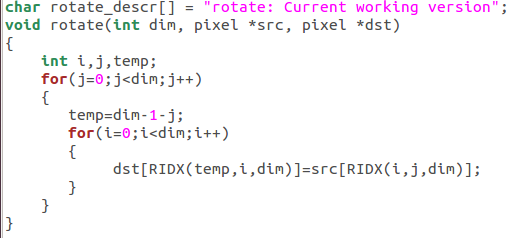
Perflab 实验日志

**优化代码1**

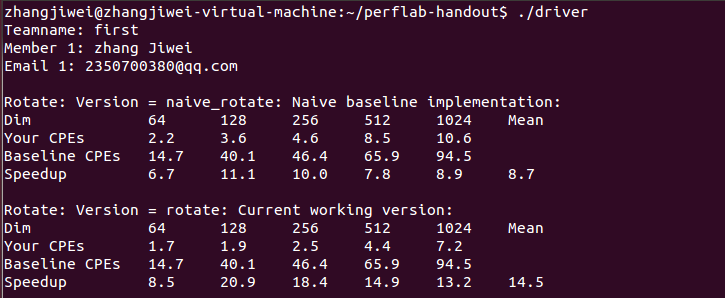


详细注释：如上优化代码所示，进行了两部优化

A:将循环 i 放在外边，循环 j 放在里边

B:提前计算需要重复计算的 dim-1-j，在循环里直接使用结果

优化结果：



优化思路及实现过程：

A:将循环 i 放在外边，循环 j 放在里边

减少了每次指针的移动弄位置，当 j 为循环外围时，每次循环都要将指针移动一个横排那么长，当 i为循环外围时，每次只需寻址位置+1 即可。

B:减少重复计算

在原函数中，中间有一步 dim-i-j 有重复运算，因此计划将其提前计算，在循环中直接使用即可。

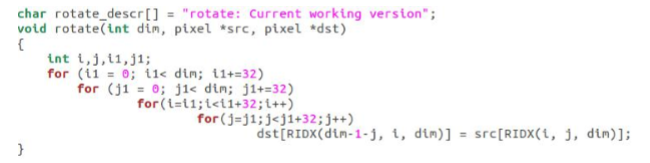
**优化代码2**

详细注释：建立临时变量储存需要重复计算的数据

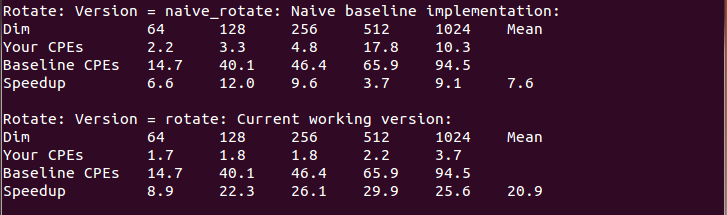
每次循环32个数据 来求cache的命中

优化思路：将原循环分成32\*32小块，提高空间的局部性，减少程序中的跳转语句。

实现过程：将其划分为32\*32像素的小块，每次转移32个数据



优化结果：



**优化代码3**

详细注释：设置转换地点的初始值

每次变换32个数据，以求cache命中

将32行作为一个划分界限，每次将这32行的第一列在一个循环内一起转换

转换开始后 操作32次

每次转换完成之后将转换源向下移动一格

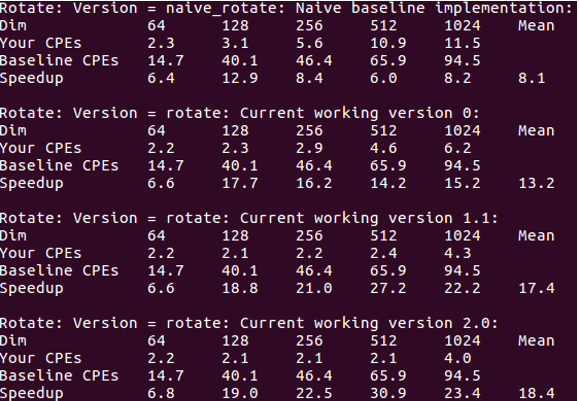
将接受转换的地点左移一位

转换源转换目标初始化 转换源向下移动32行

转化目标点和转换源相对应

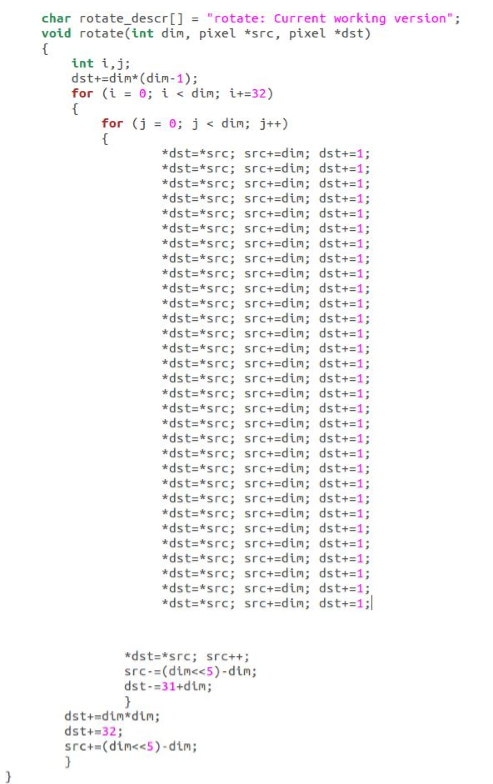
在代码二的基础上将循环转换，进一步提升效率

优化结果：



优化思路：将循环次数减少32倍。减少关键路径的长度，有效提高程序运行速度。

实现过程：将其划分每32行为一个单位，每次在32行中只移动1小列的32个数据。

****