МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

ПРАКТИКУМ ПО КУРСУ СКиПОД

Разработка параллельной версии программы для перемножения матриц с использованием ленточного алгоритма

ОТЧЕТ

о выполненном задании

студента 441 учебной группы факультета ВМК МГУ

Кемаева Юрия Юрьевича

Москва, 2016 г.

Постановка задачи

Необходимо разработать параллельную версию программы для перемножения матриц с использованием ленточного алгоритма на MPI и OpenMP, измерить характеристики работы в зависимости от количества процессов, а также исследовать масштабируемость алгоритма.

Описание алгоритма

Пусть имеются квадратные $N \times N$ матрицы A и B, которые требуется перемножить, используя p процессов.

Последовательная версия (p=1)

Напрямую перемножаются матрицы A и B по правилу $C_{ij} = \sum_{k=1}^{N} \quad A_{ik} * B_{kj}$.

OMP-версия

Процесс под номером i рассчитывает элеметы итоговой матрицы, соответствующие перемножению $\lceil \frac{N}{p} \rceil$ строк A на матрицу B.

MPI-версия

Матрица A разбивается на p блоков A_i по строкам размером $\lceil \frac{N}{p} \rceil \times N$, матрица B – на p блоков B_i по столбцам размером $N \times \lceil \frac{N}{P} \rceil$ соответственно.

Процесс под номером i рассчитывает элеметы итоговой матрицы, соответствующие перемножению блока A_i и матрицы B.

Для этого на первом шаге процесс i получает блоки A_i и B_i, перемножает их, записывая результат в свой буфер. Затем отправляет B_i процессу $i-1$ и получает B_{i+1} от процесса $i+1$.

Описанная процедура повторяется p раз, в результате в процессе i оказывается результат перемножения A_i на B.

Далее буферы всех процессов объединяются.

Оптимизации и примечания по коду

1. Матрица B транспонируется в начале перемножения, и транспонируется в конце еще раз. То есть в итоге она не меняется, а реализация метода перемножения становится проще и эффективнее (из-зе непрерывных в памяти блоков B_i).
2. Если внутри цикла по переменной i необходимо много раз обращаться к элементу A[i*N + j], то адрес A[i*N] заносится в переменную opt_A, и обращение идет к opt_A[j]. Данная оптимизация ускоряет программу в 3-4 раза.
3. Компиляция проходит с флагом –O3.
4. В последовательной и OMP версиях программы время замеряется командой omp_get_wtime(), в MPI-версии MPI_Wtime(). Измеряется непосредственно время работы функции mult_square_matrices() (память под матрицы выделена ранее).
5. Основные этапы работы алгоритма прокомментированы в коде.
6. Корректность OMP и MPI версий проверялась сравнением с последовательной версией.

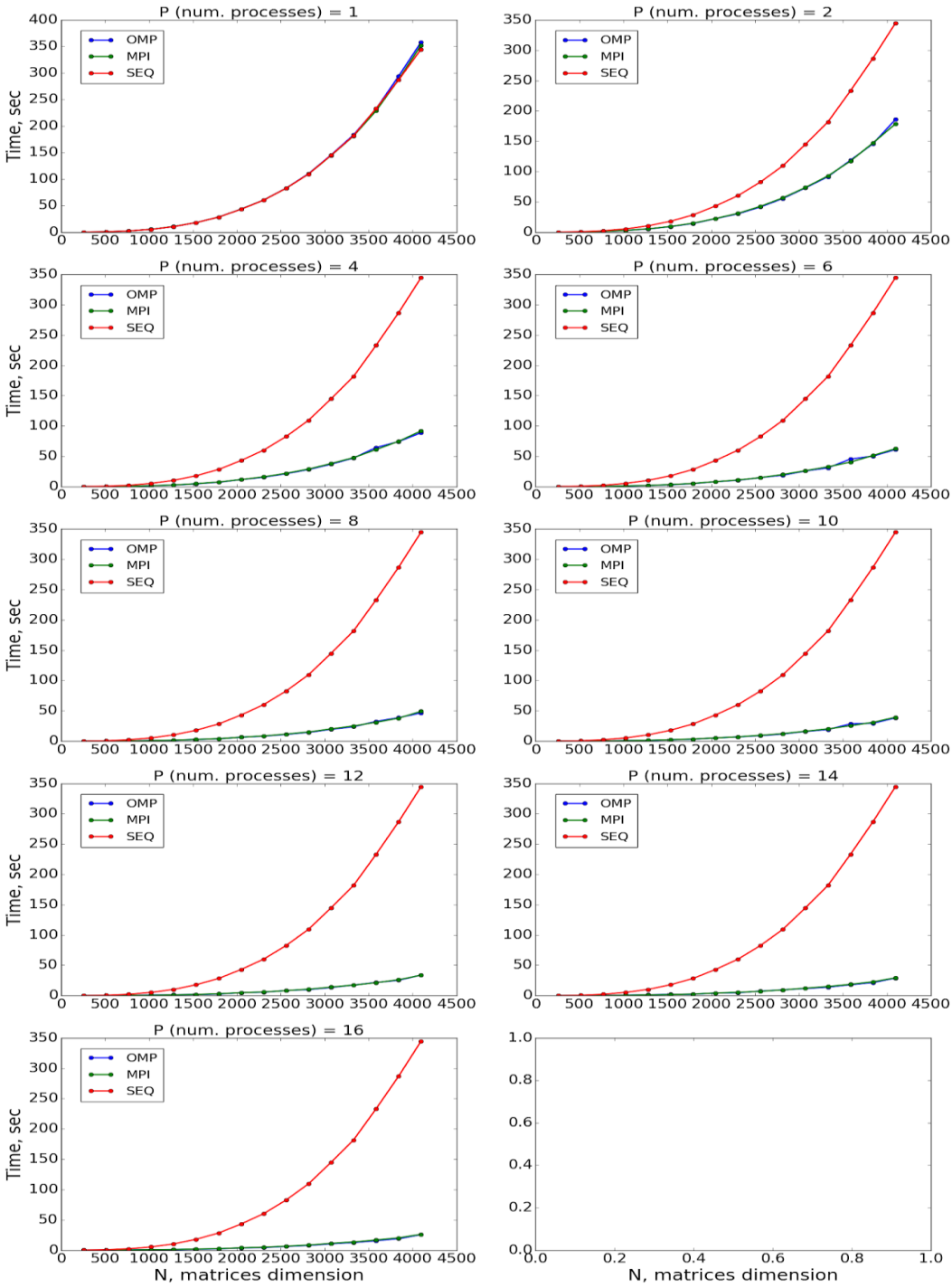
Результаты

Для каждой конфигурации берется среднее время по 5-ти запускам.

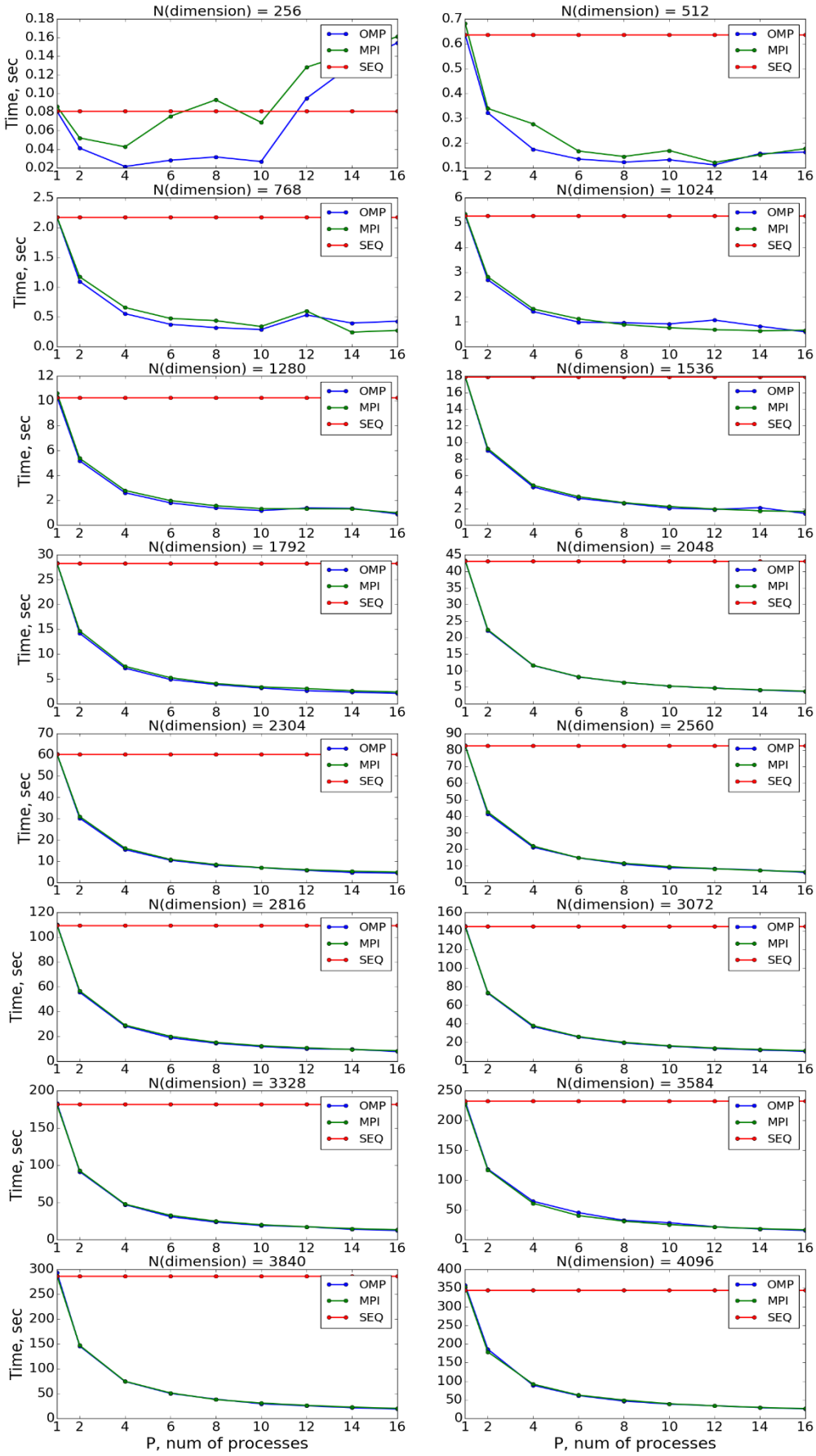
Тип элементов A и B – $float$. N **принимает величины: от 256 (0.25 Мб) до 4096 (64 Мб)** элементов с шагом 256. Отдельно были проведены эксперименты на BlueGene и Regatta.

Regatta

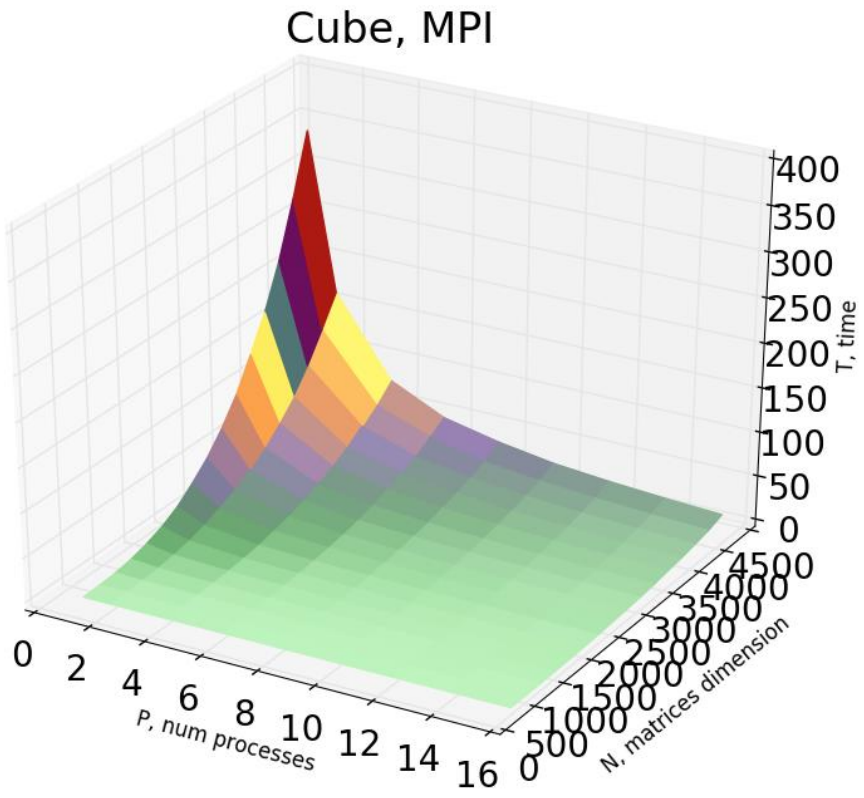
Кол-во процессов: 1, 2, 4, 6 ... 14, 16



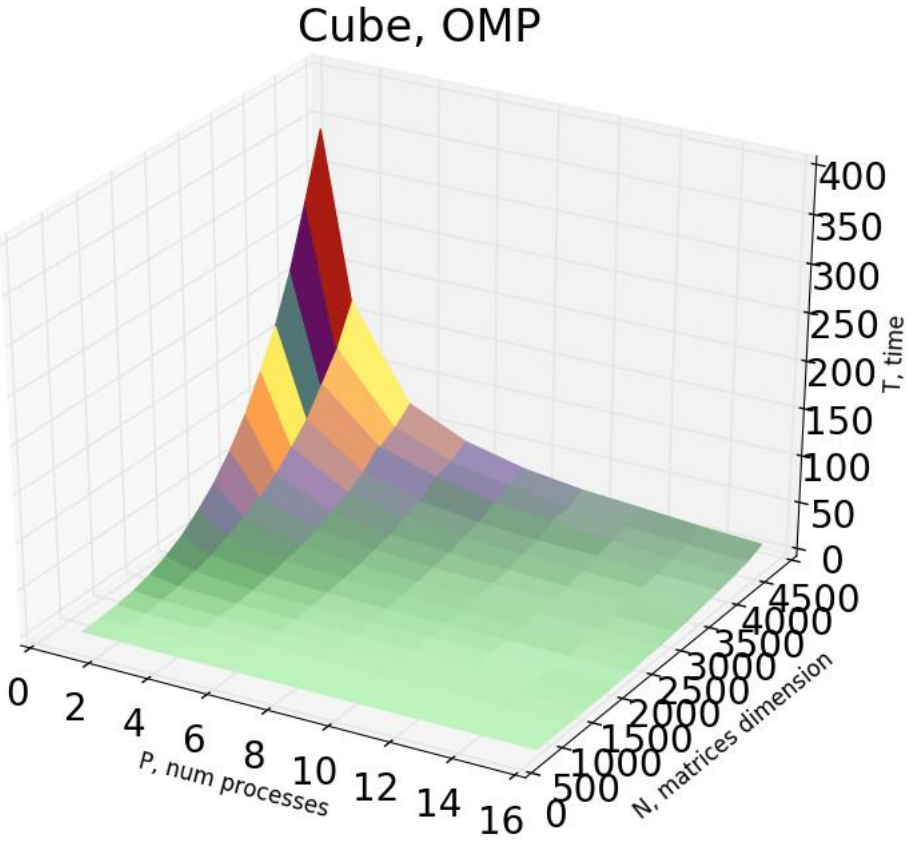
Г1, *зависимость времени работы от размера матриц*



Г2, ***зависимость времени работы от размера кол-ва процессов***



Г3, общий график, MPI-версия



Г4, общий график, OMP-версия

	1	2	4	6	8	10	12	14	16
256	0.08072	0.04116	0.02142	0.02812	0.03165	0.02673	0.09467	0.13046	0.15416
512	0.63646	0.32150	0.17475	0.13523	0.12304	0.13228	0.11186	0.15712	0.16327
768	2.16992	1.09151	0.54994	0.37159	0.31671	0.28419	0.52901	0.39309	0.42388
1024	5.28529	2.68496	1.40408	0.98182	0.95280	0.90976	1.06353	0.81276	0.58855
1280	10.33709	5.18044	2.61127	1.79123	1.38385	1.16571	1.38021	1.35085	0.89920
1536	17.93833	9.04553	4.61453	3.23689	2.65275	2.05278	1.89532	2.10906	1.40229
1792	28.35991	14.18451	7.18225	4.88754	3.88000	3.17432	2.64652	2.33704	2.09768
2048	43.31886	22.13038	11.56047	8.09793	6.44849	5.32295	4.73802	4.13723	3.71832
2304	60.36765	30.19189	15.43822	10.42480	8.02479	6.93918	5.67359	4.67531	4.34072
2560	82.82466	41.46418	21.27996	14.83354	11.04387	8.92424	8.31512	7.42580	6.01616
2816	110.20291	55.44703	28.16472	18.79681	14.39826	11.69290	9.94734	9.46833	7.64998
3072	145.41439	72.99519	36.97237	25.77685	19.44874	15.97229	13.47972	11.84865	10.49412
3328	183.33875	91.42539	47.12660	30.92286	23.66804	19.05859	17.39495	13.94978	12.46944
3584	232.15375	118.58226	64.44729	45.49248	32.52213	28.50815	21.64902	17.97528	15.59587
3840	293.74899	145.61497	74.22266	50.29865	38.61290	29.43913	25.32194	21.47659	18.79893
4096	358.09808	185.75125	88.85359	61.07451	46.51734	38.28267	33.87477	28.86052	25.46810

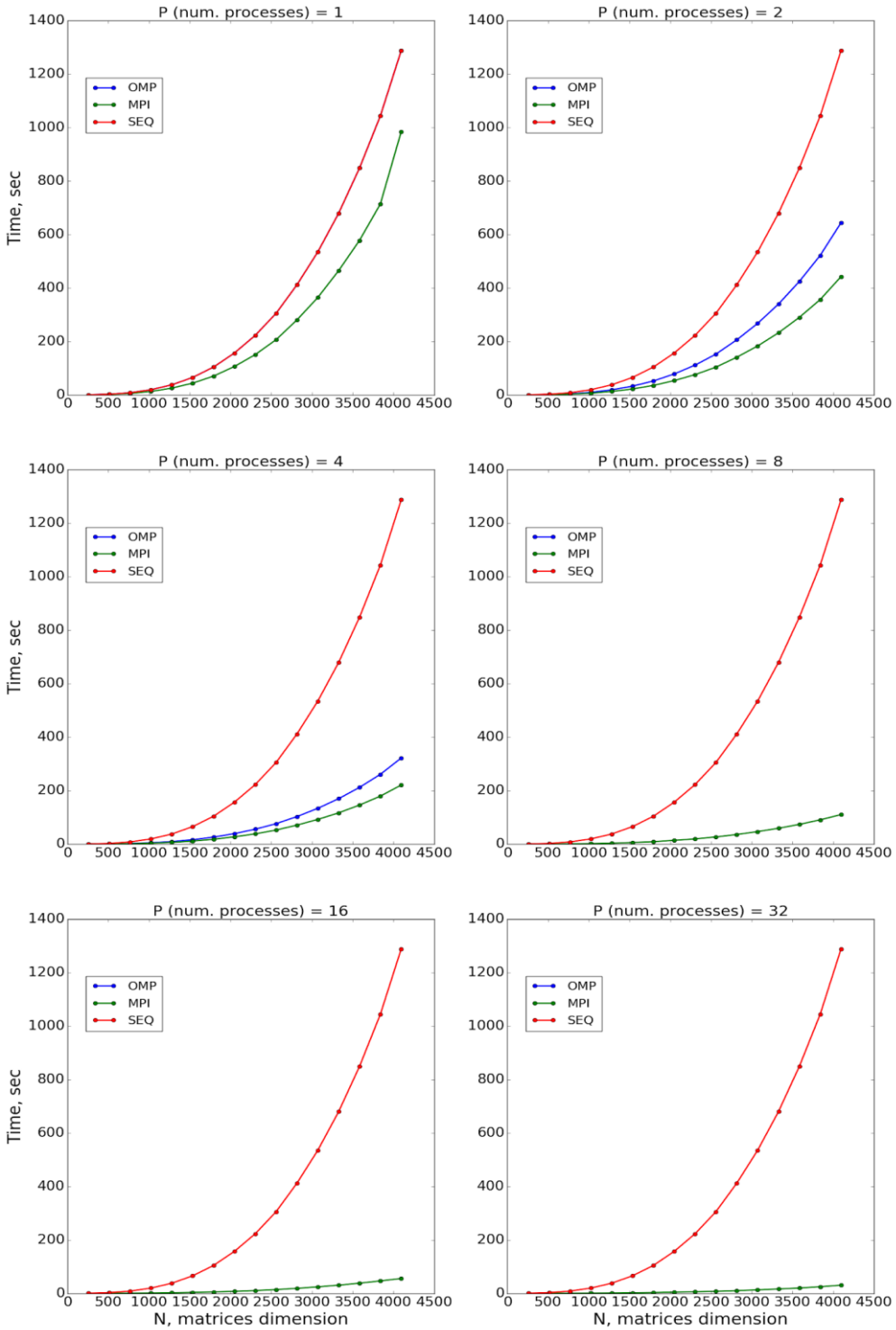
T1, таблица значений для OMP

	1	2	4	6	8	10	12	14	16
256	0.08594	0.05209	0.04251	0.07541	0.09303	0.06888	0.12804	0.14393	0.16095
512	0.68254	0.33903	0.27699	0.16747	0.14565	0.16960	0.12172	0.15246	0.17712
768	2.17212	1.16906	0.65414	0.47031	0.43329	0.33559	0.59604	0.24015	0.26923
1024	5.33622	2.80241	1.51184	1.11188	0.88102	0.75495	0.67599	0.63343	0.65443
1280	10.63210	5.37949	2.79175	1.97709	1.55901	1.33139	1.31670	1.30465	0.99693
1536	17.94890	9.24733	4.80478	3.43676	2.71173	2.24694	1.94674	1.73366	1.63609
1792	28.31840	14.66153	7.54320	5.24903	4.07452	3.40040	3.07507	2.61419	2.35798
2048	43.31026	22.38232	11.62644	8.14941	6.44226	5.36279	4.71226	4.21403	3.82793
2304	60.31617	30.96781	16.07808	10.80671	8.42069	6.95878	5.96156	5.28306	4.84649
2560	82.60776	42.50471	21.94309	14.88142	11.59115	9.50706	8.25952	7.20719	6.45776
2816	109.62862	56.50546	28.92972	19.95879	15.07026	12.33159	10.58555	9.33535	8.33252
3072	144.45580	73.64993	38.02968	26.21756	20.11484	16.36923	14.06892	12.40563	11.17944
3328	181.31261	92.65181	47.78820	32.68943	24.95484	20.09258	17.38530	14.96534	13.42079
3584	228.69640	117.17297	61.09845	40.42957	31.11264	25.41881	21.31938	18.63550	16.64675
3840	287.45403	146.99030	74.70604	51.28049	37.82418	30.89839	26.18094	22.66687	20.17896
4096	352.45140	178.51877	91.82824	62.48472	49.12167	39.33894	33.70886	29.39406	26.08617

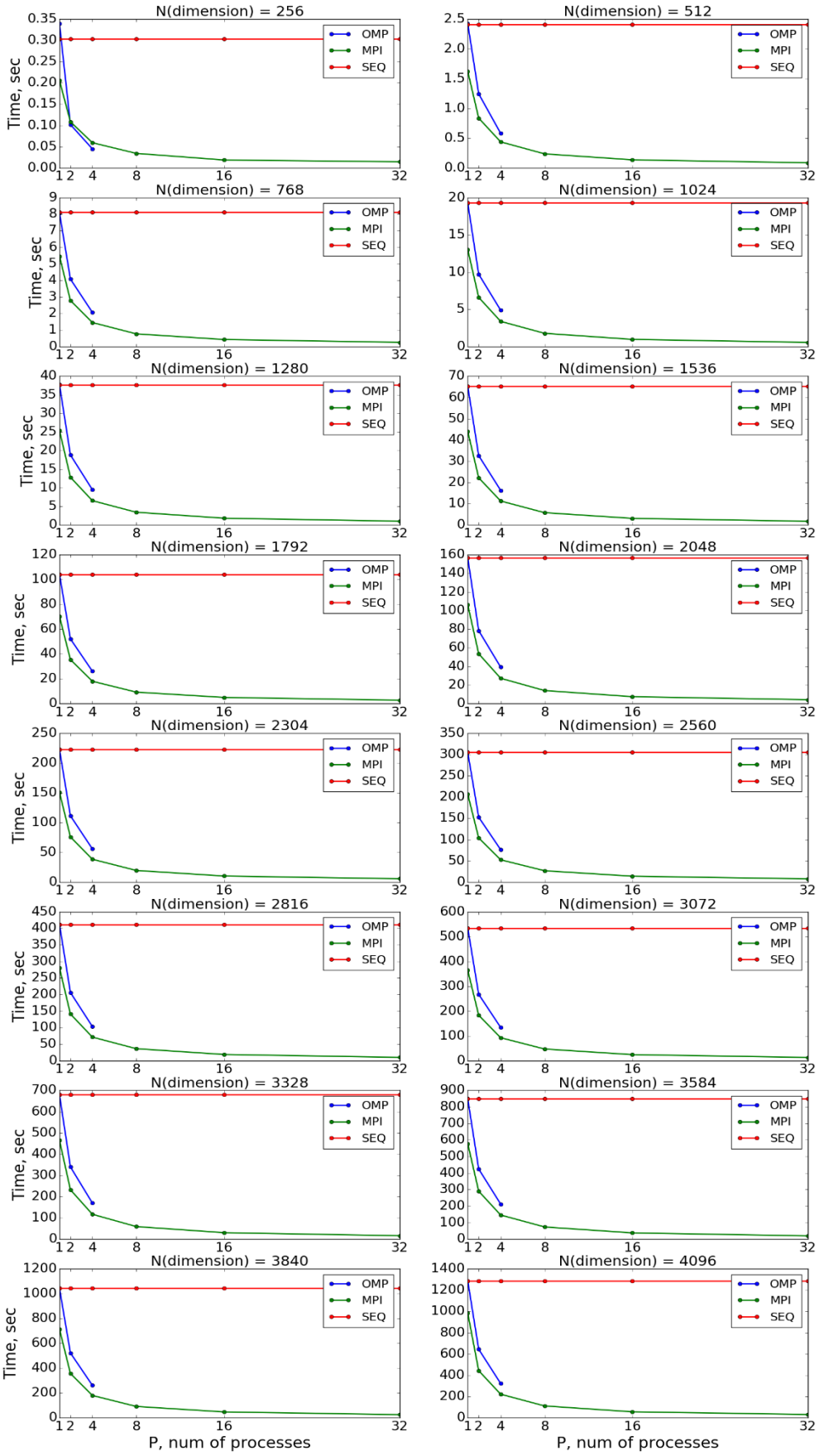
T2, таблица значений для MPI

Bluegene

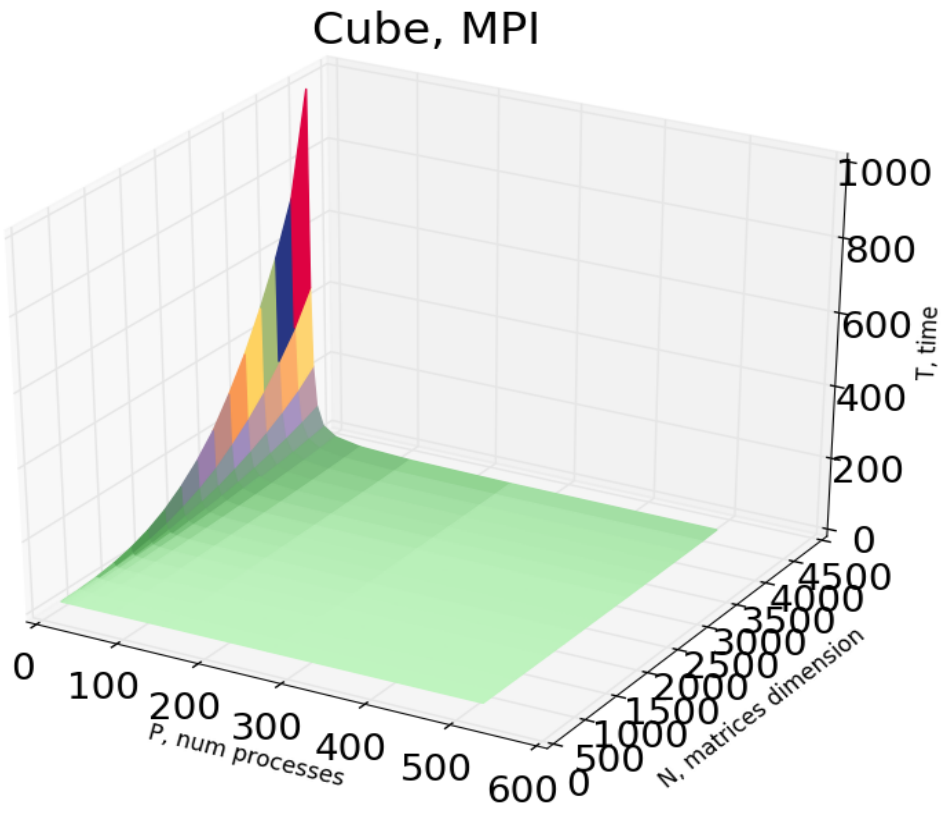
Кол-во процессов OMP: 1, 2, 4
Кол-во процессов MPI: 1, 2, 4, 8, 16 … 512



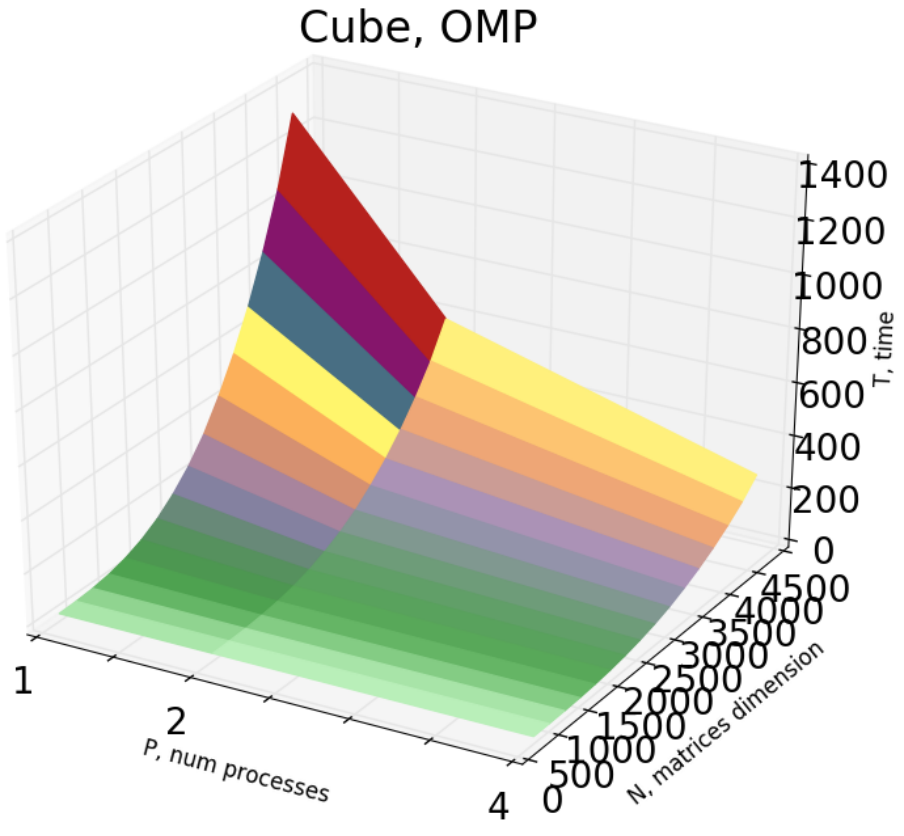
Г5: ***зависимость времени работы от размера матриц***



Г6: ***зависимость времени работы от размера кол-ва процессов***



Г7, общий график, MPI-версия



Г8, общий график, OMP-версия

	1	2	4
256	0.256818	0.151291	0.087605
512	2.445686	1.168844	0.627069
768	8.117645	4.024503	2.061009
1024	19.264020	9.705075	4.783508
1280	37.682382	18.803329	9.396931
1536	65.296926	32.638959	16.325436
1792	104.109890	52.044009	26.052569
2048	156.621751	78.362342	39.204125
2304	222.555110	111.300695	55.608112
2560	305.015192	152.501979	76.244178
2816	411.651395	205.852644	102.899624
3072	534.568222	267.297790	133.685282
3328	679.262475	339.617311	169.779493
3584	848.219644	424.164157	212.081659
3840	1043.283084	521.630669	260.797819
4096	1287.560587	643.740370	321.892513

T3, таблица значений для OMP

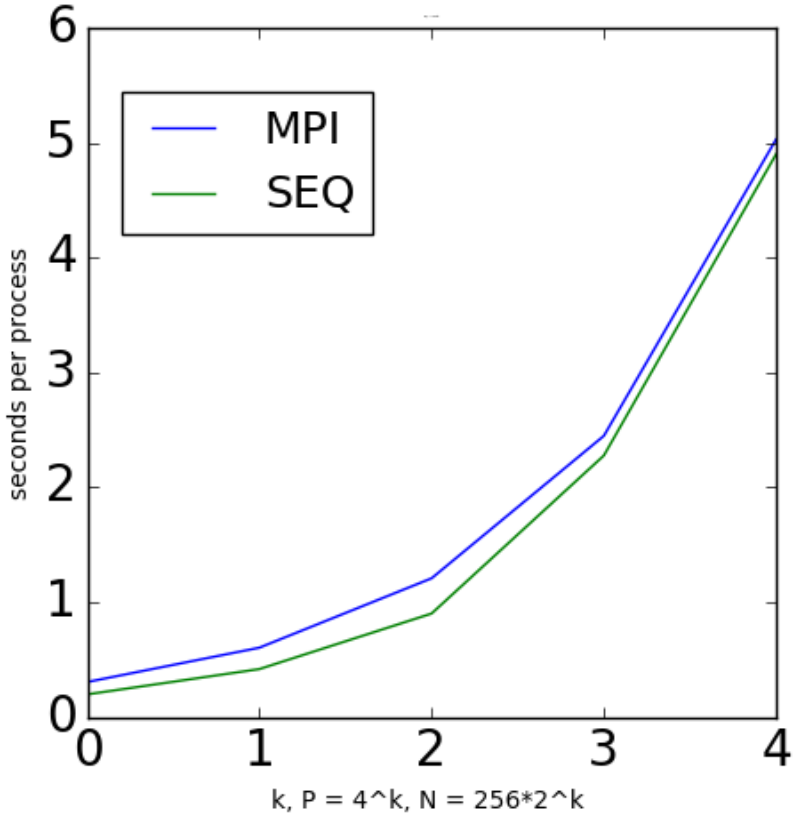
	1	2	4	8	16	32	64	128	256	512
256	0.20549	0.10839	0.05916	0.03407	0.01852	0.01430	0.01316	0.01436	0.01790	0.01848
512	1.62309	0.83591	0.43698	0.23530	0.13527	0.08495	0.05422	0.04764	0.04904	0.05711
768	5.45815	2.77805	1.44494	0.75919	0.41766	0.24615	0.16238	0.11262	0.10153	0.10853
1024	13.00617	6.58926	3.36833	1.75848	0.94484	0.53565	0.33304	0.23453	0.17948	0.16832
1280	25.37473	12.82086	6.52903	3.41571	1.81513	1.00697	0.60874	0.41016	0.30294	0.27570
1536	43.97521	22.18859	11.26634	5.78416	3.10124	1.72140	0.99795	0.65554	0.49062	0.38736
1792	70.25702	35.39147	17.93268	9.17236	4.90252	2.65298	1.53415	0.98182	0.70857	0.59292
2048	106.14546	53.47286	27.13552	13.92660	7.33139	4.05940	2.38961	1.55751	1.15048	0.93985
2304	150.67604	75.77284	38.27817	19.47961	10.09213	5.52533	3.10490	1.89720	1.30202	1.04411
2560	206.43337	103.79780	52.35749	26.60213	13.72737	7.47389	4.15749	2.49633	1.68360	1.25180
2816	280.41898	140.85878	70.99554	35.98650	18.49969	9.72139	5.55613	3.26985	2.14975	1.62321
3072	364.15303	182.85903	92.16631	46.72744	24.02876	12.64789	7.22991	4.26253	2.78991	2.02450
3328	464.63456	232.17976	116.88064	59.12310	30.26794	15.94439	8.91822	5.14508	3.28493	2.41124
3584	577.82445	289.85858	145.86215	73.73888	37.70422	19.64251	11.04580	6.33164	3.99043	2.78741
3840	713.79741	356.37582	179.27666	90.58372	46.26779	24.47539	13.48795	7.69119	4.82618	3.45080
4096	984.11997	442.32117	220.90029	110.44923	55.11103	30.10503	16.97552	10.24090	6.73823	4.92140

T4, таблица значений для MPI

Слабая масштабируемость

Для исследования слабой масштабруемости был построен график по результатам работы последовательной и MPI версий в системе Bluegene.

Так как размер обрабатываемых данных растет квадратично по N, то и количество процессов должно расти квадратично.



Г9, график при N/P = const

Выводы

Результат работы:

1) была реализована последовательная и параллельная версия программы для перемножения матриц с использованием ленточного алгоритма на MPI и OpenMP (код в последней секции)
2) было измерено время работы в зависимости от количества процессов и размеров входных данных, результаты отображены на графиках Г1-Г8. Из графиков видно, что ускорение параллельной версии примерно = 2, что является оптимумом для данной задачи.
3) была исследована сильная и слабая масштабируемость алгоритма. Алгоритм обладает слабой масштабируемостью, не уступающей последовательной версии (график Г9). Также реализация алгоритма (как на OMP, так и на MPI) показывает хорошую сильную масштабируемость (следует из Г2 и Г6).

Поведение алгоритма при вариациях N и P отражено на кубах Г3, Г4, Г7 и Г8.