LAB1: Implement and compare blocked and recursive matrix multiplication

TA: Li Yunfan

March 20, 2018

1 Requirements

- ddl:3/23 Friday 23:59:59
- please send the lab results to 2260115144@qq.com before ddl
- (the program – "yours.c" , raw test data, machine configuration, and a report of the results are required; you can only use the Linux operating system and programming language C, and can only use the standard C library, any other C library are not acceptable.)

2 Introduction

- Aim of the lab: to maximize matrix multiplication performance (by minimizing Cache Misses)

Let's start from the beginning:

3 Matrix Multiplication

3.1 Tasks

1. Implement blocked matrix multiplication.
2. Implement recursive matrix multiplication.
3. Tune both programs to run as fast as possible, and compare their performance. Write a technical report on the differences in efficiency. (Make sure you include raw test data, testing scripts, machine configuration, and drawing in the report.)

3.2 Hints

- HINT1: Tiling: Tune the block size of the sub-matrices so that they can just fit in cache.
- HINT2: Divide and Conquer.
- HINT3: Complete your lab on the baseline of file "yours.c", the methods the TA has implemented is compiled into a object file "handout.o".
- HINT4: use "gcc yours.c handout.o -o yours" to link your work to "handout.o" so that you can use the check and generate function in it.
- Finally, implement your own timing gadget, the grading of the lab will be relative. For example, if A has the fastest matrix multiplication algorithm in this class and A's running time is 60 seconds. While B uses 120 seconds. Then A's final score will be 100 and B's score will be 50. All of your programs are to be run on the same PC so don't worry about your computer configuration.