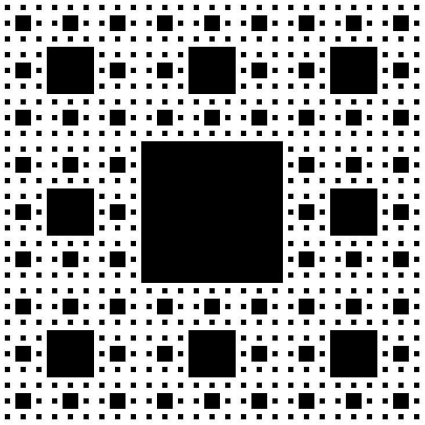

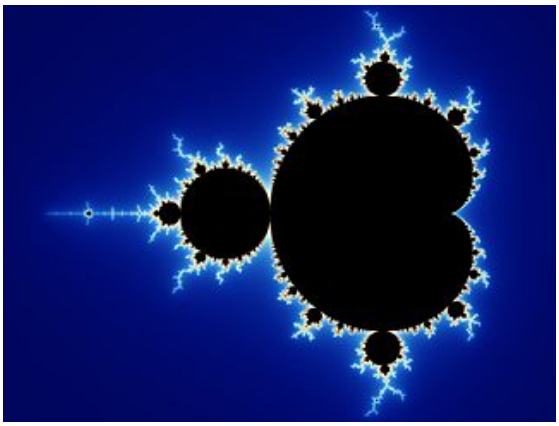

Abbildung 0.6: Links: Sierpinski-Fraktal vierter Stufe, Rechts: Mandelbrot-Menge. Quelle: Wikipedia.

IWR, Universität Heidelberg
Aufgabenblatt 9

Wintersemester 2012/13
19. Januar 2013

Übungen zur Vorlesung

Paralleles Höchstleistungsrechnen

Dr. S. Lang, J. Pods

Abgabe: 8. Januar 2013, 12 Uhr, per E-mail an `jurgis.pods@iwr.uni-heidelberg.de`

Dieses Blatt ist freiwillig. Die Punkte für dieses Blatt sind daher reine Zusatzpunkte und als Geschenke des Übungsleiters anzusehen ;) Da für die Mandelbrot-Aufgabe nichts selbst zu programmieren ist, sollten hier nur die Erfahrungen mit dem zur Verfügung gestellten Programm in der Abgabe dokumentiert werden, am besten mit Screenshots.

Übung 20 Weihnachtsaufgabe: Fraktale mit MPI (10 Punkte)

In dieser Aufgabe wollen wir zwei Fraktale mit MPI berechnen: Den Sierpinski-Teppich und die Mandelbrotmenge. Die Aufgabe ist hoch-experimentell und als Weihnachts-Schmankerl anzusehen, Punkte gibts für alle guten Ideen, Hinweise,...

Sierpinski-Teppich

Das erste Fraktal ist der sogenannte Sierpinsky-Teppich, siehe Abbildung 0.6 links. Ein Sierpinski-Teppich n.ter Stufe wird folgendermaßen erzeugt:

- Starte mit einem weißen Quadrat der Seitenlänge l.
- Unterteile das Quadrat in 9 gleichgroße Unterquadrate.
- Färbe das mittlere Quadrat schwarz.
- Fülle die anderen 8 Unterquadrate mit Sierpinski-Teppichen $(n-1)$-ter Stufe.

Der Sierpinski-Teppich 0-ter Stufe ist ein weißes Quadrat. Auf der Homepage finden Sie ein sequentielles Programm, das einen Sierpinski-Teppich n-ter Stufe erzeugt. Als Parameter erhält es die Grundlänge s des Ausgangsquadrats in Pixeln und die Stufe n , wobei gelten sollte $s = k \cdot 3^n$, $n = 1, 2, \ldots$ und $k \in \mathbb{N}_+$, d.h. s ist das Vielfache irgendeiner Dreier-Potenz und das Ausgansgquadrat lässt eine n-stufige Unterteilung zu. Das Programm erzeugt eine Datei `sierpinsky.ppm`, die Sie mit fast jedem Bildbetrachter anschauen können.

Aufgabe

Erweitern Sie das Programm, so dass es MPI-parallel arbeitet, d.h. jeder Prozess bearbeitet nur einen Teil des Feldes. Schwierigkeiten gibt es bei der Datei-Ausgabe. Im Prinzip gibt es mehrere Möglichkeiten:

1. Da die `ppm`-Dateien sehr groß werden können, lassen Sie jeden Prozessor seinen Anteil in einer eigenen Datei ausgeben. Speichern Sie aber nicht alle Dateien, Sie haben nur ca. 1 GB Platz!
2. Testen Sie das parallele Schreiben von Dateien mit den MPI-Funktionen `MPI_File_write()`, ...
3. Finden Sie ein besseres (bzgl. des Speicherverbrauchs) Format als `ppm`. Im Prinzip brauchen wir ja nur schwarz-weiss-Bilder und nicht den vollen RGB-Farbraum.

Mandelbrot-Menge

Die Mandelbrotmenge (Abbildung 0.6, rechts) ist über eine komplexe Folge $f_c^{(n)}(z) := z_n^2 + c,\ z_n \in \mathbb{C}\ \forall n \in \mathbb{N}$ definiert, wobei c eine gegebene komplexe Zahl ist. Ist für den Startwert $z_0 = 0$ die Folge beschränkt, so ist c in der Mandelbrotmenge. Es ist bekannt, dass alle c mit $|c| > 2$ nicht in der Mandelbrotmenge liegen. Ein Bild der Mandelbortmenge kann auf dem Bildschirm ausgegeben werden, in dem jedes Pixel einen Punkt der komplexen Ebene repräsentiert. Es werden dann n Iterationen berechnet und die Mandelbrotpunkte mit $|f^{(n)}(c)| < 2$ schwarz gefärbt. Interessant sind die Randpunkte der Mandelbrot-Menge. Diese kann man in Abhängigkeit davon, für welches n zuerst $|f_c^{(n)}(n)| > 2$ galt, einfärben. Der Rand kann mit immer größerer Auflösung untersucht werden. Auf der Homepage finden Sie ein sequentielles Programm, was die Begrenzungen der Intervalle und die Inkrements in der komplexen Ebene erhält und wiederum eine `ppm`-Datei exportiert.

Aufgabe

Das Programm kann MPI-parallel mit folgender Strategie behandelt werden: Die Berechnung der Farbwerte kann für jedes Pixel unabhängig erfolgen. Allerdings ist die Anzahl der Iterationen für jedes Pixel unter Umständen stark unterschiedlich. Daher wird der Bildschirm in einige Quadrate unterteilt und ein Prozess als Master bestimmt. Die Slaves fordern von ihm nach Erledigung ihrer Aufgaben neue Arbeit an (wie beim TSP-Problem). Ausgenutzt wird zusätzlich die Tatsache, dass alle Punkte in einem Quadrat die gleiche Farbe haben, falls die Randpixel auch schon alle die gleiche Farbe haben. Mit der initialen Quadrateaufteilung erhalten die Slaves nun die Aufgabe, die Farben eines ihnen zugewiesenen Quadrates zu errechnen. Sind die Punkte auf dem Rand nicht alle von der gleichen Farbe, wird das Quadrat in vier kleinere geteilt und als neue Aufgaben an den Master zurückgesendet. Die Tiefe der Verschachtelung hat natürlich eine Obergrenze. Die Implementierung ist etwas aufwändig, daher betrachten wir eine Variante von von Skjellum, Lusk und Gropp mit parallelem Rendering der Ausgabe über einen parallelen X-Server.

1. Kopieren Sie den Ordner `/export/home/dune/dunkurs/mandel` in Ihr Homeverzeichnis.
2. Schreiben Sie in Ihre `~/mpihosts` Datei nur zwei Zeilen, in jeder steht `localhost`. Auf diese Art kann man MPI-parallele Programme auf einem Multicore-Rechner ausführen, d.h. es wird das MIMD-Modell ohne echte physikalische Trennung der Rechner emuliert.
3. Legen Sie eine Date `~/.bashrc` an (oder editieren eine gegebenfalls schon bestehende Datei):

```
1 PATH=/export/home/dune/dunekurs/mpich-install/bin:$PATH;
    export PATH
```

   Das Programm funktioniert nur mit `MPICH`. Ich habe eine aktuelle Version von MPICH lokal in meinem Folder kompiliert und die Makefiles so angepasst, dass gegen diese gelinkt wird.
4. Laden Sie die Einstellungen, `source ~/.bashrc`.
5. Kompilieren Sie das Programm `pmandel` durch Aufruf von `make` im Verzeichnis `mandel`.
6. Starten Sie das Programm durch Aufruf von

```
1 mpiexec -f ~/mpihosts -n 2 ./pmandel
```

   im Verzeichnis `mandel`. Viel Spass!

Nachdem wir ein bisschen herumgespielt haben, sind besonders folgende Punkte interessant an dieser Implementierung:

- Dynamische Verteilung der Last durch einen Masterprozess,
- Der Einsatz abgeleiteter Datentypen in MPI (Methoden `MPI_Type_contiguos, MPI_Type_commit` in `pm_genproc.c`),
- Die Verwendung der Grafik-Bibliothek MPE zur visuellen Kontrolle der Berechnungen. Dabei wird ein paralleles X11 verwendet.

Nachdem Sie das Program getestet haben, suchen Sie nach Dokumentation zu den MPI-Datentypen und zur MPE-Bibliothek. Versuchen Sie, die interessanten Teile der Implementierung nachzuvollziehen. Das wars!

! Achten Sie darauf, daß Sie sich immer mit `-X` einloggen: `ssh -X phr...`, sonst haben Sie keine graphische Ausgabe!

! Nehmen Sie nach dem Test des Mandelbrot-Programms die Änderungen an `~/.bashrc` unbedingt wieder zurück, sonst arbeiten Sie immer mit MPICH!

- Richtiges paralleles Mandelbrot-Rechnen mit paralleler Ausgabe geht im Pool leider nicht, da erst die graphische Umleitung eingerichtet werden müsste (Vielleicht finden Sie ja heraus, wie man so etwas als nicht-Superuser machen kann?).
- Sie können Sich natürlich gerne auch eine eigene Version von MPICH lokal in Ihrem Home-Verzeichnis kompilieren.