

A SLICES-SC EU KUTATÁSI INFRASTRUKTÚRA HASZNÁLATÁNAK TAPASZTALATAI: AGYI BIOELEKTROMOS JELFELDOLGOZÁS MULTI-GPU SZÁMÍTÁSI KÖRNYEZETBEN

DR. JUHÁSZ ZOLTÁN
PANNON EGYETEM, VESZPRÉM

Villamosmérnöki és Információs
Rendszerek Tanszék

juhasz@virt.uni-pannon.hu

AGYI EEG JELEK FELDOLGOZÁSA



- nagy időbeli felbontás ($<ms$)
- nyugalmi vagy feladat centrikus mérések
- idő-frekvencia elemzés
- konnektivitás elemzés

Cél: új módszerek fejlesztése, amik javítják a térbeli felbontást és lehetővé teszik eddig rejtett jelenségek detektálását



AZ EEG FELDOLGOZÁS SZÁMÍTÁSI IGÉNYE

- Nagy adatmennyiség
 - **2 kHz** mintavételi frekvencia esetén: 2k adatpont/elektróda/mp → 512k adatpont 256 elektróda/mp → **10^9 adatpont** egy 30 perces mérés alatt
 - **32 kHz** mintavételi frekvencia esetén: **$> 10^{10}$ samples** for a 30 min measurement
- Néhány száz utasítás / adatpont esetén → $1-10 \times 10^{12}$ ops (Tera op/s)
- Az EEG feldolgozás meghatározó platformja a MATLAB, időigényes módszerek kizárva! Futási idő: több óra per alany

PÁRHUZAMOSÍTÁSI LEHETŐSÉGEK

A párhuzamosság több szinten is jelentkezik

- **alany:** csoport tagok egyidejű kiértékelése, 20-200 alany
- **csatorna:** 64-256 csatorna/elektróda egyidejű feldolgozása
- **minta:** adatpontok független feldolgozása
- **térbeli műveletek:** az összes elektróda egy időpillanatban
- **algoritmus:** pl. FFT

Nagyfokú párhuzamosság ($10^3 - 10^9$), ideális GPU futási „alany”

200-2000x gyorsulás MATLAB-hoz viszonyítva

A GPU PROGRAMVÉGREHAJTÁS JELLEMZŐI

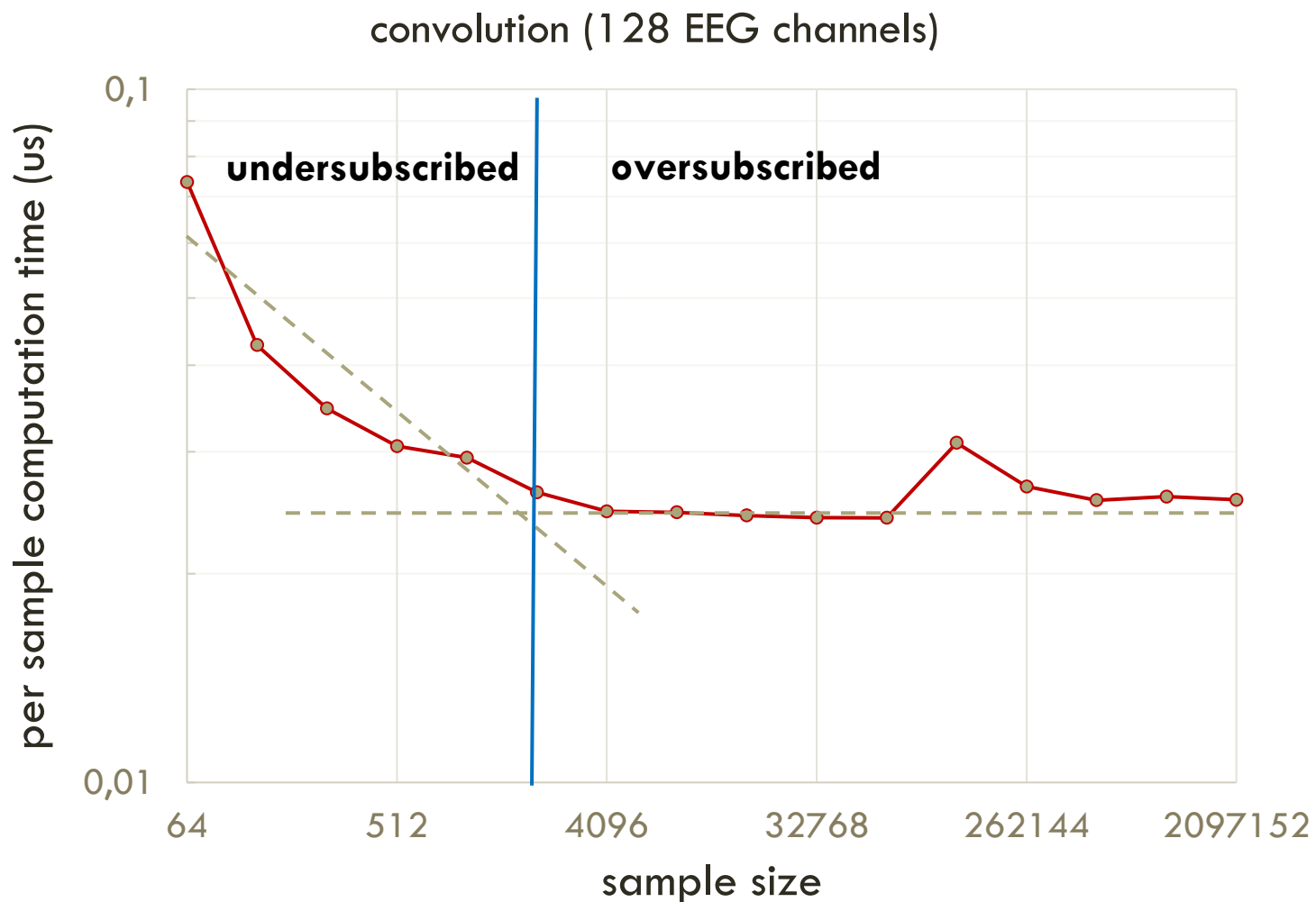
Fix számú mag és multi-processor

„Throughput” optimalizált

Kis terhelés esetén rossz hatékonyság

Nagy terhelés esetén nincs további gyorsulás

Több GPU alkalmazása?





- Dashboard
- Jobs
- All Jobs
- My Jobs
- Create Job
- Live
- Available Resources**

Available Resources Per Cluster

Cluster 6

 GPU Model

Tesla V100-SXM3-32GB
with 31 GB RAM


 Storage

/project_ghent/
/project_scratch/


Nodes

gpulab6A

 ONLINE

 **4** / 16 GPUs available

 **24** / 96 CPUs available

 **4** / 1583 GB CPU RAM available

A GPULAB RENDSZEREK HASZNÁLATA

GPULab GPU klaszterek interaktív és batch módban is használhatók

Interaktív: alapvetően az erőforrások és job-ok ellenőrzésére, lekérdezésére, adatmozgatásra, illetve interaktív futtatásokra --> JupyterHub

Batch: távoli végrehajtás a szuperszámítógépeken megszokott módon. Feladatbeküldés job leíróval történik.

A rendszer Docker konténer alapú, több előre konfigurált Docker image is elérhető

Nekünk egy kész GPU konténer felhasználásával kellett egy új Nvidia HPC SDK-t használó konténert készítenünk

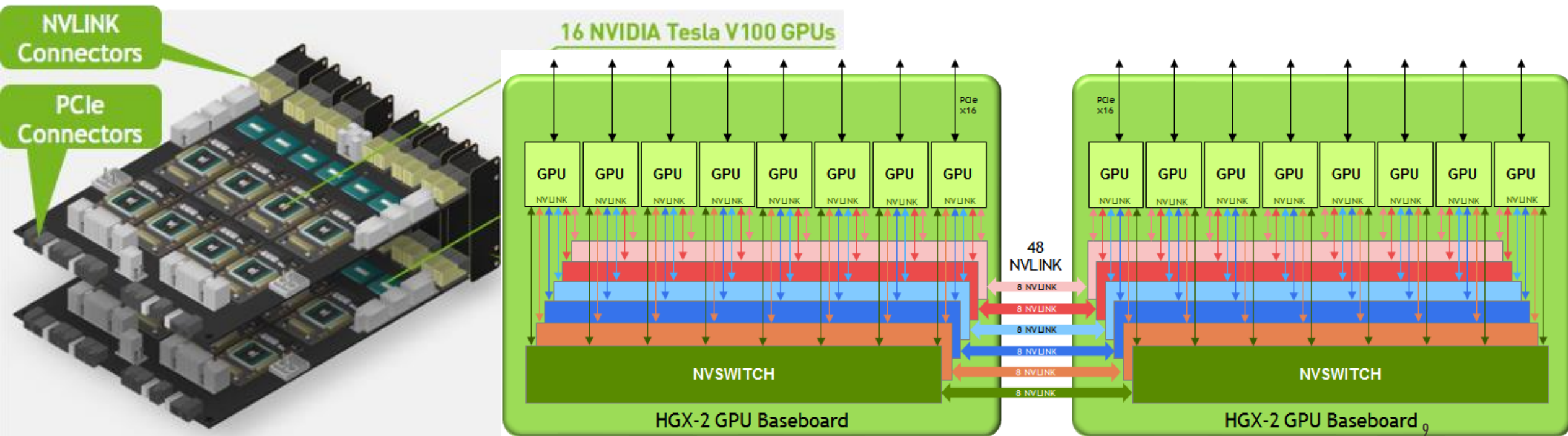
Példa egy feladat leíró fájlra

```
{
  "name": "CUDA_MEMD",
  "description": "Test_CUDA_MEMD",
  "request": {
    "docker": {
      "image": "gitlab.ilabt.imec.be:4567/ilabt/gpu-docker-stacks/code-server-notebook:latest",
      "command": "/project_ghent/startScript.sh",
      "storage": [
        {
          "hostPath": "/project_ghent",
          "containerPath": "/project_ghent"
        }
      ]
    },
    "resources": {
      "clusterId": 6,
      "cpus": 4,
      "gpus": 2,
      "cpuMemoryGb": 16,
      "minCudaVersion": 11
    },
    "scheduling": {
      "maxDuration": "1 min",
      "interactive": false
    }
  }
}
```


A GPULAB HGX-2 RENDSZER

A sok elérhető GPU klaszter közül nekünk a Cluster 6-ra volt szükségünk

- NVIDIA HGX-2 16 Tesla V100 GPU-val, 96 2.7Ghz vCPU mag + 1.5TB RAM
- V100: 5120 FP32 mag/chip, közel 15 Tflop/s FP32 számítási teljesítmény, szumma teljesítmény: 238 Tflop/s
- NVLink interconnect



A HAGYOMÁNYOS MULTI-GPU PROGRAMOZÁSI MODELL

Elosztott, message-passing programozási model

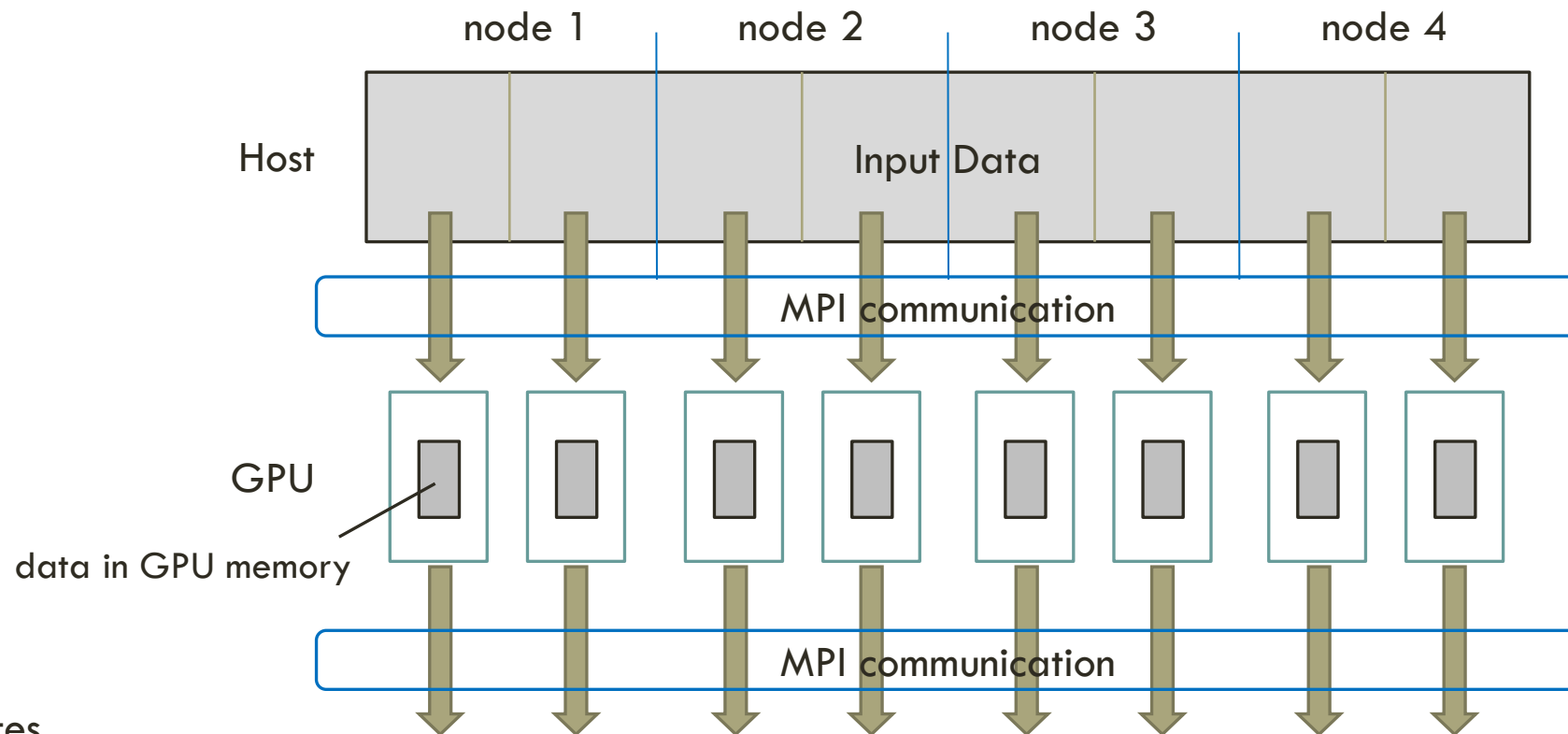
1. lépés: Adat szétosztás MPI segítségével a node-okra

2. lépés (opcionális): további MPI kommunikáció a GPU futtatás előkészítéséhez

3. lépés: GPU kód végrehajtása

4. lépés (opcionális): MPI kommunikáció az eredmények, köztes adatok GPU-k közötti szétosztására

5. lépés: eredmény összegyűjtés (redukció), másolás a host node-ra



KÖZVETLEN MULTI-GPU PROGRAMOZÁSI MÓDSZER

Shared-memory programming model with unified memory and asynchronous in-kernel communication

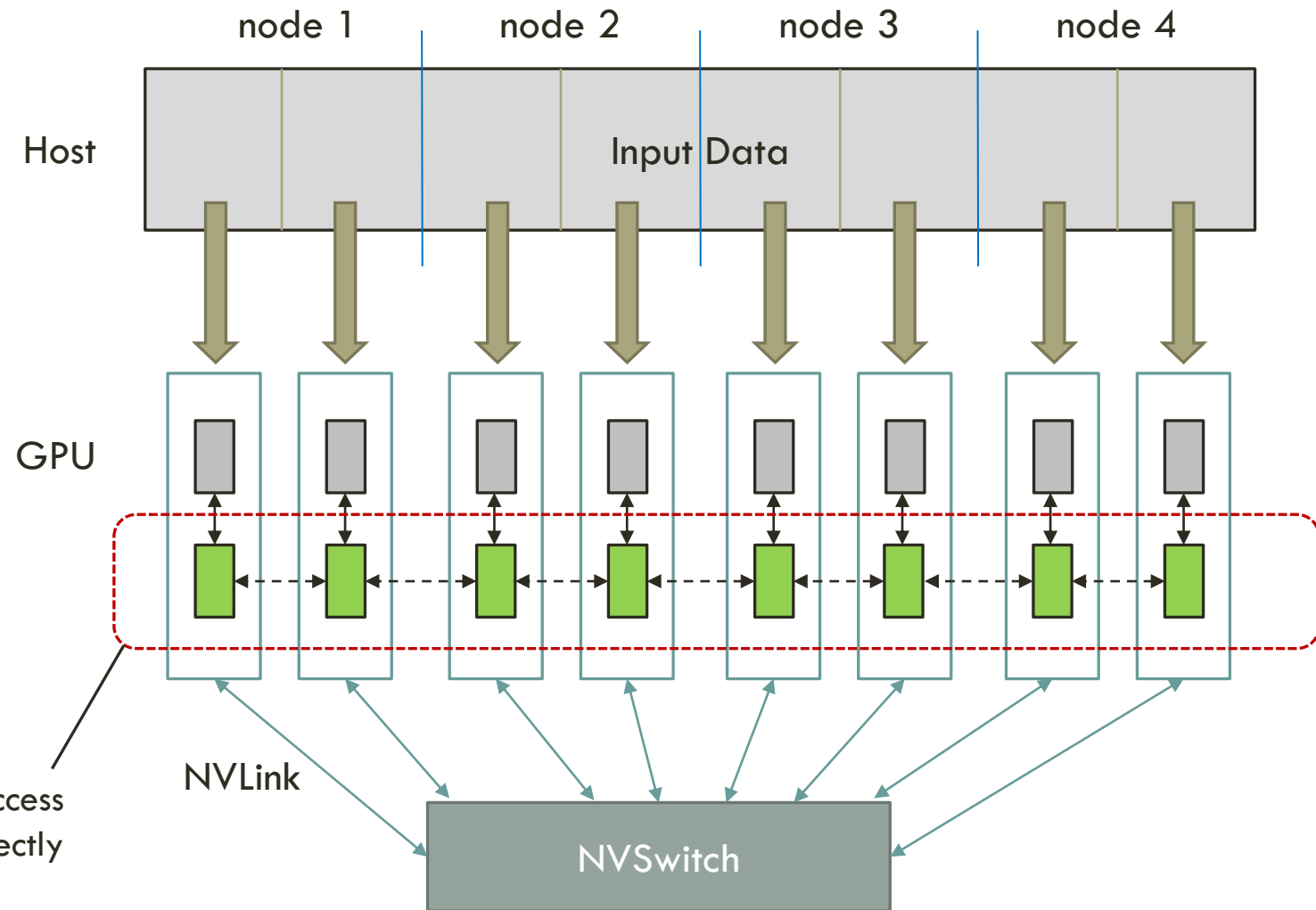
1. lépés: Adat szétosztás a node-ok között MPI segítségével

2. lépés: GPU kernel futtatás, távoli GPU memória írás/olvasás, kollektív kommunikáció NVLink csatornán keresztül

3. lépés: végeredmény másolás a host node-ra

...

GPU kernels access peer memory directly



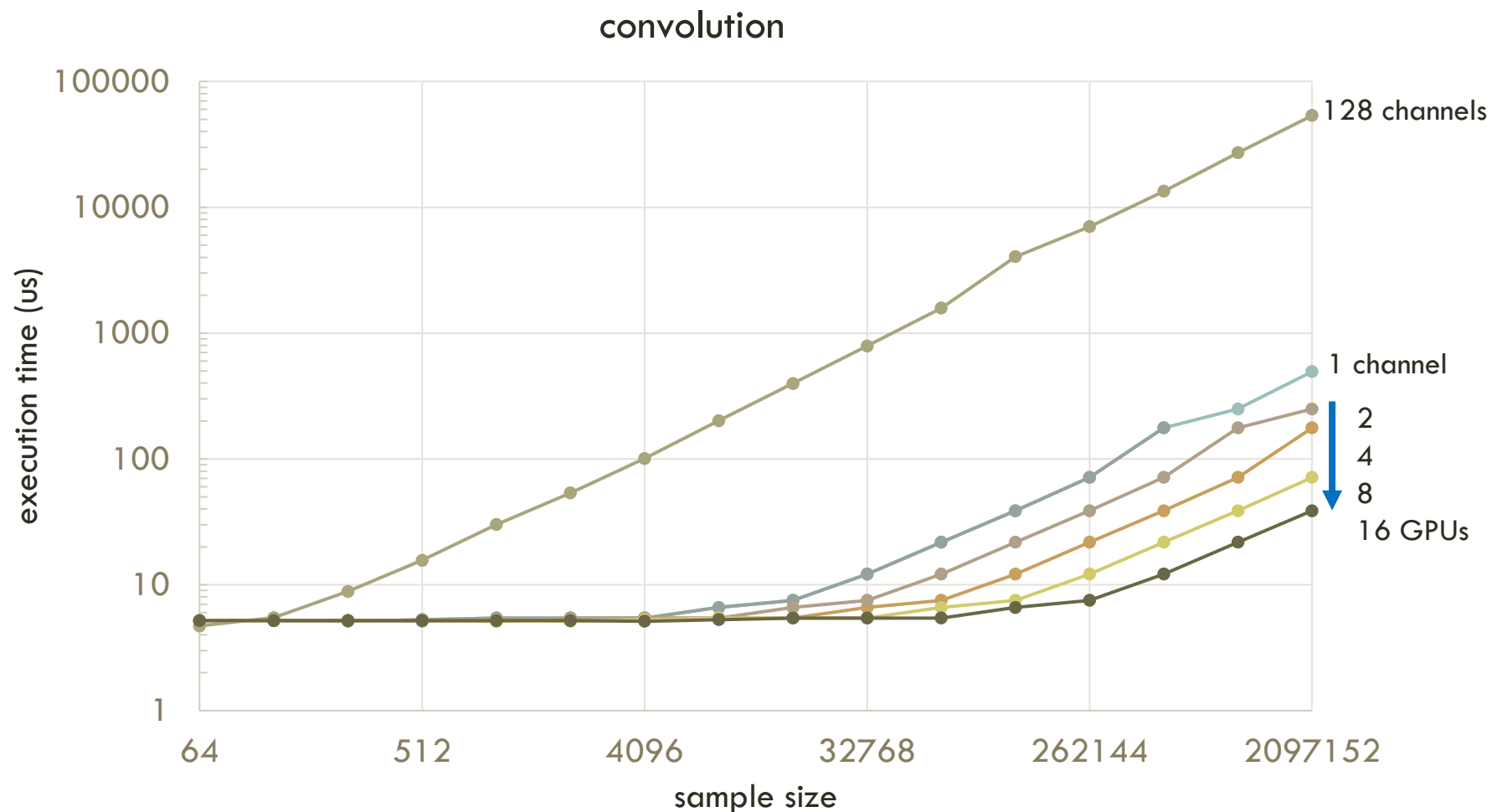
EGY ILLUSZTRATÍV PÉLDA

- GPU algoritmus futtatása és mérése egy V100 GPU-n

- 128 és 1 csatorna feldolgozási ideje az adathossz függvényében

- 1 csatorna adatainak particionálása és szétosztása 2 – 16 GPU kártyára párhuzamos végrehajtásra

- a multi-GPU változat illusztrálja a további gyorsulást még 1 csatorna párhuzamos feldolgozása esetén is



ÖSSZEFOGLALÁS

Az EEG adatfeldolgozás

A SLICES-SC projekt

Több kritikus EEG fe

Multi-GPU teljesítmé
változata.

2 doktorandusz hallg
kerete

A kutatást folytatni t
szuperszámítógép.

Folytatás a Komond



ával.

ettük el.

