TP2 C++

B3111 : Edern HAUMONT et Nicolas SIX

Mercredi 11 novembre 2015

Table des matières

1 tests

Introduction   We used a few manually generated test and we generated some bigger input files with a generation code. This script also ofently generates an expected output file.

To test or application, we made it read th input file instead of the keyboard input. We also redirect the application output to another file which is afterwards compared to the expected output file.

1.1   manually generated tests

1.1.1   TestAdd

The first test just check that the ADD query runs fine. We add several event and check each time the state of the table with STATS_C.

1.1.2   TestAddStatC and TestStatC

These tests check the comportment of the STATS_C query, especially with no values, one value, and after several ADD.

1.1.3   TestStatD7

This test checks the comportment of the STATS_D7 query, especially with no values, one value, and after several ADD.

1.1.4   TestJamDH

This test checks the comportment of the JAM_DH query, especially with no values, full jam values and multiple values.

1.1.5   TestOpt

This test checks the comportment of the OPT query. We made this request with 0 sensors, sensors with no value, 1 and 2 sensors.

1.2   machine generated tests

1.2.1   Test2

The code generates entries (all are green) for 1500 sensors and a restricted time interval. Calls JAM_DH query to test input values. Used to check that the program runs.

1.2.2   Test3

The code generates entries for 50 sensors but in bigger time intervals and multiple traffics. Calls the 3 different queries with several parameters. Used o check that the program computes datas properly.

1.2.3   Test4

The code generates some entries to test basically if the last query works with some missing values. Fast to check.

1.2.4   Test5

Our biggest test. 20M events on 1500 sensors. To compare with outputs from other groups. Sensor numbers between 0 and 1G