Rainer Spurzem, spurzem@ari.uni-heidelberg.de

Introduction to GPU Accelerated Computing

Mon-Fri, Feb. 12-16, 2024, Uni Heidelberg, GPU Block Course

Tutors:

Helena Ren helena.ren@mpi-hd.mpg.de
Vahid Amiri vahid.amiri@uni-heidelberg.de
Lorena Acuňa acuna@mpia.de

Location:

Lecture: hybrid; CIP Pool KIP INF227 1.401 and via zoom
Tutorials: CIP Pool KIP INF227 1.401 and online zoom

START: 10:15 a.m., Monday Feb. 12, 2024

1. zoom and chat access, general informations

Lecture and online tutorial for Group 2 (Vahid Amiri):
https://eu02web.zoom-x.de/j/67597507858?wd=Y05BRUlXc3N2SjhSRXUwdytvWDZrUT09

Meeting ID: 675 9750 7858/Passcode: please see Rocket Chat or uebungen web:
https://uebungen.physik.uni-heidelberg.de/chat/group/WS23-GPU-Course

Group 3 (Lorena Acuna) is hybrid, in person in KIP INF227 1.401 and online:
Meeting ID: 697 4720 4337/Passcode: please see Rocket Chat or uebungen web:
https://eu02web.zoom-x.de/j/69747204337?wd=TGVVRFNhazZPd2thenUyVFJKVGxsdz09

Time in lecture: Mon-Fri 10:15 – 13:00
Lunch Break: 13:00 – 14:15
Sometimes lecture continued a little bit and followed by Tutorials:
14:15 – 17:00   **(no afternoon session on Friday!)**

There are rocket chats open:
General course chat:
https://uebungen.physik.uni-heidelberg.de/chat/group/WS23-GPU-Course

Also for each group:
https://uebungen.physik.uni-heidelberg.de/chat/group/WS23-GPU-1

https://uebungen.physik.uni-heidelberg.de/chat/group/WS23-GPU-2

(Note: this block course takes place in 2024, but it still belongs to the winter semester 2023, therefore it is labelled with WS23)

Group 3: NO Rocket chat for technical reasons, please contact Lorena in zoom or via email.

Rocket Chat: Messages independent of zoom! Messages are permanent! In zoom: raise hand (preferred) and chat, but not permanent! Any questions raised in the general course rocket chat will be answered there, remain there, and/or discussed in the lecture.

Need to know: This course is not graded!
Certificate of successful participation only! (2 ECTS points)

Necessary condition for certificate:

Regular participation in course, all Mon-Wed exercises done on the computer! The last exercise is a homework, and will be prepared on Thursday. You have about one week time to turn it in. More time can be given on request, talk to your tutor.

This is NOT a programming course ! 1:

You should have some knowledge of a higher level programming language (like python, c, c++, fortran, …). We will learn special CUDA extensions of the c-language for programming GPUs (graphical processing unit). This is called GPGPU – general purpose GPU programming.

This is NOT a programming course! 2:

You learn CUDA, but our learning here is by using examples, running them, explaining the background in the lecture, and only make small changes to the program, if any. The course is too short to teach you writing CUDA codes from scratch and many features of CUDA will not be covered.

Data Handling!

For the homework you produce some simple data, and the task is to plot them. You can use anything you like (jupyter notebook, gnuplot, idl, …); we try to help – but again – this is not part of the course.

2. Getting access to the kepler computer

a. Every participant has to send me a public key (some have done already). You will all get an account on our kepler system with userid lecturenn where nn is an individual number which I will give to you (a list will be made).

b. It is recommended to work in a group of two persons (one also permitted). You can work on one account together if you wish. Or get two accounts, as you prefer.

c. Create an ssh key Linux/Unix Users (also possible for Windows/Mac):

`ssh-keygen -t rsa` (passphrase should NOT be empty)

It produces a private key `id_rsa` and a public key `id_rsa.pub`

Send the public key by email to spurzem@ari.uni-heidelberg.de
After this has been installed on kepler, you can login with

```
ssh lecturenn@kepler2.zah.uni-heidelberg.de
```

(lecturenn is your account on the system, such as lecture01, lecture02, …)

If you give a non-standard name to your private key, you need to do this:

```
ssh -i 'full_path_to_private_key'
lecturenn@kepler2.zah.uni-heidelberg.de
```

If it is e.g. in *home/data/xyz/.ssh/my_id.rsa* you need to use inside the '..' above.

For Windows Users:

- you can use a terminal window, which works like a Linux command window, compare https://www.wikihow.com/Use-SSH (check especially 'create encrypted keys' and the chmod command, that may cause a problem)
- Second Way: Use the putty client program for ssh; https://putty.org/ (here you need to make sure to copy and paste the openssh public key to send by email!).
- Third Way – open a Linux Shell on Windows or use a WLS installation

(ubuntu on Windows) – then you can follow the Linux informations above.

3. After login to the kepler computer

```
ls -lt
```

shows you files in your home directory, there should be **`gpu-course-master.tgz`** and **`gworknb6.tar.gz`** (later)

```
tar xvfz gpu-course-master.tgz
```

creates the subdirectory gpu-course:

```
cd gpu-course ; ls -ltr
```

shows the course exercises, in several subdirectories (0_hello, 1_add, …), next:

```
module load cuda ; module list
```

is needed to make the CUDA and NVIDIA software for GPU available.

4. How to run an exercise

Example of hello world program:

```
cd hello
nvcc -o hello.gpu hello.c
./hello.out
nvcc -o hello.gpu hello.cu
./hello.out
nvcc -o device.gpu device.cu
./device.out
```

(for those who like it: there is also a makefile). The -o option allows you to choose a name for your exectuable file (if not given Unix default name a.out). This little program produces output, saying there are no GPU devices. This is because the kepler computer has a batch system, we are on the login nodes, and not on the 12 worker nodes. Only the worker nodes have GPUs.

We need to use the slurm batch system to submit jobs to the worker nodes; the submission script is provided in gpu_script.sh :

```
sbatch gpu_script.sh
```

Information about the job and queue status (see also kepler Manual link on our course webpage):

```
squeue
sinfo
```

The result of the batch job will appear on job.nnnn.out and job.nnnn.err ; look for it with

```
ls -ltr
```

For reading and editing files you can use the Linux programs:

```
cat  ; more ; vi ; nano ; pico ; …?
```

Transfer data files to your own computer:

```
scp  lecturenn@kepler2.zah.uni-heidelberg.de:gpu-course/0_hello/job.2345.out .
```

(Note the .)