

Übungsblatt 2

(10 Punkte)

Besprechung ab Dienstag, 3. Mai 2016

2.1 OpenMP – erste Schritte (3 Punkte)

Betrachten Sie folgendes C-Programm:

```
int main()
{
    #pragma omp parallel
    {
        printf("Hallo Welt\n");
    }
    return 0;
}
```

- Übersetzen Sie das Programm und führen Sie es aus. Vergessen Sie nicht "-fopenmp" zu verwenden. Wie oft wird "Hallo Welt" ausgegeben?
- Ändern Sie den Code so, dass das Programm die Nummer des jeweiligen Threads (ID) und die Anzahl aller Threads ausgegeben wird. Die Ausgabe sollte wie die folgende sein:

```
Hallo Welt: Ich bin Thread Nummer x von y Threads
```

wobei x die ID des Threads ist und y die Gesamtzahl der Threads. Ist es möglich, die Anzahl der Threads in dem Programm zu steuern?

2.2 Schleife Parallelismus (7 Punkte)

Betrachten Sie folgendes sequentielles C-Programm:

```
int main()
{
    int size=8,i,x[]={1,2,3,4,5,6,7,8};

    for(i=0;i<size;i++)
    {
        x[i]*=10;
    }
}
```

- Fügen Sie den erforderlichen Code hinzu um die for-Schleife mit OpenMP zu parallelisieren. Messen Sie die Laufzeit des sequentiellen und parallelen Programms. Was fällt Ihnen auf?
- Parallelisieren Sie die Schleife nur mit "#pragma omp parallel" (ohne "#pragma omp parallel for" oder "#pragma omp for")
- In dem Git-Repository befindet sich eine erweiterte Version des obigen Programms. Parallelisieren Sie auch dieses Programm und messen Sie die Laufzeit. Was fällt Ihnen auf?

Allgemeine Hinweise: Die Übungstermine und weitere Informationen finden Sie unter <http://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/sommersemester-2016/rechnerarchitektur-ss16.html>. Die Übungsblätter werden in den Übungen bearbeitet und besprochen. Eine Abgabe vorher ist also nicht nötig, dennoch sollte man sich im Voraus mit den Übungen vertraut machen. Für die Teilnahme an der Klausur nach BPO 2013 / der Fachprüfung nach DPO 2001 ist der Übungsschein *nicht* erforderlich.