Revision 1

C2115 Practical introduction to supercomputing

Lesson 15

Petr Kulhanek
kulhanek@chemi.muni.cz

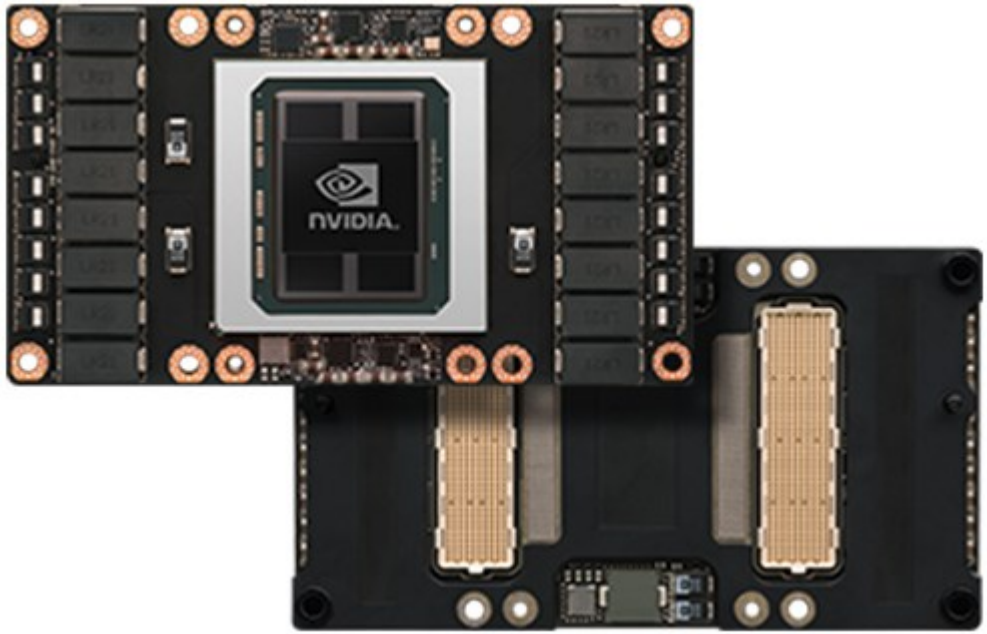
National Center for Biomolecular Research, Faculty of Science,
Masaryk University, Kotlářská 2, CZ-61137 Brno

Content

- **GPGPU**
  comparison to CPU
- **Running applications**
  pmemd on GPU

GPGPU

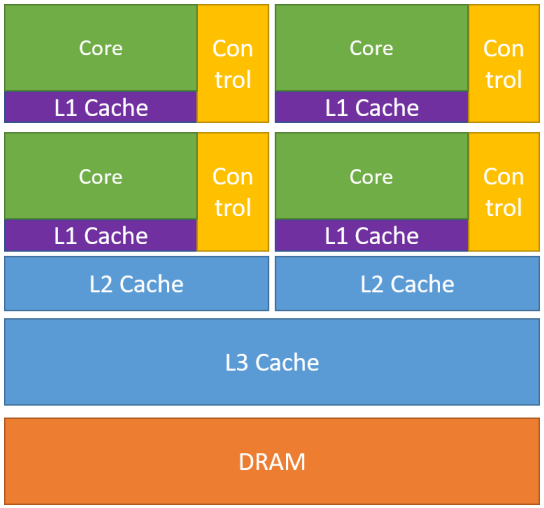
General-purpose computing on graphics processing units



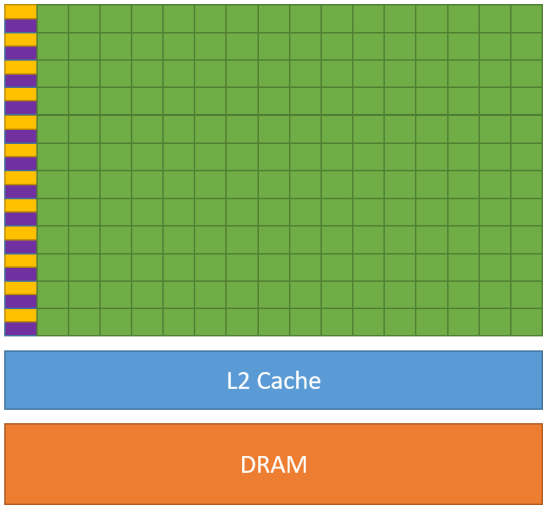
Nvidia Tesla P100

CPU vs GPU

https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html



CPU



GPU

The GPU contains a large number of GPU computing cores that are organized into groups (SM, streaming processors). The GPU performs computational operations on a group of data - vector data processing.

Example: Nvidia RTX 3070

- 5,888 CUDA colors
- 46 Streaming Multiprocessors

Demonstration video:
https://www.youtube.com/watch?v=-P28LKWTzrI

Using the GPU

Parallelization of jobs using GPU for their run requires non-trivial modifications in algorithms and the use of special development environments.

Programming method:

- Nvidia CUDA
- OpenCL (Nvidia, AMD, etc.)
- or the use of optimized libraries
  - cuBLAS, cuFFT, etc.

GPU task monitoring (Nvidia)

- In batch systems, only the assigned GPU are available to the job (affected by the CUDA_VISIBLE_DEVICES variable set by the batch system).

- The progress of job on the GPU can be monitored using the tool **nvidia-smi**.

```
[kulhanek@wolf30 main]$ nvidia-smi
Mon Mar  1 20:37:41 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  GeForce RTX 3070    Off  | 00000000:01:00.0 Off |                  N/A |
|  0%   48C    P8    15W / 240W |    257MiB /  7982MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  GeForce RTX 3070    Off  | 00000000:C1:00.0 Off |                  N/A |
|  0%   42C    P8     7W / 240W |      2MiB /  7982MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      9252      C   pmemd.cuda                       255MiB |
+-----------------------------------------------------------------------------+
```

Exercise

pmemd.cuda

- **pmemd** is a program for molecular dynamics. More detailed information can be found here: http://ambermd.org
- Script for GPU run of the application:

```
#!/bin/bash

# activate pmemd for GPU
module add pmemd-cuda:18.1

# run the application
pmemd.cuda -O -i prod.in -p 6000.parm7 \
                         -c 6000.rst7
```

Exercise 1

Input data is on the WOLF cluster in the directory:
/home/kulhanek/Documents/C2115/data/chitin

1. Determine the performance of pmemd per 1 GPU in ns per day and compare it with the most powerful run of pmemd on the CPU (exercise L14.E1).
2. Run the task on 1GPU in a separate directory in the Infinity environment, request the entire node (place=excl) and use the same computing node (props=vnode=wolf30).
3. Monitor the progress of the task on the computing node with the tool nvidia-smi.

Submitting the job into the batch system:

$ psubmit default run.sh **ngpus**=1 place=excl props=vnode=wolf30