

Visualisierung mit Jumpshot (180 Punkte)

Wir wollen jetzt das Laufverhalten unseres parallelen MPI-Programms mit einem geeigneten Werkzeug visualisieren. Wir gehen dabei in mehreren Schritten vor:

- Binden des Programms mit zusätzlichen Bibliotheken
- Ausführen des Programms; hierbei entsteht eine so genannte Spurdatei ('trace')
- Konvertierung der Spurdatei für Jumpshot
- Visualisierung der Spurdatei in Jumpshot
- Identifizieren wichtiger Merkmale des Programmlaufs anhand der Visualisierung

Zur Bearbeitung der Aufgabe müssen Sie das MPI-Modul laden: `module load mpich2/1.0.7-MPE-ufs-nfs`

Zum Binden unseres Programms:

Wir benötigen die Bibliotheken der MPE-Software, welche Funktionen zur Spurgenerierung enthält.

Wir binden sie in unser Programm folgendermaßen ein: `mpicc programm.c -mpe=mpilog`

Damit funktioniert die Spurgenerierung automatisch und läuft folgendermaßen ab:

Während des Programmablaufes entsteht knotenlokal je eine Spurdatei im Format `clog2`. Nach Programmende werden diese Dateien automatisch zu einer Datei `programm.clog2` zusammengefasst. Alle MPI-Aufrufe werden standardmäßig erfasst. Wenn Sie weitere Ereignisse aufzeichnen möchten, so finden Sie hierzu ein Beispiel in der Datei `cpilog.c`. Zu finden mit `locate cpilog.c`. Dies ist das normale "compute pi"-Programm mit zusätzlichen Protokollierungen.

Damit wir die Spurdatei visualisieren können, muss diese mittels `clog2T0slog2 programm.clog2` in die Datei `programm.slog2` konvertiert werden.

Die Visualisierung erfolgt mittels `jumpshot programm.slog2 &`. Jumpshot ist ein Java-Swing-Programm, mithin reichlich langsam vom Aufbau der graphischen Oberfläche. Man kann das Programm durchaus über das Internet hinweg verwenden, aber selbst mit DSL haben Sie daran nicht viel Freude. Am besten installieren Sie sich lokal eine Installation von MPICH2, mit dessen Jumpshot Sie arbeiten. Ansonsten sollte man es an einem Rechner der Universität über das lokale Netzwerk verwenden, da hier die Bandbreite höher ist. Ein Jumpshot-Paket zum Download finden Sie auf der Materialiensseite der Vorlesung. Dieses benötigt lediglich Java und ist auch unter Windows lauffähig. Eine eigene Installation von MPICH2 wird dann nicht benötigt.

Sie können jetzt in Jumpshot Bereiche vergrößern und verkleinern, nach links und rechts auf der Zeitachse sich bewegen und viele Dinge mehr.

Details finden sich in der Dokumentation unter `/opt/mpich2/1.0.7-MPE-ufs-nfs/doc/jumpshot-4`.

Am besten spielen Sie ein bisschen mit diesen Möglichkeiten. Wenn Sie mit der rechten Maustaste einen Block anklicken, erhalten Sie Details zur Art des Spureintrages und zur Dauer des Ereignisses.

Aufgabenstellung

1. Verfügen Sie über lauffähige Versionen des Jacobi- und Gauss-Seidel-Programms, so visualisieren Sie den Ablauf Ihres Jacobi- und Gauss-Seidel-Programms für folgende Kombinationen von Prozessen und Knoten (Prozesse/Knoten) und einer Anzahl von 20 Iterationen: (2/3) und (4/5)
2. Verfügen Sie lediglich über eine lauffähige Version des Jacobi-Programms, so visualisieren Sie den Ablauf Ihres Jacobi-Programms für folgende Kombinationen von Prozessen und Knoten (Prozesse/Knoten) und einer Anzahl von 20 Iterationen: (3/3), (4/3), (5/3), (4/4), (5/4) und (5/5).

Führen Sie die zutreffende der beiden oberen Aufgaben aus.

- Erzeugen Sie die entsprechenden Spurdateien
- Visualisieren Sie die entsprechenden Dateien mit Jumpshot
- Interpretieren Sie die Visualisierung. Zu diesem Zweck erstellen Sie am besten Detailvergrößerungen der entsprechenden Zeitabschnitte in Jumpshot und von diesem Bild dann einen Screenshot (unter KDE z.B. mit `ksnapshot`). Mit der Version von der Webseite kann unter Linux auch direkt mit dem Print Knopf in der Menüleiste ein Screenshot erstellt werden. Die folgenden drei Zeitabschnitte müssen diskutiert werden:
 1. Die Startphase des Programms bis zum Ende der ersten Iteration in allen Prozessen,
 2. Die Phase der Synchronisation zwischen den Prozessen bei Ende einer Iteration. Bei Jacobi ist das der Zeitabschnitt, in dem alle Prozesse ihre Zeilen austauschen. Bei Gauss-Seidel nehmen wir denselben Zeitabschnitt, auch wenn ja dann aber die Prozesse in unterschiedlichen Iterationen sind und
 3. Die Endphase des Programms mit dem Einsammeln der Ergebnisse.
 4. Diskutieren und Analysieren Sie hierbei insbesondere vorhandene Anomalien der Gesamtlaufzeiten.
- Bestimmen Sie die prozentualen Werte von Rechenzeit und Kommunikation über alle Iterationen hinweg. (Tipp: Iterationen mit rechter Maustaste markieren -> Statistik anklicken)

Protokollieren Sie mit, wie viel Zeit Sie benötigt haben.

Rückmeldung (+ 5-10 Punkte)

Bearbeitungszeit			
Schwierigkeit	<input type="radio"/> zu leicht	<input type="radio"/> genau richtig	<input type="radio"/> zu schwer
Lehrreich	<input type="radio"/> wenig	<input type="radio"/> etwas	<input type="radio"/> sehr
Verständlichkeit	<input type="radio"/> großteils unklar	<input type="radio"/> teilweise unklar	<input type="radio"/> verständlich
Kommentar:			

Abgabe

Abzugeben ist ein gemäß den bekannten Richtlinien erstelltes und benanntes Archiv (.tar.gz). Das enthaltene und gewohnt benannte Verzeichnis soll folgenden Inhalt haben:

- Die Spurdateien (Wenn Sie Erweiterungen bei der Spurerzeugung einbauen, dann auch den erweiterten Programmcode.)
- Die geforderten Screendumps von Jumpshot im JPEG- oder PNG-Format
- Eine Interpretation (PDF oder TXT) der Visualisierung

Senden Sie Ihre Abgabe per Mail zum obigen Abgabetermin bis spätestens 14 Uhr an:

`christian.lohse@pvs.informatik.uni-heidelberg.de`