

Инструкции за ползване на високопроизводителния изчислителен клъстер в ИИКТ-БАН

За достъп до клъстера въведете следната команда:

```
ssh <username>@gw.ipp.acad.bg
```

Въведете името на потребителския си акаунт (**student<number>**) и паролата си.

Копирайте примерите за обучението, като използвате следната команда:

```
cp -r /opt/exp_software/documentation/Feb2013Training/Examples .
```

В нашия първи пример ще изпълним следната задача, която ще отпечата „**Hello Wolrd!**” на екрана. Отидете в директорията **/home/student<number>/Examples/Hello** (**cd Examples/Hello**) и въведете следната команда:

```
cat hello.sh
```

```
#!/bin/bash
```

```
#PBS -q lifesci
```

```
echo „Hello World!”
```

За да изпратите тази първа задача, използвайте следната команда:

```
qsub hello.sh
```

Командата ще изведе следния резултат:

```
<Job_ID>.torq.hpcg
```

Вашата задача ще бъде завършена почти веднага и двата изхода, стандартния изход и стандартния изход за грешка ще бъдат в **hello.sh.o <Job_ID>** и **hello.sh.e <Job_ID>** съответно. Въведете следните команди.

```
cat hello.sh.o<Job_ID>
```

```
cat hello.sh.e<Job_ID>
```

Също така, можете да укажете изходните файлове с помощта на **-o** и **-e** като опции в **qsub**:

```
qsub -o my_output -e my_error hello.sh
```

Вторият ни пример ще отпечата изчислителните ресурси, предоставени за изпълнението на задачата. Отидете в директорията **/home/student<number>/Examples/Resources** и отпечатайте скрипта **res.sh** (използвайте командата **cat res.sh**).

```
#!/bin/bash
```

```
# PBS -q lifesci
```

```
#PBS -l nodes=2:ppn=4
```

```
cat $PBS_NODEFILE
```

Редът **#PBS -l nodes=2:ppn=4**, означава, че нашата задача ще се изпълни на 2 възела и ще използва 4 логически ядра от всеки възел. Това може да бъде модифицирано чрез промяна на скрипта **res.sh** или с помощта на опцията **-l** на командата **qsub**. Изпратете скрипта като използвате следната команда:

```
qsub -l nodes=2:ppn=4 res.sh
```

Отпечайте изхода:

```
cat res.sh.o <Job_ID>
```

С помощта на командата **qstat** можете да получите информация за всички задачи на даден потребител. Въведете следната команда:

```
qstat -u <username>
```

Както споменахме по-рано задачите са подадени до опашки и командата **qstat** ни позволява да следим опашките. Нека използваме следните команди и разгледаме техните изходи:

qstat -q ще отпечата информация за всички опашки.

qstat -q lifesci ще отпечата информация за опашката **lifesci**.

```
qdel <Job_ID>
```

Ако вашата задача е в опашката, когато изпращате тази команда, тя ще бъде премахната от опашката.

MPI задачи

Сега, когато знаем как да се изискват ресурси за нашите изчислителни задачи и как се управляват време е да видим как може да изпълнят MPI задачи на клъстера.

Отидете в директорията **/home/student<number>/Examples/HelloMPI**. Има два файла в директорията - **helloMPI.c** и **helloMPI.sh**. Разгледайте ги като използвате **cat**.

За да компилирате тази програма използвайте следната команда:

```
mpicc helloMPI.c -o helloMPI
```

Това ще създаде изпълним файл **helloMPI** на програмата. За да го стартирате, изпълнете следната команда:

```
mpirun -np 4 ./helloMPI
```

Това ще стартира 4 процеса изпълняващи вашата програма. Алтернативно, можете да изпратите задачата си с помощта на командата **qsub helloMPI.sh**.

Проверете статуса на задачата си с помощта на **qstat** и/или проверете дали резултатите са налице като използвате команда **ls**.

Отидете в директорията **/home/student<number>/Examples/MPIscatterReduce**. За да видите съдържанието въведете командата **ls**.

За да компилирате програмата използвайте командата **make**. Прегледайте **program_scatter.c**, **shell.sh**, **Makefile**. Изпратете задачата с **qsub**.

Други два примера има в директории **/Examples/MPIBcast** и **/Examples/MPIscatterReduce**.