

Korištenje grafičkih procesora

- [Pokretanje poslova](#)
 - [Korištenje jednog grafičkog procesora](#)
 - [Korištenje više grafičkih procesora na jednom čvoru](#)
 - [Korištenje više grafičkih procesora na više čvorova](#)
 - [Korištenje cijelih čvorova s grafičkim procesorima](#)
 - [Korištenje više procesorskih jezgri](#)
- [Korištenje scratch diska](#)
- [Nadzor GPU poslova](#)

Na Isabelli su dostupna tri poslužitelja Dell EMC PowerEdge C4140 s po:

- 4 grafička procesora NVIDIA Tesla V100-SXM2-16GB
- 2 procesora Intel Xeon Silver 4114 s ukupno 20 procesorskih jezgri
- 384 GB radne memorije
- Lokalnog spremišta od 3.2 TB NVMe SSD diska

Instalirani su NVIDIA alati i biblioteke:

- CUDA
- cuDNN 7.3.1
- NCCL 2.3.5.

Pripremljeni su moduli za dostupne verzije CUDA Toolkita koji postavljaju sve CUDA varijable okoline:

Verzija	Modul
9.0	cuda/9-0
10.0	cuda/10-0
10.1	cuda/10-1
10.2	cuda/10-2
11.0	cuda/11-0
11.1	cuda/11-1
11.2	cuda/11-2
11.3	cuda/11-3

Pokretanje poslova

U nastavku je opisano podnošenje tipičnih aplikacija koje koriste grafičke procesore. Više informacija o pokretanju poslova možete pronaći na stranici [Pokretanje i upravljanje poslovima](#), a o korištenju pojedinih aplikacija na [Korisničke aplikacije i knjižnice](#).



U svakom opisu posla **obavezno** treba koristiti *cuda-wrapper.sh*, kao u primjerima u nastavku.

Korištenje jednog grafičkog procesora

Za pokretanje poslova koji zahtijevaju jedan grafički procesor potrebno je u opisu posla postaviti:

```
#$ -pe gpu 1  
  
cuda-wrapper.sh aplikacija
```

Korištenje više grafičkih procesora na jednom čvoru

Za pokretanje poslova koji zahtijevaju više grafičkih procesora na jednom čvoru potrebno je koristiti paralelnu okolinu **gpusingle** te željeni broj grafičkih procesora (maksimalno 4):

```
#$ -pe gpusingle 4  
  
cuda-wrapper.sh aplikacija
```

Korištenje više grafičkih procesora na više čvorova

Korištenje više grafičkih procesora na više čvorova moguće je pomoću knjižnica MPI moduli prevedenih s podrškom za CUDA-u:

Verzija	Prevodilac	Verzija CUDA-e	Modul
MVAPICH2.2	Intel 2017	9.0	mpi/mvapich2-intel2017-cuda90-2.2-x86_64
MVAPICH2.2	Intel 2018	10.0	mpi/mvapich2-intel2018-cuda-2.2-x86_64
MVAPICH2.2	Intel 2019	10.1	mpi/mvapich2-intel2019-cuda101-2.2-x86_64
OpenMPI 1.10.7	Intel 2019	10.0	mpi/openmpi-intel-cuda-x86_64
OpenMPI 3.0.0	Intel 2019	10.0	mpi/openmpi3-intel-cuda-x86_64

U slučaju korištenja MVAPICH modula:

```
#$ -cwd  
#$ -pe gpu 8  
  
module load <mpi modul>  
  
mvapich-wrapper.sh aplikacija
```

U slučaju korištenja OpenMPI modula:

```
#$ -cwd  
#$ -pe gpu 8  
  
module load <mpi modul>  
  
openmpi-wrapper.sh aplikacija
```

Korištenje cijelih čvorova s grafičkim procesorima

Za pokretanje poslova koji zahtijevaju cijele čvorove potrebno je koristiti paralelnu okolinu **gpufull** te željeni broj grafičkih procesora (mora biti djeljitelj broja 4). Primjer skripte za zauzimanje dva čvora:

```
#$ -pe gpufull 8  
  
module load <mpi modul>  
  
mvapich-wrapper.sh aplikacija
```

Korištenje više procesorskih jezgri

Raspoređivač poslova pretpostavljeno svakom poslu na grafičkim procesorima pridjeljuje jednu CPU jezgru. Ukoliko je potrebno, moguće je tražiti veći broj jezgara:

```
#$ -l cores <broj_jezgri>
```





U svakom opisu posla **obavezno** treba koristiti odgovarajući wrapper s obzirom na vrstu posla:

- **cuda-wrapper-nopin.sh** - poslovi koji koriste jedan GPU
- **openmpi-wrapper-nopin.sh** - poslovi koji koriste više GPU-ova i OpenMPI biblioteku
- **mvapich-wrapper-nopin.sh** - poslovi koji koriste više GPU-ova i MVAPICH biblioteku

BITNO:

- navedeni broj se odnosi na broj procesorskih jezgri po zahtijevanom grafičkom procesoru. Ukoliko korisnički posao zahtijeva 6 grafičkih procesora i 24 procesorske jezgre potrebno je navesti:

```
#$ -pe gpu 6
#$ -l cores=4

mvapich-wrapper-nopin.sh aplikacija
```

- čvorovi raspolažu s 20 procesorskih jezgri pa je to potrebno uzeti u obzir prilikom navođenja parametra
- prilikom korištenja cijelih čvorova nije potrebno navoditi ovaj parametar



Broj zahtijevanih procesorskih jezgara ograničen je s obzirom na broj traženih grafičkih procesora. Maksimalne vrijednosti dane su u sljedećoj tablici:

Broj grafičkih procesora	Maksimalan broj procesorskih jezgri po grafičkom procesoru
1	17
2	9
3	6

Korištenje scratch diska

Za poslove koji zahtijevaju čest nasumičan pristup podacima na disku, kao što su TensorFlow i PyTorch poslovi, preporuča se korištenje diskova na radnim čvorovima. Općenite upute o korištenju scratch diskova mogu se pronaći na [Pokretanje i upravljanje poslovima](#).

Za svaki korisnički posao automatski se kreira privremeni direktorij na disku na radnom čvoru, do kojeg je putanja spremljena u SGE varijablu okoline **\$TMPDIR**. Primjer prebacivanja podataka na radni čvor:

moj_posao.sge

```
#!/bin/bash

#$ -N scratch_1
#$ -cwd
#$ -o output/scratch.out
#$ -j y
#$ -pe gpu 1
#$ -l scratch=100

# neka su podaci koje aplikacija koristi spremljeni u $HOME/data/data.tar

mkdir -p $TMPDIR/data
tar xf -C $HOME/data/data.tar $TMPDIR/data/

cuda-wrapper.sh python3.5 moja_aplikacija.py
```





Privremeni direktorij na /scratch disku se **automatski briše** nakon završetka posla. Ukoliko se tokom izvođenja aplikacije generiraju bitni izlazni podaci, potrebno ih je unutar skripte za opis posla kopirati u home direktorij.

Nadzor GPU poslova

Za pregled svih korisničkih **poslova koji se izvršavaju** na čvorovima s grafičkim procesorima koristi se naredba:

```
$ qstat -u "*" -q gpu.*.q
```

Ili je moguće koristiti naredbu **gpustat** koja pruža informaciju o poslovima koji se izvršavaju, na kojim procesorima se izvršavaju te kakvo opterećenje stvaraju:

```
$ gpustat
```

Za pregled poslova **u redu čekanja** za radne čvorove s grafičkim procesorima koristi se naredba:

```
$ gpuqw
```