并行与分布式计算导论 作业 5
PDC 2023s Homework 5

截止期限 2023 年 5 月 8 日 23:59

DDL: 2023 May. 8 23:59 (GMT+8)

利用 **SYCL/CUDA** 并行编程,

Programming in **SYCL or CUDA**,

本次作业需要实现并利用 SYCL/CUDA 进行异构编程（GPU）。本次作业利用限差分方法求解一个二维拉普拉斯方程。需要通过近似方法求解由有限差分方法得出的二维(N+2) * (N+2)矩阵系统 Au = b。该算法在科学计算等领域有着广泛的应用，算法输出所有$u_{i,j}$，存储为长为$(N+2)^2$的向量u。这本质上是在求解一个在 $[-1, 1]^2$ 上的二维拉普拉斯方程$-(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}) = f(x,y)$。在每个点$(x_i, y_i)$上，导数可以通过以下公式近似计算$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}$，$\frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{h^2}$，其中$h = \frac{2}{N+1}$。我们假定在矩阵系统 Au=b 的边界上，所有未知量 u 的值都被设为 0。这意味着在计算矩阵时，只考虑内部节点而不考虑边界节点。本次作业使用函数$f(x,y) = sin(\pi x)sin(\pi y)$进行测试，这个函数对应的精确解为$u(x,y) = \frac{1}{2\pi^2}sin(\pi x)sin(\pi y)$。

在近似求解过程中给向量u中元素初值设置为 1，近似结果经过多次迭代会逐渐接近精确解，我们将实验的终止条件设置为$||b - Au|| < 10^{-6}$。

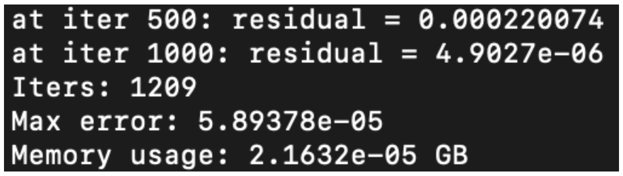
This assignment involves the implementation of a program to solve a matrix system Au = b that results from a 2D finite difference method for Laplace's equation $-(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}) = f(x,y)$ on $[-1,1]^2$, where A is an (N+2) * (N+2) 2D matrix, u is a vector of unknowns, and b is a vector of constants. The approximate solution will be accelerated using SYCL/CUDA. Derivatives at each point (x_i, y_i) are approximated using $\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}$ and $\frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{h^2}$, where $h = \frac{2}{N+1}$. Assuming zero boundary conditions $u(x,y) = 0$ reduces this to an N × N system for the interior nodes. The code will be tested using $f(x,y) = sin(\pi x)sin(\pi y)$ and initial guess u = 1. The exact solution is $u(x,y) = \frac{1}{2\pi^2}sin(\pi x)sin(\pi y)$. The program will stop when $||b - Au|| < 10^{-6}$.

请实现以下三个任务，并撰写书面报告，简述你的实现思路，测试结果和分析。将书面报告和代码源码共同打包为压缩包上传至教学网，同时在课程服务器中建立文件夹 hw5，并将代码上传至文件夹。

Please finish the following three tasks and write a report containing your programming idea, test results and analysis. You need to pack up your report and source code together and upload to Course website. Please create a directory named "hw5" on the course machine, and upload your code to the directory.

任务 1

完成环境配置。我们提供了一份满足上述要求的串行版本代码，你需要将附件中压缩包上传至教学服务器（或你要使用的机器），使用命令 tar -xzvf hw5.tgz 解压。进入 /hw5 目录，使用命令 make，编译生成可执行文件 fd。本次作业不需要使用数据集。可以在 /hw5 目录中，使用命令 ./fd N tolerance 测试可执行文件 fd 是否可以正确运行。当运行命令.fd 50 0.000001 时，你应该可以看到类似下图的运行结果。


```
at iter 500: residual = 0.000220074
at iter 1000: residual = 4.9027e-06
Iters: 1209
Max error: 5.89378e-05
Memory usage: 2.1632e-05 GB
```

测试程序的可扩展性，尝试在 N=10，100，1000，...的情况下运行，统计运行的时间和内存，分析是运行时间还是内存限制了程序进行更大规模运算。

Task1

To set up the environment, start by uploading the attached file to the course machine or your own machine. Next, untar the file using the command "tar -xzvf hw5.tgz". To generate the executable file fd, execute the command "cd <path_to_hw5>; make". To test whether the program "fd" can execute correctly, use the command "cd <path_to_hw2>; ./fd N tolerance". Upon executing the command "./fd 50 0.000001", you should see a similar result as shown in the figure above.

Next, run the code for (N + 2) * (N + 2) grids with N = 10, 100, 1000, and so on. Determine the maximum size of the matrix that you can reasonably run, and identify the limiting factors, such as memory and computational runtime. Provide timing results and estimate the memory requirements.

任务 2

使用 SYCL/CUDA 对串行版本进行加速。修改 hw5/fd.c，增加其他的必要函数和文件。可能需要对 Makefile 进行必要的修改。

再次测试程序的可扩展性，比较和串行版本的差异。

Task2

To accelerate the serial version, use SYCL or CUDA. Modify "hw5/fd.c" and add any necessary functions or files. You may also need to make modifications to the Makefile.

Rerun the code to test its scalability and compare it with the original serial version. Evaluate the effectiveness of the acceleration and document any improvements in performance.

任务 3（可选）

使用共享内存提升代码性能。可之前的实现进行比较分析，如带宽和吞吐量的差异。

Task3 (Optional)

Improve the performance of your code by implementing shared memory. Consider using a different parallelization pattern for the shared memory implementation. Compare the shared memory implementation with the previous implementation in terms of bandwidth, throughput, and other relevant performance metrics.

撰写书面报告：在完成上述任务后，撰写报告阐述您的编程思路、测试结果和分析，报告要求尽量**简洁**。**不要**粘贴代码截图。本次实验需要在 pdc01 或 pdc02 上完成。

In addition to your code, please provide a clear and concise high-level description of how your implementation works, as well as the test results and analysis. Please **refrain from** pasting your code directly in your report. Note that GPU resources are only available on pdc01 and pdc02.