2023-2-8
AVA->BSD 2023-2-10
AEB->AVA 2023-2-13
最大返回数：4
Ans: AnsNum = 4
方案 1：
flight 1:HO1860
flight 2:XQ8871
flight 3:GJ9767
Price:17840

方案 2：
flight 1:HO1860
flight 2:XQ8871
flight 3:GJ9767
Price:17930

方案 3：
flight 1:HO1860
flight 2:X24835
flight 3:GJ9767
Price:19130

方案 4：
flight 1:HO1860
flight 2:X24835
flight 3:GJ9767
Price:19220
响应时间：85ms

Request 3:
代理：全选
航段数：4
N：3
航段：
BAV->CGQ 2023-7-7
BHY->AOG 2023-7-8
BSD->BFU 2023-7-9
PEK->CGD 2023-7-10
Ans: AnsNum = 4
方案 1：
flight 1:MU1849
flight 2:GS4973
flight 3:PN4113
flight 4:HU2119
Price:24750

方案 2：
flight 1:MU1849
flight 2:GS4973
flight 3:PN4113
flight 4:HU2119
Price:24870

方案 3：
flight 1:G53144
flight 2:GS4973
flight 3:PN4113
flight 4:HU2119
Price:25830

方案 4：
flight 1:MU1849
flight 2:SC4732
flight 3:PN4113
flight 4:HU2119
Price:25740
响应时间：87ms

Request 4:
代理：全选
航段数：2
N：3
航段：
BAV->CGQ 2023-8-17
BHY->AOG 2023-8-30
Ans: AnsNum = 4
方案 1：
flight 1:X28033
flight 2:ZH6796
Price:8040

方案 2：
flight 1:X28033
flight 2:ZH6796
Price:8100

方案 3：
flight 1:X28033
flight 2:ZH6796
Price:8280

方案 4：
flight 1:X28033
flight 2:EU9195
Price:9060
响应时间：91ms

4、余座更新功能

任选两个更新文件对余座数据进行更新，然后通过文本查找验证更新成功。

1）

航班号：9C4542

更新文件：Update4.txt

更新前后：9,4,8→A,A,5

图 1-1 例 1 更新前后对比

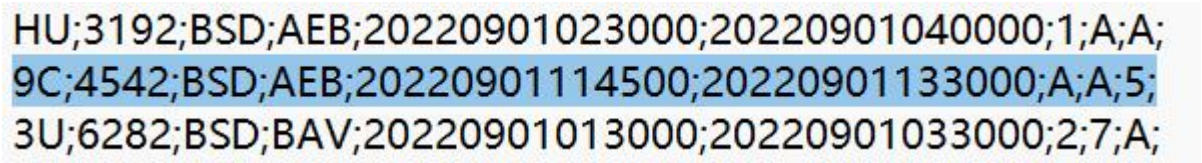

图 1-2 Update4.txt 文本查找验证

2）

航班号：HU6197
更新文件：Update2.txt
更新前后：2,A,8→9,8,9

图 1-3 例 2 更新前后对比

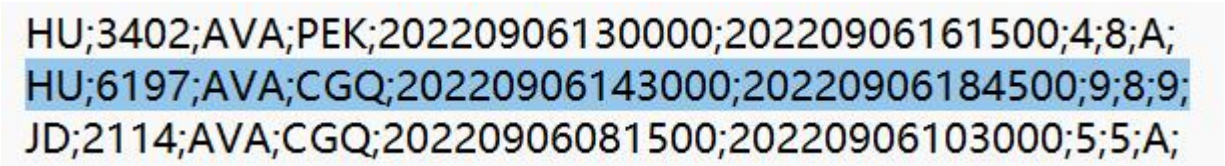

图 1-4 Update2.txt 文本查找验证

（2）性能测试

主要根据赛题描述的非功能性要求，测试作品的相关性能。

1）测试方法：

性能测试主要分为搜索性能测试和阻塞测试。

搜索性能测试主要利用 postman、jmeter 等工具向服务器发包，与此同时在服务器使用 top 指令检测相关指标，重点关注响应时间、CPU idle 值等数据。

搜索过程阻塞测试通过 python 多线程操作，在短时间内多次发包给后端，根据返回时间验证后端搜索过程中无阻塞产生。

更新余座数据阻塞测试通过 python 多线程操作，同时发送更新数据与搜索包给后端，根据返回时间验证更新数据不会阻塞搜索过程。

2)测试用例：

1、单次搜索性能

利用 postman 发包，考察搜索响应时间。

a、简单测试用例（单航段）：平均响应时间 38ms

图 1-5 简单测试用例响应时间

b、一般测试用例（三航段）：平均响应时间 40ms

图 1-6 一般测试用例响应时间

c、复杂测试用例（六航段以上）：平均响应时间 80ms

图 1-7 复杂测试用例响应时间

2、搜索请求并发测试

利用 jmeter 开启两个线程连续向服务器发送多个请求包，使用 top 指令监控 CPU idle 值。

测试条件如下：

jmeter 配置：

图 1-8　jmeter 并发配置

使用数据:

{"N":5,"M":8,"Max":40,"agency":["SHS001","XIC001","WUX001","SHE001","NNG001","DNH001","SZX001","AOG001","XUZ001","CGQ001"],"date":["20220901000000","20220902000000","20220903000000","20220904000000","20220905000000","20220906000000","20220907000000","20220908000000"],"sCity":["AQG","PEK","BAV","BHY","BFU","AQG","BHY","PEK"],"dCity":["AEB","AQG","BHY","PEK","AVA","BHY","AEB","AQG"]}

测试结果：CPU idle 值普遍低于 20%

```
%Cpu(s): 73.3 us,  2.0 sy,  0.0 ni, 23.8 id,  0.0 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu(s): 75.0 us,  0.0 sy,  0.0 ni, 25.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 76.5 us,  3.9 sy,  0.0 ni, 19.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 87.1 us,  0.0 sy,  0.0 ni, 12.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 83.0 us,  0.0 sy,  0.0 ni, 17.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 76.5 us,  1.0 sy,  0.0 ni, 22.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 77.8 us,  0.0 sy,  0.0 ni, 22.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 78.4 us,  1.0 sy,  0.0 ni, 19.6 id,  0.0 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu(s): 80.0 us,  0.0 sy,  0.0 ni, 20.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 81.4 us,  2.0 sy,  0.0 ni, 16.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 87.3 us,  0.0 sy,  0.0 ni, 12.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 83.0 us,  1.0 sy,  0.0 ni, 16.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 75.0 us,  1.0 sy,  0.0 ni, 24.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 79.2 us,  0.0 sy,  0.0 ni, 19.8 id,  0.0 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu(s): 81.2 us,  0.0 sy,  0.0 ni, 18.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 93.1 us,  0.0 sy,  0.0 ni,  6.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu(s): 78.4 us,  1.0 sy,  0.0 ni, 20.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

图 1-9 cpu 监控数据

3、阻塞测试

a. 搜索过程阻塞测试代码如图 1-10：

```python
import time
import requests
import json
from concurrent.futures import ThreadPoolExecutor, ProcessPoolExecutor  # 线程池，进程池
import threading


# 测试函数
def query_test(i):
    url = 'http://api.hustairline.xyz/api/query'
    data = """{ "N": 2, "agency":["DNH001"],"M":4,
    "date":["20220621000000","20220622000000","20220623000000","20220624000000"],
    "sCity":["CIF","BSD","DLU","BHY"],
    "dCity":["BSD","DNH","AVA","DLU"]}"""
    start = time.ctime()
    response = requests.post(url, data=data)
    end = time.ctime()
    print('test {} start at {}, end at {}'.format(i, start, end))
if __name__ == '__main__':
    thread_pool = ThreadPoolExecutor(5)  # 定义5个线程执行此任务
    for i in range(20):
        thread_pool.submit(query_test, i)   # 测试20组数据
```

图 1-10 搜索过程阻塞测试代码

测试结果如图 1-11，多次请求在同一时间返回，证明**搜索过程无阻塞**，可实现并发。

```
$ python .\query_test.py
test 0 start at Wed Jul  6 11:13:20 2022, end at Wed Jul  6 11:13:21 2022
test 2 start at Wed Jul  6 11:13:20 2022, end at Wed Jul  6 11:13:21 2022
test 1 start at Wed Jul  6 11:13:20 2022, end at Wed Jul  6 11:13:21 2022
test 5 start at Wed Jul  6 11:13:21 2022, end at Wed Jul  6 11:13:22 2022
test 6 start at Wed Jul  6 11:13:21 2022, end at Wed Jul  6 11:13:22 2022
test 3 start at Wed Jul  6 11:13:20 2022, end at Wed Jul  6 11:13:22 2022
test 7 start at Wed Jul  6 11:13:21 2022, end at Wed Jul  6 11:13:22 2022
test 8 start at Wed Jul  6 11:13:22 2022, end at Wed Jul  6 11:13:23 2022
test 10 start at Wed Jul  6 11:13:22 2022, end at Wed Jul  6 11:13:23 2022
test 4 start at Wed Jul  6 11:13:20 2022, end at Wed Jul  6 11:13:23 2022
test 12 start at Wed Jul  6 11:13:23 2022, end at Wed Jul  6 11:13:23 2022
test 13 start at Wed Jul  6 11:13:23 2022, end at Wed Jul  6 11:13:24 2022
test 9 start at Wed Jul  6 11:13:22 2022, end at Wed Jul  6 11:13:24 2022
test 11 start at Wed Jul  6 11:13:22 2022, end at Wed Jul  6 11:13:24 2022
test 14 start at Wed Jul  6 11:13:23 2022, end at Wed Jul  6 11:13:24 2022
test 15 start at Wed Jul  6 11:13:23 2022, end at Wed Jul  6 11:13:24 2022
test 16 start at Wed Jul  6 11:13:24 2022, end at Wed Jul  6 11:13:24 2022
test 17 start at Wed Jul  6 11:13:24 2022, end at Wed Jul  6 11:13:24 2022
test 18 start at Wed Jul  6 11:13:24 2022, end at Wed Jul  6 11:13:25 2022
test 19 start at Wed Jul  6 11:13:24 2022, end at Wed Jul  6 11:13:26 2022
```

图 1-11 搜索过程阻塞测试结果

b. 余座更新过程阻塞测试代码如图 1-12：

```
import time
import requests
import json
from concurrent.futures import ThreadPoolExecutor, ProcessPoolExecutor  # 线
程池，进程池
import threading


# 测试函数
def query_test(i):
    url = 'http://api.hustairline.xyz/api/query'
    data = """{ "N": 2, "agency":["DNH001"],"M":4,
    "date":["20220621000000","20220622000000","20220623000000","20220624000
000"],
    "sCity":["CIF","BSD","DLU","BHY"],
    "dCity":["BSD","DNH","AVA","DLU"]}"""
    start = time.ctime()
    response = requests.post(url, data=data)
    print(response.status_code)
    end = time.ctime()
    print('test {} start at {}, end at {}'.format(i, start, end))
def reset_query(i):
    url = 'http://api.hustairline.xyz/api/reset'
    start = time.ctime()
    response = requests.get(url)
    end = time.ctime()
    print('test {} start at {}, end at {}'.format(i, start, end))
if __name__ == '__main__':
    thread1 = threading.Thread(target=query_test, args=(1,))
    thread2 = threading.Thread(target=reset_query, args=(2,))
    thread1.start()
    thread2.start()
```

图 1-12 余座更新阻塞测试代码

测试结果如图 1-13，task1 为搜索进程，task2 为余座更新进程，task1 返回时间与 task2 相同，证明余座更新进程**未阻塞搜索服务**。

```
$ python .\query_test.py
200
test 1 start at Wed Jul  6 11:57:42 2022, end at Wed Jul  6 11:57:43 2022
test 2 start at Wed Jul  6 11:57:42 2022, end at Wed Jul  6 11:57:43 2022
```

图 1-13 余座更新阻塞测试代码

二、测试结果分析

（1）基本功能要求

低价行程搜索支持多旅客、多代理人、多航段搜索，支持设置最大结果数，所有测试用例搜索结果与预期结果完全一致；搜索结果中不存在任何错误，总票价、航班号、余座数据、乘客座位表、时间等信息完备。余座更新功能所有测试用例结果与预期一致，**满足基本功能要求**。

（2）性能

单次搜索在简单和一般测试用例下，平均响应时间为 40ms 左右；在复杂测试用例下，平均响应时间为 80ms 左右，证明**作品具有高性能，低延时的特性**；

图 2-1 服务器配置

如图 2-1，网站部署服务器使用普通 2 核 CPU，根据赛题要求，在搜索并发数为 2 的情况下，CPU 的 idle 值应低于 20%。由测试结果（图 1-9），复杂测试用例下，搜索并发数为 2 时，CPU 的 idle 值普遍低于 20%，即**作品能够充分利用硬件资源，满足该要求。**

搜索功能和余座更新功能**全程无阻塞**，满足无阻塞的非功能性要求。

因此，作品满足赛题中描述的所有非功能性要求。