

Gedächtnisprotokoll Parallele Programmierung (WiSe 23/24)

Grundlagen

- shared vs. distributed memory
 - Unterschiede bei Parallelisierung
 - Vor-/Nachteile
- NUMA
- Threads vs. Prozesse
 - Wann wofür?
- Race conditions
 - Entstehen wodurch?
 - Wie vermeiden?
- SPMD
 - Einsatzgebiet?
- system calls
 - Performance impact?
- Größenordnung von Zugriffszeiten für
 - Cache
 - RAM
 - Netzwerk
 - Storage
- 2 Leistungsmetriken für Netzwerke nennen
 - Einheit, Definition

Cache

- Pseudocode gegeben
 - was ist nach dieser Instruktion im Cache
 - umschreiben, sodass für Cachelokalität besser
 - Array of Struct (AoS) und Struct of Array (SoA)
- Warum ist Cachelokalität wichtig
 - Laden aus RAM sehr langsam

Leistungsmessungen

- was können für Fehler bei der Leistungsmesung entstehen?
 - z.B. Messfehler, Systematische Fehler, ...
- warum sollte man Test mehrfach durchführen?
- 2 Ansätze für Leistungsanalyse und -optimierung nennen
 - Online, Offline
 - Vor & Nachteile

Hardware Architektur

- Abbildungen für shared und Distributed Memory ausfüllen
 - shared: Processor, Cache, Bus, Main-Memory
 - distributed: Processor, Cache, Bus, Main-Memory, Network
- Vor und Nachteile der beiden Architekturen nennen (nochmal)

POSIX Threads

- bei vorgegebenem Code Ablauf und Ausgabe erklären
 - welche Ausgabe ist in zufälliger Reihenfolge
 - was kommt definitiv vor dem anderen
- 2 Synchronisationsmöglichkeiten für (P)Threads nennen
 - Barrier
 - ???

openMP

- gegebene Code (for-Schleife mit #pragma omp parallel for und einer Addition drin)
 - #pragma omp atomic
 - #pragma omp critical
 - #reduce(+:iter)
 - Was ist schneller? Begründung
- vorgegebenen Code parallelisieren
 - Pseudo-Code genügt
- irgendwas inkrementelle Parallelisierung

MPI

- Eigenheit von MPI_Send
 - bei kleinen Datenmengen buffered
 - bei großen Datenmengen synchron
- Fehler im Code finden
- aus vorgegebenem Code Skizze des Communicators anfertigen und potenzielle Probleme identifizieren
 - buffered Ring-Kommunikation
 - bei großer Datenmenge Buffer zu klein -> Deadlock
- "Sie sollen Broadcast implementieren" (ideale Welt und natürlich möglichst geringe Laufzeit)
 - Laufzeit des eigenen Algorithmus in O-Notation angeben

Netzwerke

- 2 Netzwerktopologien skizzieren
 - z.B. Bus, Ring, Star, Fat Tree, Torus
 - jeweils Vor- und Nachteile nennen

Scalability

- Formeln für Speedup und Effizienz angeben
- Weak/strong scaling definieren
- aus vorgegebenen Daten Speedup Graph zeichnen und interpretieren
 - linear speedup
 - superlinear speedup
 - Abfall, weil Kommunikations-Overhead
- ...

???

- ...