

Besonderheiten

- Unterscheidung der Architekturen
- Verteilung der Chunks
- Konkreten GPU-Typ anfordern

PBSPro bietet zur besseren Platzierung der Jobs einige Attribute um auf das Verhalten Einfluss zu nehmen. Jedoch versuchen wir die Anzahl dieser Optionen möglichst gering zu halten, da man sonst schnell durcheinander kommt oder fehlerhafte Anforderungen stellt.

Unterscheidung der Architekturen

Der Scheduler ist so konfiguriert, dass zunächst versucht wird einen Job auf einem der kleinen Knoten laufen zu lassen und nur dann auf die [UV2000](#) auszuweichen, wenn ansonsten kein Platz ist (auf Grund von Ressourcenauslastung oder Anforderungen).

Auf Grund der Unterschiedlichen Architekturen (Bsp.: Ivybridge vs. Sandybridge-Prozessoren) kann es vorkommen, dass ein Job obwohl er klein ist besser auf der UV2000 läuft oder umgekehrt.

Um beim Absenden des Jobs darauf Einfluss zu nehmen, gibt es das Attribut *arch*, welches folgende Werte kennt:

```
arch=ivybridge
arch=uv2000
```

Jobs welche mit *arch=uv2000* abgesendet werden, starten nur auf der UV2000 und umgekehrt bei *arch=ivybridge* startet der Job nur auf den kleinen MPI-Knoten.

Beispielhafte Anwendung

```
#PBS -l select=1:ncpus=24:mem=10GB:arch=uv2000
```

Besonders wenn man viele BLAST-Jobs abschickt, sollte der Parameter *arch=ivybridge* genutzt werden, da es zu Problemen sonst auf der UV2000 kommen kann, die die Ausführungsgeschwindigkeit deutlich reduzieren.

Verteilung der Chunks

PBSPro ermöglicht es zu bestimmen, ob die einzelnen Chunks auf demselben Host laufen dürfen oder müssen oder ob ein Knoten exklusiv benötigt wird. Dazu gibt es das Attribut *place* welches folgende Werte besitzt:

```
place = free
place = pack
place = scatter
place = vscatter
```

Wenn der Parameter nicht gesetzt wird, verwendet das Batch-System den Wert *free* und verteilt die Chunks beliebig über alle Knoten. Bei *pack* werden alle Chunks auf einen Knoten gelegt. Bei *scatter* werden alle Chunks auf unterschiedliche Knoten verteilt, was besonders für Netzwerk-Benchmark-Anwendungen interessant ist. Zusätzlich gibt es noch die Option *vscatter* um die Chunks über vNodes zu verteilen, was besonders bei der UV2000 von Bedeutung ist.

Beispiel 1

```
#PBS -l select=2:ncpus=12:mem=10GB:arch=uv2000
#PBS -l place=pack
```

Beispiel 1 würde den Job auf die UV2000 legen und dort drei vNodes beanspruchen, da die UV2000 pro vNode nur 8 Cores hat. Diese vNodes wären jedoch direkt nebeneinander, sodass es zu möglichst wenig Geschwindigkeitseinbußen käme.

Beispiel 2

```
#PBS -l select=2:ncpus=12:mem=10GB
#PBS -l place=scatter
```

Beispiel 2 würde auf zwei verschiedenen Nodes auf dem Cluster gestartet werden, obwohl die zwei Chunks auch auf einen einzelnen MPI-Knoten passen würden. Dabei ist zu beachten, dass der Job auch auf einer vNode der UV2000 und einem MPI-Knoten mit unterschiedlicher CPU-Architektur und Taktrate starten kann. Dies muss sowohl bei der Kompilierung als auch bei der Laufzeit (Sync-Barrieres zwischen den Prozessen) berücksichtigt werden. Es empfiehlt sich also in diesem Fall auch das Attribut *arch* zu verwenden.

Konkreten GPU-Typ anfordern

Das Batchsystem erlaubt es einen speziellen GPU-Typ anzufordern. Dazu muss im Select-Statement der Wert `accelerator_model` gesetzt werden.

Folgende Werte werden derzeit unterstützt:

Wert	GPU-Typ	Jahr	Anzahl
gtx1080ti	Nvidia GTX 1080 TI	2017	120
rtx8000	Nvidia Quadro RTX 8000	2020	4
rtx6000	Nvidia Quadro RTX 6000	2020	40
a100	Nvidia A100	2020	32

Beispiel 2: GTX 1080Ti anfordern

```
#PBS -l select=1:ncpus=2:ngpus=1:accelerator_model=gtx1080ti
```

Beispiel 3

```
#PBS -l select=1:ncpus=2:ngpus=4
```

In Beispiel 3 werden 4 GPUs auf einem Server angefordert