

Performance mit OPEN MP

Erkenntnisse

In beiden Parallelisierungsvarianten (siehe Dokumentation) zeigt sich, dass eine Parallelisierung bei einem 9x9 Sudokufeld in MPI keine Verbesserung der Zeit für die Lösung des Sudokus. Die Zeiten verlängern sich linear mit der Anzahl der gegebenen Prozesse (siehe bsp. RawdataParallelprogramm1.ods im Git). Der Backtracking Algorithmus braucht nur 0.45 Millisekunden, um ein 9x9 Sudoku mit normaler Schwierigkeit und einem Prozess zu lösen. Mit 6 Prozessen haben wir eine Verdopplung der Zeit. Aus dieser Erkenntnis wurde die Interesse geweckt, wie lange alle MPI Kommunikationen benötigen, um diese mit den Zeiten der Lösung des Sudokus zu vergleichen.

Es wurde gemessen das beispielsweise Variante 1 relativ betrachtet 20 % für die Breiten-suche, 40 % für das Lösen des Problems und ca. 40 % für MPI Kommunikationen bei (mit Abweichungen) der gesamten Zeit für 4 Prozesse benötigt. Bei 16 Prozessen ist die Dauer der Berechnung der Lösung nur noch bei ungefähr 20 % der gesamten Dauer. Daraus lässt sich folgern, dass die MPI Kommunikationsdauer die Zeit der Problemlösung übertrifft.

Forschungsfrage

Aus den Erkenntnissen stellte sich die Frage, ob man das 9x9 Sudoku Problem effizienter Parallelisieren könnte, damit man schneller Lösungen erhält.

Eine Möglichkeit wäre die Implementierung von OPEN MP anstatt von MPI. In diesem Fall würde man sich Kommunikationen sparen, da auf einem gemeinsam genutzten Speicher zugegriffen wird. Man bräuchte somit keinen Manager Prozess der jedem Prozesse ein Sudoku zuschicken muss und es könnte direkt jeder Thread auf eine einzige initialisierte Liste zugreifen, ohne jegliche Kommunikationen zwischen den Threads.

Die Vermutung wäre, dass sich die Implementierung von OPEN MP, verglichen mit dem jetzigen MPI Ansatz, positiv auf die Performance auswirken würde.

Vorgehen

Das Programm wird in OPEN MP überführt und es werden analog wie bei MPI die Zeiten gemessen.