

Kapitel 1

Parallele Modelle

Wie rechnet man parallel?

Vorlesung *Theorie Paralleler und Verteilter Systeme* vom 11. April 2008

Till Tantau
Institut für Theoretische Informatik
Universität zu Lübeck



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

Lernziele der heutigen Vorlesung und der Übungen.

- 1 Grundidee des parallelen Algorithmus verstehen
- 2 DAGs als paralleles Modell kennen und anwenden können
- 3 PRAMs als paralleles Modell kennen und anwenden können
- 4 Das Netzwerkmodell und Netztopologien kennen



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?
Arten der
Parallelverarbeitung
Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

Gliederung

1 Einführung

Mehr Prozessoren = Schneller?
Arten der Parallelverarbeitung
Fallbeispiel

2 Das DAG-Modell

3 Das PRAM-Modell

Die Idee
Ein Beispiel

4 Das Netzwerkmodell

Das Modell
Wichtige Topologien

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?
Arten der
Parallelverarbeitung
Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

Das Versprechen der Parallelverarbeitung.

- Die Kosten, Prozessoren schneller zu machen, steigen exponentiell.
- Die Kosten, die Anzahl der Prozessoren zu erhöhen, steigen linear.

Versprechen

Durch Parallelverarbeitung kann man Probleme billiger schnell lösen als durch schnellere Prozessoren.

Zur Diskussion

Welche Einwände lassen sich gegen dieses Argument erheben?

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung
Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

Grundsätzliche Arten von Parallelverarbeitung.

- 1 SISD: single instruction, single data
»Eine Instruktion auf einen Datensatz anwenden«
Beispiel: Normale CPU
- 2 SIMD: single instruction, multiple data
»Eine Instruktion auf mehrere Datensätze anwenden«
Beispiel: Graphikprozessor
- 3 MIMD: multiple instructions, multiple data
»Unterschiedliche Instruktionen auf unterschiedliche Datensätze anwenden«
Beispiel: Dual-Core-Prozessoren, große Parallelrechner

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

Die Matrix-Vektor-Multiplikation.

Problem

Eingabe Eine $n \times n$ Matrix A und ein Vektor v .

Ausgabe $b = A \cdot v$.

Zur Diskussion

Wie löst man das Problem, wenn man ...

- ... einen Prozessor hat?
- ... zwei Prozessoren hat?
- ... drei Prozessoren hat?
- ... $n/2$ Prozessoren hat?
- ... $n/2 + 1$ Prozessoren hat?
- ... n Prozessoren hat?
- ... $2n$ Prozessoren hat?
- ... n^2 Prozessoren hat?
- ... $2n^2$ Prozessoren hat?

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

Was bringen mehr Prozessoren?

Für das Problem Matrix-Vektor-Multiplikation hat die Verdoppelung der Prozessoranzahl folgenden Effekt:

- Für kleine Prozessorzahlen halbiert sich jedesmal die Rechenzeit.
- Für Prozessorzahlen über n gilt dies nicht mehr exakt, aber fast.
- Für Prozessorzahlen in der Nähe von n^2 verringert sich die Rechenzeit kaum.
- Für Prozessorzahlen über n^2 ändert sich die Rechenzeit nicht mehr.



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

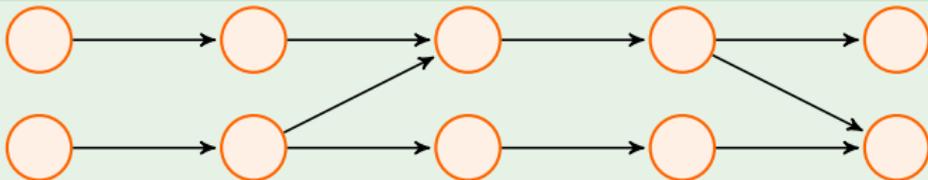
Die Idee hinter dem DAG-Modell.

- Das DAG-Modell macht **Abhängigkeiten** sichtbar.
- Das DAG-Modell blendet die Implementierung aus.

Beschreibung des DAG-Modells

- Im DAG-Modell wird ein Algorithmus durch einen DAG (directed acyclic graph) beschrieben.
- Die Eingabe liegt an den Knoten mit Eingrad 0 an.
- Die anderen Knoten führen Berechnungen durch.
- Sie können ihre Berechnung erst durchführen, wenn die Ergebnisse der Vorgänger vorliegen.
- Berechnungen können recht beliebig sein.

Beispiel



Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

Das Konzept des Schedules.

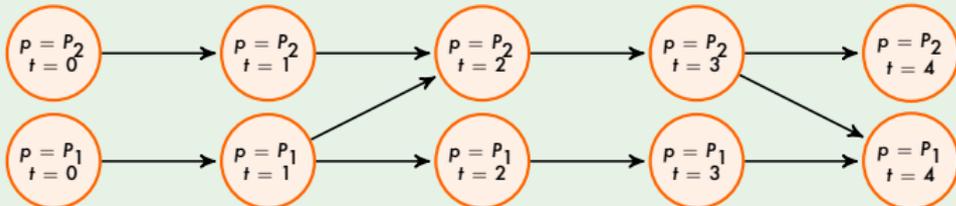
Definition (Schedule)

Sei $G = (V, E)$ ein DAG und P eine Menge von Prozessoren. Ein **Schedule** S ist eine Abbildung, die jedem Knoten von G einen Prozessor $p \in P$ zuordnet und einen Zeitpunkt $t \in \mathbb{N}$. Dabei muss gelten:

- 1 Keinen zwei Knoten darf derselbe Zeitpunkt und gleichzeitig derselbe Prozessor zugeordnet werden.
- 2 Der Zeitpunkt jedes Prozessors muss größer sein als der seiner Vorgänger.

Der maximale in S vorkommende Zeitpunkt heißt **Zeitbedarf**, abgekürzt $T(S)$.

Beispiel



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

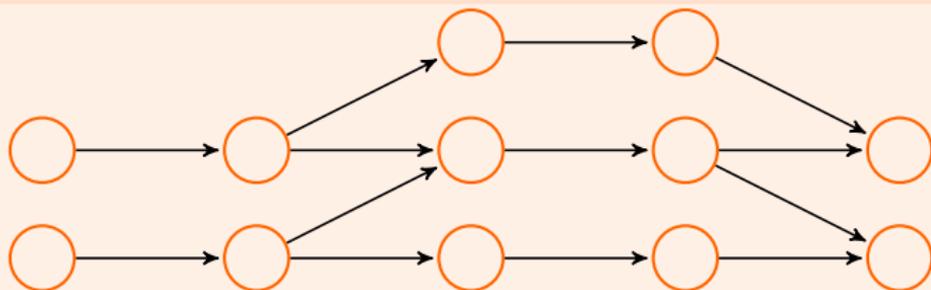
Das Modell

Wichtige Topologien

Zusammenfassung

5-Minuten-Aufgabe

Zur Übung



Gruppe 1

Geben Sie einen optimalen Schedule für den DAG mit $P = \{P_1, P_2\}$ an.

Gruppe 2

Geben Sie optimale Schedules für den DAG mit $P = \{P_1\}$ an und dann für $P = \{P_1, P_2, P_3\}$.

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

PRAMs im Überblick

Aufbau und Initialisierung

- Die Abkürzung PRAM steht für Parallel Random Access Machine.
 - Jeder Prozessor ist eine eigene RAM. Das bedeutet:
 - Eine RAM hat Zugriff auf einen Satz Register, in denen Zahlen stehen.
 - Eine RAM arbeitet ein festes Programm ab.
- Neu im Vergleich zur einfachen RAM ist:**
- Es gibt noch **globale Register**.
 - Es gibt noch ein **nur lesbares Prozessor-ID-Register**.
 - Es gibt neue Befehle, um die globalen Register zu lesen und zu beschreiben.
- Die Prozessoren arbeiten synchron (exakt gleichgetaktet).
 - Alle Prozessoren führen das gleiche Programm aus, aber bei jedem Prozessor ist PID anders initialisiert.
 - Ein- und Ausgabe liegen in bestimmten Bereichen des globalen Speichers.



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

- Die Kommunikation erfolgt mittels des globalen Speichers (shared memory).
- Man unterscheidet, ob der Lesezugriff ausschließlich (exclusive read) oder parallel (concurrent read) sein darf.
- Schreibzugriff kann man ebenfalls unterschiedlich regeln:
 - Nur ein bestimmter Prozessor darf eine bestimmte globale Zelle beschreiben (owner write).
 - Immer nur ein Prozessor darf schreiben (exclusive write).
 - Paralleles Schreiben (concurrent write) ist erlaubt. Hier gibt es die Untervarianten Common, Arbitrary und Priority.
- Insgesamt gibt es folgende sinnvolle Zugriffsarten: EROW, EREW, CROW, CREW, Common-CRCW, Arbitrary-CRCW, Priority-CRCW.



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

Beispiel eines PRAM-Programms.

Folgendes Programm berechnet die Summe der Zahlen im Array A im globalen Speicher.

```
1  for  $h \leftarrow 1$  to  $\log n$  seq do
2      if  $pid \leq n/2^h$  then
3          global read  $a \leftarrow A[2 \cdot pid]$ 
4          global read  $b \leftarrow A[2 \cdot pid - 1]$ 
5           $c \leftarrow a + b$ 
6          global write  $A[pid] \leftarrow c$ 
7  if  $pid = 1$  then
8      global write  $S \leftarrow c$ 
```

Zur Übung

Welche Schreib-/Lese-Zugriffsart wird im Programm verwendet?



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

Die Idee hinter dem Netzwerk-Modell.

- Man modelliert ein festes Kommunikationsnetzwerk durch einen Graphen.
- Die **Knoten** des Graphen sind die Prozessoren.
- Die **Kanten** des Graphen sind Kommunikationskanäle.
- Die Prozessoren arbeiten in der Regel **asynchron** und schicken sich gegenseitig Nachrichten.

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

Das Netzwerk-Modell – parallel oder verteilt?

Till Tantau



Gemeinsamkeiten mit verteilten Systemen

- Modellierung des Kommunikationsnetzwerks.
- Nachrichtenbasiertes Modell.

Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee

Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

Unterschiede zu verteilten Systemen

- Feste Topologie (fest verdrahtet)
- Sichere Kommunikation

Wichtige Parameter eines Netzwerks

Parameter Durchmesser

Maximale Entfernung zwischen Knoten.
Dies ist die Zeit, die eine Nachricht höchstens braucht.
Kleiner ist besser.

Parameter Grad

Maximale Anzahl von Nachbarn eines Knoten.
Je höher der Grad, desto mehr Kabel muss man ziehen.
Kleiner ist besser.

Parameter Zusammenhänge

Minimale Anzahl von Knoten/Kanten, die man löschen muss,
um das Netz in zwei Teile zu spalten.
Je höher der Zusammenhang, desto ausfallsicherer.
Größer ist besser.

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung

Liste wichtiger Netztopologien.

- 1 Lineares Array (Pfad)
- 2 Ring
- 3 2D-Gitter
- 4 3D-Gitter
- 5 Hyperwürfel
- 6 Binärbaum

Zur Übung

Bestimmen Sie für eine der obigen Topologien für p Prozessoren den Durchmesser, Grad und Kantenzusammenhang.

Parallele Modelle

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell

Wichtige Topologien

Zusammenfassung

- 1 **Parallele Algorithmen** führen mehrere Berechnungen koordiniert aus.
- 2 Probleme haben häufig **parallele Anteile** und **sequentielle Anteile**.
- 3 Das **DAG-Modell** beschreibt abstrakt, welche Anteile parallel und welche sequentiell sind.
- 4 Das **PRAM-Modell** beschreibt Algorithmen konkreter.
- 5 Das **Netzwerkmodell** modelliert die **Kommunikation** bei parallelen Programmen realistischer als das PRAM-Modell.

Till Tantau



Ziele und Inhalt

Einführung

Mehr Prozessoren =
Schneller?

Arten der
Parallelverarbeitung

Fallbeispiel

Das DAG-Modell

Das PRAM-Modell

Die Idee
Ein Beispiel

Das Netzwerkmodell

Das Modell
Wichtige Topologien

Zusammenfassung