Project Report

1. Brief Description of this work load

This code is an efficient implementation in C++ of the 8 puzzle algorithm, designed to be used from high performance real-time applications (video games) and with an optional fast memory allocation scheme. 8puzzle with no arguments runs with one of the boards in the cpp file, we can select the one we want changing the conditional compilation instructions. Or if we can pass in a board on the command line using digits for the tile positions, where zero is the space.

1. Steps to compile and run
2. Hardware specs of evaluation platform
3. Initial perf results

According to the description we list in part 2. We first compile this work load and run it in the platform equipped with the hardware parameters listed in part 3. Fig. 1 is the relative result of this workload.

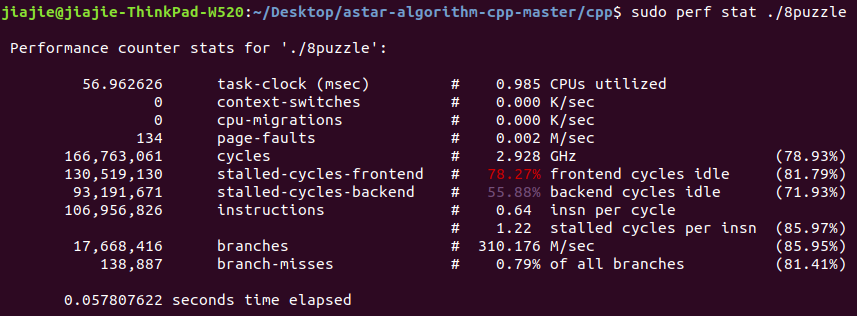


Fig. 1. Results with initial perf command

We can see from the figure that the instruction is more than 100 M. Specifically, it’s about 107 M instructions. The parameter of instruction per cycle (IPC) is 0.64. Obviously, this workload satisfy the requirement of the IPC and instructions amount. Hence, it is reasonable for us to choose it as our benchmark.

1. Initial report from running it on ESESC
2. Explanation of bottleneck and how you found it
3. What architectural parameters you changed to improve performance
4. Why the architectural parameter changes are reasonable
5. Explanation of how your changes fix the bottleneck
6. Other plots, details, or rationale you find necessary