DATE: 2018/1/9

Department of Electrical and Electronic Information Engineering		ID	M143245
Name	Yuki Takeda		

Supervisor	Shuichi Ichikawa Naoki Fujieda

Abstract

Title	Parallelization of benchmark suite MiBench with OpenMP

MiBench is a benchmark program suite for embedded applications. MiBench is designed for single core processors and consists of 35 applications in 6 categories. ParMiBench then appeared for the embedded systems equipped with multi-core processors. ParMiBench is a benchmark program suite that parallelizes MiBench using POSIX thread, and it consists of 7 applications in 4 categories.

Implementation with POSIX thread is problematic because the program description is complicated and difficult to understand, it depends on the platform and low portability. From these problems, it is desirable for ordinary users to use benchmark programs parallelized using a high level API such as OpenMP. In this research, applications implemented by POSIX thread in ParMiBench are re-implemented with OpenMP, and comparison is made between them. We also parallelize several applications including the categories that was not handled by ParMiBench by using OpenMP to evaluate their execution time. Eleven applications were parallelized, that are basicmath, bitcount, susan (smoothing), stringsearch, dijkstra, patricia, sha, qsort, jpeg, blowfish (encryption, decryption), and FFT (FFT / IFFT).

For evaluation, execution time with number of OpenMP threads 1, 2, 4, 8, and 16 was measured for each parallelized program. As an index of Speedup by parallelization, the ratio of execution time in parallel number 1 and execution time in number of OpenMP threads n is defined as Speedup in number of OpenMP threads n. Speedup was calculated from the measured execution time.

As a result of the evaluation, Speedup equal to or better than that of basicmath, patricia was obtained. In the implementation of ParMiBench, the bitcount is a value less than 1 in Speedup in many cases, whereas the Speedup also rose as the number of OpenMP threads increased. Speedup is about the same when susan is less than 4 in number of OpenMP threads , but Speedup saturates when it is over. dijkstra got a Speedup over ParMiBench, but the sequential execution time was longer than ParMiBench. For stringsearch and sha, the value of Speedup decreases for all number of OpenMP threads. For the four applications of qsort, jpeg, blowfish (encryption, decryption) , and FFT (FFT / IFFT), Speedup was maximum at qsort: 8.819, jpeg: 1.398, blowfish (encryption): 15.60, blowfish (decryption): 15.75 , FFT: 1.504, IFFT: 1.513.