MZT

Laboratorium 3

Przemysław Kleszcz

Nr albumu: 124624

Platforma testowa

Procesor	Intel Core i5-7440HQ, 2.8GHz, 6MB Cache
RAM	DDR4 - 16 GB
System operacyjny	Microsoft Windows 10 Pro
Środowisko programistyczne	Visual Studio Professional 2017 v15.5.7
Środowisko uruchomieniowe	.NET Framework 4.6.01055

Metoda klasyczna (i,j,k).(n=2000)

	Próba_1	Próba_2	Próba_3	**Średnia_Duration**
method	duration [s]	duration [s]	duration [s]	
classic	37,640	3,52030E+01	35,125	**35,989**

Próba_1	Próba_2	Próba_3	**Średnia_PERF**	**l_block**	**n**
performance [Mflops]	performance [Mflops]	performance [Mflops]			
385,564	412,255	413,171	**403,6633333**		1936

Metoda bez skoków przy pobieraniu danych z macierzy B(i,k,j).(n=2000)

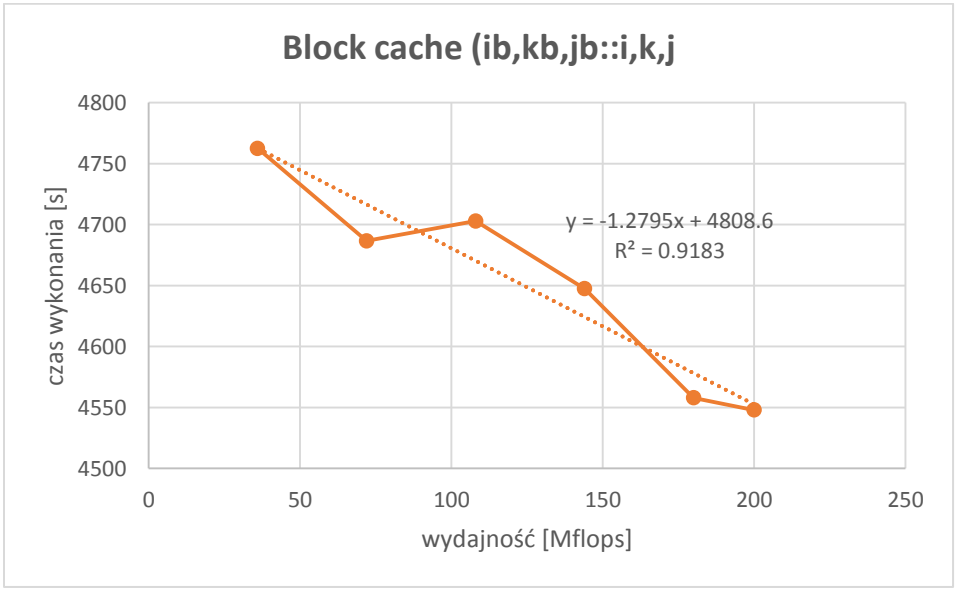
	Próba_1	Próba_2	Próba_3	**Średnia_Duration**
method	duration [s]	duration [s]	duration [s]	
HP	5,25	4,907	4,89	**5,016**

Próba_1	Próba_2	Próba_3	**Średnia_PERF**	**l_block**	**n**
performance [Mflops]	performance [Mflops]	performance [Mflops]			
2764,31	2957,54	2967,82	**2896,556667**		1936

Metoda Block Cache, podział każdej z macierzy na kwadratowe bloki o rozmiarze l_block. Kolejność indeksów dla bloków jest ib, jb, kb; indeksów w bloku: i,j,k.(dla n=2000).

	Próba_1	Próba_2	Próba_3	**Średnia_Duration**
method	duration [s]	duration [s]	duration [s]	
block_cache (ib,kb,jb::i,k,j)	3,312	3,25000E+00	3,219	**3,260**
	3,125	3,188	3,094	**3,136**
	3,157	3,172	3,047	**3,125**
	2,829	2,797	2,844	**2,823**
	3,375	3,438	3,406	**3,406**
	3,391	3,422	3,765	**3,526**

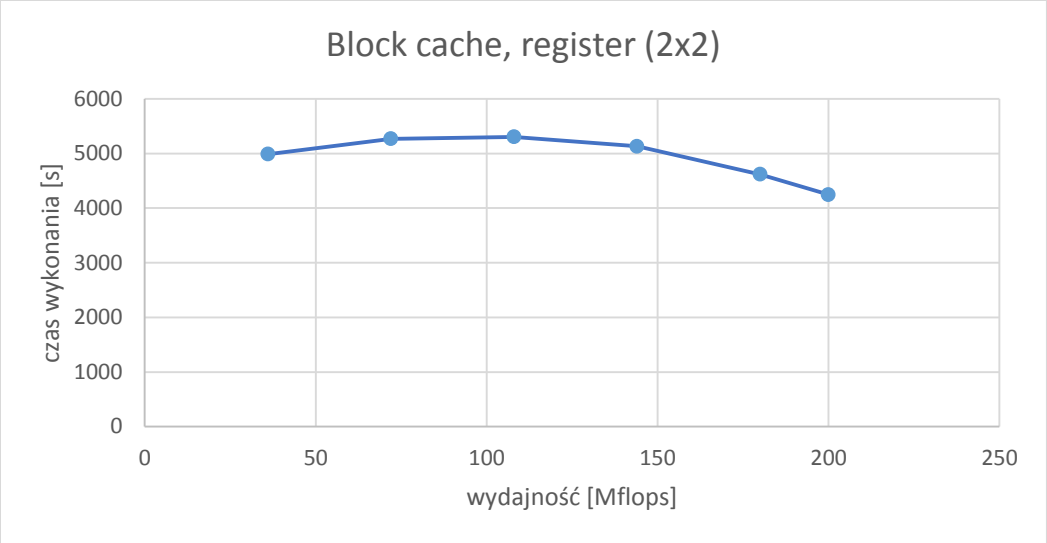
Próba_1 performance [Mflops]	Próba_2 performance [Mflops]	Próba_3 performance [Mflops]	**Średnia_PERF**	**l_block**	**n**
4687,43	4776,86	4822,86	**4762,383333**	36	1980
4701,85	4608,93	4748,96	**4686,58**	72	1944
4654,19	4632,18	4822,21	**4702,86**	108	1944
4637,83	4690,89	4613,37	**4647,363333**	144	1872
4599,84	4515,64	4558,07	**4557,85**	180	1980
4718,37	4675,63	4249,67	**4547,89**	200	2000



Metoda Mlock cache, register (2x2), pack (BLAS 1989). Zastosowane blokowanie na poziomie RAM – cache (podział każdej macierzy na bloki o rozmiarze l_block x l_block), blokowanie rejestrów 2x2, pakowanie bloku macierzy A tak, żeby uniknąć skoków w danych przy pobieraniu elementów tej macierzy. Kolejność indeksów: kb, ib, jb :: j,i,k.(dla n=2000)

	Próba_1	Próba_2	Próba_3	**Średnia_Duration**
method	duration [s	duration [s]	duration [s]	
block_cache, registers (2x2), pack (BLAS 1989)	3,047	3,06300E+00	3,235	**3,115**
	2,719	2,937	2,719	**2,792**
	2,828	2,688	2,797	**2,771**
	2,703	2,562	2,421	**2,562**
	3,516	3,25	3,328	**3,365**
	3,89	3,766	3,656	**3,771**

Próba_1 performance [Mflops]	Próba_2 performance [Mflops]	Próba_3 performance [Mflops]	**Średnia_PERF**	**l_block**	**n**
5095,1	5068,49	4799,01	**4987,533333**	36	1980
5403,93	5002,82	5403,93	**5270,226667**	72	1944
5195,64	5466,25	5253,23	**5305,04**	108	1944
4854,2	5121,16	5419,42	**5131,593333**	144	1872
4415,47	4776,86	4664,9	**4619,076667**	180	1980
4113,11	4248,54	4376,37	**4246,006667**	200	2000



Metoda Block cache, block XMM registers (2x4x8), SSE2, SSE3, pack A,B(similar to Intel MKL library).

	Próba_1	Próba_2	Próba_3	dnia_Dura
method	duration [s]	duration [s]	duration [s]	
block_cache, (kb, jb=1, ib), block registers, vectorization AVX, pack A, B(like Intel MKL)	0,687	6,88000E-01	0,672	**0,682**
	0,468	0,485	0,453	**0,469**
	0,438	0,454	0,468	**0,453**
	0,359	0,375	0,375	**0,370**
	0,422	0,406	0,437	**0,422**
	0,406	0,406	0,406	**0,406**
	0,328	0,328	0,312	**0,323**
	0,313	0,313	0,328	**0,318**

Próba_1 performance [Mflops]	Próba_2 performance [Mflops]	Próba_3 performance [Mflops]	**Średnia_PERF**	**l_block**	**n**
22597,9	22565,1	23102,4	**22755,13333**	36	1980
31295,9	20295,4	32435,5	**28008,93333**	72	1944
33546,3	32364,1	31395,9	**32435,43333**	108	1944
36547,1	34987,8	34987,8	**35507,56667**	144	1872
36788,6	38238,4	35525,8	**36850,93333**	180	1980
36190,3	36190,3	36190,3	**36190,3**	216	1944
33469,7	33469,7	35186,1	**34041,83333**	252	1764
32969,8	32969,8	31462,1	**32467,23333**	288	1728