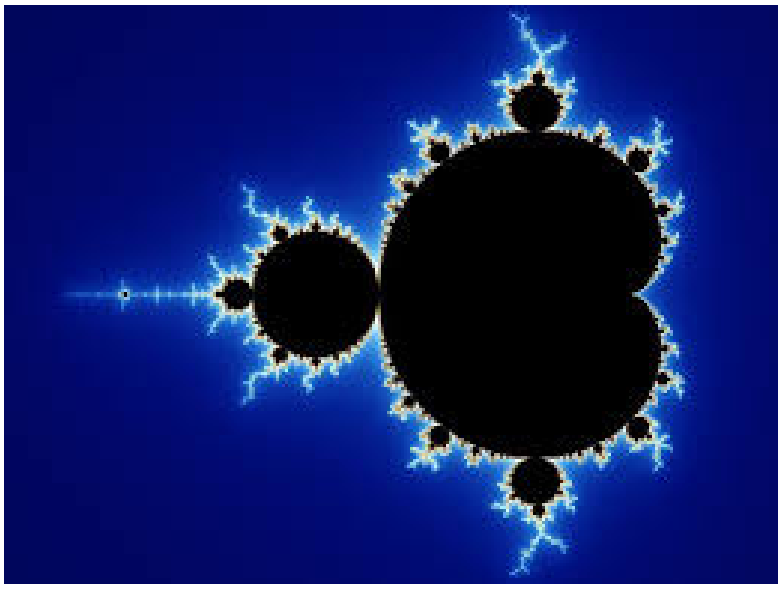
PARALLELISM LABORATORY 5

David Morais
david.morais

Laura van Dinteren
laura.van.dinteren



	Number of threads (elapsed in seconds)					
Version	1	4	8	12	16	20
1D Block Geometric Data Decomposition by columns	2.86	1.80	1.25	0.91	0.73	0.61
1D Block-Cyclic Geometric Data Decom-position by columns	2.858	0.7212	0.338	0.2691	0.2095	0.16
1D Cyclic Geometric Data Decomposition by rows	2.85	0.7620	0.3837	0.2644	0.1994	0.160208
	Number of threads (L2 Cache Misses per thread)					
1D Block Geometric Data Decomposition by columns	184454	93139	93225	93064	93407	93284
1D Block-Cyclic Geometric Data Decom-position by columns	1642609	1358022	206380	138037	103568	83112
1D Cyclic Geometric Data Decomposition by rows	1642385	412451	206881	138530	103856	83762

1D Block Geometric Data Decomposition by columns

Modelfactor tables

Overview of whole program execution metrics											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Elapsed time (sec)	2.88	2.09	1.83	1.58	1.28	1.10	0.94	0.86	0.76	0.71	0.64
Speedup	1.00	1.38	1.58	1.82	2.25	2.62	3.08	3.34	3.77	4.04	4.49
Efficiency	1.00	0.69	0.39	0.30	0.28	0.26	0.26	0.24	0.24	0.22	0.22

Table 1: Analysis done on Wed Dec 11 04:45:46 PM CET 2024, par4115

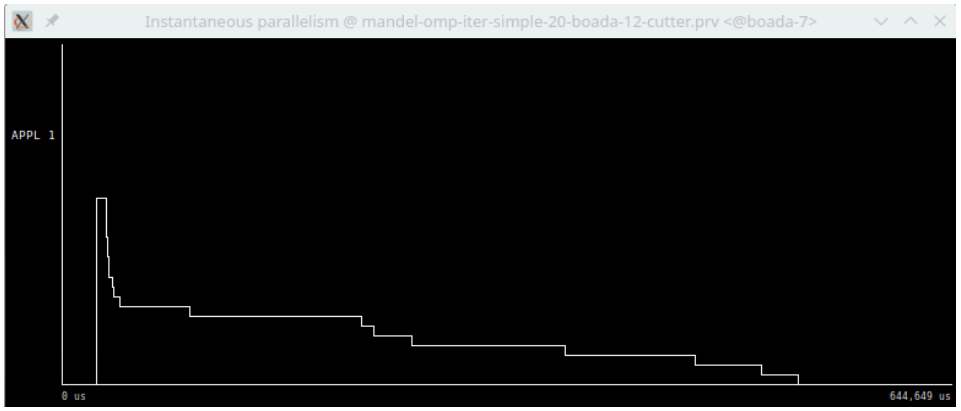
Overview of the Efficiency metrics in parallel fraction, ϕ=99.10%											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Global efficiency	100.00%	69.08%	39.60%	30.60%	28.50%	26.55%	26.15%	24.42%	24.16%	23.05%	23.19%
Parallelization strategy efficiency	100.00%	69.16%	39.71%	31.31%	29.76%	28.22%	27.96%	26.46%	26.43%	25.43%	25.72%
Load balancing	100.00%	69.16%	39.72%	31.32%	29.78%	28.25%	28.00%	26.50%	26.48%	25.49%	25.80%
In execution efficiency	100.00%	99.99%	99.96%	99.95%	99.93%	99.90%	99.85%	99.84%	99.81%	99.77%	99.70%
Scalability for computation tasks	100.00%	99.88%	99.73%	97.72%	95.76%	94.06%	93.53%	92.28%	91.39%	90.64%	90.15%
IPC scalability	100.00%	100.01%	100.00%	100.04%	100.04%	100.05%	100.06%	99.91%	99.90%	99.77%	99.73%
Instruction scalability	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.99%	99.99%	99.99%	99.99%	99.99%
Frequency scalability	100.00%	99.87%	99.73%	97.69%	95.73%	94.02%	93.47%	92.37%	91.49%	90.85%	90.41%

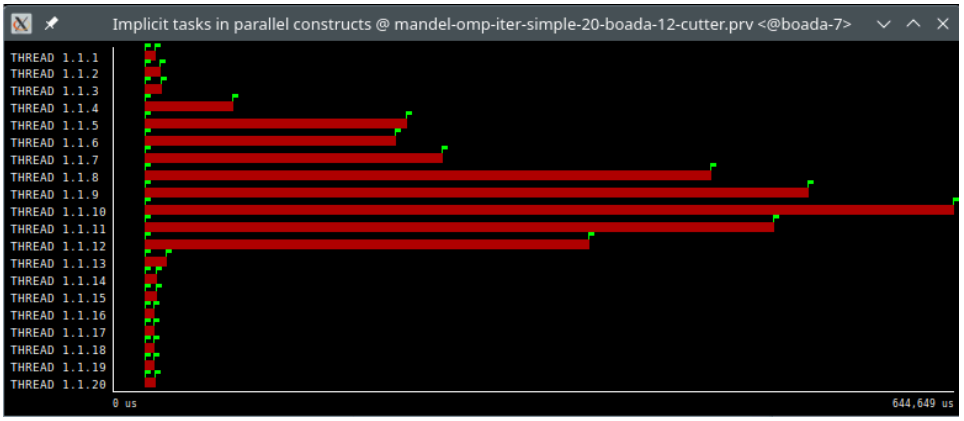
Table 2: Analysis done on Wed Dec 11 04:45:46 PM CET 2024, par4115

Statistics about explicit tasks in parallel fraction											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Number of implicit tasks per thread (average us)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Useful duration for implicit tasks (average us)	2858756.12	1431055.85	716647.01	487559.46	373151.23	303918.19	254713.15	221275.21	195496.16	175221.67	158550.03
Load balancing for implicit tasks	1.0	0.69	0.4	0.31	0.3	0.28	0.28	0.27	0.26	0.25	0.26
Time in synchronization implicit tasks (average us)	0	0	0	0	0	0	0	0	0	0	0
Time in fork/join implicit tasks (average us)	31.7	0	0	0	0	0	0	0	0	0	0

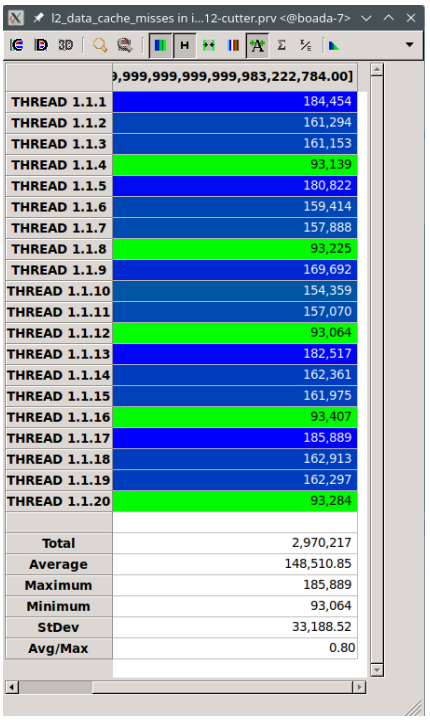
Table 3: Analysis done on Wed Dec 11 04:45:46 PM CET 2024, par4115

Paraver analysis



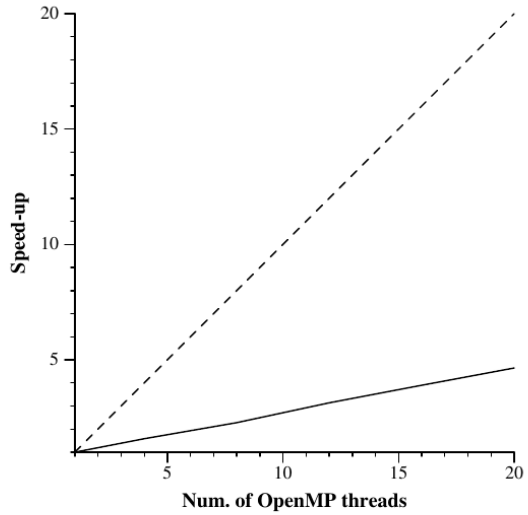


Memory analysis



	9,999,999,999,999,983,222,784.00]
THREAD 1.1.1	184,454
THREAD 1.1.2	161,294
THREAD 1.1.3	161,153
THREAD 1.1.4	93,139
THREAD 1.1.5	180,822
THREAD 1.1.6	159,414
THREAD 1.1.7	157,888
THREAD 1.1.8	93,225
THREAD 1.1.9	169,692
THREAD 1.1.10	154,359
THREAD 1.1.11	157,070
THREAD 1.1.12	93,064
THREAD 1.1.13	182,517
THREAD 1.1.14	162,361
THREAD 1.1.15	161,975
THREAD 1.1.16	93,407
THREAD 1.1.17	185,889
THREAD 1.1.18	162,913
THREAD 1.1.19	162,297
THREAD 1.1.20	93,284
Total	2,970,217
Average	148,510.85
Maximum	185,889
Minimum	93,064
StDev	33,188.52
Avg/Max	0.80

Strong scalability



par4115
Speed-up wrt sequential time (mandel funtion only)
Generated by par4115 on Wed Dec 11 04:49:07 PM CET 2024

Analysis

If we take a look at the modelfactor tables we can quickly see that the block geometric data decomposition is not a good strategy for 20 threads, the speed-up is not even 5, the efficiency is 22%, and the load balancing is really poor (26%). The paraver graphics back this theory, the load balancing is awful, as a result the instantaneous parallelism decreases rapidly as the execution goes. It is also noticeable that the graphic matches the Mandelbrot computing areas. The areas that are white require more computation and the threads that execute said areas have more work to do, this explains the load unbalance. About the memory analysis, threads with lower misses might be benefiting from better spatial or temporal locality of data, whereas threads with high misses may be accessing data in less cache-friendly patterns. Finally, for the strong scalability graphic, we can see that it is lineal and it is approximately 5 for 20 threads, which is really poor.

1D Block-Cyclic Geometric Data Decomposition by columns

Modelfactor tables

Overview of whole program execution metrics											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Elapsed time (sec)	2.88	1.46	0.75	0.54	0.42	0.35	0.30	0.26	0.24	0.21	0.20
Speedup	1.00	1.97	3.86	5.33	6.82	8.33	9.73	11.25	12.13	13.64	14.70
Efficiency	1.00	0.99	0.97	0.89	0.85	0.83	0.81	0.80	0.76	0.76	0.74

Table 1: Analysis done on Wed Dec 18 04:24:29 PM CET 2024, par4115

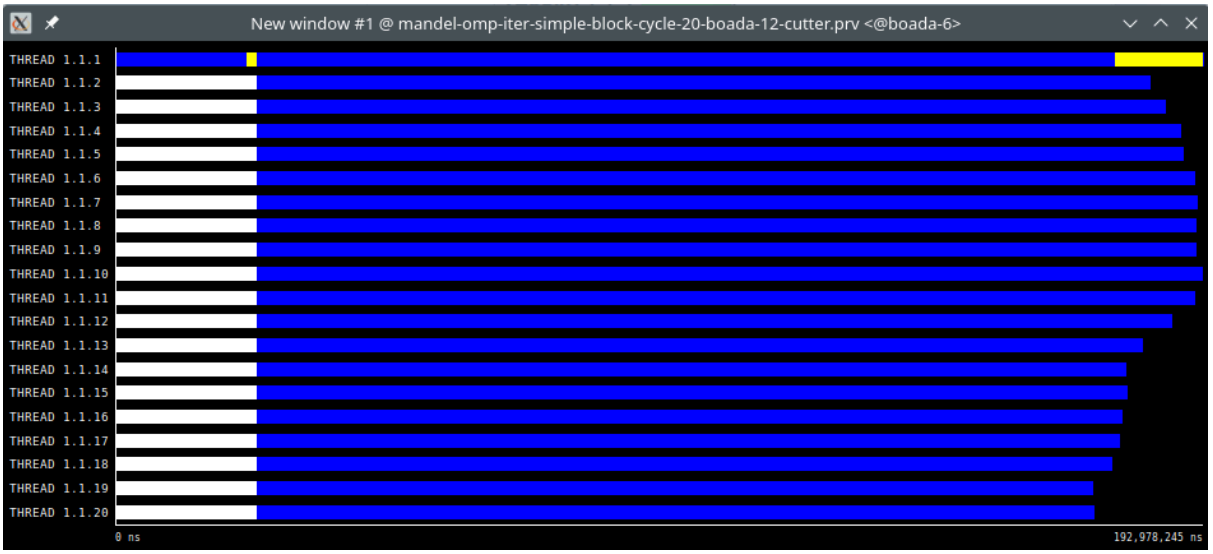
Overview of the Efficiency metrics in parallel fraction, ϕ=99.14%											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Global efficiency	100.00%	99.60%	98.88%	92.46%	89.74%	89.35%	88.11%	88.10%	84.49%	85.48%	83.97%
Parallelization strategy efficiency	100.00%	99.92%	99.50%	98.97%	97.44%	99.06%	97.78%	97.90%	94.26%	95.56%	93.94%
Load balancing	100.00%	99.94%	99.57%	99.10%	97.64%	99.37%	98.24%	98.38%	94.89%	96.48%	94.87%
In execution efficiency	100.00%	99.98%	99.93%	99.87%	99.79%	99.69%	99.54%	99.51%	99.33%	99.04%	99.03%
Scalability for computation tasks	100.00%	99.68%	99.37%	93.42%	92.10%	90.20%	90.11%	89.99%	89.64%	89.45%	89.38%
IPC scalability	100.00%	99.90%	99.78%	99.82%	99.78%	99.81%	99.85%	99.81%	99.72%	99.71%	99.67%
Instruction scalability	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.99%	99.99%	99.99%	99.99%	99.99%
Frequency scalability	100.00%	99.77%	99.60%	93.60%	92.30%	90.38%	90.25%	90.17%	89.90%	89.72%	89.69%

Table 2: Analysis done on Wed Dec 18 04:24:29 PM CET 2024, par4115

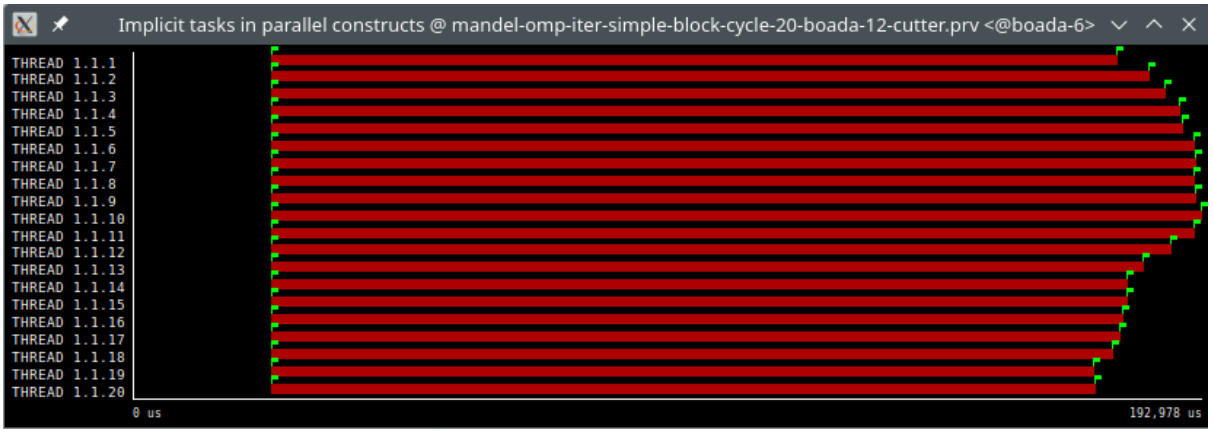
Statistics about explicit tasks in parallel fraction											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Number of implicit tasks per thread (average us)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Useful duration for implicit tasks (average us)	2859609.21	1434408.88	719402.17	510166.47	388114.21	317018.05	264469.51	226983.02	199384.16	177600.92	159964.54
Load balancing for implicit tasks	1.0	1.0	1.0	0.99	0.98	0.99	0.98	0.98	0.95	0.96	0.95
Time in synchronization implicit tasks (average us)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Time in fork/join implicit tasks (average us)	25.06	273.16	2956.71	3110.13	6235.34	1975.26	6960.78	9463.25	22273.89	6446.69	17526.13

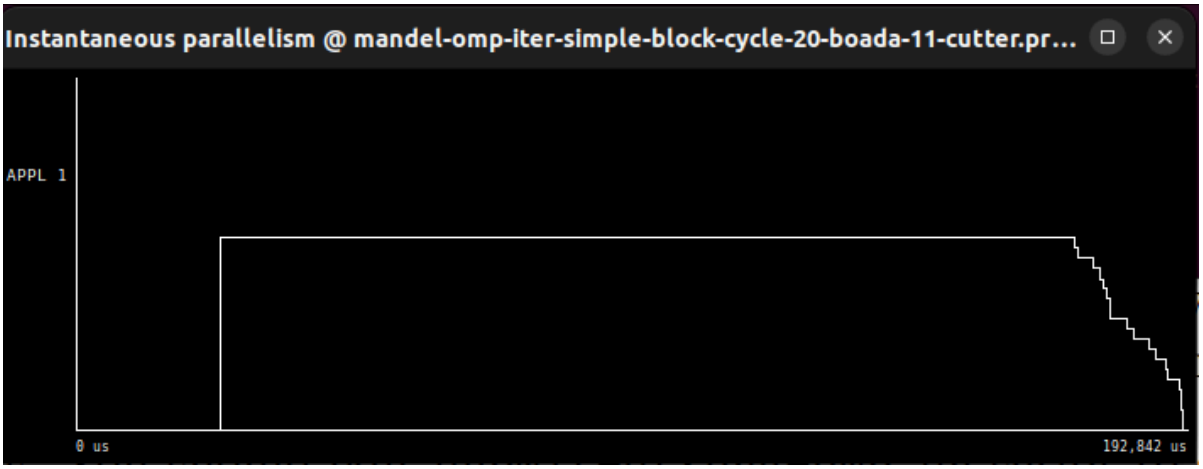
Table 3: Analysis done on Wed Dec 18 04:24:29 PM CET 2024, par4115

Paraver analysis

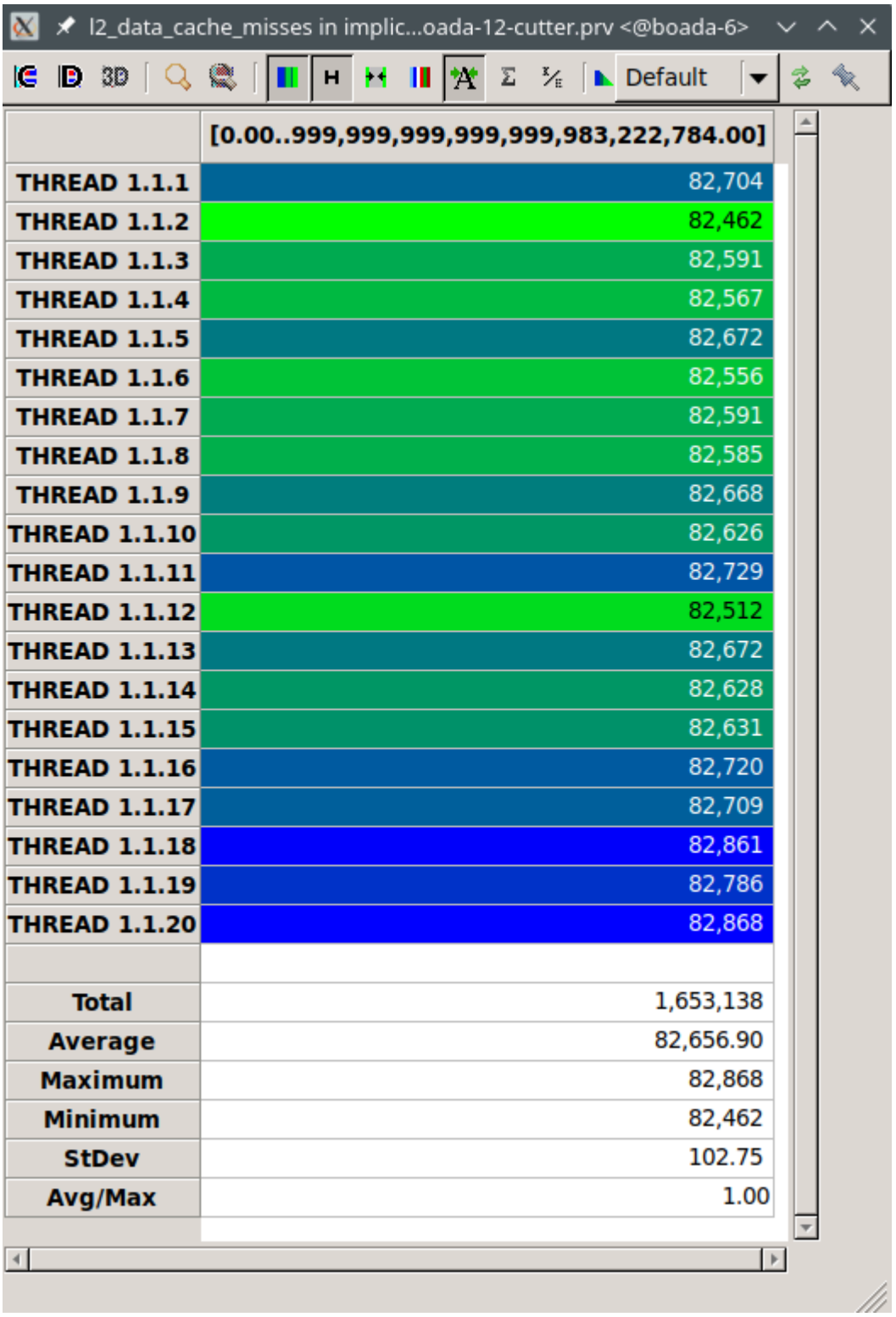


Blau ejecuta, blanc no s'ha creat, groc wait



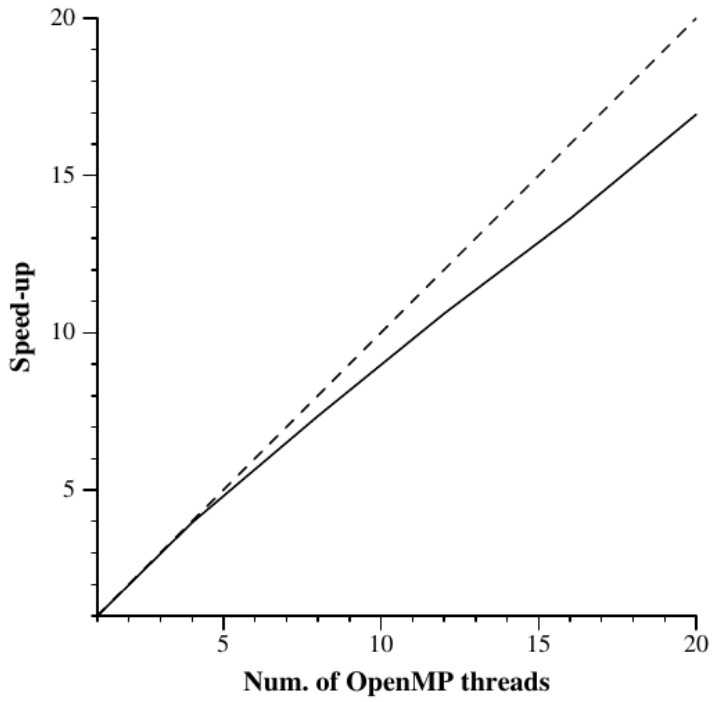


Memory analysis



	[0.00..999,999,999,999,999,983,222,784.00]
THREAD 1.1.1	82,704
THREAD 1.1.2	82,462
THREAD 1.1.3	82,591
THREAD 1.1.4	82,567
THREAD 1.1.5	82,672
THREAD 1.1.6	82,556
THREAD 1.1.7	82,591
THREAD 1.1.8	82,585
THREAD 1.1.9	82,668
THREAD 1.1.10	82,626
THREAD 1.1.11	82,729
THREAD 1.1.12	82,512
THREAD 1.1.13	82,672
THREAD 1.1.14	82,628
THREAD 1.1.15	82,631
THREAD 1.1.16	82,720
THREAD 1.1.17	82,709
THREAD 1.1.18	82,861
THREAD 1.1.19	82,786
THREAD 1.1.20	82,868
Total	1,653,138
Average	82,656.90
Maximum	82,868
Minimum	82,462
StDev	102.75
Avg/Max	1.00

Strong scalability



par4115
Speed-up wrt sequential time (mandel funtion only)
Generated by par4115 on Wed Dec 18 04:16:38 PM CET 2024

Analysis

For this strategy, we can see that the modelfactor tables show much better numbers than the previous strategy; for 20 threads the speed-up is almost 15, the efficiency is almost 75% and the load balance almost reaches 95%. In the paraver graphics we can see how the load balance improves a lot, being almost a 100% parallelizable. The little fork & join time is explained by the few unbalances. About the misses in the L2 cache, all the threads have more or less the same amount of misses, being the maximum 82868 misses and the minimum 82462 ($\approx$ 400 difference). The block size chosen was 16, because the size of the cache line is 64 and each element occupies 4 bytes, so 64/4 = 16. About the strong scalability graphic, we can say that it is pretty good, although a bit could be improved.

1D Cyclic Geometric Data Decomposition by rows

Modelfactor tables

Overview of whole program execution metrics											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Elapsed time (sec)	2.88	1.46	0.75	0.54	0.42	0.35	0.30	0.26	0.24	0.21	0.20
Speedup	1.00	1.98	3.86	5.34	6.92	8.23	9.70	10.95	12.01	13.45	14.48
Efficiency	1.00	0.99	0.96	0.89	0.87	0.82	0.81	0.78	0.75	0.75	0.72

Table 1: Analysis done on Wed Dec 18 04:36:58 PM CET 2024, par4115

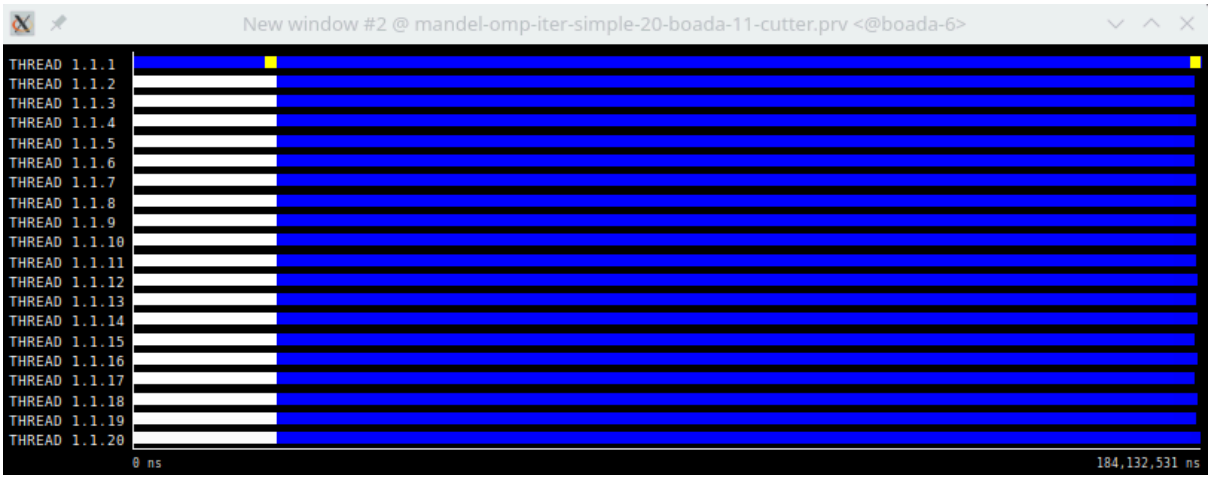
Overview of the Efficiency metrics in parallel fraction, ϕ=99.14%											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Global efficiency	100.00%	99.77%	99.09%	92.76%	91.50%	87.48%	86.91%	85.68%	83.55%	83.28%	81.50%
Parallelization strategy efficiency	100.00%	99.96%	99.82%	99.72%	99.29%	99.10%	99.05%	98.65%	97.27%	98.35%	97.90%
Load balancing	100.00%	100.00%	99.88%	99.85%	99.47%	99.41%	99.43%	99.25%	97.97%	99.11%	98.86%
In execution efficiency	100.00%	99.96%	99.93%	99.87%	99.82%	99.69%	99.62%	99.40%	99.28%	99.24%	99.03%
Scalability for computation tasks	100.00%	99.81%	99.27%	93.02%	92.15%	88.28%	87.75%	86.84%	85.89%	84.68%	83.25%
IPC scalability	100.00%	99.87%	99.68%	99.41%	98.64%	97.68%	97.19%	96.35%	95.49%	94.19%	92.94%
Instruction scalability	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.99%	99.99%	99.99%	99.99%	99.99%
Frequency scalability	100.00%	99.94%	99.59%	93.58%	93.42%	90.38%	90.29%	90.14%	89.96%	89.91%	89.59%

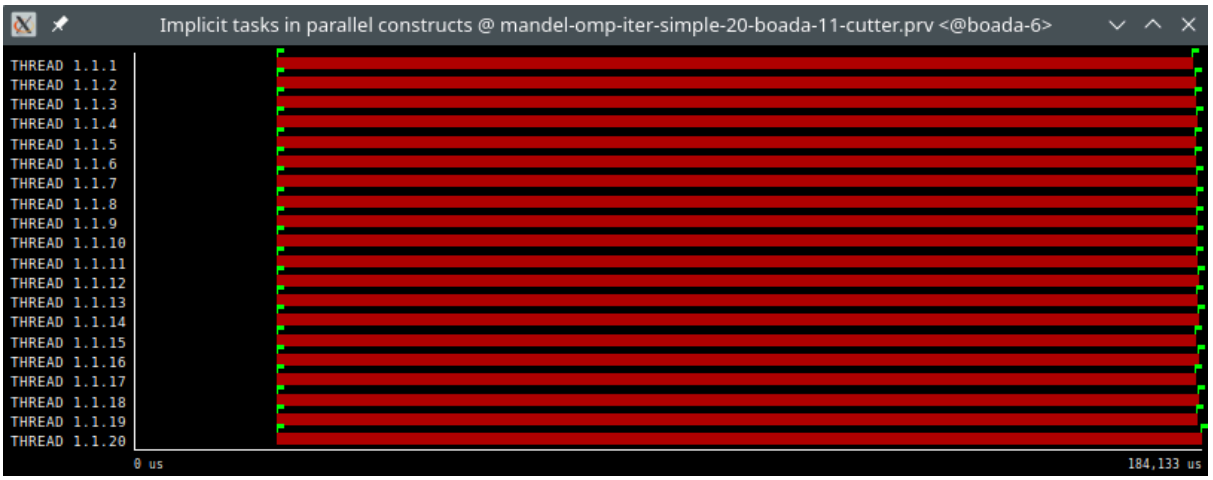
Table 2: Analysis done on Wed Dec 18 04:36:58 PM CET 2024, par4115

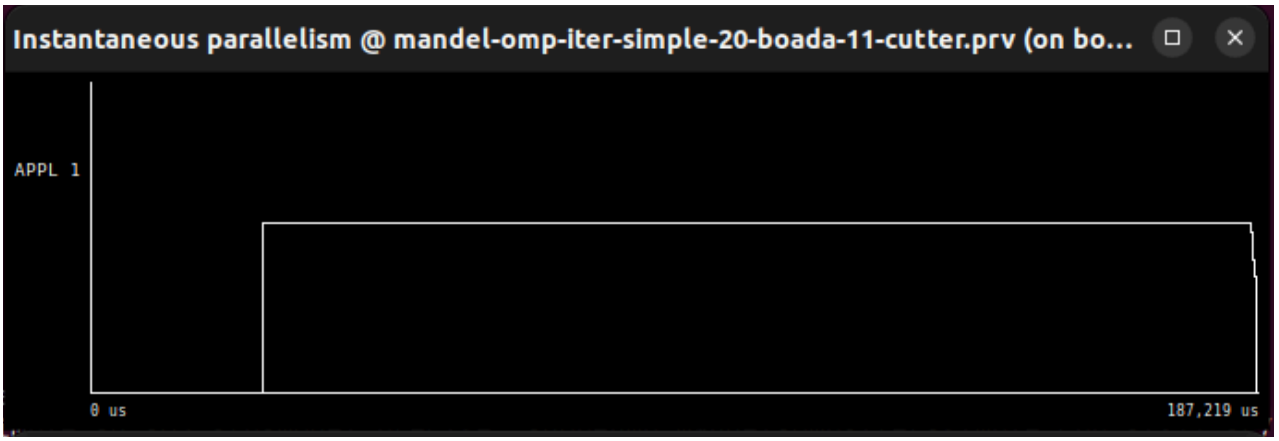
Statistics about explicit tasks in parallel fraction											
Number of processors	1	2	4	6	8	10	12	14	16	18	20
Number of implicit tasks per thread (average us)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Useful duration for implicit tasks (average us)	2858507.23	1431964.62	719892.19	512152.4	387755.05	323815.48	271467.67	235111.83	207996.45	187544.46	171683.58
Load balancing for implicit tasks	1.0	1.0	1.0	1.0	0.99	0.99	0.99	0.99	0.98	0.99	0.99
Time in synchronization implicit tasks (average us)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Time in fork/join implicit tasks (average us)	117.16	615.67	1723.37	2136.6	6551.52	8107.53	8298.9	8931.88	10503.7	10842.22	12828.34

Table 3: Analysis done on Wed Dec 18 04:36:58 PM CET 2024, par4115

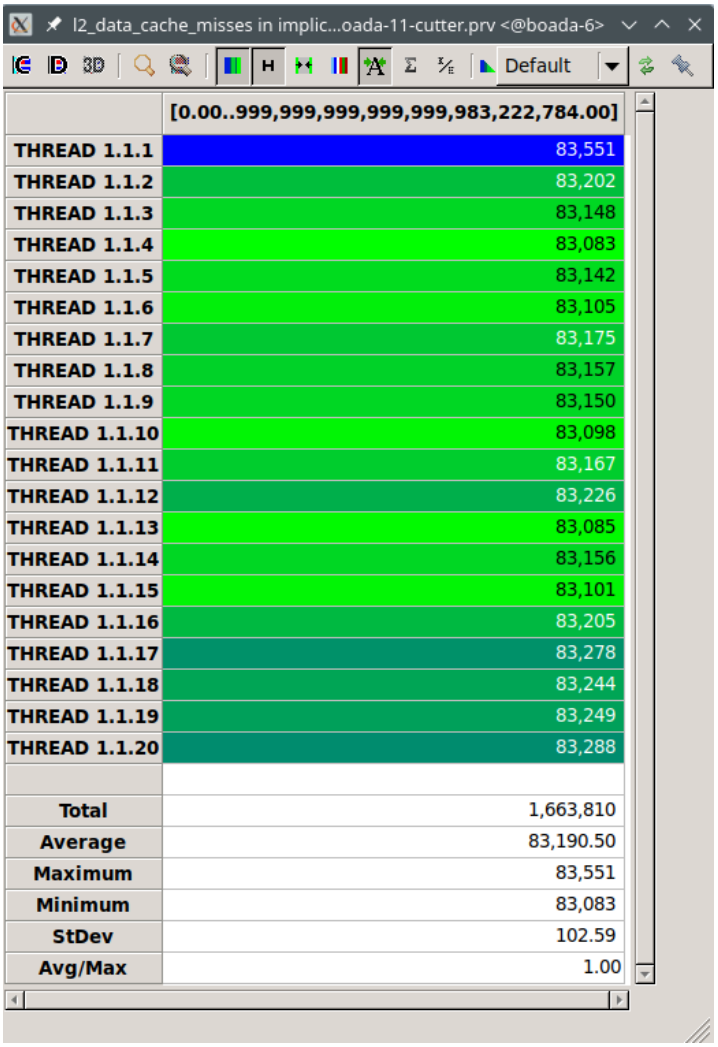
Paraver analysis





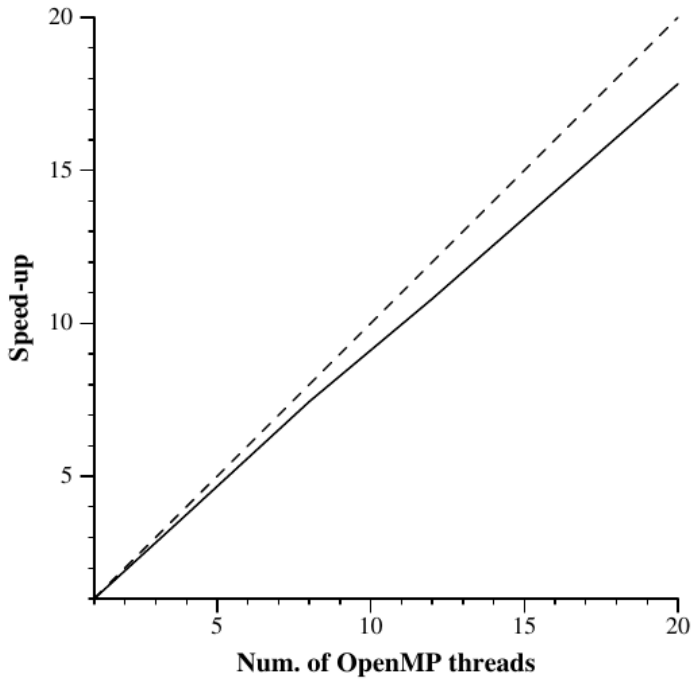


Memory analysis



	[0.00..999,999,999,999,999,983,222,784.00]
THREAD 1.1.1	83,551
THREAD 1.1.2	83,202
THREAD 1.1.3	83,148
THREAD 1.1.4	83,083
THREAD 1.1.5	83,142
THREAD 1.1.6	83,105
THREAD 1.1.7	83,175
THREAD 1.1.8	83,157
THREAD 1.1.9	83,150
THREAD 1.1.10	83,098
THREAD 1.1.11	83,167
THREAD 1.1.12	83,226
THREAD 1.1.13	83,085
THREAD 1.1.14	83,156
THREAD 1.1.15	83,101
THREAD 1.1.16	83,205
THREAD 1.1.17	83,278
THREAD 1.1.18	83,244
THREAD 1.1.19	83,249
THREAD 1.1.20	83,288
Total	1,663,810
Average	83,190.50
Maximum	83,551
Minimum	83,083
StDev	102.59
Avg/Max	1.00

Strong scalability



par4115
Speed-up wrt sequential time (mandel funtion only)
Generated by par4115 on Wed Dec 18 04:44:00 PM CET 2024

Analysis

The cyclic geometric data decomposition by rows is the best strategy, although it is pretty similar to the previous one. Looking at the modelfactor tables, we can see that the speed-up is almost 14.5, the efficiency is also good being 82% and the load balancing is almost 99% which is also shown in the paraver graphics where we can see that it has a perfect parallelization, the load balance is perfect and it is also reflected in the fork & join time, which is really low. The misses in the L2 cache differ just a tiny bit from each other and the strong scalability graphic backs it all, reaching a speed-up of almost 18 for 20 threads.