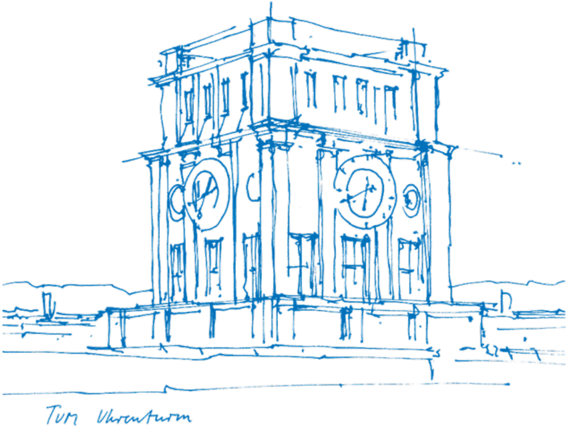
Please adjust group name in tumuser.sty.
Department of Electrical and Computer Engineering
Technical University of Munich

Grundlagenpraktikum: Rechnerarchitektur

Gruppe 104 - Abgabe zu Aufgabe A325



Viktor Bayo, Mikhail Lykov, Georgy Chomakhashvili

Lehrstuhl für Rechnerarchitektur und Parallelsysteme
Fakultät der Informatik
Technische Universität München

März 15, 2023

Outline

1 Problemstellung

2 Lösungsansatz

3 Genauigkeit

4 Performanz

Problemstellung

- Aufgabe:
  - $\sqrt{2}$ in wählbarer Genauigkeit zu berechnen
  - Wahlweise in dezimaler oder hexadezimaler Darstellung
- Zum Beispiel:
  - Darstellung: Dezimal, Genauigkeit: 5 $\Longrightarrow$ 1.41421
  - Darstellung: Hexadezimal, Genauigkeit: 10 $\Longrightarrow$ 1.6A09E667F3

Problemstellung

Theoretischer Teil

- Datenstruktur bignum
- Arithmetik
  - Addition
  - Subtraktion
  - Multiplikation
    - Karazuba Multiplikation
  - Division
    - Newton-Raphson-Verfahren

Praktischer Teil

- Berechnung der Konstante $\sqrt{2}$
  - Binnary Splitting
  - Newton-Raphson-Verfahren
- Analyse
  - Genauigkeit
  - Performanz

Outline

1 Problemstellung

2 Lösungsansatz

3 Genauigkeit

4 Performanz

Lösungsansatz
Rahmenprogramm

- Rahmenprogramm I/O-Operationen in C
- 3 Versionen:
  - Hauptimplementierung: Binary Splitting
  - Version 1: Binary Splitting + SIMD
  - Version 2: Newton–Raphson-Verfahren
- Standardwerte
- Tests

Lösungsansatz

Datenstruktur bignum

- Ganzzahlen beliebiger Größe zu speichern
- Format, um die Fixkommazahlen zu speichern
- bignum
  - mantissa
  - exponent

$$\overset{0123\,45678}{a = 6031.15357} \rightarrow$$

a.mantissa

0	1	2	3	4	5	6	7	8
6	0	3	1	1	5	3	5	7

$$\uparrow$$

$$a.exponent = 4$$

$$[0].\boldsymbol{mantissa \cdot base^{exponent}}$$

Lösungsansatz
Addition

- Beispiel:
  - $a = 13000$ $\quad a.mantissa = [1,3]$ $\quad a.exponent = 5$
  - $b = 720$ $\quad b.mantissa = [7,2]$ $\quad b.exponent = 3$
  - $result = a + b = 13720$
- Vorgehen:
  - a_shift = 2
  - b_shift = 0
  - result.mantissa = 4
  - result.exponent = 5

$$
\begin{array}{rcccc}
 & 0 & 0 & 0 & 0 \\
+ & 1 & 3 & & \\
\hline
 & 1 & 3 & 0 & 0 \\
+ & & & 7 & 2 \\
\hline
 & 1 & 3 & 7 & 2
\end{array}
$$

- Ergebnis: $[0].1372 \cdot 10^5 = 13720$

Lösungsansatz
Subtraktion

- Beispiel:
  - $a = 13.4$ $\quad a.mantissa = [1, 3, 4]$ $\quad a.exponent = 2$
  - $b = 0.25$ $\quad b.mantissa = [2, 5]$ $\quad b.exponent = 0$
  - $result = a - b = 13.15$
- Vorgehen:
  - Extra 1
  - a_shift = 1
  - b_shift = 0
  - result.mantissa = 4
  - result.exponent = 2

```
   [1] 0 0 . 0 0
+        1 3 . 4
-----------------
     1   1 3 . 4 0
-          0 . 2 5
-----------------
   [1] 1 3 . 1 5
```

- Ergebnis: $[0].1315 \cdot 10^2 = 13.15$

Lösungsansatz
Multiplikation

- Beispiel (Dezimal-Zahlensystem):
  - $a = 127$ $a.mantissa = [1, 2, 7]$ $a.exponent = 3$
  - $b = 254$ $b.mantissa = [2, 5, 4]$ $b.exponent = 3$
  - $result = a \cdot b = 32258$
- Vorgehen:
  - result.mantissa = 6
  - result.exponent = 6
  - l, r - Grenzen von Mantissen

$$result.mantissa[i] = \sum_{j=l}^{r} a.mantissa[j] \cdot b.mantissa[i - j] \tag{1}$$

- Laufzeit: $\Theta(n^2)$

Lösungsansatz
Multiplikation

$a = 127, b = 254, carry_i = result[i]/base$

- $result[0] = a_0 \cdot b_0 = 7 \cdot 4 = 28 \mod 10 = 8, c_0 = 2$
- $result[1] = a_0 \cdot b_1 + a_1 \cdot b_0 = 35 + 8 = 43 + c_0 = 45 \mod 10 = 5, c_1 = 4$
- $result[2] = a_0 \cdot b_2 + a_1 \cdot b_1 + a_2 \cdot b_0 = 14 + 10 + 4 = 28 + c_1 = 32 \mod 10 = 2, c_2 = 3$
- $result[3] = a_1 \cdot b_2 + a_2 \cdot b_1 = 4 + 5 = 9 + c_2 = 12 \mod 10 = 2, c_3 = 1$
- $result[4] = a_2 \cdot b_2 = 2 + c_3 = 3 \mod 10 = 3, c_4 = 0$
- $result[5] = 0 + c_4 = 0$

0 1 2 3 4 5

- $result = 8 \cdot 10^0 + 5 \cdot 10^1 + 2 \cdot 10^2 + 2 \cdot 10^3 + 3 \cdot 10^4 + 0 \cdot 10^5 = 32258$

$[0].032258 \cdot 10^6$

$32258 \cdot 10^5$

Lösungsansatz
Vektorisierte Multiplikation

- Vektor a:

128 Bit

[ ][a0]	[ ][a1]	[ ][a2]	[ ][a3]	[ ][a4]	[ ][a5]	[ ][a6]	[ ][a7]

- Vektor b:

[ ][b7]	[ ][b6]	[ ][b5]	[[b4]	[ ][b3]	[ ][b2]	[ ][b1]	[ ][b0]

- Multiplikationsvektor:

[a0 · b7]	[a1 · b6]	[a2 · b5]	[a3 · b4]	[a4 · b3]	[a5 · b2]	[a6 · b1]	[a7 · b0]

Lösungsansatz
Karazuba-Multiplikation

- $a = ah + al \cdot base^{-half\_size}$
- $b = bh + bl \cdot base^{-half\_size}$

$$ah \cdot bh + ((ah + al)(bh + bl) - ah \cdot bh - al \cdot bl)base^{-half\_size} + al \cdot bl \cdot base^{-2 \cdot half\_size} \quad (2)$$

- Laufzeit: $O(n^{1.59})$

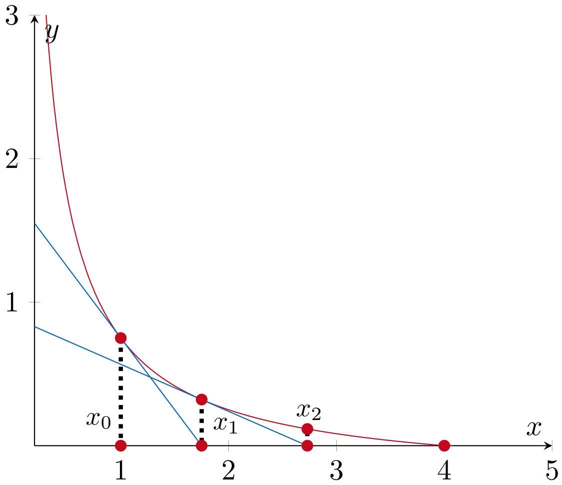
Lösungsansatz
Division

- $\frac{a}{b} = a \cdot \frac{1}{b}$, $\frac{1}{b}$ - Kehrwert
- Kehrwert wird mittels Newton-Raphson-Verfahren approximiert
- Vorgehen:
  - $f(x) = \frac{1}{x} - b$
  - $x_{i+1} = x_i + \frac{f(x)}{f'(x)}$
  - $x_{i+1} = x_i + (1 - b \cdot x_i)x_i$
  - $x_0 = 1$, da $[0].b \in [0.1; 1]$

- Laufzeit: $O(n^{1.59})$

Lösungsansatz

Newton-Raphson-Verfahren



Lösungsansatz
Binary Splitting

$$\boxed{\sqrt{2} = 1 + \sum_{i=1}^{\infty} \prod_{k=1}^{i} \frac{2k-1}{4k}} \tag{3}$$

$$S_{n_1,n_2} = \sum_{n=n_1}^{n_2-1} \prod_{k=n_1}^{n} \frac{2k-1}{4k} \qquad P_{n_1,n_2} = \prod_{k=n_1}^{n_2-1} 2k-1 \tag{4}$$

$$Q_{n_1,n_2} = \prod_{k=n_1}^{n_2-1} 4k \qquad T_{n_1,n_2} = Q_{n_1,n_2} \cdot S_{n_1,n_2} \tag{5}$$

Lösungsansatz
Binary Splitting

- $P_{n_1,n_2} = \begin{cases} p(n_1) = 2n_1 - 1 & \text{if } n_1 = n_2 - 1, \\ P_{n_1,m}P_{m,n_2} & \text{otherwise} \end{cases}$

- $Q_{n_1,n_2} = \begin{cases} q(n_1) = 4n_1 & \text{if } n_1 = n_2 - 1, \\ Q_{n_1,m}Q_{m,n_2} & \text{otherwise} \end{cases}$

- $T_{n_1,n_2} = \begin{cases} p(n_1) = 2n_1 - 1 & \text{if } n_1 = n_2 - 1, \\ T_{n_1,m} \cdot Q_{m,n_2} + P_{n_1,m} \cdot T_{m,n_2} & \text{otherwise, wobei } m = \left\lfloor \frac{n_1+n_2}{2} \right\rfloor \end{cases}$

- Laufzeit: $\mathcal{O}((n \log n)^{1.59})$

Lösungsansatz

Berechnung $\sqrt{2}$ mittels Binary Splitting

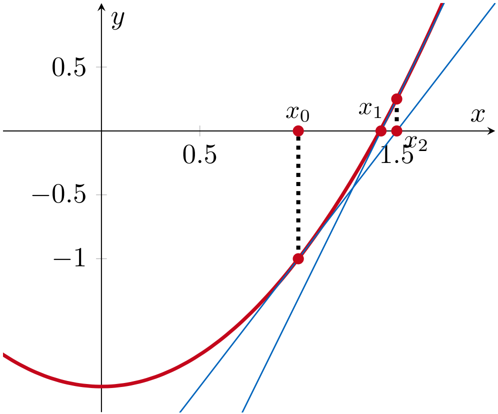
- $S_{n_1,n_2} = \frac{T_{n1,n2}}{Q_{n1,n2}}$
- $\sqrt{2} \approx 1 + S_{1,n+1}$ - ergibt n binäre Nachkommastellen.

- Laufzeit: $\mathcal{O}((n \log n)^{1.59})$

Lösungsansatz

Berechnung $\sqrt{2}$ mittels Newton-Raphson-Verfahren

- $f(x) = x^2 - 2.$
- $x_{i+1} = x_i + \frac{f(x)}{f'(x)}$
- $x_{i+1} = x_i - \frac{x^2-2}{2x}$
  $= x_i + \frac{1}{2}\left(\frac{2}{x_i} - x_i\right)$
- $x_0 = 1$, da $\sqrt{2} \geq 1$
- Laufzeit: $\mathcal{O}(n^{1.59})$



Outline

1 Problemstellung

2 Lösungsansatz

3 Genauigkeit

4 Performanz

Genauigkeit

$$\sqrt{2} - (1 + S_{1,to}) = 1 + \sum_{i=1}^{\infty} \prod_{k=1}^{i} \frac{2k-1}{4k} - (1 + S_{1,to}) =$$

$$= \sum_{i=to}^{\infty} \prod_{k=1}^{i} \frac{2k-1}{4k} \leq \sum_{i=to}^{\infty} \left(\frac{1}{2}\right)^{i+1} = \left(\frac{1}{2}\right)^{to} \tag{6}$$

$$\prod_{k=1}^{i} \frac{2k-1}{4k} = \frac{1}{4} \prod_{k=2}^{i} \frac{2k-1}{4k} \leq \frac{1}{4} \prod_{k=2}^{i} \frac{2k}{4k} = \frac{1}{4} \prod_{k=2}^{i} \frac{1}{2} \leq \left(\frac{1}{2}\right)^{i+1} \tag{7}$$

Outline

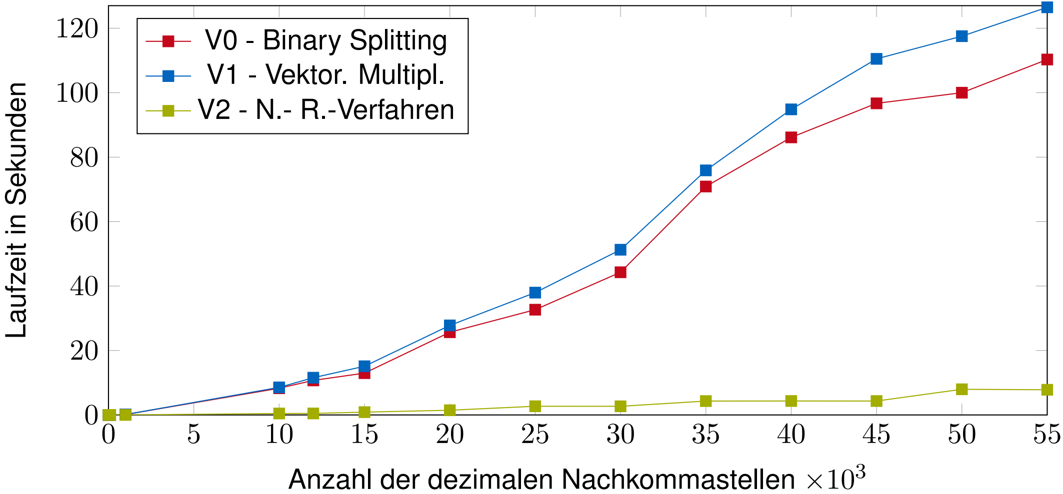
1 Problemstellung

2 Lösungsansatz

3 Genauigkeit

4 Performanz

Performanz



V0 - Binary Splitting
V1 - Vektor. Multipl.
V2 - N.- R.-Verfahren
Laufzeit in Sekunden
0
20
40
60
80
100
120
0
5
10
15
20
25
30
35
40
45
50
55
Anzahl der dezimalen Nachkommastellen $\times 10^3$