

Übungen zur Vorlesung
"Paralleles Höchstleistungsrechnen"

Prof. Dr. P. Bastian, Ch. Engwer

Besprechung am 04. 11. 2008 in der Übung

ÜBUNG 1 HIGH PERFORMANCE COMPUTING.

In dieser Übung wollen wir für verschiedene numerische Anwendung (sogenannte "kernel") ausprobieren wieviele Rechenoperationen pro Sekunde heute möglich sind. Implementieren Sie folgende mathematische Operationen:

1. *Matrixmultiplikation*. Gegeben zwei Matrizen $A, B \in \mathbb{R}^{n \times n}$ dann ist das Matrixprodukt $C = AB$ gegeben durch

$$c_{i,j} = \sum_{k=1}^n a_{i,k} b_{k,i} \quad .$$

2. *Gauß-Seidel 2d*. Es sei $\Omega_n^d = \{(i_0, \dots, i_{d-1}) \in \mathbb{Z}^d \mid \forall 0 \leq k < d : 0 \leq i_k < n\}$. Für die Gitterfunktionen $u^m : \Omega_n^2 \rightarrow \mathbb{R}$ definiere die Iteration

$$u^{m+1}(i, j) = \frac{1}{4} (u^{m+1}(i-1, j) + u^{m+1}(i, j-1) + u^m(i, j+1) + u^m(i+1, j)) \quad (i, j) \in [1, n-1]^2.$$

3. *Gauß-Seidel 3d*. Für die Gitterfunktionen $u^m : \Omega_n^3 \rightarrow \mathbb{R}$ definiere die Iteration

$$u^{m+1}(\alpha) = \frac{1}{26} \left(\sum_{\substack{\beta \in [0,1]^d \\ \beta \neq (0,0,0)}} u^{m+1}(\alpha - \beta) + u^m(\alpha + \beta) \right) \quad \alpha \in [1, n-1]^3.$$

Implementieren Sie diese Operationen in einer Programmiersprache ihrer Wahl mit einer Datenstruktur ihrer Wahl (z. B. mehrdimensionale Felder) und bestimmen Sie für jede Operation die Anzahl der Fließkommaoperationen aus und ermitteln Sie die Geschwindigkeit des Programmes in "Millionen Fließkommaoperationen pro Sekunde".

Zur Zeitmessung können Sie die Funktionen, welche in `timer.h` zur Verfügung gestellt werden, verwenden. Den header finden Sie auf der Vorlesungshomepage. Hinweise zur Verwendung finden Sie auf der nächsten Seite. Bei kleinem n sollten Sie für eine genaue Zeitmessung jede Operation mehrmals hintereinander in einer Schleife ausführen, um die Messfehler zu verringern.

Vergessen Sie nicht die Felder mit sinnvollen Daten (nicht 0.0) zu initialisieren, z. B. $u(i, j) = i + j$.

Übersetzen Sie das Programm mit maximaler Optimierungsstufe. Für den GNU C/C++ Compiler ist etwa `-O3 -funroll-loops` empfehlenswert.

Verbessern Sie die Cachenutzung durch Kacheln und ermitteln Sie die Beschleunigung.

Stellen Sie die Ergebnisse in graphischer Form, d. h. MFLOPS über Problemgröße dar. Diskutieren Sie die Form der Kurven, insbesondere warum die MFLOP-Rate wann absinkt. Vergleichen Sie die Ergebnisse von Matrixmultiplikation und Gauß-Seidel-Iteration.

Bei der Gauß-Seidel-Iteration können Sie mit anderen Durchlaufreihenfolgen experimentieren. Z. B. im 2d-Fall das Schachbrettmuster: Bearbeite zuerst alle Indizes mit $i + j$ gerade, dann die mit $i + j$ ungerade. Falls möglich, testen Sie ihre Programme auf verschiedenen Rechnern.

Hinweise zur Zeitmessung

Die verschiedenen Zeiten

Bei der Zeitmessung am Computer ergibt sich das Problem, dass die Zeit, die ein Programm benötigt von der Auslastung des Systems abhängt. Sind viele Prozesse tätig, bekommt der einzelne nur wenig Zeit und nützt dementsprechend lange.

Was konstant bleibt ist die sogenannte Prozessorzeit. Diese gibt an, wie viele Prozessorsekunden das Programm verbraucht hat. D. h. solange das Programm läuft tickt die Uhr weiter, wenn das Betriebssystem das Programm warten lässt steht die Uhr.

timer.h

Um Ihnen die Arbeit zu erleichtern haben wir für Sie eine Headerdatei mit Hilfsbefehlen um die verbrauchte Prozessorzeit auszulesen.

timer.h stellt Ihnen drei Befehle zur Verfügung:

- `void reset_timer(struct timeval * timer):` Zähler zurücksetzen / initialisieren.
- `double get_timer(struct timeval timer):` verbrauchte Sekunden auslesen.
- `void print_timer(struct timeval timer):` verbrauchte Sekunden ausgeben.

Beispiel

```
/* Headerfile zur Zeitmessung */
#include "timer.h"

int main()
{
    /* Variable zur Zeitmessung */
    struct timeval timer;
    /* Zähler zurücksetzen / initialisieren */
    reset_timer(&timer);
    // ... was tun und Zeit verbrauchen ...
    /* Zähler ausgeben */
    print_timer(timer);
}
```

Alle die sich für die Interna interessieren können in dem manpage zu `getrusage` (2) weitere Informationen finden.