

# Запуск задач через планировщик

2022

# Запуск задачи

Скрипт:

```
#!/bin/sh  
echo "Hello world"
```

Запуск:

```
$ chmod +x job.sh      # (2) сделать скрипт исполняемым  
$ sbatch job.sh       # (3) поставить задачу в очередь  
Submitted batch job 35014  
$ cat slurm-35014.out  
Hello world
```

# Переменные среды

Скрипт:

```
#!/bin/sh  
env | grep SLURM | sort
```

Запуск:

```
$ chmod +x job.sh  
$ sbatch job.sh  
Submitted batch job 35015  
$ cat slurm-35015.out  
...  
SLURM_JOB_ID=35015  
SLURM_JOB_NUM_NODES=1  
SLURM_JOB_NODELIST=m1  
...
```

# Запрос ресурсов внутри скрипта

Запрос 2-х узлов по 4 ядра на каждом:

```
#!/bin/sh
#SBATCH --nodes=2 --ntasks=8 --cpus-per-task=1
echo $SLURM_JOB_NODELIST
echo $SLURM_JOB_NUM_NODES
```

Вывод:

```
$ chmod +x job.sh
$ sbatch job.sh
Submitted batch job 35023
$ cat slurm-35023.out
m[4-5]
2
```

# MPI hello world (код)

Программа:

```
#include <mpi.h>
#include <stdio.h>
int main(int argc, char** argv) {
    int rank, nprocs, namelen;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    char name[MPI_MAX_PROCESSOR_NAME];
    MPI_Get_processor_name(name, &namelen);
    printf("Hello world: processor %s, rank %d/%d\n",
           name, rank+1, nprocs);
    MPI_Finalize();
    return 0;
}
```

Скрипт:

```
#!/bin/sh
#SBATCH --nodes=2 --ntasks=8
module load mpi/openmpi-local
mpirun hello-world
```

Компиляция и запуск:

```
$ module load mpi/openmpi-local
$ mpicc hello.c -o hello-world
$ sbatch job.sh
Hello world: processor m4, rank 1/8
Hello world: processor m4, rank 3/8
Hello world: processor m4, rank 4/8
Hello world: processor m4, rank 2/8
Hello world: processor m5, rank 6/8
Hello world: processor m5, rank 8/8
Hello world: processor m5, rank 5/8
Hello world: processor m5, rank 7/8
```

# OpenFOAM

Настройка:

```
$ rm -rf $HOME/damBreak
$ export SINGULARITY_IMAGE=/box/singularity/openfoam.sqsh
$ singularity shell
Singularity openfoam.sqsh:~> . /opt/openfoam6/etc/bashrc
Singularity openfoam.sqsh:~> cp -r \
    $FOAM_TUTORIALS/multiphase/interFoam/laminar/damBreak \
    $HOME
$ cd $HOME/damBreak
$ mv 0/alpha.water.org 0/alpha.water
$ ls
0 constant system
```

# OpenFOAM

Скрипт:

```
#!/bin/sh
#SBATCH --nodes=1 --ntasks=2
cd $HOME/damBreak
cat > system/decomposeParDict << EOF
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       decomposeParDict;
}
numberOfSubdomains $SLURM_NTASKS;
method scotch;
EOF
```



```
export SINGULARITY_IMAGE=/box/singularity/openfoam.sqsh
singularity run blockMesh
singularity run decomposePar -force
module load mpi/openmpi-local
mpirun singularity run interFoam -parallel
```

Запуск:

```
$ sbatch job.sh
Submitted batch job 35033
```

Просмотр очереди задач:

```
$ squeue
JOBID PARTITION  NAME  USER ST TIME  NODES NODELIST
35025      ant  job.sh myuser  R 0:01      2 m[4-5]
```

# OpenFOAM

Вывод:

```
$ tail slurm-35033.out
time step continuity errors : sum local = 4.21498e-32, global = 2.46346e-48, cumulative = 2.3875e-47
DICPCG: Solving for p_rgh, Initial residual = 2.09176e-12, Final residual = 2.09176e-12, No Iterations 0
time step continuity errors : sum local = 4.21505e-32, global = 1.77753e-48, cumulative = 2.56525e-47
DICPCG: Solving for p_rgh, Initial residual = 2.09176e-12, Final residual = 2.09176e-12, No Iterations 0
time step continuity errors : sum local = 4.21505e-32, global = 2.1862e-48, cumulative = 2.78387e-47
ExecutionTime = 1.48 s  ClockTime = 7 s

End

Finalising parallel run
```

# Выводы

- ▶ Для одной задачи (task) по умолчанию выделяется одно ядро процессора.
- ▶ Скрипт всегда запускается только на одном из узлов.
- ▶ Список узлов автоматически передается в команду `mpirun`.
- ▶ Предполагается, что каждый узел «видит» одни и те же файлы в рабочей директории.

© 2018–2022 Ivan Gankevich [i.gankevich@spbu.ru](mailto:i.gankevich@spbu.ru)

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. The copy of the license is available at <https://creativecommons.org/licenses/by-sa/4.0/>.