

Instrukcja obsługi klastra Slurm

Instrukcja dla użytkownika

Spis treści

Spis treści	1
Zlecanie zadań	2
Wstęp	2
Srun	2
Sbatch	2
Sprawdzanie stanu zleconych zadań	4
Squeue	4
Scontrol	4
Anulowanie zadania	6
Sprawdzanie logów	6
Sprawdzanie stanu klastra	7

Zlecanie zadań

Wstęp

Istnieją dwa podstawowe mechanizmy zlecania zadań w klastrze slurm. Różnią się od siebie przeznaczeniem i sposobem działania.

Srun

Pierwszy mechanizm wykorzystuje polecenie *srun* i pozwala na natychmiastowe, interaktywne wykonywanie komend. Wynik działania komendy wyświetla się od razu po jej zakończeniu w terminalu, z którego została uruchomiona. Za pomocą polecenia *srun* można na przykład rozpocząć zdalną sesję:

```
srun --pty /bin/bash
```

Podłączyć się do już trwającego zadania celem modyfikacji jego parametrów:

```
srun --jobid=job_id --pty /bin/bash
```

Lub wykonać dowolne polecenie, na przykład celem wykonania testów na węzłach:

```
srun hostname
```

Dokładny opis dostępnych możliwości można przeczytać na oficjalnej stronie z dokumentacją Slurma pod adresem <https://slurm.schedmd.com/srun.html>

Sbatch

Drugi mechanizm zlecania zadań w Slurm wykorzystuje polecenie *sbatch*. Za pomocą tego polecenia można odpalać bardziej zaawansowane zadania, których treść mamy zapisaną w skrypcie powłoki. Zadania zlecone za pomocą *sbatch* wykonują się w tle i nie wymagają aktywnego nadzoru użytkownika. Zadania *sbatch* zwykle nie

wykonują się od razu tylko trafiają do kolejki, z której zostaną przekazane do wykonania w momencie pojawienia się wymaganych zasobów.

Zlecenie przykładowego zadania za pomocą polecenia sbatch może wyglądać następująco:

```
sbatch /sciezka/do/skryptu.sh
```

Zlecając zadania za pomocą sbatch dobrą praktyką jest określić ich zapotrzebowanie na zasoby oraz limit czasowy na wykonanie zadania. Opcjonalnie można również określić ścieżki do plików, do których trafią wyniki działania skryptu.

Wyżej wspomniane parametry można doprecyzować za pomocą argumentów dodanych do wywołania polecenia sbatch:

```
sbatch /sciezka/do/skryptu.sh --time=00:02:00 --mincpus=1 --mem=100M
```

Lub w nagłówku odpalanego skryptu:

```
#!/bin/bash
#SBATCH --job-name=NAME
#SBATCH --output=/tmp/job.out
#SBATCH --error=/tmp/job.err
#SBATCH --time=00:02:00
#SBATCH --ntasks=1
#SBATCH --mincpus=1
#SBATCH --mem=100M

echo "Tutaj umiesc swoje zadanie";

exit 0;
```

Wywołanie polecenia sbatch powinno zwrócić ID zadania, którego można następnie użyć przy sprawdzaniu jego stanu.

```
user@lab-net-58 $ sbatch /tmp/test.sh
Submitted batch job 138
```

Dokładny opis dostępnych możliwości można przeczytać na oficjalnej stronie z dokumentacją Slurma pod adresem <https://slurm.schedmd.com/sbatch.html>

Sprawdzanie stanu zleconych zadań

Squeue

Aby sprawdzić stan zleconych zadań można posłużyć się poleceniem `squeue`, które pokaże zadania aktywnie wykonywane oraz te, które czekają w kolejce:

```
user@lab-net-58 $ squeue
JOBID PARTITION NAME      USER      ST  TIME  NODES  NODELIST(REASON)
169    dcc      task123  inf12345  R   2:13   7      lab-net-56
```

Polecenie `squeue` można wykonywać z różnymi argumentami, na przykład

```
squeue --start
```

pokaże listę zadań, wraz z ich przewidywanym czasem rozpoczęcia, a polecenie

```
squeue -u inf12345 -i 5
```

pokaże listę zadań użytkownika `inf12345`, która będzie się automatycznie odświeżać co 5 sekund.

Pełną dokumentację polecenia `squeue` można znaleźć na stronie Slurma pod adresem <https://slurm.schedmd.com/squeue.html>

Scontrol

Aby pozyskać szczegółowe informacje o uruchomionym za pomocą `sbatch` zadaniu można posłużyć się poleceniem `scontrol`, jako parametr polecenia podając ID zadania, które nas interesuje. Polecenie to może być przydatne przy próbie debugowania.

```
user@lab-net-58 $ scontrol show job 138
```

```
JobId=138 JobName=test_sleep
  UserId=inf12345(17449) GroupId=students(120) MCS_label=N/A
  Priority=4294901753 Nice=0 Account=ldap QOS=normal
  JobState=RUNNING Reason=None Dependency=(null)
  Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
  RunTime=00:15:08 TimeLimit=01:00:00 TimeMin=N/A
  SubmitTime=2025-01-15T22:08:12 EligibleTime=2025-01-15T22:08:12
  AccrueTime=2025-01-15T22:08:12
  StartTime=2025-01-15T22:08:12 EndTime=2025-01-15T23:08:12 Deadline=N/A
  SuspendTime=None SecsPreSuspend=0 LastSchedEval=2025-01-15T22:08:12
Scheduler=Backfill
  Partition=dcc AllocNode:Sid=lab-net-58:9905
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=lab-net-56
  BatchHost=lab-net-56
  NumNodes=1 NumCPUs=2 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
  ReqTRES=cpu=1,mem=256M,node=1,billing=1
  AllocTRES=cpu=2,node=1,billing=2
  Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
  MinCPUsNode=1 MinMemoryNode=256M MinTmpDiskNode=0
  Features=(null) DelayBoot=00:00:00
  OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
  Command=/tmp/test_sleep.sh
  WorkDir=/home/inf12345
  StdErr=/home/inf12345/slurm-138.out
  StdIn=/dev/null
  StdOut=/home/inf12345/slurm-138.out
  Power=
```

Pełną dokumentację polecenia `scontrol` można znaleźć na stronie Slurma pod adresem <https://slurm.schedmd.com/scontrol.html>

Anulowanie zadania

Aby anulować zlecone zadanie można posłużyć się poleceniem *scancel*

```
scancel <jobID>
```

Pełną dokumentację polecenia *scancel* można znaleźć pod adresem <https://slurm.schedmd.com/scancel.html>

Sprawdzanie logów

Slurm loguje zdarzenia w plikach, znajdujących się w folderze */var/log/slurm/*. W przypadku problemów z uruchomieniem zadania można otworzyć logi celem diagnozy problemu, na przykład za pomocą polecenia:

```
tail -n 30 -f /var/log/slurm/slurmctld.log
```

Sprawdzanie stanu klastra

Aby sprawdzić aktualny stan klastra można posłużyć się utworzonym specjalnie do tego celu interaktywnym zestawem wykresów, dostępnym pod adresem <http://lab-net-58/nodes> jak również, alternatywnie, wykorzystać polecenie *sinfo*:

```
user@lab-net-58 $ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
dcc*      up       9:00:00      2   idle lab-net-[56-57]
```

W celu otrzymania listy węzłów klastra, które aktualnie borykają się z problemami, wraz ze skrótowym wyjaśnieniem ich powodu można posłużyć się poleceniem:

```
sinfo -R
```

Pełna dokumentacja polecenia *sinfo* jest dostępna pod adresem <https://slurm.schedmd.com/sinfo.html>