GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
INSTITUT FÜR INFORMATIK

Practical Course on Parallel Computing · SoSe 2017

Prof. Yahyapour · Prof. Grabowski · Prof. Quadt

Assignment Sheet 9 · Submission Deadline: 19.06.2018, 3 pm

Organizational Issues

Organizational hints were already given in the lecture and can also be found as slides in StudIP. Please follow the following guidelines carefully:

- Solutions (including source code) must be send via E-Mail to christian.koehler[at]gwdg[dot]de **before** the deadline.
- Solution must be provided as either a **tar** or **zip** archive.
- If the assignment sheet contains questions (besides practical exercises), they must be answered in text form and provided as pdf inside the archive.
- Solutions must be submitted by groups of two students.

Hint: For resources aiding you in fulfilling these assignments, please refer to the lecture slides provided in Stud.IP. In particular, consult the *OpenACC Programming and Best Practices Guide*[1] and the *OpenACC API Quick reference guide*[2] for the practical exercises.

You can use your own hardware or the Scientific Compute Cluster[3] to run your code for the assignments.

Assignment 1 – Distributed heat equation using MPI+CUDA (*7 Points*)

a) For $n = 1024$, initialize a two-dimensional `float` array describing the $n \times n$-matrix A which is given in the following way

$$A_{ij} = \begin{cases} 1 & \text{if } i,j < \frac{n}{2} \text{ or } \frac{n}{2} \leq i,j \\ 0 & \text{otherwise} \end{cases}$$

distributed across 4 GPU devices being controlled by one MPI process each. (3P)

b) Like previously on a single device, apply the Jacobi iteration

$$A_{ij} \leftarrow (A_{i-1,j} + A_{i+1,j} + A_{i,j-1} + A_{i,j+1})/4$$

for all $0 < i < n-1, 0 < j < n-1$ for the entire array. This involves exchanging "halo" data, i.e. A_{ij} for $i,j = \frac{n}{2} - 1, \frac{n}{2}$ between iterations. Again, the pointwise difference should not exceed $\epsilon = 0,01$ before the iteration process is stopped. (4P)

[1] `https://www.openacc.org/sites/default/files/inline-files/OpenACC_Programming_Guide_0.pdf`

[2] `http://www.nvidia.com/docs/IO/116711/OpenACC-API.pdf`

[3] `https://info.gwdg.de/dokuwiki/doku.php?id=en:services:application_services:high_performance_computing:start`

Assignment 2 – Asynchronous matrix multiplication (*8 Points*)

a) Implement the multiplication of two matrices that are specified on the host and parallelize your code using OpenACC directives. (2P)

b) For an $n \times n$ matrix A_{ij} given by the formula

$$A_{ij} = \cos\left(12\frac{i+j}{n}\right)\sin\left(24\frac{i-j}{n}\right)$$

modify your code from a) to initialize the matrix on the device directly and compute $A * A$. Use clauses for the `data` directive to avoid unnecessary copy operations. (3P)

c) Split the parallel matrix multiplication from a) into equally sized chunks, such that the entire matrix does not need to be kept in the device memory at the same time. The chunks should be transferred to the device, operated on and the results transferred back to the host in an asynchronous way. [4] (3P)

[4] Use the `async(id)` directive with a chunk-dependent `id`.