

Übungen zur Vorlesung: Praktische Parallelprogrammierung Blatt 0 (Einführung)

Für die Parallelprogrammierung stehen 2 unterschiedliche Systeme zur Verfügung:

- Der Cluster **hermes**, der aus dem Frontend-Rechner `hermes.fim.uni-passau.de` und z.Z. 6 Arbeitsknoten besteht. Das Frontend und die Arbeitsknoten haben je 2 PentiumIII CPUs mit 1 GHz und 512 MB RAM.
- Die SMP-Maschine **hydra** mit 8 Quadcore AMD Opteron CPUs mit 2,3 GHz und insgesamt 64 GB RAM. Der Zugriff auf **hydra** erfolgt über den Frontend-Rechner `ravel.fim.uni-passau.de`.

Aufgabe 1 (Einloggen auf dem Parallelrechner)

Nachdem Ihre Kennung auf dem Cluster **hermes** und im Infosun-Pool (für **ravel** und **hydra**) eingerichtet worden ist (Benutzeranträge erhalten Sie bei Armin Größlinger, R 103 IM) können Sie sich mit

```
ssh Kennung@hermes.fim.uni-passau.de
```

auf Hermes einloggen. Das Frontend **ravel** für **hydra** erreichen Sie mit

```
ssh Kennung@ravel.fim.uni-passau.de
```

Legen Sie in Ihrem Homeverzeichnis eine Datei `.forward` mit dem Inhalt

```
Kennung@fim.uni-passau.de
```

o. ä. an, damit E-Mails, die Ihnen von **hermes** geschickt werden, an eine Kennung weitergeleitet werden, die Sie regelmäßig abfragen. E-Mails von **ravel/hydra** werden automatisch an die CIP-Pool Kennung geleitet.

Aufgabe 2 (Übersetzen auf den Frontends)

Auf den Frontendrechnern **hermes** bzw. **ravel** können Sie Anwendungen wie Texteditor etc. starten (z.B. **xemacs**); ggf. müssen Sie sich per **ssh -X ...** einloggen um die X11-Weiterleitung zu aktivieren.

Ebenfalls auf dem Frontend können Sie verschiedene Compiler aufrufen. Probieren Sie (die Datei **minibeispiel.c** finden Sie in Stud.IP) auf **hermes**:

```
gcc4 -Wall -fopenmp -o minibeispiel minibeispiel.c
```

bzw.

```
gcc -Wall -fopenmp -o minibeispiel minibeispiel.c
```

auf **ravel**.

Damit sollten Sie das Executable **minibeispiel** erhalten, das Sie mit **./minibeispiel** starten können.

Aufgabe 3 (Jobs starten)

Um (exklusive) Rechenzeit auf Arbeitsknoten von **hermes** anzufordern, verwenden Sie das Kommando **qsub**:

```
qsub -I -lnodes=3          # 3 Arbeitsknoten anfordern
```

Analog können auf **ravel** Kerne (cores) von **hydra** zur exklusiven Nutzung¹ angefordert werden:

```
qsub -I -lnodes=hydra:ppn=7 # 7 Cores auf hydra anfordern
```

Stehen die angeforderten Knoten bzw. Cores zur Verfügung, erhalten Sie eine Shell auf einem der angeforderten Knoten bzw. **hydra** und können Prozesse starten. Beachten Sie, dass die Dauer einer interaktiven Sitzung auf 30 Minuten beschränkt ist.

Wenn ein Job auf einem Arbeitsknoten bzw. auf **hydra** läuft, dann können Sie sich zusätzlich per SSH auf dem Arbeitsknoten (von **hermes** aus) bzw. auf **hydra** (über den Namen **haydn.fim.uni-passau.de**) einloggen. Dies ist beispielsweise nützlich, wenn ein Prozess „hängen“ bleibt und per Hand beendet werden muss.

Hinweis: Auf **ravel** und **hydra** ist ein 64-bit System (Ubuntu 10.04) installiert, auf **hermes** und seinen Arbeitsknoten ein 32-bit System (CentOS 4).

Weitere Informationen zu **hermes** finden Sie unter:

```
http://www.infosun.fim.uni-passau.de/cl/hpcline
```

Bei Fragen zu oder Problemen mit den Parallelrechnern wenden Sie sich bitte ohne zu zögern an Armin Größlinger (armin.groessleriger@uni-passau.de).

Zur Programmiersprache C finden Sie umfangreiche Informationen in „The C Book“ unter:

```
http://publications.gbdirect.co.uk/c\_book/
```

¹Die angeforderte Anzahl an Kernen steht dem gestarteten Job exklusiv zur Verfügung; Arbeitsspeicher, IO-Bandbreite etc. teilen sich alle gleichzeitig laufenden Jobs auf **hydra**.