Pracownia Technik Obliczeniowych

SLURM

Paweł Daniluk

Wydział Fizyki

Wiosna 2016

Przetwarzanie wsadowe

Zasoby superkomputera z reguły dzielone są pomiędzy wielu użytkowników i wiele zadań.

Użytkownik przygotowuje zlecenie – wsad, który jest następnie przetwarzany.

System kolejkowy odpowiada za szeregowanie zgłaszanych wsadów i sukcesywne ich wykonywanie.

Zasoby

Wsady mogą różnić się wymaganiami:

- czas
- liczba węzłów klastrowych
- procesorów
- rdzeni
- pamięć operacyjna
- zasoby dyskowe
- urządzenia specjalne (GPU itp.)
- dostępne oprogramowanie

Rolą systemu kolejkowego jest wybierać zadania, które mogą być wykonane i "sprawiedliwie" przydzielać im zasoby.

SLURM

Simple Linux Utility for Resource Management

Podstawowe polecenia

- `srun` – uruchamia zadanie w trybie interaktywnym
- `sbatch` – uruchamia zadanie w trybie wsadowym
- `squeue` – wyświetla kolejkę zadań
- `scancel` – anuluje zadanie
- `scontrol` – służy do zarządzania wszystkim
- `sinfo` – wyświetla informacje o dostępnych zasobach

Pojęcia

Węzeł (node)

Serwer w klastrze. Pojedynczy fizyczny komputer połączony z pozostałymi siecią.

Partycja (partition)

Logiczny wycinek klastra obejmujący wskazane węzły. Może mieć określone dodatkowe ograniczenia, priorytety, zasady i uprawnienia.

Zadania interaktywne

```
srun [OPTIONS...] executable [args...]
```

```
[pawel@sh ~]$ srun hostname
msys28
```

W szczególności można w ten sposób uruchomić instancję shella.

```
[pawel@sh ~]$ srun --pty bash -l
[pawel@msys28 ~]$
```

Specyfikowanie zasobów

Wymagania wobec węzła

- `--sockets-per-node=<sockets>` – liczba procesorów w węźle
- `--cores-per-socket=<cores>` – liczba rdzeni w procesorze
- `--threads-per-core=<threads>` – liczba wątków na rdzeniu
- `--cpu-freq =<requested frequency in kilohertz>` – minimalna częstotliwość taktowania
- `--mincpus=<n>` – minimalna liczba logicznych procesorów
- `--mem=<MB>` – minimalna ilość pamięci
- `--mem-per-cpu=<MB>` – minimalna ilość pamięci na CPU

Wybór maszyn i partycji

- `-w, --nodelist=<host1,host2,... or filename>` – wybór konkretnych węzłów
- `-p, --partition=<partition_names>` – wybór partycji

Specyfikowanie zasobów c.d.

Opis wsadu

- `-n, --ntasks=<number>` – liczba zadań
- `-c, --cpus-per-task=<ncpus>` – liczba procesorów na zadanie
- `--time-min=<time>` – czas allokacji
- `-N, --nodes=<minnodes[-maxnodes]>` – liczba węzłów
- `-D, --chdir=<path>` – katalog obliczeń
- `--exclusive` – wyłączność allokacji węzłów
- `-J, --job-name=<jobname>` – nazwa wsadu
- `--pty` – symulacja terminala

Zadania wsadowe

```
sbatch [options] script [args...]
```

Skrypt opisujący wsad

```
#!/bin/sh
#
#SBATCH --job-name=testJob
#SBATCH --time=01:00:00
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --partition=dragon-default
#
# Display all variables set by slurm
env | grep "^SLURM" | sort
```

Podgląd stanu systemu

squeue

```
[pawel@sh ~]$ squeue
     JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
     48546    normal DT3b1a_1 charzews  R 12-02:13:30      1 msys16
     48547    normal DT3b1a_1 charzews  R 12-02:13:30      1 msys17
     48548    normal DT3b1a_1 charzews  R 12-02:13:30      1 msys17
     48549    normal DT3b1a_1 charzews  R 12-02:13:30      1 msys17
     48550    normal DT3b1a_1 charzews  R 12-02:13:30      1 msys18
     48551    normal DT3b1a_1 charzews  R 12-02:13:30      1 msys18
     48552    normal DT3b1a_1 charzews  R 12-02:13:30      1 msys18
     49368    normal DT3b1aWa charzews  R  5-19:14:04      1 msys28
```

sinfo

```
[pawel@sh ~]$ sinfo
PARTITION    AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal*         up 21-00:00:0      1  drain msys15
normal*         up 21-00:00:0     20    mix msys[1-8,11-14,16-18,21-23,27-28]
normal*         up 21-00:00:0      3   idle msys[24-26]
preemptible     up 21-00:00:0      1  drain msys15
preemptible     up 21-00:00:0     20    mix msys[1-8,11-14,16-18,21-23,27-28]
preemptible     up 21-00:00:0      3   idle msys[24-26]
```

Zarządzanie zasadami

scancel

`scancel [OPTIONS...] [job_id[_array_id][.step_id]] [job_id[_arr`

Przerywa zadania i usuwa je z kolejki.

scontrol

Wybrane komendy:

- `hold`, `release` – zatrzymuje i zwalnia zadanie w kolejce
- `resume`, `suspend` – zatrzymuje i uruchamia pracujące zadanie
- `show` – wyświetla informacje o wskazanym elemencie systemu (zadaniu, partycji)
- `requeue` – zwraca zadanie do kolejki
- `update` – zmienia właściwości zadania

Zależności

Można zdefiniować wsady, których wykonanie zależy od innych obliczeń. W tym zastosowaniu bardzo przydatna jest możliwość nazywania wsadów.

`-d, --dependency=<dependency_list>`
Możliwe zależności:

- `after:job_id[:jobid...]` – po rozpoczęciu wsadów `job_id`
- `afterany:job_id[:jobid...]` – po zakończeniu wsadów `job_id`
- `afternotok:job_id[:jobid...]` – po nieudanym zakończeniu wsadów `job_id`
- `afterok:job_id[:jobid...]` – po udanym zakończeniu wsadów `job_id`
- `singleton` – po zakończeniu wsadów o tej samej nazwie

Zadanie 0

"Zaloguj" się na węzeł klastra zlecając uruchomienie powłoki systemu.

Zadanie 1

Uruchom trwające 60 sekund zadanie, które wyświetla i zapisuje do pliku nazwę maszyny, na której zostało uruchomione.

Zadanie 2

Uruchom skrypt z poprzedniego zadania na 12 procesorach.

Zadanie 3

Uruchom skrypt z poprzednich zadania na 3 różnych maszynach.

Zadanie 4

Uruchom dwa wsady: ten z zadania 2 i dodatkowy, który po zakończeniu pierwszego zliczy maszyny, na które trafiły zadania.