

Falls du noch nicht fertig bist, bearbeite lieber die Aufgaben der anderen Zettel.

Aufgabe 1. Installiere `openmpi` oder `MPICH` und schreibe ein kleines MPI-Hallo-Welt Programm. Probiere die Funktionen `MPI_Comm_rank` und `MPI_Comm_size` aus.

Aufgabe 2. Lasse zwei MPI-Prozesse Ping-Pong spielen. Das heißt schreibe ein MPI-Programm, in dem Prozess 0 an Prozess 1 eine Nachricht schickt und danach Prozess 1 an Prozess 0. Iteriere dies, bis 42 Nachrichten verschickt worden sind. Wenn Prozess 1 seine Nachricht empfangen hat, soll er 'Ping' ausgeben und Prozess 0 'Pong', wenn er seine Nachricht empfangen hat.

Zusatzaufgaben:

1. Erweitere dein Programm so, dass alle ungeraden Ränge mit allen geraden Rängen Ping-Pong spielen.
2. Lasse alle Prozesse Rundlauf Ping-Pong spielen, das heißt Prozess 0 sendet an 1, 1 an 2 und so weiter und Prozess $n - 1$ sendet an 0. Hierbei musst du besonders aufpassen, dass sich die Prozesse nicht gegenseitig blockieren.

Aufgabe 3. Implementiere eine skalierbare Version der Vektor-Norm. Schreibe also ein MPI-Programm, in dem jeder Prozess einen Teil eines Vektors speichert (mache dir vorher gute Gedanken, wie du den Vektor aufteilst). Berechne dann die Norm des Vektors, in dem jeder Prozess lokal alle Werte quadriert und aufsummiert und diese dann mit `MPI_Reduce` auf Prozess 0 aufsummiert werden.

Aufgabe* 4. Mache dich mit Conway's Game of Life vertraut und implementiere eine parallelisierte Version davon. Auf https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens steht eine Beschreibung des Spieles mit Regeln. Überlege dir, wie du das Spielfeld parallel speichern kannst und welche Kommunikationen während des Berechnens notwendig sind. Lasse zum Testen mit 3 Prozessen einen Gleiter im Game of Life fliegen.