

## Aufgabe 5: einfache MPI Übungsaufgaben

Dieses Übungsblatt umfasst drei einfache Aufgabe zur Einarbeitung in die Programmierung mit MPI.

Sollten Probleme auftauchen schreibt bitte an die Mailingliste:

`PPG-18@wr.informatik.uni-hamburg.de`

### Aufgabe 5A: Iterative Berechnung von $\pi$ (90 Punkte)

In der ersten Aufgabe soll die Zahl  $\pi$  mittels der Integration der Funktion

$$f(x) = 4/(1 + x^2)$$

berechnet werden. Dabei wird die Kurve für den Wertebereich zwischen 0 und 1 in  $n$  Teilbereiche aufgeteilt, für die jeweils die Berechnungen durchzuführen sind. Für die MPI Umsetzung sollen die Teile der Kurve auf 4 Prozesse aufgeteilt werden und die berechneten Ergebnisse danach zum Gesamtintegral zusammengefügt werden. Das Gesamtintegral soll dabei einmal individuell aus den Teilergebnissen der einzelnen Prozessoren aufaddiert werden und einmal mittels des kollektiven Befehls `MPLSUM` ermittelt werden. Die Berechnungen sind in double precision durchzuführen. Dabei soll die Anzahl der Stützstellen, um das Intervall zwischen 0 und 1 zu berechnen, bei  $10^9$  liegen.

### Aufgabe 5B: Ring - Send und Receive (90 Punkte)

In der zweiten Aufgabe sollen die Prozesse innerhalb eines Kommunikators Nachrichten in einer Ring-Anordnung verschicken. Dabei sendet Prozess 0 seine Prozess-ID an Prozess 1, dieser seine an Prozess 2 u.s.w. Nach dem Empfang wird die empfangene Prozess ID von jedem Prozess individuell in einer Summe aufaddiert. Danach wird die empfangene Prozess ID an den nächsten Prozess weiter verschickt und wieder aufaddiert, solange bis jeder Prozess seine eigene Prozess-ID empfangen hat. Dann sollen für alle Prozesse die aufaddierten Summen ausgegeben werden.

Das Programm soll für eine beliebige Anzahl von Prozessen ausgelegt sein (mindestens 4, 8 und 11 testen).

### Aufgabe 5C: Visualisierung von Prozessen (120 Punkte)

In der letzten Aufgabe soll ein Programm mit MPI Befehlen visualisiert werden.

Schreibt ein Programm in dem die Befehle:

1. `SEND` und `RECEIVE`
2. `BROADCAST` eines einzelnen Integers
3. `BROADCAST` (aus `Send` und `Receive` Befehlen nachgestellt)

angewendet werden. Dabei soll jeder der drei Teile mit einem BARRIER Befehl vom weiteren Programmablauf getrennt werden. Das Programm soll auf 8 Prozessen laufen und anschließend mit Score-P und Vampir visualisiert werden. (Dazu Informationen im Kapitel 'Visualisierung')

## Visualisierung

Für die Visualisierung der Ergebnisse muss die Weiterleitung der graphischen Ausgabe aktiviert werden (X-Forwarding). Auf Linux-Systemen kann dies erreicht werden, indem beim Einloggen mit der zusätzlichen Angabe von `-X` der Grafikkmodus aktiviert wird.

```
ssh -X username@cluster.wr.informatik.uni-hamburg.de
```

Um die entsprechenden Module und Umgebungsvariablen zu laden müssen folgende Befehle eingegeben werden:

```
spack load -r mpi scorep
```

```
export SCOREP_ENABLE_TRACING=true
```

Um die für die Visualisierung nötigen Informationen zu erzeugen, wird das Programm gegen die Score-P Library gelinkt. Das geschieht mit Hilfe eines Wrapper Programms, das dem Compileraufruf beim Linken und Kompilieren vorangestellt wird. Im Makefile passiert das an der Stelle, wo das Executable erstellt wird.

```
scorep mpif90 -o test.x test.f90
```

Danach wird das Programm wie gewohnt aufgerufen. Nach dem Lauf wurde ein extra Directory angelegt, welches im Namen mit 'scorep' beginnt, dann das Datum und Uhrzeit anzeigt und danach eine 16-stellige Zahlenfolge, z.B:

```
scorep-201705111615-1234567890123456
```

Geht man in das aktuelle scorep-Directory, so findet sich eine Reihe von Dateien. Eine davon hat die Endung **.otf2** (für open trace format2).

Das Visualisierungsprogramm Vampir wird dann unter Einbindung der otf2-Datei, z.B. test.otf2, in folgender Form

```
$ vampir test.otf2
```

auf dem Cluster gestartet und dabei die otf2-Datei in das Programm geladen. Die Timeline-Darstellung soll dann auf die Programmteile mit Kommunikation fokussiert und zur Auswertung dieser Bereiche mittels Screenshots dokumentiert werden. Die Abgabe umfaßt die Darstellung dieser Programmteile mit einer kurzen Beschreibung der MPI Abläufe. Die Beschreibung inkl. Bilder soll als PDF Dokument (Vampir-Test.pdf) abgeliefert werden.

## Abgabe

Die auf dem Cluster lauffähigen FORTRAN Programme sollen als Quellcode mit der Angabe der Gruppe (Personen in der Gruppe) bis zum Dienstag den 29.5.2018 geschickt werden an:

```
ppg-abgabe@wr.informatik.uni-hamburg.de
```

Bitte dabei die übliche Form der Abgabe wählen.

Als Subject im Kopf der Mail bitte die Angabe: PPG-18 Blatt5 und die Liste der Familiennamen der Personen in der Übungsgruppe.