Concurrent solution for cube puzzle

December 7, 2016

1 Introduction

Description of the problem:
http://www.acta.sapientia.ro/acta-info/C4-1/info41-6.pdf

2 Solutions

In this section the number of threads is N, were N = std::thread::hardware_concurrency() - command which gives the number of concurrent threads supported. The value should be considered only a hint. There are multiple solution with different technologies:

- Using std::future Solutinon can be seen here
- Using std::thread Solutinon can be seen here

2.1 Using N + 1 threads

The performance of the program was tested in the following way:

1. Using N threads + main thread, using std::thread
2. Using N threads + main thread, using std::future
3. Using 100 threads, using std::thread
4. Using 100 threads, using std::future
5. Using N - 1 threads + main thread, with using std::thread
6. Using N - 1 threads + main thread, with using std::future
7. Using 1000 threads + main thread, with using std::thread
8. Using 1000 threads + main thread, with using std::future

Computer configuration: Processor: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz Installed memory(RAM): 4.00 GB Compiler: Cygwin 32 bit, 5.4 GCC.

The result of the performance test can be seen in the following table:

Test case	Test 1	Test 2	Test 3	Average
1	28.1	28.8	28.0	28.30
2	29.9	31.8	31.7	31.13
3	27.9	31.6	30.7	30.06
4	28.3	28.9	28.7	28.30
5	27.7	28,5	27,1	27.76
6	29.6	29.0	28.7	29.10
7	28.9	29.8	31.7	30.13
8	27.7	27.3	27.5	27.50

3 Conclusion

On this computer it is better to use std::future and then std::thread. What is more, we can use without the maximum number of allowed threads. if the main thread is not working.