Fakultät für Informatik und Mathematik
Lehrstuhl für Programmierung
Prof. C. Lengauer, Ph.D.

WS 2010/11

Übungen zur Vorlesung: Praktische Parallelprogrammierung
Blatt 4 (Physikalische Simulation)

Aufgabe 4 (Projekt n-Körper-Simulation) [18+6+2+2+2 Punkte]

In diesem Projekt wollen wir die Bewegung von Körpern unter dem Einfluss ihrer gegenseitigen Gravitationskräfte simulieren. Das Problem ist i. a. nicht analytisch lösbar, sodass die Simulation praktische Bedeutung besitzt.

Gegeben sind n Körper $K_1, \ldots, K_n$, die die Massen $m_1, \ldots, m_n$ besitzen. Unsere Simulation soll 2-dimensional sein, d. h. jedem Körper sind ein 2-dimensionaler Ortsvekter $\vec{x}$ und ein 2-dimensionaler Geschwindigkeitsvektor $\vec{v}$ zugeordnet.

Die Simulation verläuft in diskreten, gleichlangen Schritten der Länge Δt. In jedem Schritt s werden aus den Orts- und Geschwindigkeitsvektoren $\vec{x}_i^{(s)}$, $\vec{v}_i^{(s)}$ die Orts- und Geschwindigkeitsvektoren $\vec{x}_i^{(s+1)}$, $\vec{v}_i^{(s+1)}$ für den nächsten Zeitpunkt $s+1$ berechnet; siehe dazu unten angegebene Formeln.

In Stud.IP finden Sie Dateien mit Daten für den Ausgangszustand; Formatbeschreibung siehe unten.

(a) Implementieren Sie die n-Köper Simulation sequentiell und mit MPI+OpenMP. Der Benutzer soll die Anzahl der Simulationsschritte, den Parameter Δt und die Ausgangssituation (per Datei) angeben können. Der Endzustand der Simulation soll dann wieder in einer Datei gleichen Formats abgespeichert werden.

Lassen Sie das Programm am Ende die Interaktionsrate (für n Körper, Schrittezahl S und Gesamtrechenzeit T)

$$R(n, S, T) = n \cdot (n-1) \cdot \frac{S}{T}$$

d. h. die durchschnittliche Anzahl an Interaktionen je Zeiteinheit, ausgeben. Optimieren Sie die Interaktionsrate ihres parallelen Programms! Geben Sie am Anfang und am Ende der Simulation auch den Gesamtimpuls $\sum_{i=1}^{n} m_i \vec{v}_i$ aus; er sollte konstant bleiben.

Messen Sie Berechnungszeit, Speedup und Interaktionsrate mit dem Beispiel `spiralgalaxie.dat`.

(b) Benutzen Sie Newtons drittes Gesetz

$$\vec{F}_{i,j} = -\vec{F}_{j,i}$$

um die Zahl der Berechnungen zu halbieren.

(c) Newtons drittes Gesetz für alle Paarungen i, j zu verwenden ist nicht immer optimal, da der Kommunikationsaufwand steigt. Wenden Sie nun Newtons drittes Gesetz nur dort an, wo Berechnungen gespart werden können, *ohne* zusätzliche Kommunikation zu erzeugen.

(d) Erweitern Sie Ihr Programm um die Möglichkeit, Bilder mit den aktuellen Positionen der Körper abzuspeichern (z. B. als PBM oder PGM). Sehen Sie die Möglichkeit vor, nicht nur den Endzustand, sondern auch Zwischenzustände (beispielsweise alle d Schritte, mit vom Benutzer festlegbarem d) abzuspeichern. So können Sie kleine Animationen erzeugen, die Sie z. B. mit dem Kommando `animate` abspielen können.

(e) Überlegen Sie sich (eine) Möglichkeit(en), die Berechnung weiter zu beschleunigen, indem ein Verlust an Simulationsgenauigkeit in Kauf genommen wird. Skizzieren Sie Ihre Idee(n). Halten Sie ein solches Vorgehen für zulässig?

Hinweise zu C:

Benutzen Sie, um die Rechenfehler zu reduzieren, mindestens `double` Genauigkeit (64 Bit), besser `long double` (96 Bit). Bei MPI entspricht das `MPI_DOUBLE` bzw. `MPI_LONG_DOUBLE`.

Datenformat:

Die Anfangs- und Endkonfiguration des Systems wird in Dateien mit folgendem Format gespeichert. In der ersten Zeile steht die Anzahl der Körper, danach steht in jeder Zeile die Beschreibung eines Körpers: Masse (in kg), Position in x- und y-Richtung (in m), Geschwindigkeit in x- und y-Richtung (in $\frac{m}{s}$). Mit `#` werden einzeilige Kommentare eingeleitet. Beispiel:

```
2
1.98e30   0          0  0  0        # Sonne
5.977e24  1.496e11   0  0  29716    # Erde
```

Mechanik:

Ein Körper K_j übt nach Newtons Gravitationstheorie auf den Körper K_i ($i \neq j$) zum Zeitpunkt s die Kraft $\vec{F}_{i,j}^{(s)}$ aus:

$$\vec{F}_{i,j}^{(s)} = G \cdot \frac{m_i \cdot m_j \cdot (\vec{x}_j^{(s)} - \vec{x}_i^{(s)})}{|\vec{x}_j^{(s)} - \vec{x}_i^{(s)}|^3}$$

Die Gravitationskonstante G hat dabei den Wert $G \approx 6,674 \cdot 10^{-11} \frac{m^3}{kg \cdot s^2}$. Die Gesamtkraft $\vec{F}_i^{(s)}$ auf den Körper K_i ist

$$\vec{F}_i^{(s)} = \sum_{\substack{1 \leq j \leq n \\ i \neq j}} \vec{F}_{i,j}^{(s)}$$

Auf K_i wirkt damit die Beschleunigung $\vec{a}_i^{(s)}$:

$$\vec{a}_i^{(s)} = \frac{\vec{F}_i^{(s)}}{m_i}$$

Die Näherungen für die neue Position $\vec{x}_i^{(s+1)}$ und die neue Geschwindigkeit $\vec{v}_i^{(s+1)}$ ergeben sich aus $\vec{x}_i^{(s)}$, $\vec{v}_i^{(s)}$, $\vec{a}_i^{(s)}$ und Δt unter der Annahme, dass die Beschleunigung, die auf K_i wirkt, während des gesamten Intervalls Δt konstant $\vec{a}_i^{(s)}$ ist:

$$\vec{v}_i^{(s+1)} = \vec{v}_i^{(s)} + \vec{a}_i^{(s)} \cdot \Delta t$$
$$\vec{x}_i^{(s+1)} = \vec{x}_i^{(s)} + \vec{v}_i^{(s)} \cdot \Delta t + \frac{1}{2} \cdot \vec{a}_i^{(s)} \cdot (\Delta t)^2$$

Abgabe der Lösung: E-Mail an `groessli@fim.uni-passau.de` bis Dienstag, 21.12.2010 um 12.00 Uhr und Ausdruck in der Vorlesung am 21.12.2010.