**Инструкция по запуску задач с помощью SLURM**

Запуск задач на кластере проводится с помощью системы очередей SLURM.

Порядок действий:

1. командой *sinfo*определяем загруженность узлов

*$* ***sinfo***

*PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST*

*normal\*      up   infinite      1    mix n2-3*

*normal\*      up   infinite      7   idle n1-[1-4],n2-[1-2,4]*

Откуда следует, что очередь (*PARTITION*) имеет название *normal*.

Узлы доступны для использования и имеют состояние *up*.

Время исполнения задачи (*TIMELIMIT*) не ограничено (*infinite*)

Состояние узлов:

*n1-[1-4],n2-[1-2,4]* – свободны (*idle*)

*n2-3* – частично загружен (*mixed state*).

Максимальное число логических ядер (CPU) на узле – 48.

Допустим, программа будет исполняться на числе ядер < 48.

Целесообразно загружать узлы максимально плотно с тем, чтобы другие пользователи могли при необходимости задействовать максимальное количество свободных узлов. Для этого необходимо проверить частично загруженный узел на предмет количества свободных ядер командой *scontrol show node n2-3*:

*$* ***scontrol show node n2-3***

*NodeName=n2-3 Arch=x86\_64 CoresPerSocket=24*

***CPUAlloc=12****CPUTot=48 CPULoad=11.74*

*AvailableFeatures=(null)*

*ActiveFeatures=(null)*

*Gres=(null)*

*NodeAddr=n2-3 NodeHostName=n2-3 Version=20.02.5*

*OS=Linux 4.18.0-193.19.1.el8\_2.x86\_64 #1 SMP Mon Sep 14 14:37:00 UTC 2020*

*RealMemory=1 AllocMem=0 FreeMem=122655 Sockets=2 Boards=1*

***State=MIXED****ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS\_label=N/A*

*Partitions=normal*

*BootTime=2020-11-30T14:38:50 SlurmdStartTime=2020-11-30T14:40:04*

*CfgTRES=cpu=48,mem=1M,billing=48*

*AllocTRES=cpu=12*

*CapWatts=n/a*

*CurrentWatts=0 AveWatts=0*

*ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s*

Из полученных данных следует, что на узле можно еще выделить еще максимум 36 CPU. Для этого в скрипте ***Start.sh***  необходимо указать

***#!/bin/bash***

***#SBATCH --nodes=1****# Number of nodes*

***#SBATCH --nodelist=n2-3****# Node name*

***#SBATCH --ntasks-per-node=1****# Number of processes per node*

***#SBATCH --cpus-per-task=32****# Number of threads per process*

***#SBATCH --job-name=MPI\_prog*** *# Будет отображаться в очереди*

#***SBATCH --output=out.txt #*** *Сюда попадет вывод на экран*

*#****SBATCH --error=error.txt #*** *Сюда попадут ошибки запуска*

***mpirun ./a.out*** *# в случае запуска MPI задачи*

***./a.out*** *# в случае запуска OpenMP задачи*

Пояснения:

*a.out* – исполняемый файл, получающийся после компиляции по умолчанию.

*#SBATCH --nodes=1             #*  число выделяемых узлов

*#SBATCH --nodelist=n2-3   #*  имя выделяемого узла

*#SBATCH --ntasks-per-node=1   #*  количество процессов для MPI задачи.

*#SBATCH --cpus-per-task=32    #*  количество потоков на MPI процесс. Важно при запуске OpenMP или гибридной задачи.

*#SBATCH --job-name=* *MPI\_prog*  # название задачи в очереди

Запуск задачи на исполнение в случае использование скрипт-файла:

*$* ***sbatch ./Start.sh***

*Submitted batch job 294*

Задачи можно ставить и без использования скрипт-файла. Но со скрипт-файлом удобнее. Например

***srun -n1 ./a.out*** – постановка задачи в очередь в составе одного процесса.

Состояние загрузки узла:

*$****sinfo***

*PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST*

*normal\*      up   infinite      1****alloc****n2-3*

*normal\*      up   infinite      7   idle n1-[1-4],n2-[1-2,4]*

***$ scontrol show node n2-3***

*NodeName=n2-3 Arch=x86\_64 CoresPerSocket=24*

*CPUAlloc=48 CPUTot=48 CPULoad=11.68*

*AvailableFeatures=(null)*

*ActiveFeatures=(null)*

*Gres=(null)*

*NodeAddr=n2-3 NodeHostName=n2-3 Version=20.02.5*

*OS=Linux 4.18.0-193.19.1.el8\_2.x86\_64 #1 SMP Mon Sep 14 14:37:00 UTC 2020*

*RealMemory=1 AllocMem=0 FreeMem=122655 Sockets=2 Boards=1*

*State=****ALLOCATED****ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS\_label=N/A*

*Partitions=normal*

*BootTime=2020-11-30T14:38:50 SlurmdStartTime=2020-11-30T14:40:04*

*CfgTRES=cpu=48,mem=1M,billing=48*

*AllocTRES=cpu=48*

*CapWatts=n/a*

*CurrentWatts=0 AveWatts=0*

*ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s*

Таким образом, узел *n2-3* оказывается загруженным полностью.

Если нужны все ядра узла, что в скрипте надо задать:

***--cpus-per-task=48***

А также указать имя свободного узла (например, n2-4):

***--nodelist=n2-4***

Проверить состояние запущенной задачи можно с помощью команды *squeue***:**

*$* ***squeue***

*JOBID PARTITION   NAME   USER ST       TIME  NODES NODELIST(REASON)*

*291    normal  MPI\_prog yukul****R****15:24:31      1 n2-3*

*294    normal  MPI\_prog yukul****R****7:39      1 n2-3*

Для удаление своей задачи из очереди необходимо использовать команду *scancel*, после которой нужно написать номер своей задачи*:*

*$* ***scancel JOBID***

----------------------------------------------------------------------------------------------------

Полная документация доступна на сайте проекта:

<https://www.schedmd.com/>