Universität Hamburg, Fachbereich Informatik
Arbeitsbereich Wissenschaftliches Rechnen
Prof. Dr. T. Ludwig, Anna Fuchs, Jannek Squar

Übungsblatt 6 zur Vorlesung
Hochleistungsrechnen
im WiSe 2023/2024
Abgabe: 02.12.2022, 23:59

Dieses Übungsblatt soll Ihnen die Möglichkeit geben, Ihre ersten Schritte in der Programmierung und Fehlersuche mit MPI zu machen. Die erworbenen Fertigkeiten werden auf späteren Übungsblätter für komplexere Aufgaben benötigt.

1 Das erste MPI-Programm (100 Punkte)

Erweitern Sie das gegebene Programm `timempi.c` um MPI-Kommunikation damit es eine ähnliche Ausgabe erzeugt wie das parallele Script von Übungsblatt 3. Dabei sind folgende Vorgaben zu beachten:

- Bei n Prozessen sollen die Prozesse mit den Rängen 0 bis n-2 folgendes bei sich erzeugen:
  `[Rang] HOSTNAME: TIMESTAMP`
  und per MPI an den Prozess mit dem letzten Rang (n-1) senden, welcher die komplette Ausgabe übernimmt.
- Diese Ausgabe soll nach Rang der Prozesse geordnet erfolgen.
- Die Prozesse sollen alle erst beenden, wenn die Ausgabe komplett erfolgt ist. Das Programm ist falsch, wenn ein Prozess zu früh beenden könnte!
- Direkt vor dem Beenden soll jeder Prozess einen Text ausgeben: „[Rang] beendet jetzt!“
- Das Programm muss mit beliebig vielen Prozessen lauffähig sein. Überlegen Sie dabei eine Handhabung für 1 Prozess.
- Das Programm muss ohne Warnungen und Fehler kompilieren. Entfernen Sie keine Flags aus dem Makefile.

Die Ausgabe könnte wie folgt aussehen:

```
[0] west1: 2021-11-10 13:15:57.968558
[1] west2: 2021-11-10 13:15:57.968557
[1] beendet jetzt!
[2] beendet jetzt!
[0] beendet jetzt!
```

Hinweis: Binden Sie den Header `mpi.h` ein und benutzen Sie den Compiler `mpicc`.

2 Ergebnisse sammeln im MPI-Programm (50 Punkte)

Erweitern Sie Ihr MPI-Programm um folgende Funktion:

- Direkt nach der Ausgabe der empfangenen Daten soll der Prozess mit dem letzten Rang noch den kleinsten Mikrosekunden-Anteil sowie die größte Differenz der Mikrosekundenwerte zwischen allen Prozessen ausgeben.
- Hierfür bieten sich kollektive Operationen an.

Die Ausgabe könnte dann wie folgt aussehen:

```
[0] west1: 2021-11-10 13:15:57.968558
[1] west2: 2021-11-10 13:15:57.968557
[2] Kleinster MS-Anteil: 968557
[2] Größte Differenz: 1
[2] beendet jetzt!
[0] beendet jetzt!
[1] beendet jetzt!
```

3 Paralleles Debugging mit DDT (60 Punkte)

In dieser Aufgabe sollen Sie sich mit dem parallelen Debugger DDT vertraut machen. Verwenden Sie dazu Ihr gerade geschriebenes paralleles Programm.

Unter folgendem Link finden Sie eine kleine Einführung **und Hinweise zum Starten von DDT**:

`https://wr.informatik.uni-hamburg.de/teaching/ressourcen/debugging#ddt`

Hinweis: Für die Benutzung von DDT benötigen Sie X-Forwarding. Alternativ können Sie X2Go nutzen, welches üblicherweise eine höhere Leistung erreicht.

Die Lizenz auf dem Cluster ist für 16 gleichzeitige Prozesse gültig. Bedenken Sie bei Ihren Tests, dass sich alle Benutzer diese Anzahl teilen! Das Aufrufen von DDT ist nur vom Login-Knoten möglich, wobei ein Programm durchaus auf mehreren Knoten analysiert werden kann.

Machen Sie sich ein wenig mit DDT vertraut, indem Sie Ihr Programm mit vier Prozessen debuggen. Dokumentieren Sie folgende Punkte mit jeweils einer kurzen Beschreibung und einem Screenshot; markieren Sie die relevanten Stellen der Screenshots:

- Wie kann man in DDT die Programmparameter angeben? Geben Sie zwei Wege an.
- Setzen Sie in einer Zeile einen Breakpoint. Welche Step-Möglichkeiten gibt es und wie unterscheiden sich diese?
- Schauen Sie sich die Werte der Variable an, in der Sie den Rang des aktuellen Prozesses gespeichert haben. Erklären Sie die Linien in der Darstellung. Vergleichen Sie die Werte aller Prozesse mit Hilfe des Rechtsklickmenüs.
- Machen Sie sich mit der Funktion des Evaluate-Fensters in der rechten unteren Ecke vertraut.
- In der oberen Leiste finden Sie eine Übersicht aller Prozesse und Threads Ihres Programmes. Wechseln Sie zwischen den einzelnen Prozessen und beobachten Sie das Evaluate-Fenster.
- Erweitern Sie Ihr Programm um ein Array und initialisieren Sie es mit beliebigen Zahlenwerten. Lassen Sie sich die Werte anzeigen. Visualisieren Sie diese Array in DDT. (**Hinweis:** Das Array wird nur für diese Aufgabe benötigt und soll im abgegebenen Programmen `timempi` **nicht** enthalten sein.)

Abgabe

Als Abgabe erwarten wir ein gemäß den Vorgaben benanntes komprimiertes Archiv, das ein gemäß den Vorgaben benanntes Verzeichnis mit folgendem Inhalt enthält:

- Die Quellen der C-Programme `timempi.c`.
  - Ein `Makefile` derart, dass `make timempi`, `make clean` und `make` erwartungsgemäße Binärdateien erzeugen bzw. löschen.
  - **Keine** Binärdateien!
- Ein PDF `ddt.pdf` mit den Antworten und Screenshots.

Senden Sie das Archiv an `hr-abgabe@wr.informatik.uni-hamburg.de`.