

Aufgabe 10: MPI - Darstellung der Kommunikation mit Sunshot

Dieses Übungsblatt umfasst die Darstellung der internen MPI Kommunikation mittels des Programms Sunshot. Sollten Probleme auftauchen, schreiben Sie bitte an die Mailingliste:

`PPG-12@wr.informatik.uni-hamburg.de`

Aufgabe 10A: Tracen eines Beispielprogramms für die MPI Kommunikation (200 Punkte)

Erstellen Sie ein kleines MPI Programm das mit 4 Prozessen läuft, welches die folgende Befehle enthält:

1. BROADCAST
2. SCATTER
3. SEND und RECEIVE
4. REDUCE

und stellen Sie die interne MPI Kommunikation mittels Sunshot dar. Dazu wird aus dem Programm eine HDTrace Datei erzeugt, welche mittels Sunshot gelesen und dann graphisch dargestellt wird. Eine kurze Einführung der Nutzung von HDTrace und Sunshot erfolgt am Ende des Übungsblattes.

Erzeugen Sie zu jedem dieser Befehle einen Screenshot (innerhalb von Sunshot) und beschreiben Sie den Ablauf der Kommunikation in einer kurzen Darstellung.

Hinweis In Sunshot sollte die Option `'MPI communication'` und `'nested Operation'` eingeschaltet sein.

Aufgabe 10B: Darstellung der Kommunikation von ALLREDUCE (60 Bonus Punkte)

Auf der Basis von Aufgabe 10A soll das Programm um den Befehl **ALLREDUCE** erweitert und mit Sunshot dargestellt werden. Diskutieren Sie anhand der Screenshots die Unterschiede zum REDUCE Befehl, gefolgt von einem BROADCAST.

Abgabe

Die auf dem Cluster lauffähigen FORTRAN Programme, sowie die **Screenshots und die schriftliche Auswertung** sollen bis zum Mittwoch den 11.7.2012 geschickt werden an:

hermann.lenhardt@informatik.uni-hamburg.de

Bitte dabei folgende Form wählen

1. bitte **NUR den Quellcode und das Makefile, die Screenshots und die schriftliche Auswertung** schicken,
2. bitte für **jede Aufgabe ein separates Verzeichnis anlegen** und
3. alles **als komprimiertes Archiv .tgz oder zip** schicken! D.h. es soll wirklich nur **ein einzelnes Archiv** geschickt werden!

Als Subject im Kopf der Mail bitte die Angabe: PPG-12 Blatt10 und die Liste der Familiennamen der Personen in der Übungsgruppe.

Informationen zur Nutzung der HDTrace Bibliothek

Zunächst einmal findet man auf der Webseite der WR-Gruppe dazu Infos:

<http://wr.informatik.uni-hamburg.de/teaching/ressourcen/profiling>

Der für unsere Zwecke ausreichende Teil wird hier aufgeführt.

Trace Modul laden:

```
-----  
$module load mpich2  
$module load hdtrace/1.0-complete
```

Umgebungsvariablen setzen:

```
-----  
$export HDTRACE_BUFFER_SIZE_KB=50000  
$export HDTRACE_ENABLE_LIKWID=0
```

```
$make run -> (mpiexec -n 2 ./executable)
```

Einbinden der trace files in eine Projektdatei:

```
-----  
$project-description-merger.py -o trace.proj trace_cluster*.info
```

Die Endung .py weist darauf hin, dass die Projektdatei `project-description-merger.py` ein Pythonscript ist. Der * ist ein sogenanntes wildcard Zeichen und ersetzt beliebig lange Zeichenketten. Das führt dazu, dass alle .info Dateien in das Projekt eingebunden werden. Theoretisch könnte auch *.info eingesetzt werden.

Hier gilt es eine Besonderheit für den Dateinamen zu beachten. Unter C kann der Name des Programms für die trace-Dateien übernommen werden. Unter Fortran beginnt der Name standardmäßig mit trace und enthält Angaben über den Rechnerknoten auf dem der Job lief. Daher muss der Projektname trace.proj heißen.

Wenn das Programm im Batchbetrieb mit Hilfe eines scripts gestartet wird, muss der Name der .info Dateien angepasst werden. Dazu muss man sich allerdings mit der Syntax der Batch-Jobs befassen. Das ist für unsere Zwecke nicht unbedingt nötig. Wenn der Job mehrfach gestartet wird, muss nicht jedesmal die Projektdatei neu erstellt werden (es muss lediglich das timeline-Fenster geschlossen werden und die Project-Datei im Sunshot-Fenster mit Enter neu bestätigt werden).

Der folgende Befehl startet das Programm sunshot, mit dem die Kommunikation des MPI-Programms grafisch dargestellt werden kann.

```
sunshot $PWD/trace.proj &
```

Mit Enter den Dateinamen bestätigen, daraufhin wird ein weiteres Fenster geöffnet, und anschließend in diesem Legende-Fenster in der Spalte 'V' bei 'MPI communication' ein Häkchen setzen und auf den timeline Button, links vom Button mit dem orangefarbenen Punkt klicken, um das timeline-Fenster zu öffnen. Das Häkchen bei 'MPI communication' führt dazu, dass Verbindungspfeile erstellt werden, die den Ablauf der Kommunikation aufzeigen.

screenshots können aus sunshot heraus erstellt werden, indem man auf den Button links vom grünen redraw Button klickt. Das .png file wird üblicherweise im Verzeichnis /tmp gespeichert und im Fenster, aus dem sunshot gestartet wurde, ausgegeben:

```
Wrote screenshot to /tmp/jumpshot-1-TimeLinetrace.proj.png
```

In der Menüzeile des TimeLine Fensters gibt es eine Checkbox mit der Bezeichnung 'Nested'. Wenn hier das Häkchen gesetzt ist, wird z.B. die Kommunikation innerhalb einer Barrier angezeigt. Das gibt zusätzliche Informationen darüber, wie einzelne Kommandos unter der Haube arbeiten.