Karlsruher Institut für Technologie

Übungsbetrieb Parallele Algorithmen

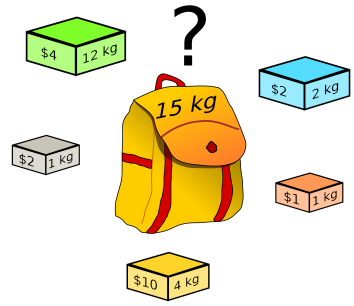
Lukas Hübner, Marvin Williams | WS 21/22

Sie haben die Wahl...

- Programmieraufgabe (C++): Knapsack Problem
  - Shared Memory (TBB, OpenMP, ...) ⟵ empfohlen
  - MPI
  - CUDA
- Miniseminarvortrag (8 Plätze)
- Wikipedia-Artikel erstellen (2 Plätze)

Knapsack Problem

- Gegeben: Menge von Objekten U, Gewichtsfunktion w, Nutzenfunktion v und Kapazität C
  - $w : U \mapsto \mathbb{N}$
  - $v : U \mapsto \mathbb{N}$
  - $C : \mathbb{N}$
- Gesucht: $X \subseteq U$ von Objekten, die das Gewicht C nicht überschreiten und den Nutzen maximieren
  - $\sum_{x \in X} w(x) \leq C$
  - $\sum_{x \in X} v(x)$ maximal



- Entscheidungsproblem (schwach) NP-Vollständig
- In der Programmieraufgabe wird das Optimalwertproblem gelöst

Knapsack Lösungsansätze

- Branch-and-Bound
  - Nebenläufig Baum der Lösungskandidaten abarbeiten
  - Obere/Untere Schranken finden, um den Baum zu beschneiden
- Dynamic Programming
  - Tabelle mit Lösungen für Teilprobleme
  - Tabelle wird nebenläufig schrittweise ausgefüllt

Liste von relevanten Papers auf https://algo2.iti.kit.edu/sanders/courses/paralg21/

Knapsack Framework (Nur für Shared Memory!)

- Framework zu finden unter https://git.scc.kit.edu/paralgws2122
  - Arbeiten im eigenen Fork
  - Die Datei `algorithm/algorithm.hpp` enthält die relevanten Definitionen und Datenstrukturen
  - NUR die Datei `algorithm/algorithm_impl.hpp` anpassen (Eigene Dateien können hinzugefügt werden)
  - Die README enthält die detaillierte Anleitung
  - Lösen des 0/1-Knapsack Problems
- Ihr Code wird (hoffentlich bald) automatisch ausgeführt und evaluiert
- Ergebnisse können (hoffentlich bald) eingesehen werden unter https://algo2.iti.kit.edu/paralgws2122/ranking

Implementierung mit MPI oder CUDA

- Implementierung mit MPI oder CUDA möglich
- Keine Möglichkeit, das Framework zu verwenden
- Erfahrung mit der Technologie sollte vorhanden sein
- Kein automatisches Testen und Evaluieren
- Wir stellen keine Grafikkarte und keinen Clusterzugang

Organisatorisches zur Programmieraufgabe

- 1-seitiger Abschlussbericht
  - Beschreibung des Ansatzes
  - Erklärung und Begründung der Tuning-Schritte
  - Kurze Auswertung
- Bewertung des Quellcodes
- Anmeldung bis zum 13. November mit SCC-Git Benutzername an williams@kit.edu
- Abgabe inkl. Abschlussbericht bis zum Ende der Vorlesungszeit (12.02.22)

Wikipedia-Artikel und Miniseminar

- Begrenzte Plätze
- Themenliste auf https://algo2.iti.kit.edu/sanders/courses/paralg21/
- Eigene Themenvorschläge möglich
- Bitte eine geordnete Liste mit Präferenzen bis 13. November an huebner@kit.edu

Note und Ilias-Quiz

- Note der Übungsaufgabe fließt zu 15% in die Abschlussnote ein
- In der Woche vor Weihnachten wird es ein Quiz per Ilias geben
- Note des Quizzes fließt zu 5% in die Abschlussnote ein
- Quiz dient hauptsächlich der Selbstkontrolle