

1. Übungsblatt

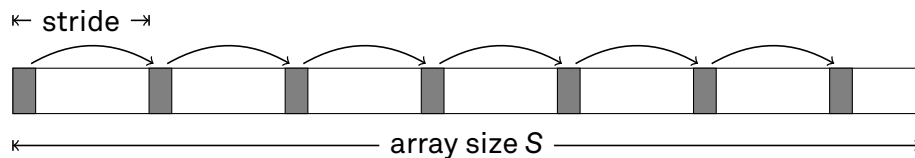
Ausgabe: 24. April 2017 · Besprechung: 8. Mai 2017

Einleitung

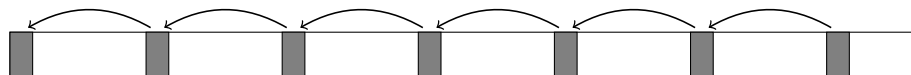
In diesem Aufgabenblatt soll es darum gehen, die *Latenzen* des Hauptspeichers und der verschiedenen Caches sichtbar zu machen und sie zu messen. Dazu soll ein *Micro-Benchmark* entwickelt werden, der mit definierten Zugriffsmustern auf den Speicher des Systems zugreift.

Grundsätzlich geschieht dies, indem wir ein Array im Hauptspeicher anlegen und geeignet initialisieren. Anschließend soll auf die Elemente des Arrays in definierter Form zugegriffen werden, z. B.

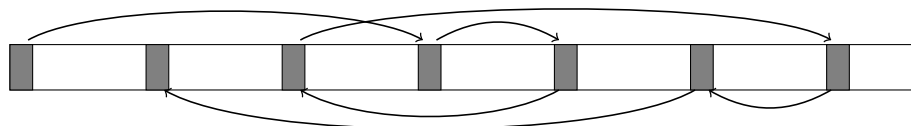
(i) sequentiell:



(ii) sequentiell, jedoch rückwärts:



(iii) "zufällig":



("Zufällig" ist hier vor allen Dingen aus der Sicht der Hardware zu sehen, das das Zugriffsmuster nicht "vorhersehen" können soll.)

Konfigurationsparameter des Benchmarks sind in allen Fällen die *stride size* s , d. h. der Abstand zwischen zwei Array-Elementen, sowie die Gesamtgröße des Arrays S .

Aufgaben

- (a) Entwickeln Sie ein C-Programm, um einen solchen Micro-Benchmark zu implementieren.
- (b) Lassen Sie den Benchmark mit verschiedenen Werten für s und S auf Ihrem System laufen (z. B. $s \in [8 \text{ bytes} \dots 16 \text{ kilobytes}]$ und $S \in [4 \text{ kilobytes} \dots 512 \text{ megabytes}]$).
- Welche Beobachtungen machen Sie?
 - Welches Verhalten hätten Sie erwartet (deckt sich das Experiment mit Ihren Erwartungen)?
 - Wie lässt sich das beobachtete Verhalten erklären?
 - Welche Rückschlüsse lassen sich aufgrund der Ergebnisse auf Ihre Hardware machen?

Sollten Sie mehrere verschiedene Rechner zur Verfügung haben (z. B. unterschiedliche Architekturen – AMD vs. Intel vs. ..., ältere vs. neuere Hardware, unterschiedliche Cache- oder Hauptspeicherkonfigurationen), sind Sie herzlich eingeladen, auch darauf zu experimentieren. Vielleicht haben Sie ja auch eine besonders exotische Hardware, die wir dann in der Übung ebenfalls diskutieren können.

Hinweise

Im Artikel *“What Every Programmer Should Know About Memory”* von Ulrich Drepper¹ finden Sie viele Hinweise zur Implementation des Benchmarks sowie zur Deutung der Ergebnisse.

¹<http://www.akkadia.org/drepper/cpumemory.pdf>