

1 Parallelisierung mit MPI (Gauß-Seidel: 500 Punkte)

Parallelisieren Sie das GS-Verfahren in dem `partdiff`-Programm gemäß dem Parallelisierungsschema.

Beachten Sie dabei folgende Anforderungen:

Abbruch

- Es gibt hier zwei Fälle, die auf Korrektheit der Parallelisierung zu prüfen sind:
 1. Abbruch nach fester Iterationszahl (beide Störfunktionen)
 2. Abbruch nach Genauigkeit (beide Störfunktionen)
- Dabei soll nach gleicher Iterationszahl das Ergebnis (Matrix und Fehlerwert) identisch bleiben.
- Beim Abbruchkriterium „erreichte Genauigkeit“ muss die parallele Variante nicht unbedingt bei derselben Iteration wie die sequentielle abbrechen. Es kann bis zu $|\text{nprocs}|$ Iterationen mehr gerechnet werden. Die Iteration, in der abgebrochen wird, muss aber deterministisch und reproduzierbar sein. Die Matrix und der Fehlerwert für $i + |\text{nprocs}|$ Iterationen verglichen mit dem sequentiellen Programm, welches nach $i + |\text{nprocs}|$ Iterationen abbricht, müssen gleich sein!
- Überprüfen Sie, dass die Ergebnisse mit 24 Prozessen auf zwei Knoten richtige Ergebnisse liefern.
- Sie dürfen keine mathematischen Abkürzungen nutzen, die es erlauben würden, nur auf Grundlage eines prozesslokalen Residuums über den Abbruch nach Genauigkeit zu entscheiden. Der Abbruch darf lokal nur auf Grundlage des globalen maximalen Residuums über alle Prozesse aus der gleichen Iteration entschieden werden.

Code

- Zu keinem Zeitpunkt darf ein Prozess die gesamte Matrix im Speicher halten. Die Matrix soll auf alle Prozesse gleichmäßig verteilt werden.
- Das Programm muss mit beliebigen Prozesszahlen funktionieren (auch für $\text{nprocs} == 1$ oder $\text{nprocs} \gg N$). Einfach das ganze Programm kontrolliert abzubrechen, reicht nicht.
- Erstellen Sie eine eigene Funktion für die MPI-Parallelisierung des GS-Verfahrens.
- Jacobi muss dabei weiterhin **parallel** funktionieren.

- Verwenden Sie die Funktion `displayMatrixMPI`, die in den Materialien bereitgestellt wird, als Grundlage für die parallele Ausgabe der Matrix.
Hinweis: Verwenden Sie diese parallele Ausgabe mit Bedacht (d.h. am Ende), da dadurch die Teilmatrix vom ausgebenen Rang überschrieben wird! Zwischenausgaben mit dieser Matrix führen also zu einem falschen Endergebnis und sollten nur zu Testzwecken verwendet werden.
- Dokumentieren Sie Ihren Code ausreichend.

Laufzeit

- Das Programm darf mit zwei Prozessen nicht langsamer als die sequentielle Variante sein.
- Es muss ein zufriedenstellender Speedup erreicht werden. Ermitteln Sie diesen nach den üblichen Konventionen schriftlich und geben Sie die verwendete Konfiguration an.

Kommunikation

- Nutzen Sie synchronisierende Funktionen (`MPI_Ssend` und `MPI_Issend`) um ihr Programm auf mögliche Verklemmungen zu untersuchen. Für die finale Abgabe können sie diese Funktionen gegen die Standard-Funktionen wieder austauschen.
- Jeder nichtblockierende Kommunikationsaufruf muss mit einem passenden `MPI_Wait` oder einem erfolgreichen `MPI_Test` abgeschlossen werden. Anderenfalls ist der Aufruf falsch.

1.1 Trace

Erstellen Sie mit `Vampir` je eine Visualisierung eines Traces ihrer Implementation für Abbruch nach Iterationen als auch Abbruch nach Genauigkeit.

- Verwenden Sie hierfür eine Ausführung mit insgesamt 12 Prozessen auf zwei Knoten.
- Die Programmlaufzeit sollte mindestens 10 Sekunden betragen.
- Die Zahl der ausgeführten Iterationen soll im hohen einstelligen bzw. niedrigen zweistelligen Bereich liegen (In der Visualisierung sollten einzelne Iterationsphasen erkennbar sein).
- Geben Sie die jeweils verwendete Konfiguration an.

Abgabe

Abzugeben ist ein gemäß den bekannten Richtlinien erstelltes und benanntes Archiv. Das enthaltene und gewohnt benannte Verzeichnis soll folgenden Inhalt haben:

- Alle Quellen, aus denen Ihr Programm besteht; gut dokumentiert (Kommentare bei geänderten Code-Teilen!)
 - Erwartet werden die Dateien `Makefile`, `askparams.c`, `partdiff.c` und `partdiff.h`.
- Ein Dokument `antworten.pdf` mit Ihren Antworten/Messungen und Visualisierungen
- **Keine** Binärdateien!

Achten Sie darauf, dass ihre Programme ohne Warnungen oder Fehler bauen und Valgrind keine Probleme anzeigt. Packen Sie ein komprimiertes Archiv (`.tar.gz`) aus dem sauberen Verzeichnis (**ohne Binärdateien oder versteckte Dateien**). Benennen Sie das Archiv nach den Nachnamen der Gruppenmitglieder (z. B. `MustermannMusterfrau.tar.gz`).

Ein Mitglied Ihrer Gruppe legt das Archiv dann in

`$HOME/HR-Abgaben-2526/Blatt-10`

ab und schickt nach erfolgter Abgabe eine E-Mail an `jannek.squar@uni-hamburg.de`, in der Sie einfach den absoluten Pfad zu Ihrem Archiv angeben:

`$HOME/HR-Abgaben-2526/Blatt-10/MustermannMusterfrau.tar.gz`.

Hinweis: `$HOME` ist eine Umgebungsvariable, die auf Ihr Heimatverzeichnis zeigt. Es sollte also nicht wortwörtlich so in der E-Mail stehen.