

HLR OPENMP, Blatt 4

Arne Struck, Jonathan Werner

15. November 2014

1 Optimierung

Sequentiell, ohne `-fopenmp`:

`./partdiff-seq 1 2 512 2 2 200` braucht **120.29s**.

1 Thread, mit `-fopenmp`:

`./partdiff-seq 1 2 512 2 2 200` braucht **120.65s**.

12 Threads, mit `-fopenmp`:

`./partdiff-seq 1 2 512 2 2 200` braucht **14.9s**.

Mit verschiedenen Scheduling, jw **12 Threads**:

- `schedule(dynamic, 1)`:
 - `./partdiff-seq 1 2 512 2 2 200` braucht **11.39s**.
 - Speedup zu ohne OMP: **10.59**. The winner! :)
- `schedule(dynamic, 4)`:
 - `./partdiff-seq 1 2 512 2 2 200` braucht **14.2s**.
 - Speedup zu ohne OMP: **~8.5**.
- `schedule(guided)`:
 - `./partdiff-seq 1 2 512 2 2 200` braucht **12.3s**.
 - Speedup zu ohne OMP: **9.8**.
- `schedule(static, 1)`:
 - `./partdiff-seq 1 2 512 2 2 200` braucht **14.1s**.
- `schedule(static, 2)`:
 - `./partdiff-seq 1 2 512 2 2 200` braucht **14.7s**.
- `schedule(static, 16)`:
 - `./partdiff-seq 1 2 512 2 2 200` braucht **14.57s**.

Zeit benötigt: 2h. Fehlersuche? Ja, zuerst haben wir die simple interference function genommen und dadurch nur ein Speedup von 4 erreicht.

2 Messungen

2.1 Messung 1

procs	speedup	time in sec
1 (no OMP)	1	120.29
1	1	120.65
2	2.044222298	59.02
3	2.960009814	40.76
4	4.045942321	29.82
5	4.868845843	24.78
6	5.952146029	20.27
7	6.710233593	17.98
8	7.65060241	15.77
9	8.431167016	14.31
10	9.1332324	13.21
11	9.913722268	12.17
12	9.995857498	12.07

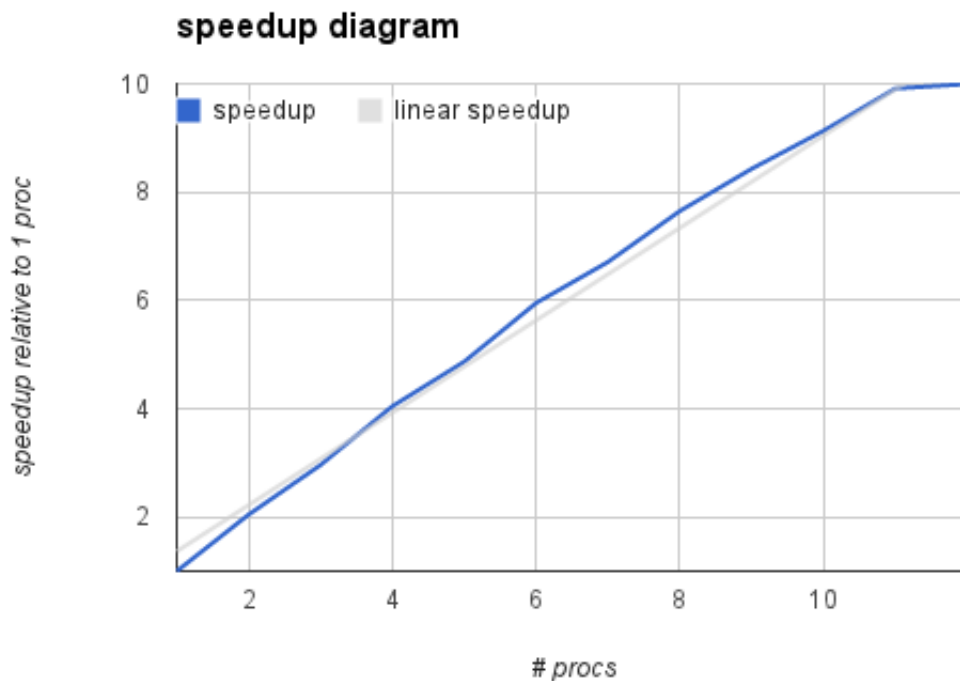
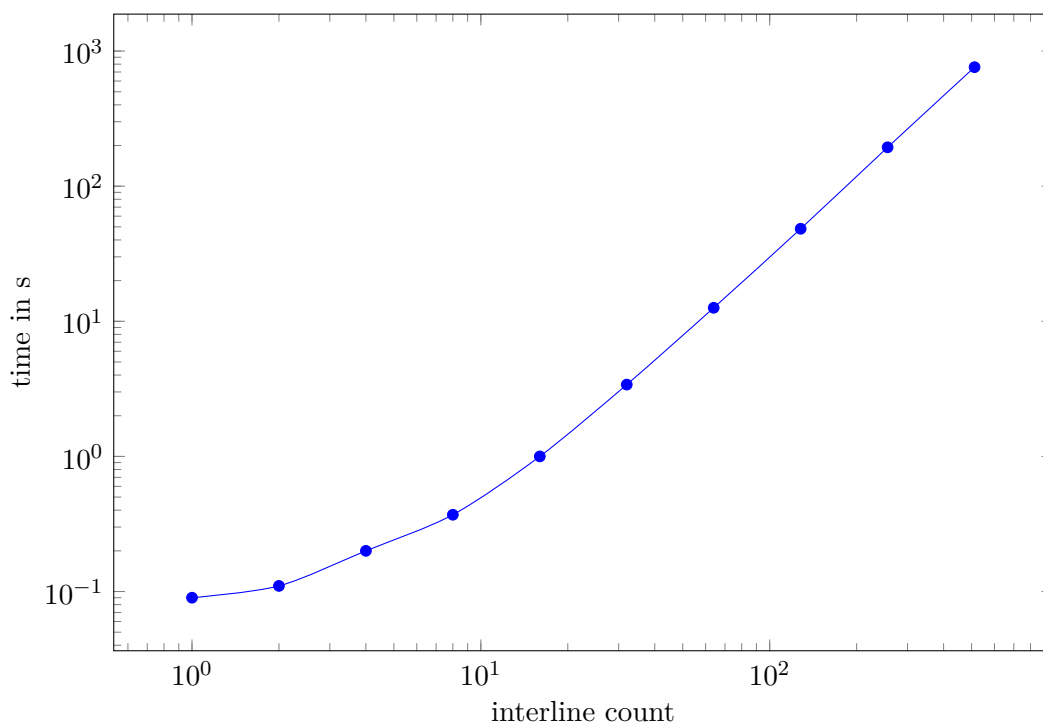


Abbildung 1:

Der Speedup des Programmes ist zu Anfang Linear (scheinbare Superlinearität ist auf messungs-genauigkeiten zurückzuführen). Dies deckt sich mit dem Amdahlschen Gesetz. Das Resultat deckt sich also mit den Erwartungen an ein gut zu parallelisierendes Programm. Um weitere Aussagen treffen zu können, müsste die Thread-Anzahl drastisch erhöht werden. Leider stößt Open-MP hier an seine Grenzen, zu mindest in Zusammenspiel mit der uns zur Verfügung stehenden Architektur.

2.2 Messung 2



Verwendete Werte:

interlines	time
1	0.09
2	0.11
4	0.2
8	0.37
16	1
32	3.4
64	12.59
128	48.4
256	194
512	760

Die benötigte Zeit nimmt mit der Erhöhung der Interlinegröße zu. Diese Zunahme sollte aufgrund des Aufbaus der Matrix quadratisch-exponentiell zunehmen, da beide Dimensionen mit den Interlines um die selbe Größe wachsen. Da allerdings die Berechnung auf 12 threads aufgeteilt werden, ist nur ein Zwölftel der Zeit des rein sequentiellen Ansatzes theoretisch nötig. Diese Vermutung wird durch unsere Tests nur partiell getützt. Zu Anfang folgt die zeitliche Zunahme ungefähr diesem Muster, bei dem Übergang von 256 auf 512 Interlines nimmt die benötigte Zeit leider über die Maßen zu. Dies liegt vermutlich sowohl daran, dass die Parallelisierung nicht perfekt ist, als auch an der Tatsache dass auch Open-MP keine perfekten Parallelisierungen schafft.