CSE6730 Modeling & Simulation Project Report

Project 1: Airport Simulation

Zhihang Liu
School of Computer Science and Engineering
Georgia Institute of technology
Atlanta GA, USA
Zliu354@gatech.edu

Abstract—**This document is a brief report of CSE6730 course project 1: Airport simulation. Part 1 introduce the frame of the code and basic strategy used in the code. Part 2 walk through the difference added to the provided frame. Part 3 gives the validation results for testing the correctness of the data. Part 4 shows the results with different scale of problems to understand our model. Part 5 talked about the future work plans.**

Keywords—DES; Java; Simulation; Airport;

I. Introduction

This Java based software was developed for the simulation of airtraffic of five seleced airports in United states. A object oriented coding strategy have been applied, so all aiports and airplane are treated as objects of the corresponding class. For airport objects, the related events are: TAKEOFF, ARRIVES, LANDED and DEPARTS. Airplane objects take care of passenger number, airplane speed and destination and original airport. All events for airport and airplanes are handled by an EventHandler class by using a comman Descrete Event Simulation (DES) method. The details of the software structure can be found in part II.

II. Software Structure

A. Airmap.java

This class stores latitude and lontitude corrdinates of all airports (table 1.).

Table 1 Airports coordinates

	Latitude	*Longitude*
Atlanta(ATL)	*33.641211*	*-84.427769*
New York (JFK)	*40.641033*	*-73.778245*
Boston (BOS)	*42.365621*	*-71.009517*
Chicago (ORD)	*41.974402*	*-87.907321*
Los Angles (LAX)	*33.941410*	*-118.410485*

The distance between two airports are calculated using Haversine Formula[1].

B. Airplane.java

Airplane class stores the information of airplanes. For simplicity, only one type of airplane has been considered: Boeing 737-700, which is also the most used airplanes used for flight within united states. The desination and origin of each airplane are initialized when creaing airplane objects. The cruise speed of Boeing 737-700 is 838 km/h and passenger capacity is 150. The airplane name, also considered as flight number is equales to the orighin airport code * 10000 + destination airport code * 1000 + airplane number. For instance flights from ATL (code = 1) from JFK (code = 1) are named as 12xxx.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

C. Airport.java

This class is developed based on the provided frame. The difference is in the airport Handler. I add a new case called AIRPLANE_TAKEOFF, which means the airplane has left the runway and flying to the destination. By doing this, the departing and lading "queue" can be more easily handled. In my code the arriving and ready-to-depart have been treated by a time flag *nextWindow.* This parameter records the next available time of the runway. The method of calculating *nextWindow* is described below:

- When an airplane arrives or ready to depart, check the current nextWindow;
- If the currentTime > nextWindow: TAKFOFF or LANDED event is scheduled to the currentTime.
- If the currentTime < nextWindow: TAKEOFF or LANDED event is scheduled to the nextWindow + runwayTime.
- The delay of the next LANDED or TAKEOFF aircraft can be calculated as nextWindow – currentTime.

By using this parameter, we don't need to queue to store the pending events, since all events are scheduled with proper delay.

The circle time of each airport were calculated by accumulate every landed airplane's circle time. The circle time for landed airplane equals: arriveTime – landedTime – runwayTime.

The passenger number was generated using uniform random distribution between 130 and 150, which is reasonable guess for busy day.

D. AirportEvent.java

A new Airplane type parameter has been added to the airportEvent to let the event know which airplane it is related to.

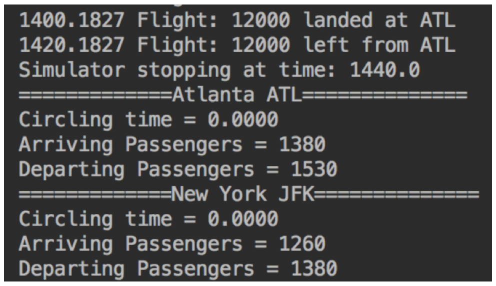
E. AirportSim.java

Contains the Main function which used for initialize airports, airplanes and do output for all events

All other classes in the project remains the same with the provided frame.

III. Validation

Before we proceed to use our model to simulation large scale problems. We begin with only 2 airports and 1 airplane to test the correctness of our model. Total simulation time is 1 day (1440 mins). The results are shown in Figure 1



```
1400.1827 Flight: 12000 landed at ATL
1420.1827 Flight: 12000 left from ATL
Simulator stopping at time: 1440.0
============Atlanta ATL=============
Circling time = 0.0000
Arriving Passengers = 1380
Departing Passengers = 1530
============New York JFK=============
Circling time = 0.0000
Arriving Passengers = 1260
Departing Passengers = 1380
```

Figure 1 Results of 2 airports and 1 airplane

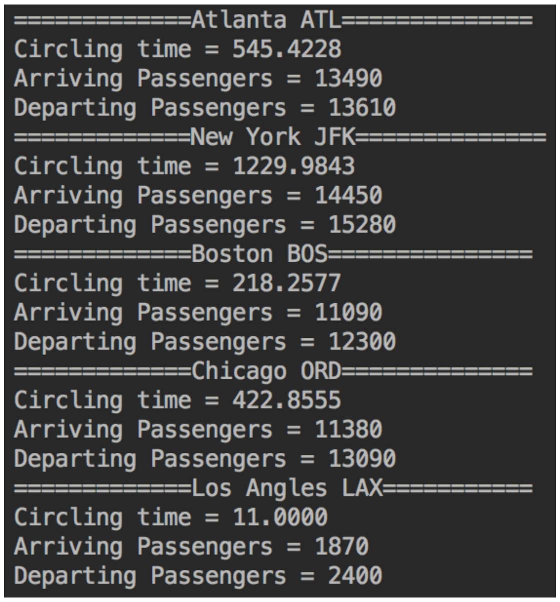
From Figure 1 we can see with only one airplane the total circling time is 0 and arriving passenger number of ATL equals the departing passenger number. The difference of departing passenger from ATL to JFK is because the simulation stops at a departed flight from ATL.

From this validation case, we can say the result of our model is reasonable.

IV. Using the model for large casees

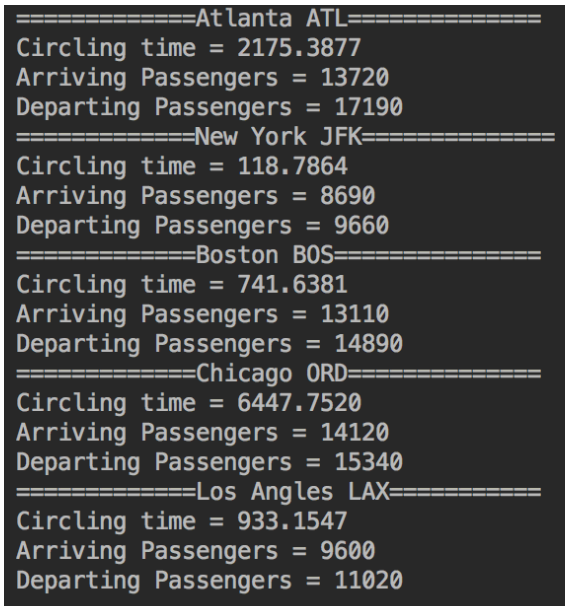
The airplanes are initialized with radomly selected (original airport, desination airport) pairs. And the initial events for all airplanes are assigned as AIRPLANE_DEPART and handled by the original airport.

A. Results of increasing number of airplanes



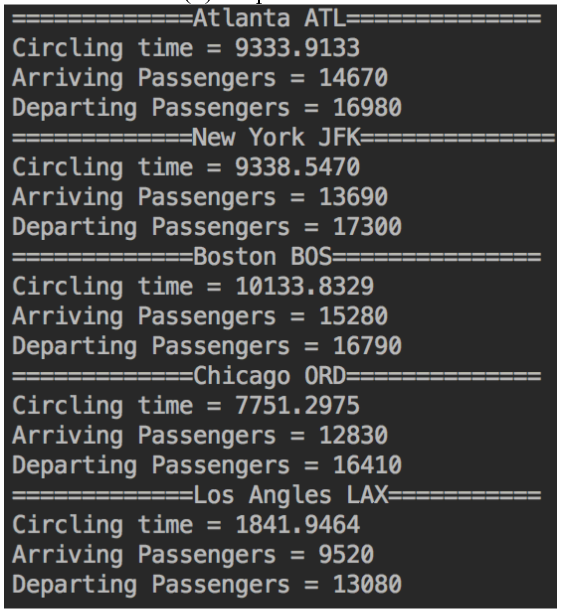
```
=============Atlanta ATL==============
Circling time = 545.4228
Arriving Passengers = 13490
Departing Passengers = 13610
=============New York JFK==============
Circling time = 1229.9843
Arriving Passengers = 14450
Departing Passengers = 15280
=============Boston BOS===============
Circling time = 218.2577
Arriving Passengers = 11090
Departing Passengers = 12300
=============Chicago ORD==============
Circling time = 422.8555
Arriving Passengers = 11380
Departing Passengers = 13090
=============Los Angles LAX==========
Circling time = 11.0000
Arriving Passengers = 1870
Departing Passengers = 2400
```

(a) Airplanes = 25



```
=============Atlanta ATL==============
Circling time = 2175.3877
Arriving Passengers = 13720
Departing Passengers = 17190
=============New York JFK==============
Circling time = 118.7864
Arriving Passengers = 8690
Departing Passengers = 9660
=============Boston BOS===============
Circling time = 741.6381
Arriving Passengers = 13110
Departing Passengers = 14890
=============Chicago ORD==============
Circling time = 6447.7520
Arriving Passengers = 14120
Departing Passengers = 15340
=============Los Angles LAX==========
Circling time = 933.1547
Arriving Passengers = 9600
Departing Passengers = 11020
```

(b) Airplanes = 50



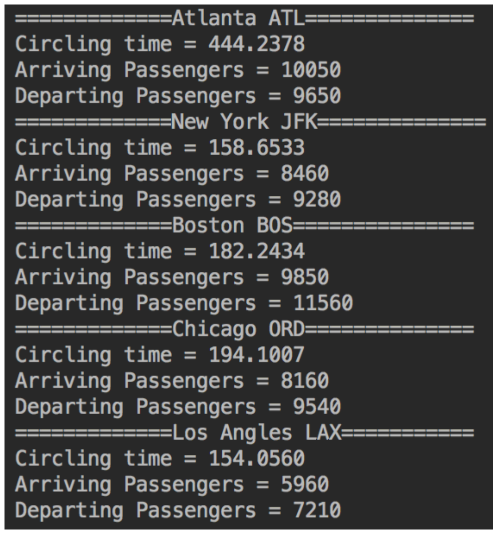
```
=============Atlanta ATL==============
Circling time = 9333.9133
Arriving Passengers = 14670
Departing Passengers = 16980
=============New York JFK==============
Circling time = 9338.5470
Arriving Passengers = 13690
Departing Passengers = 17300
=============Boston BOS===============
Circling time = 10133.8329
Arriving Passengers = 15280
Departing Passengers = 16790
=============Chicago ORD==============
Circling time = 7751.2975
Arriving Passengers = 12830
Departing Passengers = 16410
=============Los Angles LAX==========
Circling time = 1841.9464
Arriving Passengers = 9520
Departing Passengers = 13080
```

(3) airplanes = 100

Figure 2. Simulation of increasing airplanes number

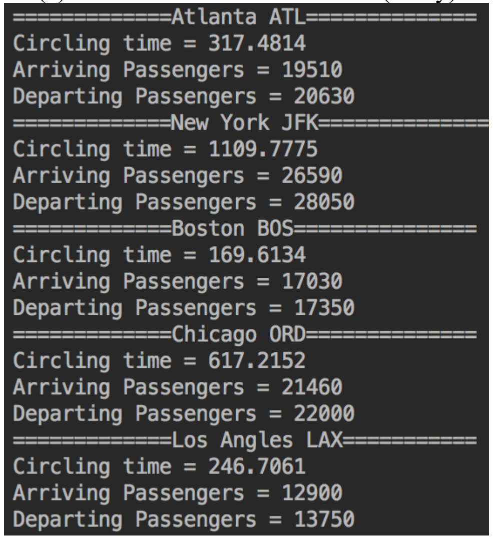
From Figure 2, we can see with increasing number of airplanes the circling time of each airport increase dramatically. Since we only have 5 runways, the circle time increasing dramatically when total number of airplanes reaches 100.

B. *Results of increasing number of simulation time*



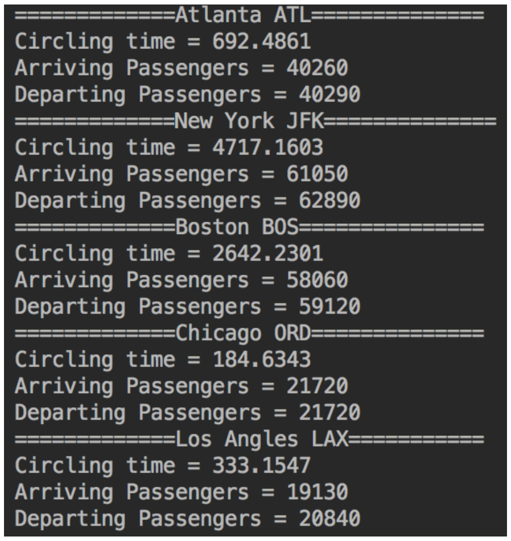
```
=============Atlanta ATL==============
Circling time = 444.2378
Arriving Passengers = 10050
Departing Passengers = 9650
=============New York JFK==============
Circling time = 158.6533
Arriving Passengers = 8460
Departing Passengers = 9280
=============Boston BOS===============
Circling time = 182.2434
Arriving Passengers = 9850
Departing Passengers = 11560
=============Chicago ORD==============
Circling time = 194.1007
Arriving Passengers = 8160
Departing Passengers = 9540
=============Los Angles LAX===========
Circling time = 154.0560
Arriving Passengers = 5960
Departing Passengers = 7210
```

(a) Simulation time = 1440 min (1 day)



```
=============Atlanta ATL==============
Circling time = 317.4814
Arriving Passengers = 19510
Departing Passengers = 20630
=============New York JFK==============
Circling time = 1109.7775
Arriving Passengers = 26590
Departing Passengers = 28050
=============Boston BOS===============
Circling time = 169.6134
Arriving Passengers = 17030
Departing Passengers = 17350
=============Chicago ORD==============
Circling time = 617.2152
Arriving Passengers = 21460
Departing Passengers = 22000
=============Los Angles LAX===========
Circling time = 246.7061
Arriving Passengers = 12900
Departing Passengers = 13750
```

(b) Simulation time = 2880 min (2 days)



```
=============Atlanta ATL==============
Circling time = 692.4861
Arriving Passengers = 40260
Departing Passengers = 40290
=============New York JFK==============
Circling time = 4717.1603
Arriving Passengers = 61050
Departing Passengers = 62890
=============Boston BOS===============
Circling time = 2642.2301
Arriving Passengers = 58060
Departing Passengers = 59120
=============Chicago ORD==============
Circling time = 184.6343
Arriving Passengers = 21720
Departing Passengers = 21720
=============Los Angles LAX===========
Circling time = 333.1547
Arriving Passengers = 19130
Departing Passengers = 20840
```

(b) Simulation time = 5760 min (3 days)

Figure 3. Simulation results with increase of simulation time

From Figure 3, we can see the circling time is at most increased by factor of 4 (since every simulation the airplanes are initialized randomly). The number of passengers arrive and departed were also increased linearly. The results also show that the simulation reaches stable condition, which means all airplanes did several TAKEOFF and LANDED events.

V. Conclusion and Future Work

Since the time pressure of approaching deadline, some detailed experiments haven't done. For future work, I plan to add more types of airplanes to see the effects of airplanes data. Also, we can add weather condition to affect the airport take-off and landing time.

References

[1] Haversine formula: http://andrew.hedges.name/experiments/haversine/

[2] Boeing 737-700 data: http://www.boeing.com/commercial/737ng/