OpenMP

Proseminar Effiziente Programmierung in C

Niklas Beck

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

21.6.2021

Gliederung

Was ist OpenMP? [1][4]

- Open Multi-Processing
- Oracle, IBM, Intel u.a.
- C/C++, Fortran
- Shared-Memory-Programmierung

Warum OpenMP? [2][3][4]

- parallelisierung oft sehr wichtig
- einfach zu erlernen
- funktioniert auch ohne OpenMP wissen vom Compiler
- Wirtschaft, Industrie bis zu High-Supercomputing-Systemem

Hauptverwendung [5][9]

- Schleifen
  - Daten-Parallelismus
  - Task-Parallelismus
- Auslagerung von Code mit Accelerator

Fork-Join Modell [8]

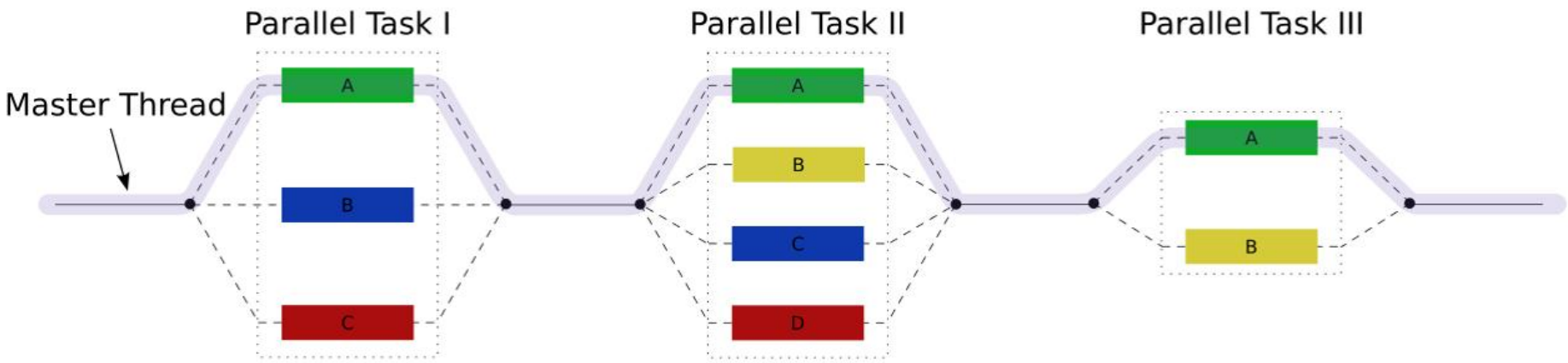

Abbildung: Fork-Join Modell

Daten-Parallelismus [6]

- Eine Operation
- mehrere Daten
- Aufteilung der Daten auf Threads
  - gleichmäßige Verteilung

Beispiel Daten-Parallelismus [6][8]

- Arrays a,b,c mit länge 12
- Parallelismus mit Schleife
  - c[i]=b[i]+a[i]
  - c[i]= c[i]*2
  - c[i]= C[i]+10

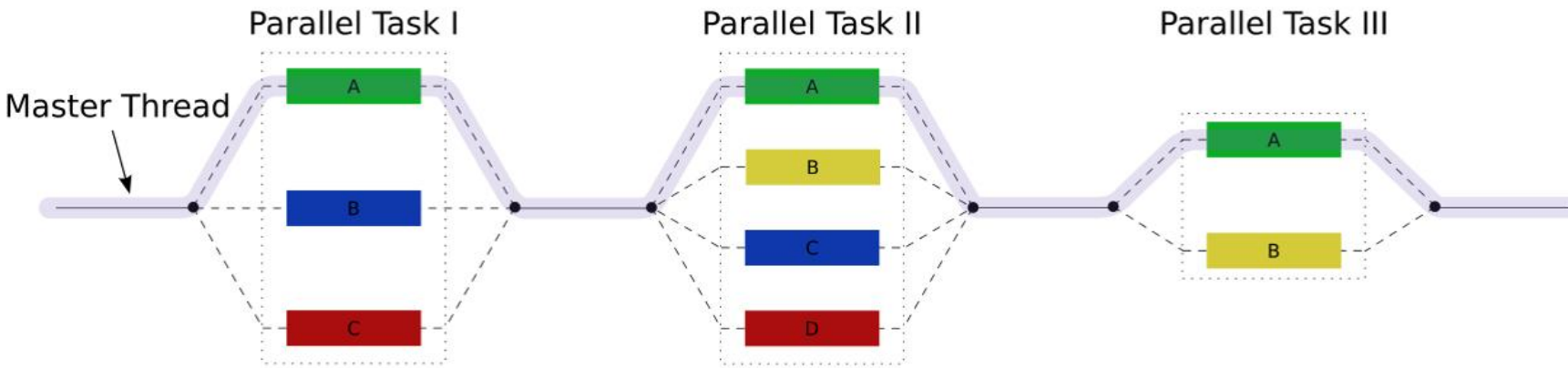

Abbildung: Fork-Join Modell

Task-Parallelismus[6]

- Mehrere Operation
- die selben Daten
- Verteilung von einzelnen Operationen auf Threads

Beispiel Task-Parallelismus [6][11]

- (a-b)*(a+b)
- Makierung der Parallelen Region für OpenMP
- Makierung der Aufgabe für OpenMP
  - (a-b)=c Thread 0
  - (a+b)=d Thread 1
- einbauen einer Barriere
- es soll nur ein Thread weiter rechnen
  - c*d

Accelerator-Support [9]

- OpenMP unterstützt Accelerator
- Auslagerung von Rechenleistung auf GPU
- Makierung für OpenMP was ausgelagert werden soll
  - #pragma omp parallel for
  - #pragma omp target teams distribute parallel for

Kompilieren[12][13]

- gcc -fopenmp (Name der C-Datei)
- bedingte Kompilierung
  - omp_get_thread_num()
  - #ifdef _OPENMP
  - #endif

Implementation [5]

- #include <omp.h>
- #pragma omp Direktive [Klauselliste]
- Beispiel Direktiven
  - parallel
  - parallel for
- Beispiel Klauseln
  - private, shared

Beispiel 1 (abgewandelt von [14])

```
1  #include <stdio.h>
2  #include <omp.h>
3
4  int main()
5  {
6      #pragma omp parallel
7      {
8          printf("Hello World");
9      }
10      return 0;
11 }
```

Listing 1: Hello World

Beispiel 2 (abgewandelt von [14])

```
1  #include <stdio.h>
2  #include <omp.h>
3
4  int main()
5  {
6      omp_set_num_threads(4);
7      #pragma omp parallel
8      {
9          int i = omp_get_thread_num();
10         printf("Hello World(%d)\n", i);
11     }
12     return 0;
13 }
```

Listing 2: Hello World erweitert

Beispiel 3 (abgewandelt von [15])

```
1  int main()
2  {
3      omp_set_num_threads(4);
4      int n = 12;
5      int a[] = {1,2,3,4,5,6,7,8,9,10,11,12};
6      int b[] = {1,2,3,4,5,6,7,8,9,10,11,12};
7      int c[] = {0,0,0,0,0,0,0,0,0,0,0,0};
8      #pragma omp parallel for
9          for(int i = 0;i<n;i++)
10         {
11             c[i] = a[i] + b[i];
12             printf("%d\n", c[i]);
13         }
14     return 0;
15  }
```

Listing 3: for-Schleife(Daten-Parallelismus)

Beispiel 4

```
1  #pragma omp parallel
2  {
3      #pragma omp task
4      {c = a-b;}
5      #pragma omp task
6      {d = a+b;}
7      #pragma omp barrier
8      #pragma omp single
9      {
10         e = c*d;
11         printf("Das Ergebnis ist:%d", e);
12     }
13 }
```

Listing 4: Task-Paralellismus

Nützliches [5][8]

- master
- critical
- atomic
- omp_get_num_threads()
- omp_get_max_threads()

Zusammenfassung

- Was ist OpenMP?
- Funktionen
  - Task-Parallelismus
  - Daten-Parallelismus
  - Accelerator-Support
- Implementation
  - parallele Region
  - parallelisierung von Schleifen
  - verteilung der Aufgabe

[1] OMPAPI.General.01. URL: https://www.openmp.org/about/openmp-faq/ Zugriff 18.06.2021

[2] OMPAPI.General.03. URL: https://www.openmp.org/about/openmp-faq/ Zugriff 18.06.2021

[3] OMPAPI.General.05. URL: https://www.openmp.org/about/openmp-faq/ Zugriff 18.06.2021

[4] OpenMP Wikipedia. URL: https://de.wikipedia.org/wiki/OpenMP Zugriff 17.06.2021

[5] OpenMP. URL: https://de-academic.com/dic.nsf/dewiki/1052687 Zugriff 18.06.2021

[6] Data parallelism vs Task parallelism. URL: https://www.tutorialspoint.com/data-parallelism-vs-task-parallelism Zugriff 17.06.2021

[7] Direktiven. URL: https://docs.microsoft.com/de-de/cpp/parallel/openmp/reference/openmp-directives?view=msvc-160 Zugriff 18.06.2021

[8] Fork-Join URL: https://www.programmersought.com/article/18904878560/ Zugriff 15.06.2021

[9] OpenMP Accelerator Support. URL: https://www.openmp.org/updates/openmp-accelerator-support-gpus/ Zugriff 16.06.2021

[10] OpenMP Methoden. URL: https://docs.microsoft.com/de-de/cpp/parallel/openmp/reference/openmp-functions?view=msvc-160 Zugriff 18.06.2021

[11] OpenMP Task. URL: https://www.ibm.com/docs/en/zos/2.1.0?topic=SSLTBW_2.1.0/com.ibm.zos.v2r1.cbclx01/prag_omp_task.html Zugriff 18.06.2021

[12] Compelieren. URL: https://www.dartmouth.edu/ rc/classes/intro_openmp/compile_run.html Zugriff 18.06.2021

[13] 2.4 OMP-Elemente. URL: https://homepages.uni-regensburg.de/ brf09510/EDV/kurs_info/brf09510/hpc/openMP/openmp.html Zugriff 18.06.2021

[14] 2. URL: https://www.geeksforgeeks.org/openmp-hello-world-program/ Zugriff 18.06.2021

[15] Seite 10. URL: https://numerik.uni-koeln.de/sites/numerik/uebungen/hpc_16/Grundlagen_OpenMP.pdf Zugriff 18.06.2021