

Aufgabe 5: MPI-Matrix Operationen und Visualisierung

In den folgenden Aufgaben sollen kollektive MPI Befehle und spezielle Matrix-Operationen geübt und die Darstellung der internen MPI Kommunikation visualisiert werden.

Sollten Probleme auftauchen, schreibt bitte an die Mailingliste:

`PPG-17@wr.informatik.uni-hamburg.de`

Aufgabe 5A: MPI-Matrix Operationen mit Scatter und Gather (90 Punkte)

In der Aufgabe ist eine Integer-Matrix der Größe 24×24 zu erstellen. Diese Integer-Matrix $M(24, 24)$ wird in absteigender Reihenfolge, d.h. von $M(1, 1) = 576$ bis $M(24, 24) = 1$, auf dem Masterprozess initialisiert.

Danach wird diese Matrix je nach Prozessoranzahl mittels dem Befehl SCATTER in gleich große Teilmatrizen N aufgeteilt. Pro Teilmatrix wird die Operation $N(i, j) = N(i, j) * (\text{ProcessorID} + 1)$ ausgeführt. Danach berechnet jeder Prozess die Summe über alle Elemente seiner Teilmatrix.

Anschließend werden mittels des MPI Befehls GATHER die Inhalte der Teilmatrizen in die bestehende Gesamtmatrix $M(24, 24)$ neu zusammengeführt. Auch für die Gesamtmatrix wird die Summe über alle Elemente errechnet und diese Gesamtsumme mit der Summe verglichen, welche als Gesamtsumme der Teilmatrizen ermittelt wird.

Dieses Programm soll für jeweils 4, 6 und 8 Prozessoren ausgeführt werden.

Aufgabe 5B: Visualisierung von Prozessen (120 Punkte)

In der zweiten 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 Prozessoren laufen und anschließend mit Score-P und Vampir visualisiert werden. (Dazu Informationen im Kapitel 'Visualisierung')

Aufgabe 5C: Visualisierung von Prozessen (Bonus 90 Punkte)

Als Bonusaufgabe besteht die Möglichkeit durch die Visualisierung der Aufgabe 4A Bonuspunkte zu sammeln. Visualisiert dafür die Lösung von 4A und beschreibt mögliche Optimierungen. Eventuell mit dem zusätzlichen Hinweis: Hierfür nehmen wir an, dass die Prozesse nach dem Berechnen von Pi möglichst zeitnah noch andere Aufgaben übernehmen sollen. Im Quelltext kann dies durch einen `call sleep(1)` Aufruf vor dem `MPI_finalize` emuliert werden.

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 Grafikmodus aktiviert wird.

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

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

```
export MPICH_NEMESIS_NETMOD=tcp
```

```
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 16.5.2016 geschickt werden an:

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

Als Subject im Kopf der Mail bitte die Angabe: PPG-17 Blatt5 und die Liste der Familiennamen der Personen in der Übungsgruppe.