

## 实践项目 P1: Matrix Multiplication Autotuner

布置周: 第 3 周

提交周: 第 7 周 2023 年 10 月 25 日 23:59

设计和实现一个程序性能的自动调优器 Autotuner, 可以自动测试 Matrix Multiplication 程序的可配置参数, 并根据性能选择最优配置输出。要求:

1. Autotuner 的实现框架可以参考课上介绍的 OpenTuner (项目地址: <https://github.com/jansel/opentuner>), 也可以自行设计和实现, 需要包括三个明确的接口实现: (1) 输入的目标程序; (2) 输入的配置参数值组合; (3) 输入的参数值搜索算法。请在报告中说明所设计框架和实现的优劣。
2. 输入的目标程序为 Matrix Multiplication, 统一采用附 1 所列 C 语言版本。
3. 输入的配置参数值组合包括如下:
  - a) 循环分块大小: 测试循环分块大小  $s$  为 8, 16, 32, 64, 128 五种情况。
  - b) 编译优化级别: 测试 O0, O1, O2, O3 四种级别。
4. 实现两种参数值搜索算法:
  - 一种为 Grid Search, 遍历上述所有配置参数值的组合, 对实验结果进行详细分析, 并尝试解释不同组合的性能差异产生的原因;
  - 另一种可以自行设计和实现任一种课上介绍的 Design of Experiments 算法, 根据实验过程和结果, 比较和说明上述两种算法的优劣。

附 1: Matrix Multiplication 程序 (C 语言)

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/time.h>
#include <assert.h>

#define n 4096
double A[n][n];
double B[n][n];
double C[n][n];

float tdiff(struct timeval *start,
            struct timeval *end) {
    return (end->tv_sec-start->tv_sec) +
        1e-6*(end->tv_usec-start->tv_usec);
}

int main(int argc, const char *argv[]){
    assert(argc==2);
    int s = atoi(argv[1]);
    if (s < 1 || s > 4096) {
        printf("Invalid input values.\n");
        return -1;
    }
    for (int i=0; i<n; ++i){
        for (int j = 0; j<n; ++j){
            A[i][j] = (double)rand() / (double)RAND_MAX;
            B[i][j] = (double)rand() / (double)RAND_MAX;
            C[i][j] = 0;
        }
    }

    struct timeval start, end;
    gettimeofday(&start, NULL);
    for (int ih = 0; ih < n; ih += s)
        for (int jh = 0; jh < n; jh += s)
            for (int kh = 0; kh < n; kh += s)
                for (int il = 0; il < s; ++il)
                    for (int kl = 0; kl < s; ++kl)
                        for (int jl = 0; jl < s; ++jl)
                            C[ih+il][jh+jl] += A[ih+il][kh+kl] * B[kh+kl][jh+jl];
    gettimeofday(&end, NULL);
    printf("%.6f\n", tdiff(&start, &end));
    return 0;
}
```