

Zentralübung Rechnerstrukturen:

Parallelismus und Parallele Programmierung

5. Übungsblatt

1 Parallelismus

1.1 Leistungsbewertung

Gegeben sei ein Multiprozessorsystem mit 16 Prozessoren. Die Leistungssteigerung gegenüber einem Einprozessorsystem sei $S(16) = 8$. Die Ausführungszeit auf dem Einprozessorsystem sei $T(1) = 800$ und die Anzahl der auszuführenden Einheitsoperationen auf dem Multiprozessorsystem sei $P(16) = 1200$.

- Berechnen Sie die Effizienz $E(16)$, die parallele Ausführungszeit $T(16)$ und den Parallelindex $I(16)$.
- Interpretieren Sie den berechneten Parallelindex.
- Ermitteln Sie anhand von Amdahls Gesetz den Bruchteil des Programms, der nur sequentiell ausführbar ist.

1.2 Leistungsbewertung

Die Ausführungszeit einer sequentiellen Anwendung betrage T_{seq} . Von dieser Anwendung lassen sich 20 % nicht parallelisieren. Die verbleibenden 80 % werden zwischen den Prozessoren gleichmäßig verteilt. Das bedeutet, dass jeder Prozessor ungefähr den gleichen Anteil der zu parallelisierenden Aufgabe bearbeitet und jeder gleichviel Zeit benötigt. Beispielsweise betrage die Ausführungszeit der parallelen Anteile der Anwendung 20 % der sequentiellen Ausführungszeit, wenn vier Prozessoren sie ausführen.

- Setzen Sie voraus, dass die Parallelisierung keinen Aufwand verursacht. Berechnen Sie die Beschleunigung und die Effizienz bei einer unterschiedlichen Anzahl von Prozessoren. Fügen Sie die Ergebnisse in die vorgegebenen Tabelle ein.

Prozessoren	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$
Beschleunigung					
Effizienz					

- b) Evaluieren Sie zusätzlich die Skalierbarkeit mit Hilfe der berechneten Werte.
- c) Zur Steigerung der Genauigkeit der Berechnung nehmen Sie nun an, dass durch jeden verwendeten Prozessor eine zusätzliche Bearbeitungszeit von 1 % der sequentiellen Ausführungszeit benötigt wird. Berechnen Sie Beschleunigung und Effizienz für 64 Prozessoren.

1.3 Leistungsbewertung

Ein Einprozessorsystem soll erweitert werden. Dabei existieren folgende, in der Anschaffung gleich teure Alternativen:

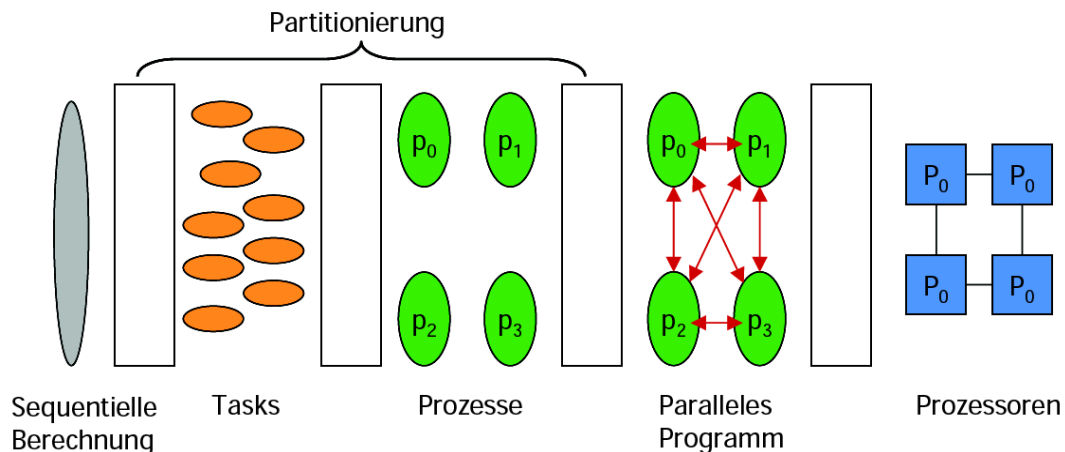
- Ausbau zu einem 2-fach SMP-System,
d.h. die Installation eines zweiten identischen Hauptprozessors mit Zugriff auf einen gemeinsamen Hauptspeicher. Auf Grund der notwendigen Synchronisation beim Zugriff auf den gemeinsamen Speicher wird die Ausführung des Programms um 2 % der unparallelisierten Ausführungszeit erhöht.
- Einsetzen eines mathematischen Coprozessors.
Dieser bietet eine 10x schnellere Ausführung der Gleitkommaarithmetik als der vorhandene Systemprozessor. Es ist allerdings keine parallele Verarbeitung, d.h. der gleichzeitige Einsatz von Haupt- und Coprozessor, möglich.

Das zu bearbeitende Problem ist zu 74 % parallelisierbar. Der Anteil der Gleitkommaarithmetik am Gesamtprogramm beträgt 40 %. Bestimmen Sie, welche der beiden Möglichkeiten unter dem Gesichtspunkt der Ausführungszeit zu bevorzugen ist.

2 Parallele Programmierung

2.1 Parallelisierungsprozess

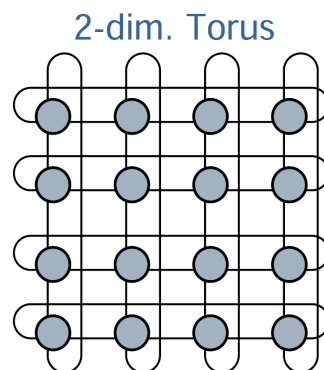
Als Programmierer sollten Sie ein gegebenes Problem möglichst effizient auf einem Parallelrechner umsetzen. Vervollständigen Sie die Grafik, indem Sie die beim Parallelisierungsprozess durchzuführenden Schritte in die jeweils dafür vorgesehene Box eintragen.



2.2 Parallelisierung

Eine Methode aus der Numerischen Mathematik arbeitet auf einem 2D-Torus mit $n \times n$ Knoten. In jeder Iteration werden zwei Schritte durchgeführt:

- Im ersten Schritt werden die Zustände der Knoten, basierend auf ihrem aktuellen Zustand und den Zuständen ihrer Nachbarknoten, aktualisiert. Dazu müssen zuerst diese Zustände aus dem Speicher gelesen werden.
- Im zweiten Schritt werden die neuen Zustände zurück in den Speicher geschrieben.



a) Gegeben sei ein SMP-Knoten mit P Prozessoren:

Berechnen Sie die Beschleunigung der Anwendung bei der Ausführung auf dem SMP-Knoten. Welches Problem tritt hierbei auf?

- b) Um eine Leistungssteigerung zu erhalten, wird der SMP-Knoten durch eine NUMA-Architektur mit P Prozessoren und P Speichern ersetzt.

Beachten Sie dabei folgende vereinfachende Annahmen:

- Erfolgt ein Speicherzugriff auf einen entfernten Speicher, so benötigt ein Speicherzugriff drei Zeiteinheiten.
- Wird nur ein Prozessor verwendet, so befinden alle zur Berechnung notwendigen Daten im lokal zum Prozessor gehörenden Speicher.

Welche Beschleunigung lässt sich erzielen, wenn die Zustände der Nachbarknoten immer aus entfernten Speichern abgerufen werden müssen?

3 Klausuraufgaben

In der Übung werden sofern die Zeit ausreicht, zusätzlich die Aufgaben „Parallelverarbeitung“ der Klausuren des Wintersemesters 201/11 und Sommersemesters 2010 besprochen. Die Aufgabenstellungen finden Sie auf der Homepage zur Vorlesung Rechnerstrukturen.

3.1 Wintersemester 2010/11

In dieser Übung werden von der Aufgabe 1 die Teilaufgaben a)-c) sowie g)-i) durchgesprochen.

3.2 Sommersemester 2010

In dieser Übung werden von der Aufgabe 4 die Teilaufgaben d) und e) durchgesprochen.