



# Slurm – Workload Manager



ФИНКИ Компјутерски Центар

# Пристап и најава

---

1. За да пристапите најпрво треба да сте најавени преку [EDUVPN](#) системот на ФИНКИ
2. Преку [PuTTY](#) или друг SSH/Telnet клиент се најавувате на **gpu.finki.ukim.mk**
  - за да можете да се најавите потребно е претходно да ви е обезбеден пристап преку SSH клуч
  - за да побарате пристап треба да [пополните барање на info.finki.ukim.mk](#)
  - доколку се најавувате прв пат, можно е да ви побара лозинка и потребно е да внесете Laboratorija11!

# Перформанси и инсталирани верзии

Постојат четири CUDA сервери со следните графички картички инсталирани на нив до кои се пристапува преку Slurm Workload Manager

Хост	GPU	OS
gpu.finki.ukim.mk	/	Ubuntu 18.04 / Slurm 17.11.2
cuda1	#1: Quadro RTX 5000 16GB #2: Quadro RTX 5000 16GB	Ubuntu 18.04 / Slurm 17.11.2
cuda2	#1: Quadro RTX 5000 16GB #2: Quadro RTX 5000 16GB	Ubuntu 18.04 / Slurm 17.11.2
cuda3	#1: Quadro RTX 5000 16GB #2: Quadro RTX 5000 16GB	Ubuntu 18.04 / Slurm 17.11.2
cuda4	#1: Quadro RTX <b>8000 48GB</b> #2: Quadro RTX <b>8000 48GB</b>	Ubuntu 18.04 / Slurm 17.11.2

# Поднесување скрипта во Slurm преку Sbatch

---

- Се користи командата за доставување на скрипти во Slurm: **sbatch**  
[OPTIONS(0)...] [ : [OPTIONS(N)...]] script(0) [args(0)...]
- Sbatch ќе прикаже Error Code доколку постои грешка/проблем
- Скриптите потребно е да се напишат во .sh формат, во директориумот на корисникот (/home/users/<*user\_name*>) каде што понатаму се повикуваат на извршување

# (Најкористени) Параметри

Параметри и нивно значење:

```
#!/bin/bash

#SBATCH --ntasks-per-node=2      # Broj na zadaci po fizicko jadro na CPU
#SBATCH --time=1:00:00          # Vremetraenje na skripta (days-hrs:min:sec)
#SBATCH --job-name=test_job     # Ime na Job-ot
#SBATCH --mem=1G                # Ram memorija za izvrshvanje (primer 1G, 2G, 4G)
#SBATCH --error=testerror_%j.error # Pecatenje na greskite koi se pojavuvaat pri izvrshvanje na job-ot
#SBATCH --cpus-per-task=1       # Broj na procesori potrebni za edna zadaca
#SBATCH --output=testoutput_%j.out # Pecatenje na izlezot od skriptata I vrednostite koi gi izvrshva
#SBATCH --gres=gpu:2           # Broj na karticki na eden nod alocirani za job-ot
#SBATCH --nodelist=cuda4       # Izvrshvanje na specificni nodovi, pr. cuda4 e za izvrshvanje samo na cuda4 hostot
```

# Python - Креирање на виртуелна околина

---

Доколку е потребно да креирате виртуелна околина со специфична верзија на Python, извршете ја следната команда:

```
conda create -n virtualenv python=3.8
```

Преку оваа команда се креира виртуелна околина на корисникот во неговиот домашен директориум (/home/users/<user\_name>)

**\*Внимавајте, доколку користите една скрипта повеќе пати можно е да ви јави грешка доколку ја креирате истата виртуелна околина повеќе пати**

# Python – Користење на веќе креирана виртуелна околина

---

За да ја активирате креираната виртуелна околина, искористете ја следната команда:

```
conda activate virtualenv
```

За правилно да се користат виртуелните околинати при дефинирање на одреден job, потребно е да се вклучи патеката до anaconda и командите со кои работи conda, преку следните команди:

```
export PATH="/opt/anaconda3/bin:$PATH"  
source /opt/anaconda3/etc/profile.d/conda.sh
```

Потоа можете да започнете со користење на python во скриптата преку командата

```
python <ime>.py
```

# Пример со извршување на едноставна скрипта

```
#!/bin/bash

#SBATCH --job-name=test_job
#SBATCH --time=1:00:00
#SBATCH --ntasks-per-node=1
#SBATCH --error=testerror_%j.error
#SBATCH --output=testoutput_%j.out

export PATH="/opt/anaconda3/bin:$PATH"
source /opt/anaconda3/etc/profile.d/conda.sh
conda create -n virtualenv python=3.8
conda activate virtualenv

echo "FINKI FCC"
```

Скриптата се извршува преку `sbatch <иметонаскриптата>.sh`

На сликата подолу ни е прикажано дека скриптата е закажана:

```
marince@gpu:~$ sbatch echoscript.sh
Submitted batch job 121
marince@gpu:~$
```



# Пример параметри наменети за GPU

```
#!/bin/bash

#SBATCH --ntasks-per-node=2      # Broj na zadaci po fizicko jadro na CPU
#SBATCH --time=1:00:00          # Vremetraenje na skripta (days-hrs:min:sec)
#SBATCH --job-name=test_job     # Ime na Job-ot
#SBATCH --mem=1G                # Ram memorija za izvrshvanje (primer 1G, 2G, 4G)
#SBATCH --error=testerror_%j.error # Pecatenje na greskite koi se pojavuvaat pri izvrshvanje na job-ot
#SBATCH --cpus-per-task=1       # Broj na procesori potrebni za edna zadaca
#SBATCH --output=testoutput_%j.out # Pecatenje na izlezot od skriptata i vrednostite koi gi izvrshva
#SBATCH --gres=gpu:2            # Broj na karticki na eden nod alocirani za job-ot
#SBATCH --nodelist=cuda4        # Izvrshvanje na specifični nodovi, pr cuda4 e za izvrshvanje samo na cuda4 hostot

export PATH="/opt/anaconda3/bin:$PATH"
source /opt/anaconda3/etc/profile.d/conda.sh
conda create -n virtualenv python=3.8
conda activate virtualenv

echo "FINKI FCC"
```

Доколку вашата скрипта има потреба од повеќе GPU меморија за извршување искористете ја **cuda4** при дефинирање на **nodelist**, во спротивно користете ги **cuda1**, **cuda2** или **cuda3** хостовите.

# Опции за селекција на GPU меморија

Постојат 4 можности за селекција на GPU меморија и тоа може да се направи преку комбинирање на дел од командите во скриптата

GPU Memory	Код кој треба да се внесе во скриптата
16 GB GDDR6	<code>#SBATCH --gres=gpu:1</code> <code>#SBATCH --nodelist=cuda1 (или cuda2 или cuda3)</code>
32 GB GDDR6	<code>#SBATCH --gres=gpu:2</code> <code>#SBATCH --nodelist=cuda1 (или cuda2 или cuda3)</code>
48 GB GDDR6	<code>#SBATCH --gres=gpu:1</code> <code>#SBATCH --nodelist=cuda4</code>
96 GB GDDR6	<code>#SBATCH --gres=gpu:2</code> <code>#SBATCH --nodelist=cuda4</code>

# Примери со селекција GPU меморија

## Пример со 16 GB GPU

```
#!/bin/bash

#SBATCH --ntasks-per-node=2
#SBATCH --time=1:00:00
#SBATCH --job-name=test_job
#SBATCH --mem=1G
#SBATCH --error=testerror_%j.error
#SBATCH --cpus-per-task=1
#SBATCH --output=testoutput_%j.out
#SBATCH --gres=gpu:1
#SBATCH --nodelist=cuda1

export PATH="/opt/anaconda3/bin:$PATH"
source /opt/anaconda3/etc/profile.d/conda.sh
conda create -n virtualenv python=3.8
conda activate virtualenv

echo "FINKI FCC"
```

## Пример со 48 GB GPU

```
#!/bin/bash

#SBATCH --ntasks-per-node=2
#SBATCH --time=1:00:00
#SBATCH --job-name=test_job
#SBATCH --mem=1G
#SBATCH --error=testerror_%j.error
#SBATCH --cpus-per-task=1
#SBATCH --output=testoutput_%j.out
#SBATCH --gres=gpu:1
#SBATCH --nodelist=cuda4

export PATH="/opt/anaconda3/bin:$PATH"
source /opt/anaconda3/etc/profile.d/conda.sh
conda create -n virtualenv python=3.8
conda activate virtualenv

echo "FINKI FCC"
```

## Пример со 32 GB GPU

```
#!/bin/bash

#SBATCH --ntasks-per-node=2
#SBATCH --time=1:00:00
#SBATCH --job-name=test_job
#SBATCH --mem=1G
#SBATCH --error=testerror_%j.error
#SBATCH --cpus-per-task=1
#SBATCH --output=testoutput_%j.out
#SBATCH --gres=gpu:2
#SBATCH --nodelist=cuda1

export PATH="/opt/anaconda3/bin:$PATH"
source /opt/anaconda3/etc/profile.d/conda.sh
conda create -n virtualenv python=3.8
conda activate virtualenv

echo "FINKI FCC"
```

## Пример со 96 GB GPU

```
#!/bin/bash

#SBATCH --ntasks-per-node=2
#SBATCH --time=1:00:00
#SBATCH --job-name=test_job
#SBATCH --mem=1G
#SBATCH --error=testerror_%j.error
#SBATCH --cpus-per-task=1
#SBATCH --output=testoutput_%j.out
#SBATCH --gres=gpu:2
#SBATCH --nodelist=cuda4

export PATH="/opt/anaconda3/bin:$PATH"
source /opt/anaconda3/etc/profile.d/conda.sh
conda create -n virtualenv python=3.8
conda activate virtualenv

echo "FINKI FCC"
```

# Проверка на статус на job-от

Статусот на job-от може да се провери преку командата **squeue** која ни ги покажува следните информации:

- **JOB ID**
- **Partition** – партиција на задачата
- **Name** – име на задачата
- **USER** – име на корисникот кој ја извршува задачата
- **ST** – состојба на job-от (најчести се **PD** – Pending, **R** – Running, **S** – Suspended, **CG** – Completing, **CD** – Completed)
- **NODES** – Број на nodes поврзани со задачата
- **TIME** – Поминато време за извршување на задачата
- **NODELIST (REASON)** – Покажува каде задачата се извршува или причина зошто е сеуште на чекање. (најчести се Resources (се чека ресурсите да станат достапни или доколку на некој од job-овите има подесено dependency)

```
marince@gpu:~$ squeue
JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
122_1  /home/use serial_j goran.pe R      5:28     1  cuda3
122_[2] /home/use serial_j goran.pe PD     0:00     1  (Resources)
122_1  /home/use serial_j goran.pe R      5:34     1  cuda3
marince@gpu:~$
```

# Дополнителни команди

---

- **sinfo** – Проверка на статусот на нодовите и партициите во кластерот
- **scancel** – Запирање на задачата (или повеќе задачи), со дефинирање на id
- **sacct** – Информации за завршените и тековните задачи, како и корисниците кои ги стартувале
- **sstat** – Податоци за тековните задачи и информации за корисниците

# Ограничувања и фер политика

---

За оптимално искористување на ресурсите се моментално се подесени следните ограничувања:

- еден корисник може да извршува скрипта во максимално времетраење од 24 часа, по истекот на времето скриптата ќе биде исклучена од страна на системот
- еден корисник може да извршува два “job”-ови во исто време кои ќе се активни (ќе процесираат), доколку има закажани повеќе од два останатите ќе бидат во статус PENDING, односно ќе се чека да завршат првите два

Зависно од искористеноста на ресурсите, истите е можно да претрпат измени кои ќе бидат објавени на [info.finki.ukim.mk](http://info.finki.ukim.mk)

# Корисни линкови и документација

---

- **Документација за SLURM 17.11.2** - [https://slurm.schedmd.com/archive/slurm-17.11.2/man\\_index.html](https://slurm.schedmd.com/archive/slurm-17.11.2/man_index.html)
- **SLURM SBATCH команди** - <https://slurm.schedmd.com/archive/slurm-17.11.2/sbatch.html>
- **Проверка на статус на Job** - <https://slurm.schedmd.com/archive/slurm-17.11.2/queue.html>
- **Инсталација на PyTorch со conda** - <https://www.gcptutorials.com/post/how-to-install-pytorch-with-conda>