

1. Übungsblatt

MPI

Ein “Hello-World-Programm” in MPI könnte etwa so aussehen:

```
#include <stdio.h>
#include <mpi.h>
main(int argc, char *argv[]){
    int rank, size;
    MPI_Init(&argc, &argv);           // MPI starten
    MPI_Comm_size(MPI_COMM_WORLD, &size); // Anzahl der Prozesse ermitteln
    MPI_Comm_rank(MPI_COMM_WORLD, &rank); // Nummer des aktuellen Prozesses
    if(rank==0){
        printf("Hello World from Master of %d Processes.\n", size);
    }else{
        printf("Hello World from Slave %d.\n", rank);
    }
    MPI_Finalize();                   // MPI beenden
}
```

Übersetzen: Prompt> mpicc helloworld.c -ohelloworld.out

MPI-Implementation: z. B. Open MPI: <http://www.open-mpi.org/>

Aufgabe 1

Berechnen Sie das Skalarprodukt zweier Vektoren

$$\vec{x} = (1, 2, 3, \dots, n), \quad \vec{y} = (n, n-1, n-2, \dots, 1)$$

indem Sie die Vektoren gleichmäßig auf mehrere Prozesse verteilen (insbesondere auch lokal erzeugen), die Teilsummen lokal berechnen und an den Masterknoten senden, der diese aufsummiert.

a) Verwenden Sie dazu folgende MPI-Befehle:

```
MPI_Status status;                               // Statusvariable

MPI_Send(&var, Anzahl, MPI_DOUBLE, Ziel, Tag, MPI_COMM_WORLD); // Nachricht senden
MPI_Recv(&var, Anzahl, MPI_DOUBLE, Quelle, Tag, MPI_COMM_WORLD, &status); // ... empfangen
```

Quelle und *Ziel* sind die mit `MPI_Comm_rank` ermittelten Prozessnummern (fortlaufend von 0 bis `#Prozesse-1`).

&var ist die Adresse der Nachricht im jeweiligen lokalen Speicher.

Anzahl ist die Nachrichtengröße (bezogen auf den übergebenen Datentyp).

Tag ist eine Nachrichtenidentifikationsnummer (für unsere Zwecke reicht zunächst `Tag=0`).

- b) Konstruieren Sie nun eine baumartige Berechnung mit `send` und `receive`, um das Ergebnis am Wurzelknoten des Baums zu erhalten. Testen Sie einen Binärbaum. Welche Zahl von Kinderknoten könnte optimal sein?
- c) Modifizieren Sie nun die baumartige Berechnung um das Ergebnis auf allen Prozessoren zu erhalten.

Berechnen Sie die theoretischen Kenngrößen Speedup (S), Effizienz (E) und Scaleup (SC). Diskutieren Sie welche Ergebnisse Sie mit MPI erwarten und vergleichen Sie diese mit den entsprechenden praktischen Meßwerten Ihres Programms.