Karlsruher Institut für Technologie

Institut für Technische Informatik
Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung
Prof. Dr. rer. nat. Wolfgang Karl

Zentralübung Rechnerstrukturen: Parallelismus und Parallele Programmierung

5. Übungsblatt – Musterlösung

1 Parallelismus

1.1 Leistungsbewertung

a)

$$E(n) = \frac{S(n)}{n} \Rightarrow E(16) = \frac{S(16)}{16} = \frac{8}{16} = 0,5$$

$$S(n) = \frac{T(1)}{T(n)} \Rightarrow T(16) = \frac{T(1)}{S(16)} = \frac{800}{8} = 100$$

$$I(16) = \frac{P(n)}{T(n)} \Rightarrow I(16) = \frac{P(16)}{T(16)} = \frac{1200}{100} = 12$$

b) Der Parallelindex gibt den mittleren Grad der Parallelität an, d.h. die Anzahl der parallelen Operationen pro Zeiteinheit. Der berechnete Wert ist kleiner als die Anzahl der Prozessoren. Das bedeutet, dass zeitweise einige Prozessoren keinen Befehl ausführen.

Besser verdeutlicht wird dies durch Berechnung der Auslastung U, die angibt wieviel Operationen jeder Prozessor im Durchschnitt pro Zeiteinheit ausführt.

$$U(n) = \frac{I(n)}{n} \quad \Rightarrow \quad U(16) = \frac{12}{16} = \frac{3}{4} = 75\,\%$$

c) Amdahls Gesetz:

$$\begin{aligned} T(n) &= \frac{(1-a)}{n} * T(1) + a * T(1) \\ &= T(1) * \left(\frac{(1-a)}{n} + a \right) \\ &= T(1) * \frac{1-a+na}{n} \end{aligned}$$

$$\begin{aligned} \Rightarrow \quad 100 &= 800 * \frac{1 - a \ + \ 16a}{16} \\ &= 800 * \frac{1 \ + \ 15a}{16} \\ &= 50 * (1 \ + \ 15a) \\ \Rightarrow \quad 15a &= 1 \quad \Rightarrow \quad a = \frac{1}{15} \approx 6,7\,\% \end{aligned}$$

$\approx 6,7\,\%$ des Programmcodes sind nur sequentiell ausführbar.

1.2 Leistungsbewertung

a) Parallele Ausführungszeit von n Prozessoren:

$$\begin{aligned} T(n) &= \frac{80\,\%}{n} * T_{seq} \ + \ 20\,\% * T_{seq} \\ \text{Beschleunigung:} \quad S(n) &= \frac{T_{seq}}{T(n)} \\ \text{Effizienz:} \quad E(n) &= \frac{S(n)}{n} \end{aligned}$$

Prozessoren	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$
Beschleunigung	$5/3 = 1,67$	$5/2 = 2,50$	$10/3 = 3,33$	4	$40/9 = 4,44$
Effizienz	$5/6 = 0,83$	$5/8 = 0,625$	$10/24 = 0,42$	$1/4 = 0,25$	$5/36 = 0,14$

Achtung: In der Klausur bei Berechnungen wenn möglich Brüche statt gerundete Zahlen als Ergebnisse angeben!

b) Die Skalierbarkeit ist sehr schlecht. Grund dafür ist, dass ein großer Anteil des Programms nicht parallelisierbar ist. Die Ausführungszeit dieses Anteils bestimmt mit steigender Anzahl von Prozessoren einen zunehmenden Anteil der gesamten Ausführungszeit.

$$\text{Beschleunigung:} \quad S(n) = \frac{T_{seq}}{\left(20\,\% + \frac{80\,\%}{n}\right) * T_{seq}} \xrightarrow{n \text{ sehr groß}} \frac{1}{20\,\%} = 5$$

c) Kommunikationskosten:

$$\begin{aligned} \text{Beschleunigung:} \quad S(n) &= \frac{T_{seq}}{\left(20\,\% + \frac{80\,\%}{n} \ + \ 1\,\% * n\right) * T_{seq}} \\ \Rightarrow \quad S(64) &\approx 1,17 \\ \text{Effizienz:} \quad E(64) &= \frac{1,17}{64} \ \approx \ 0,018 \end{aligned}$$

1.3 Leistungsbewertung

Die Ausführungszeiten lassen sich in beiden Fällen wiederum mit Hilfe von Ahmdahls Gesetz ausrechnen und damit auch die erreichte Beschleunigung vergleichen. T_{seq} ist hierbei die Zeit ohne Parallelisierung oder Koprozessorunterstützung.

- SMP-System:

$$\begin{aligned} T_{SMP} &= 74\,\% * \frac{1}{2} * T_{seq} + 26\,\% * T_{seq} + 2\,\% * T_{seq} = 65\,\% * T_{seq} \\ S_{SMP} &= \frac{T_{seq}}{T_{SMP}} = \frac{1}{65\,\%} \approx 1{,}5385 \end{aligned}$$

- System mit Koprozessor:

$$\begin{aligned} T_{CP} &= 40\,\% * \frac{1}{10} * T_{seq} + 60\,\% * T_{seq} = 64\,\% * T_{seq} \\ S_{CP} &= \frac{T_{seq}}{T_{CP}} = \frac{1}{64\,\%} \approx 1{,}5625 \end{aligned}$$

Die Koprozessorlösung ist für diese Anwendung deshalb dem SMP-System vorzuziehen. Ohne Beachtung der benötigten Synchronisationszeit würde die Berechnung ein fehlerhaftes Ergebnis liefern.

2 Parallele Programmierung

a) SMP-Knoten mit P Prozessoren und einem gemeinsamen Speicher

Sequentielle Zeit $T(1)$:

$$\underbrace{5n^2}_{\text{Daten aus Speicher holen}} + \underbrace{n^2}_{\text{Berechnung}} + \underbrace{n^2}_{\text{Zurückschreiben}} = 7n^2$$

Parallele Zeit $T(P)$:

$$\underbrace{5n^2}_{\text{Daten aus Speicher holen}} + \underbrace{\frac{n^2}{P}}_{\text{par. Berechnung}} + \underbrace{n^2}_{\text{Zurückschreiben}} = 6n^2 + \frac{n^2}{P}$$

Beschleunigung $S(P)$:

$$S(P) = \frac{T(1)}{T(P)} = \frac{7n^2}{6n^2 + \frac{n^2}{P}} = \frac{7}{6 + \frac{1}{P}} < \frac{7}{6} = 1,1\overline{6}$$

Das Problem ist hierbei, dass der Speicher als limitierender Faktor wirkt und damit die Beschleunigung stark beschränkt ist.

In der Realität benötigten Synchronisationen jedoch in SMP-Systemen ebenfalls Speicherzugriffe von allen Prozessoren und damit sehr viel Zeit! Die erreichbare Beschleunigung wäre deshalb noch geringer.

b) SMP-Knoten, jedoch ohne Synchronisation vor der Berechnung

Durch ein Verzicht auf die Synchronisation kann die Berechnung auf den Prozessoren mit den Datenzugriffen überlappend erfolgen. Bei der parallelen Ausführungszeit spart man sich dadurch die Zeit für die Berechnung $\frac{n^2}{P}$ ein. Dies gilt aber nur, wenn die Berechnungszeit kürzer ist, als die Zeit für die Datenzugriffe.

Es gilt dann:

$$T(P) = 5n^2 + n^2 = 6n^2$$

$$S(P) = \frac{T(1)}{T(P)} = \frac{7n^2}{6n^2} = \frac{7}{6} = 1,1\overline{6}$$

Insgesamt überwiegt auch hier die Zeit für die nicht parallelisierbaren Lese- und Schreibzugriffe. Der gemeinsame Speicher des SMP-Systems ist der limitierende Faktor.

c) NUMA-Architektur mit P Prozessoren und P lokalen, aber gemeinsamen Speichern

Sequentielle Zeit $T(1)$:

$$\underbrace{5n^2}_{\text{Daten aus Speicher holen}} + \underbrace{n^2}_{\text{Berechnung}} + \underbrace{n^2}_{\text{Zurückschreiben}} = 7n^2$$

Parallele Zeit $T(P)$:

$$\underbrace{\underbrace{\underbrace{\frac{4*3*n^2}{P}}_{\substack{4 \text{ Werte der} \\ \text{Nachbarknoten} \\ \text{aus entferntem} \\ \text{Speicher}}} + \underbrace{\frac{n^2}{P}}_{\substack{\text{eigener Wert} \\ \text{im lokalen} \\ \text{Speicher}}}}_{\text{Daten aus Speicher holen}} + \underbrace{\frac{n^2}{P}}_{\substack{\text{parallele} \\ \text{Berechnung}}} + \underbrace{\frac{n^2}{P}}_{\substack{\text{Zurückschreiben} \\ \text{in lokalen} \\ \text{Speicher}}} = \frac{15n^2}{P}$$

Beschleunigung $S(P)$:

$$S(P) = \frac{T(1)}{T(P)} = \frac{7n^2}{\frac{15n^2}{P}} = \frac{7}{15}P$$

$$\text{für } P=2: \quad S(2) = \frac{7}{15}*2 = \frac{14}{15} \approx 0,93$$

$$\text{für } P=3: \quad S(3) = \frac{7}{15}*3 = \frac{21}{15} \approx 1,40$$

$$\text{für } P=4: \quad S(4) = \frac{7}{15}*4 = \frac{28}{15} \approx 1,87$$

$\Rightarrow$ Die Beschleunigung skaliert mit $\frac{7}{15}$ der Prozessoren-/Speicherzahl (linear).