

并行计算 II 2023 春季第三次作业优化 Gaussian Blur

丁明朔 刘翠玉

1 Gaussian Blur

Gaussian blur (又被称作 Gaussian smoothing), 是图像处理领域的经典算法, 有着非常广泛的应用。具体来讲, 本次作业中的 Gaussian blur 是将一个卷积核:

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

作用在一个 $N \times M$ 的单通道图片上, 并获得一个 $(N - 2) \times (M - 2)$ 规模的模糊后的图片, 也即不进行 padding 操作。

作业要求基于给定的 Gaussian blur 串行算法实现 (gaussian_blur.cpp), 对于十张图片进行数据处理, 自选使用 **MPI+Open MP** 或 **CUDA** 实现, 并提供优化报告。

2 思考题

Gaussian blur 是属于运算密集型还是访存密集型应用? 为什么? 请在报告中说明。

3 作业要求

根据附件中给定的 Attention 串行算法实现 (attention.cpp), 使用 MPI 进行优化。具体要求如下:

- 输入图像文件列表为 images.txt, 其中共有十个算例。代码需在十个算例下均运行通过, 最后计算十个算例的总用时作为排名;
- 可改变读入方式, 此部分无需进入耗时;
- 串行实现中除 verification 函数外, 其他均可以改动, 如需改动 verification 函数, 请在报告中说明;
- 对于 MPI+OpenMP 的实现: 基于数院集群, 最多使用 4 个节点, 进程数与线程数不限;
- 对于 CUDA 的实现: 基于数院集群, 最多使用 1 个 GPU 卡。
- 不限制编程语言、编译选项和编译器版本 (推荐 C/C++ 实现);
- 提交完整的并行程序代码, 要求程序中包括并行代码/计时模块/正确性验证, 提供与非优化版本的性能加速比较和分析;

- 提交报告中要求说明硬件型号、编译器版本、划分方法以及每个进程的任务负载；
- 鼓励报告中提供分阶段优化或者多版本优化的分析和比较。
- GPU 资源紧张，请同学们合理规划作业时间！

4 提交要求和评分标准

- **提交：**将代码和报告打包后发邮件至：parcoii2023@163.com，邮件主题为“学号-第 X 次作业”，在7月7日23点59分前提交，可更新结果至多3次。
- **评分：**(1) 基本要求：基于串行实现完成 Gaussian blur 算法的并行程序，代码运行通过，在报告中给出并行划分方法、运行时间和正确性验证；(2) 分析对比优化前后，哪些策略带来了明显性能收益，代码实际加速效果越好分数越高，MPI+OpenMP 与 CUDA 两种实现方法分别排序记分；(3) 其他加分项：根据报告内容丰富程度、代码质量酌情加分。

References