

C2115

Praktický úvod do superpočítání

15. lekce

Petr Kulhánek
kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta,
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno

➤ GPGPU

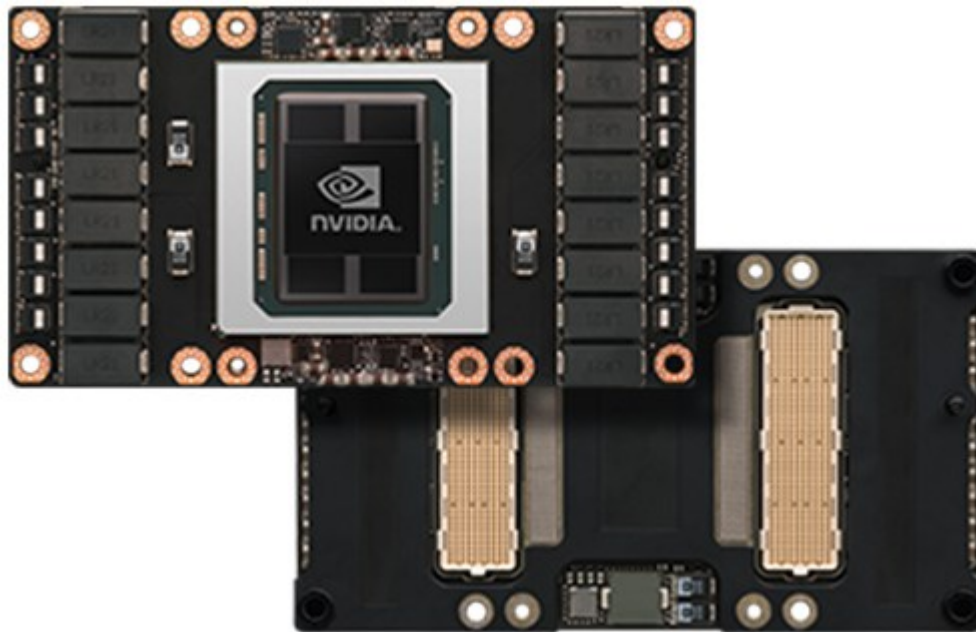
srovnání s CPU

➤ Spouštíme aplikace na GPU

- pmemd (MD simulace - chitinové vlákno)
- alphafold (predikce 3D struktury z primární sekvence)

GPGPU

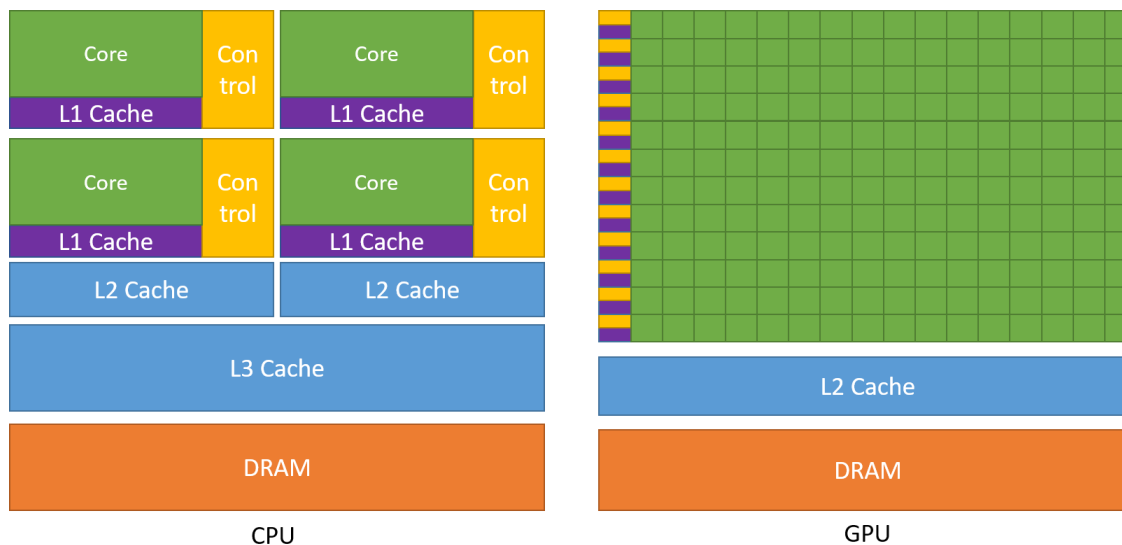
General-purpose computing on graphics processing units



Nvidia Tesla P100

CPU vs GPU

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



GPU obsahuje velké množství GPU výpočetních jader, které jsou organizované do skupin (SM, streaming processors). GPU vykonává výpočetní operace nad skupinou dat, jedná se tedy o vektorové zpracování dat.

Příklad: Nvidia RTX 3070

- 5,888 CUDA cores
- 46 Streaming Multiprocessors

Demonstrační video:

<https://www.youtube.com/watch?v=-P28LKWTzrl>

Použití GPU

Paralelizace úloh využívajících k běhu GPU vyžaduje netriviální úpravy v algoritmech a použití speciálních vývojových prostředí.

Způsob programování:

- Nvidia CUDA
- OpenCL (Nvidia, AMD, apod.)
- nebo využití optimalizovaných knihoven
 - cuBLAS, cuFFT, apod.

Monitoring GPU úloh (Nvidia)

- V dávkových systémech jsou úloze dostupné pouze přidělené GPU (ovlivňuje proměnná `CUDA_VISIBLE_DEVICES`, kterou nastavuje dávkový systém).
- Běh úloh na GPU je možné monitorovat pomocí nástroje **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 |                     |
| 0%   48C    P8      15W / 240W | 257MiB / 7982MiB |      0%   Default   |
|                                           |                     | N/A   |
+-----+-----+-----+
|  1   GeForce RTX 3070      Off          | 00000000:C1:00.0 Off |                     |
| 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
+-----+-----+-----+
```

Cvičení

pmemd.cuda

- **pmemd** je program určen pro molekulovou dynamiku. Podrobnější informace lze nalézt zde: <http://ambermd.org>
- Skript pro běh aplikace na GPU:

```
#!/bin/bash

# aktivovat pmemd pro GPU
module add pmemd-cuda:18.1

# spusteni aplikace
pmemd.cuda -O -i prod.in -p 6000.parm7 \
           -c 6000.rst7
```


Cvičení 1

Vstupní data jsou na klastru WOLF v adresáři:
</home/kulhanek/Documents/C2115/data/chitin>

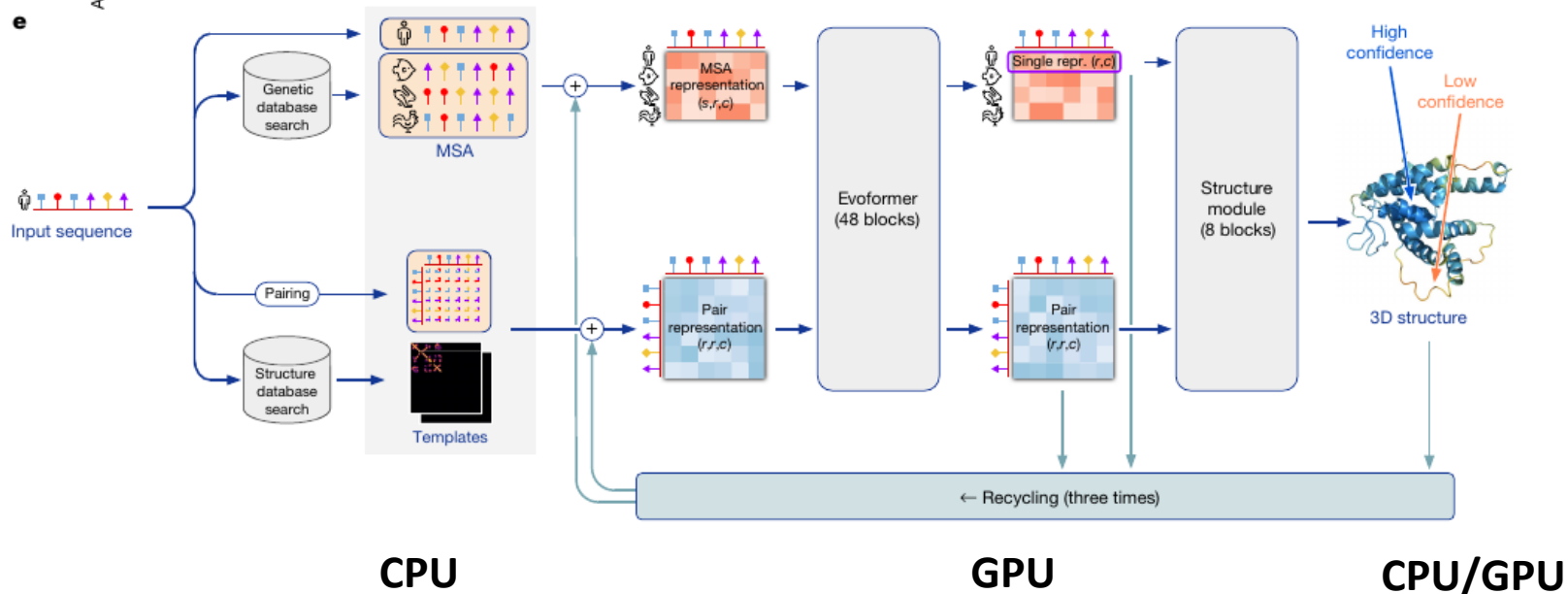
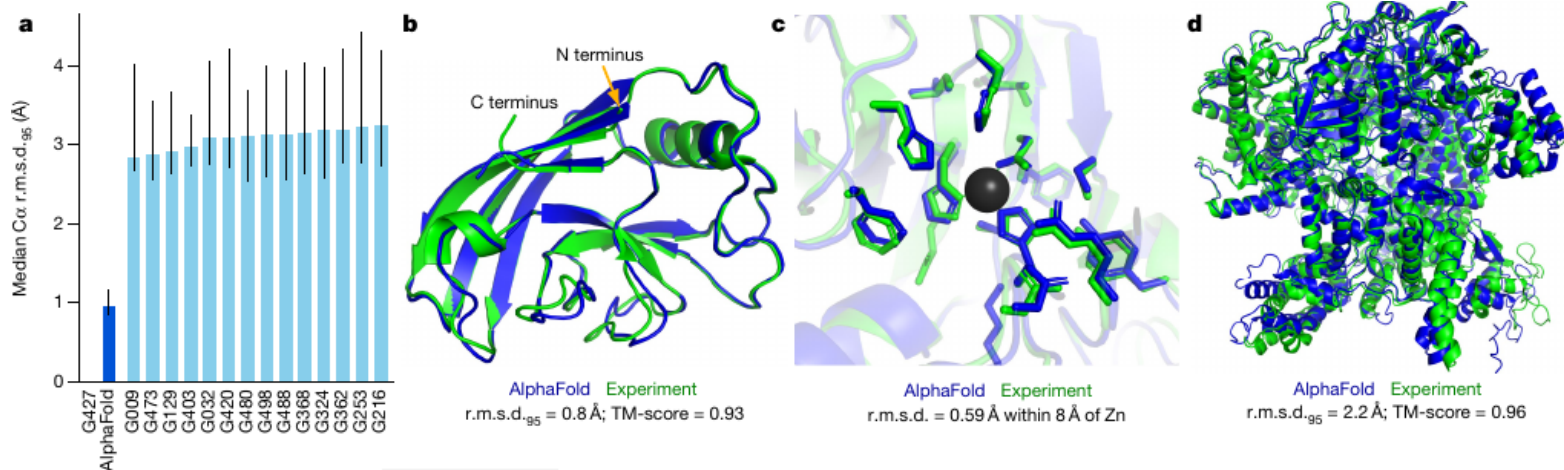
1. Určete výkon pmemd na 1 GPU v ns za den a srovnejte s nejvýkonnějším během pmemd na CPU (cvičení L14.C1).
2. Úlohu spusťte na 1 GPU v samostatném adresáři v prostředí Infinity, požadujte celý uzel (place=excl) a používejte stejný výpočetní uzel (vnode=wolf33).
3. Běh úlohy na výpočetním uzlu monitorujte nástrojem nvidia-smi.

Zadaní úlohy do dávkového systému:

```
$ qsub default run.sh ngpus=1 place=excl vnode=wolf33
```

AlphaFold

<https://doi.org/10.1038/s41586-021-03819-2>



Spuštění úlohy

- alphafold vyžaduje CPU i GPU
 - MSA běží na CPU
 - predikce struktury využívá GPU (pouze jedno GPU)
 - optimalizace geometrie využívá CPU/GPU
 - podrobnosti: module help alphafold

```
#!/usr/bin/env infinity-env
```

```
# aktivovat alphafold  
module add alphafold
```

```
# spuštění predikce pro input.fasta  
alphafold -f input.fasta
```

Cvičení 2

Vstupní data jsou na klastru WOLF v adresáři:
</home/kulhanek/Documents/C2115/data/chitin>

1. Proveďte predikci 3D struktury z primární sekvence aminokyselin pomocí programu Alphafold. Úlohu zadávejte na klastr WOLF.
2. Běh úlohy na výpočetním uzlu monitorujte nástrojem nvidia-smi.

Zadaní úlohy do dávkového systému:

```
$ qsub default alphafold.sh ncpus=8 ngpus=1 mem=40gb
```