

Paralleler Löser mit hybrider Programmierung (240 Punkte)

Nachdem Sie nun getrennte parallele Löser mit MPI und OpenMP erstellt haben, sollen diese beiden Varianten nun kombiniert werden. Innerhalb der einzelnen Knoten soll diesmal keine MPI-Kommunikation durchgeführt, sondern mit von OpenMP erzeugten Threads gearbeitet werden.

Kompilieren Sie Ihr entsprechend geändertes Programm mit `mpicc`, diesmal allerdings mit der Option `-fopenmp`. Das so erzeugte Programm enthält dann MPI- und OpenMP-Code. Es kann dann mit der bekannten Kombination aus `mpdboot`, `mpiexec` und `mpdallexit` oder über die Ressourcenverwaltung unter Verwendung der `mpiexec`-Option `-pernode` einmal pro Knoten gestartet werden (siehe auch Wiki).

Aufgabenstellung

Kombinieren Sie Ihre MPI- und OpenMP-Parallelisierungen. Wiederum muss die kombinierte Variante das selbe Ergebnis liefern wie die sequentielle.

Ermitteln Sie die Leistungsdaten Ihres Programmes für die folgenden Konfigurationen aus **K**notenanzahl, **P**rozessanzahl und **T**hreads pro Prozess (K, P, T): (1, 1, 1), (1, 1, 2), (2, 2, 1), (2, 2, 2), (3, 3, 1), (3, 6, 1), (3, 3, 2), (4, 4, 1), (4, 4, 2), (4, 8, 1). Wählen Sie die Laufzeiten so das die Läufe sinnvoll miteinander vergleichbar sind. Erstellen Sie ein Diagramm mit diesen Ergebnissen und Vergleichen Sie die Leistungsdaten der verschiedenen Konfigurationen.

Bitte protokollieren Sie mit, wieviel Zeit Sie benötigt haben. Wieviel davon für die Fehlersuche?

Vergleich der Lösungen (40 Punkte)

Ermitteln Sie die Leistungsdaten Ihres Programmes für die selben Eingaben aus dem OpenMP Übungszettel und fügen Sie die Daten in ihr Diagramm mit den bisherigen Ergebnissen ein. Vergleichen Sie die Leistungsdaten der hybriden Lösung mit der MPI- und OpenMP Lösung.

Abgabe

Die folgenden Informationen sind als tar.gz-Datei an

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

zu senden:

- Alle Quellen, aus denen Ihr Programm besteht; gut dokumentiert (Kommentare im Code!)
Achten Sie darauf, dass Ihr paralleles Programm dieselben Kommandozeilenparameter wie das sequentielle Vorgabeprogramm akzeptiert (d. h. dieselbe Anzahl an Parametern in derselben Reihenfolge)
- Ein Makefile welches mittels `make partdiff-hybrid` automatisch eine Binärdatei `partdiff-hybrid` erzeugt.
- Das Diagramm mit den ermittelten Laufzeiten

Rückmeldung (+ 5-10 Punkte)

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