# 小作业七：单机性能优化

## 汇报测量结果

### 任务零

任务零是矩阵乘法的简单实现，我们会使用不同的编译器参数 (`-O0`, `-O1`, `-O2`, `-O3`, `-fast`) 来编译该程序。

你需要测试不同编译参数下的程序性能，并汇报测试结果。

| 编译器参数 | 用时 (s) | 性能 (GFlops) | 加速比 |
| --- | --- | --- | --- |
| `-O0` | 0.9668 | 0.2776 | 1.00 |
| `-O1` | 0.3300 | 0.8134 | 2.93 |
| `-O2` | 0.3333 | 0.8055 | 2.90 |
| `-O3` | 0.0483 | 5.5610 | 20.03 |
| `-fast` | 0.0398 | 6.7386 | 24.27 |

### 任务一

在任务一中，我们给出了一个矩阵乘法内核 ($4 \times 8 \times k$) 的向量化实现，并通过反复执行来测试该内核的性能峰值。
函数 `matmul_4x8` 中使用的预处理指令 `unroll` 表示循环展开，括号内的数表示循环展开的程度。

请测试函数 `matmul_4x8` 中循环展开的程度分别设置为 $1, 2, 4, 8, 16$ 时，程序的性能。可以在编译时使用 `make UNROLL_N=4` 来设置该参数为 $4$。

| 循环展开程度 | 用时 (s) | 性能 (GFlops) | 加速比 |
| --- | --- | --- | --- |
| $1$ | 2.0677 | 15.8473 | 1.00 |
| $2$ | 1.9547 | 16.7641 | 1.06 |
| $4$ | 1.8049 | 18.1547 | 1.15 |
| $8$ | 1.7802 | 18.4065 | 1.16 |
| $16$ | 1.8251 | 17.9541 | 1.13 |

## 回答问题

- 请参考 ICC 手册 并简述参数 (`-O0`, `-O1`, `-O2`, `-O3`, `-fast`) 分别进行了哪些编译优化。每种参数罗列几个优化技术即可。

> - `-O0`：不进行任何优化。

> - `-O1`：启用基本的优化技术，例如数据依赖分析、代码移动、强度拆减等。
> - `-O2`：在`-O1`的基础上进行更多的优化，例如函数内联、常数折叠、复制传播、死代码消除等。
> - `-O3`：在`-O1`的基础上进行更高级的优化，例如循环融合、合并分支判断等。
> - `-fast`：多个优化选项的组合，通常包括`-O3`、`-xHost`、`-ipo`等，启用所有可能的优化。

- 请简述任务一中循环展开带来的好处。

> 循环展开减少了循环变量`K`的比较次数，可以减少循环控制的开销，减少了执行分支跳转指令的次数，增加指令级并行性，提高 CPU 的利用率和吞吐量。