"Work hard, play hard"

Accelerate Your Code - November 2012

web : www.Intel-Software-Academic-Program.com
email : contact@Intel-Software-Academic-Program.com
twitter : @IntelSwAcademic

Problem :

Let's imagine you have to go to a software conference in San Francisco (lucky you !), you live in Berlin. Booking the cheapest flight is easy when you want to go from point A to point B and you know the dates. Several websites do that already.
But if you want to pay a little more to extend your flight and do some tourism, it's getting complicated : you are interested in a lot of places (airports in Brazil, Mexico, Colombia, ...), and you are pretty flexible about the dates (before of after your conference). You just want to find a good opportunity : If I can get Berlin-SanFrancisco-SaoPaulo-Berlin for only 100 euros more than just Berlin-SanFrancisco-Berlin, I'll spend a week there on vacation !

But most flights requires a layover (a stop), you have to deal with multiple airline companies, and the price of a single flight is varying depending on the company used for the other flights. If you take all the segments of a flights from a single company, it's 30% cheaper.
Plus, companies are forming "alliances" to offer cheaper flights to the customers booking only inside the alliance. If you take all the segments of a flight from a single alliance, it's 20% cheaper.

Even today, this task requires human knowledge and skills because reservation software can't handle the huge combinations. It could be automated. Customers would have more options, spend less time searching on websites, airlines would fill empty planes.

For this contest, you are given a first input file listing all the flights, and a second input file listing the alliances of companies (a company can be in several alliances).
A source file solving the problem, input files and expected outputs are given to help you start.
http://www.intel-software-academic-program.com/contests/ayc/2012-11/problem/
The goal is to optimize and parallelize the software. You can start from scratch if you want, but it's probably safer to optimize and parallelize this reference version.
The goal is to optimize and parallelize for a wide range of use cases (few/many flights, few/many alliances, short/wide date ranges, ...).

Goal 1 : "Work Hard"

Find the cheapest flights between 2 cities.
Example : find the cheapest flight between Berlin and San Francisco given a time range for departure (leave after 2012-11-08 05:00 GMT - arrival before 2012-11-08 23:00 GMT) and return (leave after 2012-11-15 02:00 GMT - arrival before 2012-11-15 18:00 GMT), maximum layover time of 4h.

Goal 2 : "Play Hard"

Given a minimal and maximal vacation time (before of after your work trip) and a list of airports propose the cheapest flight for each vacation airport.
Example : how to modify the Berlin - San Francisco flight to propose a vacation of 7 days (before or after) in the following airports of interest : SaoPaulo, Panama, Cayman.
give the cheapest result for each of the 3 airports.

Benchmarking :

We have an automated cluster to compile and benchmark your solutions during the contest and for the final evaluation. It will be available a few days after the beginning of the contest.
We'll only check the code manually to validate the winning teams during the final evaluation.
You can upload your solution as often as you want to see if you code compiles correctly and evaluate the scalability on our reference hardware. Upload is done with a web form http://www.intel-software-academic-program.com/contests/ayc/2012-11/upload or automatically from your makefile.
You'll receive the results by email. You can't access the machine itself.

Professors will also receive a frequent report of their students results.

Your software has to compile and run on our reference linux platform.
(Intel Compiler version 12.1.0, gcc version 4.6.2, Debian version 4.6.2-12).
Using intel compiler is encouraged (read the doc for performance flags !).
Upload often to make sure it works fine. We'll use the last version uploaded before the end of the contest as final.

We'll provide small input files at the beginning, then bigger files to help you test scalability later during the contest. for the final evaluation, we'll use our own set of files. Check our twitter feed for the latest news **@IntelSwAcademic**

Example input files, command and output file :

input file flights.txt (first column is flight ID, times is in "epoch" format) :

```
1;Paris;11012012050000;New York;11012012130000;450;Cheap Airlines
2;New York;11012012160000;Chicago;11012012173000;100;America Flights
3;Chicago;11012012180000;Los Angeles;11012012220000;300;America Flights
4;Los Angeles;11082012070000;New York;11082012123000;450;America Flights
5;New York;11082012130000;Paris;11082012200000;600;France Airlines
6;New York;11012012140000;Los Angeles;11012012213000;1200;America Flights
7;Paris;10242012000000;Rio;10242012150000;1800;Expensive Airlines
8;Rio;11012012050000;New York;11012012153000;950;Expensive Airlines
```

input file alliances.txt (1 line per alliance) :

```
France Airlines;America Flights
```

command :

```
./run -nb_threads 2 -from Paris -to Los\ Angeles -departure_time_min 10302012000000 -
departure_time_max 11022012000000 -arrival_time_min 11082012000000 -arrival_time_max
11112012000000 -max_layover 14400 -vacation_time_min 432000 -vacation_time_max 604800 -
vacation_airports Rio -flights flights.txt -alliances alliances.txt -work_hard_file
work_hard.txt -play_hard_file play_hard.txt
```

output file 1 : work hard (2 airports) :

```
"Work Hard" Proposition :
Price : 1525
Cheap Airlines-1-Paris (11/1 5h0min)/New York (11/1 13h0min)-450$-100%
America Flights-2-New York (11/1 16h0min)/Chicago (11/1 17h30min)-100$-70%
America Flights-3-Chicago (11/1 18h0min)/Los Angeles (11/1 22h0min)-300$-70%
America Flights-4-Los Angeles (11/8 7h0min)/New York (11/8 12h30min)-450$-70%
France Airlines-5-New York (11/8 13h0min)/Paris (11/8 20h0min)-600$-80%
```

output file 2 : play hard (2 airport + third airport propositions)

```
"Play Hard" Proposition 1 : Rio
Price : 3000
Expensive Airlines-7-Paris (10/24 0h0min)/Rio (10/24 15h0min)-1800$-70%
Expensive Airlines-8-Rio (11/1 5h0min)/New York (11/1 15h30min)-950$-70%
America Flights-2-New York (11/1 16h0min)/Chicago (11/1 17h30min)-100$-70%
America Flights-3-Chicago (11/1 18h0min)/Los Angeles (11/1 22h0min)-300$-70%
America Flights-4-Los Angeles (11/8 7h0min)/New York (11/8 12h30min)-450$-70%
France Airlines-5-New York (11/8 13h0min)/Paris (11/8 20h0min)-600$-80%
```

Check the other test input files for other test cases at
http://www.intel-software-academic-program.com/contests/ayc/2012-11/problem/

Ressources :

Courses on parallel programming :
http://intel-software-academic-program.com/courses

Forums on Intel Developer Zone :
- English http://software.intel.com/fr-fr/forums/accelerate-contest-english-forum
- French http://software.intel.com/fr-fr/forums/concours-accelerate-your-code
- Russian http://software.intel.com/ru-ru/forums/accelerate-2012

Note that social activity, like blog and forum posts are part of the final evaluation ! (25/175 points)

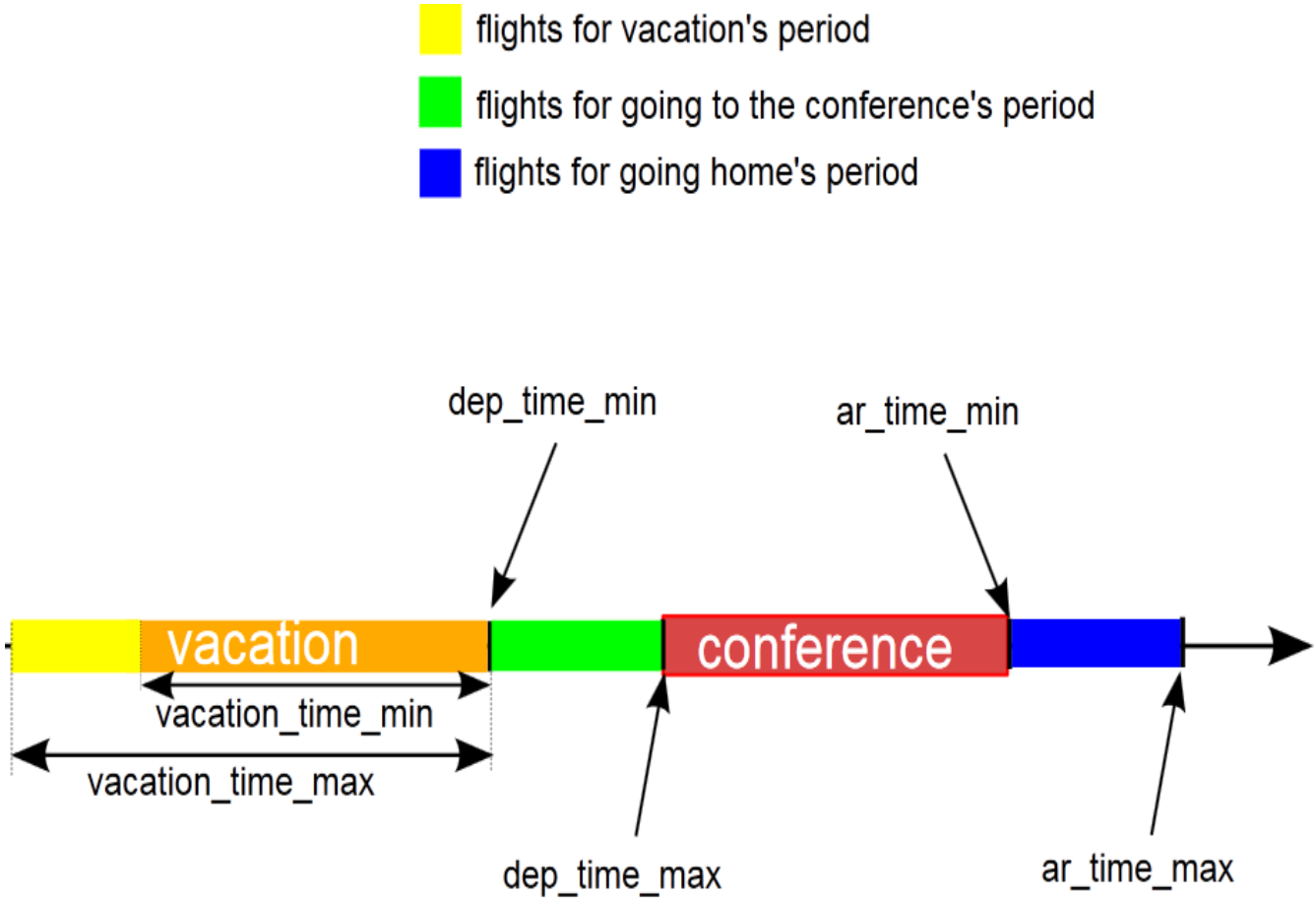
Intel Developer Zone “IDZ” - Server section :
http://software.intel.com/en-us/server-developer

Free intel tools on linux for non commercial tools (compiler, parallel programming tools) :
http://software.intel.com/en-us/non-commercial-software-development

Terms and conditions :
http://www.intel-software-academic-program.com/contests/ayc/2012-11/tc

Dates timeline :

Case 1 : Vacation before the conference :



Case 2 : Vacation after the conference :