

Hochleistungsrechnen 2019/2020

Übung

Michael Kuhn, Michael Blesel, Ruben Felgenhauer, Julius Plehn

Wissenschaftliches Rechnen
Fachbereich Informatik
Universität Hamburg

michael.kuhn@informatik.uni-hamburg.de
3blesel@informatik.uni-hamburg.de
4felgenh@informatik.uni-hamburg.de
3plehn@informatik.uni-hamburg.de

14. und 15.10.2019

Arbeitsbereich - Was kann man bei WR sonst noch tun?

- Praktikum C-Programmierung
- Programmierung für Naturwissenschaftler (PfN)
- Praktikum Einführung in die Parallele Programmierung für Geowissenschaftler (PPG)
- Master Vorlesung: Hochleistungs-Ein-/Ausgabe (HEA)

Weitere Veranstaltungen:

- Seminare z.B. Softwareentwicklung in der Wissenschaft, Effiziente Programmierung
- Praktikum: Aufbau eines Clusters / Parallele Programmierung
- Student Cluster Competition (SCC)

Studierende - Wer seid ihr?

Kleine Umfrage

- Studiengang, Semester:
- Vorkenntnisse:
 - Programmierkenntnisse?
 - C?
 - Bash?
 - Linuxgrundlagen?
 - Module bei WR?

Die Übung

- Was erwartet euch?
 - Viel Programmieren in C
 - Spaß!
- Vorkenntnisse nicht nötig, aber von Vorteil
- Ausreichende Vorbereitung in den Übungen
- 3 mögliche Übungstermine:
 - Montags: 14-16 Uhr
 - Dienstags: 12-14 Uhr
 - Dienstags: 16-18 Uhr

Termine, Inhalte und vorläufige Punkteverteilung

15./16.10.	Ablauf und Einführung Linux-Cluster	200
22./23.10.	Debugging	300
29./30.10.	Leistungsoptimierung sequentieller Anwendungen	310
05./06.11.	OpenMP	360 + 300
12./13.11.	POSIX-Threads	360
19./20.11.	Einführung MPI	180
26./27.11.	PDE-Parallelisierungsschema	200
03./04.12.	PDE-Parallelisierung (Jacobi)	300 + 150
10./11.12.	PDE-Parallelisierung (Gauß-Seidel)	600
17./18.12.	Zwischenbesprechung (Gauß-Seidel)	
14./15.01.	PDE-Auswertung	240
21./2.01.	PDE-Spurdatenanalyse	180
28./29.01.	Nachbesprechung	
		<hr/> <hr/>
		3200+

Änderungen vorbehalten.

Übungsblätter

- Ein Übungsblatt pro Woche
 - Viel Programmieren und Rückmeldung
- Abgabe in **Gruppen** je 3 Personen
- Abgabe **pünktlich**: spätestens Samstag um 23:59
- Abgabe an `hr-abgabe@wr.informatik.uni-hamburg.de`
 - Konventionen einhalten!
 - Siehe Übungsblatt 0
- Ausgabe des neuen Zettels am Sonntag
- Besprechung in der nächsten Übung

Bewertung der Übung

- Nicht pünktliche oder den Konventionen nicht entsprechende Abgaben werden **nicht** korrigiert = 0 Punkte
- Korrektheit
- Vollständigkeit
- Einhaltung der Forderungen an das Programm (zum Beispiel Speedup)
- Gute Dokumentation des Quelltextes

Bestehen der Übungen

- Mindestens 1 Mal präsentieren!
 - Potentielle Vorrechner werden ggf. vorher per Mail informiert
 - **Jedes** Gruppenmitglied muss in der Lage sein vorzurechnen
 - Wer nicht vorrechnen kann/will bekommt **keine** Punkte
- Mindestens 50% der Gesamtpunkte
- 9 Blätter - mindestens je 25%
- Blätter zu Gauß-Seidel und Jacobi verpflichtend
- Maximal 2x unentschuldigt fehlen

Wichtige Mail-Adressen

Anliegen	Ansprechpartner	Mail-Adresse
Ich bin nicht angemeldet!	Studienbüro	studienbuero@inf... ¹
Ich komme bei Aufgabe x nicht weiter!	Mailingliste	hr-1920@wr.inf... ¹
Ich möchte abgeben!	Abgabebot	hr-abgabe@wr.inf... ¹
Sonstiges	Michael Kuhn Michael Blesele	michael.kuhn@inf... ¹ 3blesele@inf... ¹

¹informatik.uni-hamburg.de

Codekonventionen: C Coding Standard

- Variablen: `int example_variable` bzw `int exmp_var`
- Globale Variablen: möglichst wenige.
- Konstanten: `GLOBAL_CONSTANT_LIKE_THIS`
- 78 Zeichen pro Zeile
- `.h`: header files; `.c`: source files
- Goto: möglichst selten
- Magic numbers: s. Goto

"Continue and break like goto should be used sparingly as they are magic in code. With a simple spell the reader is beamed to god knows where for some usually undocumented reason."

Nicht verhandelbare Codekonventionen

- **Kommentare!**
- **Fehlerbehandlung!**
- **Variableninitialisierung!**

Compilerwarnungen

- Compiler-Warnungen sind meistens sinnvoll.
- Es ist nicht erlaubt, das Makefile zu editieren, um Warnungen zu unterdrücken. Ausnahme: Neuerer C-Standard.
- Warnungen sind vor der Abgabe wie Fehler zu fixen.
- Aber: Wenn die Zeit knapp ist, sorgt erst mal dafür, dass alles läuft.

Hilfreiche Kommentare?

```
1  /* -----  
2  * void finalize(RHC* rhc)  
3  * finalizing an RHC structure.  
4  * rhc: a pointer to the RHC structure to finalize  
5  * -----*/  
6  void finalize(RHC* rhc){  
7      int i;  
8      // iterate over all IdGs  
9      // and free their pointers  
10     for( i = 0; rhc._mIdG->c > i; i= i+1){  
11         free(rhc._mIdG->p[i]);  
12     }  
13 }
```

Sprechende Bezeichnungen!

```
1 void freeAllGlobalIDs(RawHandlerContext* rhc){  
2     for(int i = 0; i < rhc.GlobalID->count; ++i){  
3         free(rhc.globalID->buffer[i]);  
4     }  
5 }
```

Hilfreiche Kommentare!

```
1  /* Unfortunately this code is necessary because there
2  seems to be a glitch in RawHandlerManager that otherwise
3  leaks GlobalID buffers.
4  Hope we can fix that bug soon and then get rid of this workaround.
   ↪ 2008-07-23 */
5
6  void freeAllGlobalIDs(RawHandlerContext* rhc){
7      for(int i = 0; i < rhc.GlobalID->count; ++i){
8          free(rhc.globalID->buffer[i]);
9      }
10 }
```

Problematisch: Nicht aktualisierte Kommentare

"Es bewährt sich, beim Lesen eigener wie fremder Kommentare allgemein nicht davon auszugehen, dass die Kommentare in irgendeinem sinnvollen Verhältnis zum Code stehen. Betrachten Sie sie wie die Auskünfte eines Betrunkenen: interessiert, aber skeptisch."

Katrin Passig (Weniger schlecht programmieren)

Highlights der Kommentare?!

- "total_hours_wasted_here = 16"
- "Now, God only knows"
- "drunk, fix later"
- "Temporary my ass!"
- "please work"
- "too scared to delete"
- "Dear future me. Please forgive me."
- "it was hard to write so it should be hard to read..."
- "TODO: Fix this. Fix what?"

Übungsblatt 1

- Ziel der ersten Übung:
 - Kennenlernen der Command Line und des Clusters
 - Einführung in C
- WR-Cluster Eckdaten:
 - 10 Knoten
 - Pro Knoten 2 Prozessoren mit jeweils 6 Kernen und 12 GB Hauptspeicher
 - Betriebssystem Ubuntu 18.04
- Einloggen per SSH
 - Eure Aufgabe: Account holen!
 - Domain: `cluster.wr.informatik.uni-hamburg.de`

Materialien

- NICHT in STiNE!
- WR-Homepage: https://wr.informatik.uni-hamburg.de/teaching/wintersemester_2019_2020/hochleistungsrechnen
- Mailingliste: <https://wr.informatik.uni-hamburg.de/listinfo/hr-1920>
- Beginners' Guide: https://wr.informatik.uni-hamburg.de/teaching/ressourcen/beginners_guide

Tutorials und Bücher

- Liste empfohlener Bücher und Tutorials zu C:
<http://www.iso-9899.info/wiki/Books>
- The C Book:
http://publications.gbdirect.co.uk/c_book/
http://publications.gbdirect.co.uk/c_book/thecbook.pdf

Todo's für diese Woche

- **Mailingliste und Cluster Zugang!!!**
- 3er Gruppen bilden (am besten jetzt)
- Erstes Übungsblatt