

Tipps & Tricks

Es gibt eine Reihe von kleinen Tools, um Informationen zum Cluster-Status zu bekommen. Diese Tools findet man in dem [Modul](#) hpc-tools.

Cluster-Auslastung

Mit dem Skript cluster_usage.py kann man sich die Auslastung des Clusters (oder einer Queue oder einer speziellen Architektur) anzeigen lassen. Hier einige Beispiele:

```
$ module load hpc-tools
$ cluster_usage.py himem
$ cluster_usage.py a100
$ cluster_usage.py rtx6000
$ cluster_usage.py skylake
```

- [Cluster-Auslastung](#)
- [GPUs Available](#)
- [Queue-Warteschlange](#)
- [Snapshots finden](#)
- [Job Logs](#)

GPUs Available

Das Skript gpus_available gibt eine Liste von GPU-Typen und deren Verfügbarkeit aus

```
$ gpus_available
{'rtx6000': 4, 'teslat4': 4, 'rtx8000': 4, 'gtx1080ti': 103, 'rtx2080': 19, 'a100': 0}
```

Queue-Warteschlange

Das Skript queue_priority.py gibt eine Liste aller wartenden Jobs und deren aktuelle Priorität aus. Diese Prioritäten steigen von alleine mit der Zeit an und der oberste Job wird vom PBS als nächstes abgearbeitet. Bei wichtigen Jobs kann man allerdings auch über den Einsatz von [Bonuspunkten](#) die Priorität erhöhen und dadurch einen Job schneller starten lassen.

```
$ queue_priority.py
1104935.hpc-batch = 720.82
1104936.hpc-batch = 720.82
1104937.hpc-batch = 720.82
1104938.hpc-batch = 720.82
1105085.hpc-batch = 719.56
1107616.hpc-batch = 674.2
1107625.hpc-batch = 674.04
1107626.hpc-batch = 674.03
1107627.hpc-batch = 674.03
...
```

Snapshots finden

Das Tool locate_snapshot gibt einem die Möglichkeit alle Snapshots mit entsprechender Datumsangabe zu einem bestimmten Pfad zu lokalisieren. Dies funktioniert für die aktiven Filesysteme /home und /gpfs. Weitere Informationen zu Snapshots finden Sie hier: [Datensicherung per Snapshots](#)

```
$ locate_snapshot /gpfs/project/phreh100/test.0.0
Path: /gpfs/project/phreh100/test.0.0
abs Path: /gpfs/project/phreh100/test.0.0
Base Location: /gpfs/project/.snapshots
/gpfs/project/.snapshots/8hours_-2021.09.24-02.01.09/phreh100/test.0.0
/gpfs/project/.snapshots/week-2021.09.19-01.00.46/phreh100/test.0.0
/gpfs/project/.snapshots/week-2021.09.12-01.00.44/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.16-22.00.54/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.23-22.00.55/phreh100/test.0.0
/gpfs/project/.snapshots/8hours_-2021.09.23-10.01.09/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.21-22.00.55/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.18-22.00.54/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.19-22.00.54/phreh100/test.0.0
/gpfs/project/.snapshots/8hours_-2021.09.23-18.01.09/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.22-22.00.55/phreh100/test.0.0
/gpfs/project/.snapshots/week-2021.09.05-01.00.43/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.17-22.00.54/phreh100/test.0.0
/gpfs/project/.snapshots/day-2021.09.20-22.00.54/phreh100/test.0.0
/gpfs/project/.snapshots/week-2021.08.22-01.00.41/phreh100/test.0.0
/gpfs/project/.snapshots/week-2021.08.29-01.00.42/phreh100/test.0.0
```

Job Logs

Üblicherweise bekommt man erst nach Abschluss eines Jobs die Möglichkeit zur Einsicht in seine Ausgaben. Aber auch während ein Job läuft, ist es möglich in dessen Ausgaben (StdOut und StdErr) zu schauen. Die entsprechenden Skripte dafür heißen `job_live_log` und `job_live_log_error`. Beide benötigen die Job-ID des zu inspizierenden Jobs als Parameter:

Beispiel

```
$ job_live_log 1377439
$ job_live_log_error 1377439
```