

Proseminar Rechnerarchitektur

Aufgabenzettel 14

Wintersemester 2021/22

27. Januar 2022

Zu bearbeiten bis Donnerstag, den **4. Februar**.

1 Amdahlsches Gesetz

Ein Programm besteht aus einer `main`-Funktion sowie zwei Funktionen `func1` und `func2`.

- Die `main`-Funktion braucht alleine 17.000 Instruktionen.
- `func1` wird 32 Mal aufgerufen und braucht jedes Mal 2.600 Instruktionen.
- `func2` wird 15 Mal aufgerufen und braucht jedes Mal 6.100 Instruktionen.

- a) Sie können nun eine Absolventin der Universität Innsbruck anstellen, die sich während ihres Informatikstudiums auf die Verwendung von x86-Vektorinstruktionen spezialisiert hat. Nach ihrer Schätzung kann sie `func1` auf 2.000 Instruktionen reduzieren und `func2` auf 4.550 Instruktionen.

Aufgrund Ihres Budgets können Sie die Absolventin nur mit der Optimierung einer der beiden Funktionen beauftragen. Berechnen Sie, welche der beiden Optimierungsmöglichkeiten sich mehr lohnt. Wie hoch ist die Beschleunigung S der besseren Variante gegenüber dem Ausgangszustand?

- b) Sie haben nun gemeinsam auch ein zweites Programm optimiert. Das Programm besteht zu 50 % aus Speicherzugriffen und zu 30 % aus parallelisierbarem Code.

Durch eine Optimierung der Speicherzugriffe sind diese nun 40 % schneller als vorher. Der nun parallelisierte Code läuft auf allen 8 Kernen mit perfekter Auslastung und ist dementsprechend 8 Mal so schnell. Die restlichen 20 % konnten nicht beschleunigt werden.

Wie groß ist die Beschleunigung dieses zweiten Programms insgesamt?

(Fortsetzung auf der nächsten Seite)

2 Cache Me If You Can

Gegeben sei folgendes ARM-Programm:

```
1 .global scalar_product
2 .align
3 .text
4 scalar_product:
5     STMFD sp!, {r4-r5, lr}
6     MOV r3, r0
7     MOV r0, #0
8     CMP r2, #0
9 scalar_loop:
10    LDMLEFD sp!, {r4-r5, pc}
11    LDR r4, [r3], #4
12    LDR r5, [r1], #4
13    MLA r0, r4, r5, r0
14    SUBS r2, r2, #1
15    B scalar_loop
```

Gehen Sie davon aus, dass Ihr System 256 KB Speicher unterstützt, der Byte-genau adressierbar ist. Es sei außerdem mit 4 KB Cache ausgestattet, der in Zeilen der Länge 32 Bytes aufgeteilt ist. Die beiden Vektoren v_0 und v_1 bestehen aus jeweils 32 32-Bit-Ganzzahlen (128 Byte insgesamt).

Nehmen Sie an, der Cache ist vollasoziativ (a–c).

- Wie viele Bits sind die Tags breit?
- Wie viele Vergleiche müssen pro Lesezugriff gemacht werden?
- Wie hoch ist die Trefferrate h bei der Ausführung von `scalar_product`, wenn der Cache zuvor leer (bzw. mit anderen Daten belegt) ist?

Gehen Sie für die restlichen Teilaufgaben (d–g) von einem direkt abgebildeten Cache aus.

- Skizzieren Sie, wie sich die m Bits des Adressbusses auf
 - Dekoder
 - Tag und
 - Offset innerhalb der Zeileaufteilen.
- Angenommen v_0 liegt ab Adresse 0x1000 und v_1 ab Adresse 0x3000 im Speicher. Wie hoch ist die Trefferrate h bei der Ausführung von `scalar_product`, wenn der Cache zuvor leer (bzw. mit anderen Daten belegt) ist?
- Wie hoch wäre h , wenn v_1 ab Adresse 0x3010 im Speicher läge? (unter sonst gleichen Bedingungen)
- Wie könnte man durch geschickte Programmänderungen die Trefferrate erhöhen?