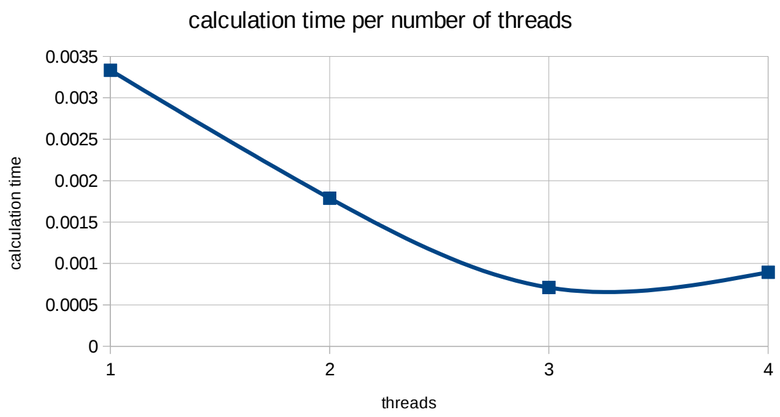
64*64	1 thread	2 threads	4 threads	8 threads
	0.001488	0.001932	0.00071	0.001116
	0.003631	0.000851	0.000631	0.000548
	0.004074	0.001736	0.000972	0.001276
	0.003834	0.001853	0.001098	0.000685
	0.001479	0.001994	0.001098	0.001074
	0.003408	0.001868	0.000432	0.000914
	0.003905	0.002014	0.000621	0.001097
	0.003843	0.001834	0.000553	0.000885
	0.003898	0.002088	0.000433	0.000479
	0.003757	0.001712	0.000544	0.000857
Average	0.003332 sec	0.001788 sec	0.0007092 sec	0.000893 sec

در اینجا با افزایش تعداد نخ ها به ۸ ، میزان **overhead** بیشتر از زمان محاسبات می شود. چون حجم محاسبات کم است

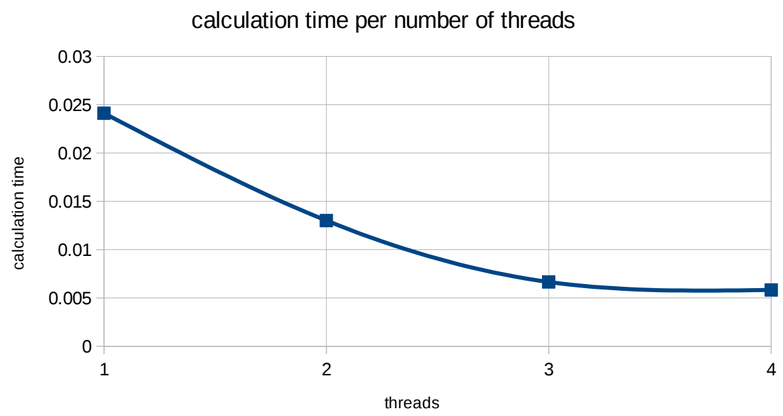
1	0.003332
2	0.001788
4	0.000709
8	0.000893



در اینجا افزایش تعداد نخ ها از ۴ به ۸ افزایش قابل ملاحظه ای در کارایی به وجود نمی آورد و تنها وقت پردازنده صرف **overhead** های ناشی از نخ های اضافه می شود
پس در این حالت نیز تنها ۴ نخ برای محاسبات کافی است.

128*128	1 thread	2 threads	4 threads	8 threads
	0.010816	0.008971	0.004493	0.007615
	0.020489	0.014493	0.005919	0.007884
	0.023435	0.009613	0.007519	0.005299
	0.019563	0.012618	0.007516	0.004306
	0.027737	0.014369	0.006995	0.007364
	0.028016	0.014067	0.007467	0.006778
	0.028461	0.012697	0.007573	0.007236
	0.027151	0.014457	0.008213	0.003115
	0.027684	0.014054	0.007909	0.00525
	0.027841	0.014738	0.002955	0.003407
Average	0.024119 sec	0.013008 sec	0.006656 sec	0.005825 sec

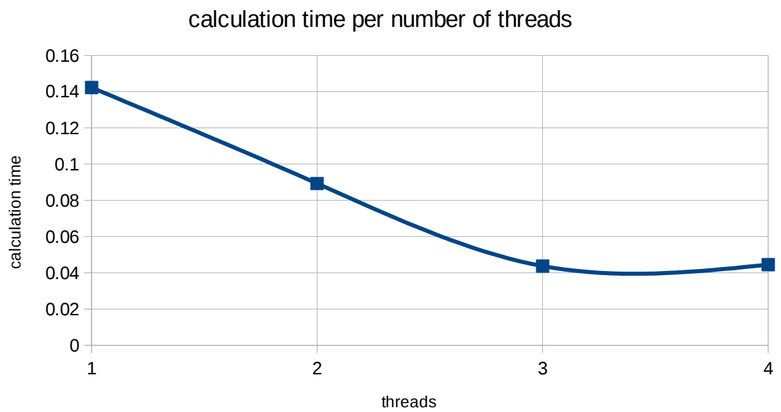
1	0.024119
2	0.013008
4	0.006656
8	0.005825



در این حالت نیز مثل حالت قبل ممکن است برای ۴ و ۸ نخ مقادیر زمان صرف شده نزدیک به هم باشد. پس باز هم ۴ نخ کافی خواهد بود.

256*256	1 thread	2 threads	4 threads	8 threads
	0.159045	0.081131	0.027268	0.027679
	0.108759	0.059986	0.037144	0.035542
	0.158745	0.099745	0.035301	0.027742
	0.096783	0.100475	0.052148	0.044976
	0.145932	0.088943	0.054477	0.056479
	0.162572	0.093477	0.053558	0.051832
	0.161035	0.095612	0.04615	0.054449
	0.15347	0.094077	0.050107	0.044535
	0.161976	0.085335	0.026643	0.04951
	0.113809	0.093836	0.054302	0.051827
Average	0.142213 sec	0.089262 sec	0.04371 sec	0.044457 sec

1	0.142213
2	0.089262
4	0.04371
8	0.044457

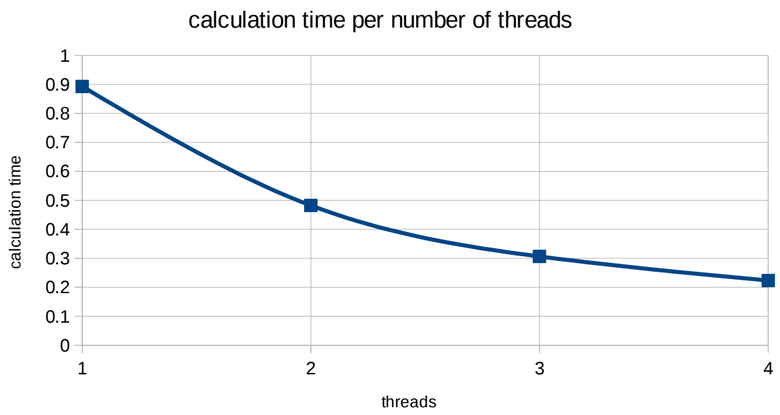


512

512*512	1 thread	2 threads	4 threads	8 threads
	0.911439	0.488989	0.222611	0.208865
	0.874398	0.428283	0.269243	0.212406
	0.916571	0.496361	0.395709	0.211628
	0.87067	0.510615	0.39989	0.211547
	0.83732	0.487643	0.224375	0.213852
	0.916439	0.431403	0.473763	0.255867
	0.935021	0.494385	0.233416	0.21099
	0.902766	0.511786	0.397476	0.211495
	0.849026	0.50093	0.225307	0.246809
	0.915232	0.472781	0.222081	0.248017
Average	0.892888 sec	0.482318 sec	0.306387 sec	0.223148 sec

از این حالت به بعد با سنگین تر شدن محاسبات، وجود ۸ نخ تاثیر مثبت خود را نشان می دهد. البته **speed up** حاصل اط ۸ نخ برای ماتریس های ۵۱۲ تایی خیلی زیاد نیست.

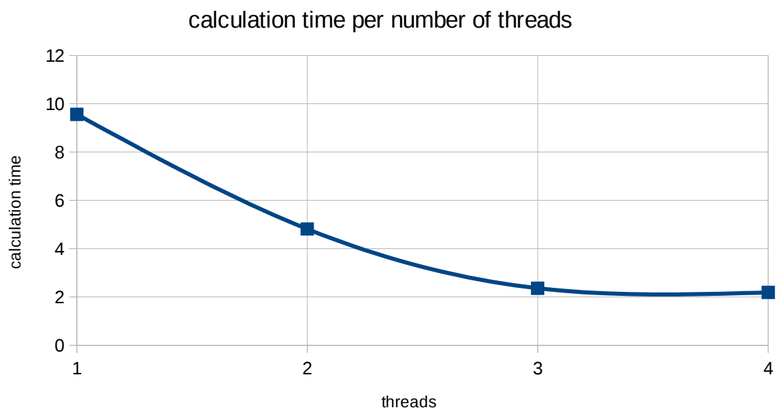
1	0.892888
2	0.482318
4	0.306387
8	0.223148



با دو برابر شدن ابعاد ماتریس ها زمان صرف شده برای محاسبات آن ها ده برابر می شود. این موضوع از طریق مقایسه عدد ها مشخص می شود.
در اینجا مانند حالت ۵۱۲*۵۱۲ استفاده از ۸ نخ زیاد در بهبود زمان پاسخ موثر نیست چون همه ی هسته های سیستم را درگیر میکند اما حتی به میزان ۱.۵ برابر نیز **gain** نمی دهد.

1024*1024	1 thread	2 threads	4 threads	8 threads
	9.57191	4.561515	2.310196	2.188885
	9.675899	4.85921	2.393382	2.166379
	9.46823	4.760934	2.334367	2.173508
	9.461341	4.863726	2.449701	2.195609
	9.628691	4.84372	2.317895	2.202297
	9.527498	4.837622	2.313041	2.199797
	9.350195	4.787525	2.298817	2.138248
	9.497831	4.894487	2.358085	2.229367
	9.712412	4.825344	2.425856	2.216009
	9.68401	4.893461	2.418072	2.169596
Average	9.557802 sec	4.812754 sec	2.361941 sec	2.18797 sec

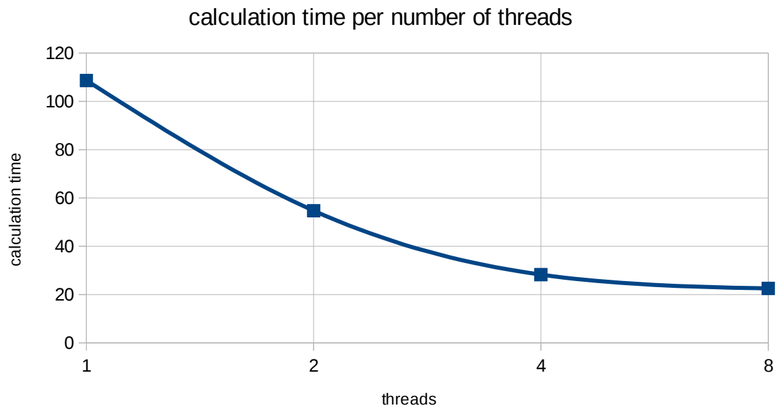
1	9.557802
2	4.812754
4	2.361941
8	2.18797



2048*2048	1 thread	2 threads	4 threads	8 threads
	110.0572	54.04372	28.55386	22.36227
	107.3672	53.20581	28.04674	23.24048
	108.4549	54.63209	28.09978	21.94025
	108.7951	54.62604	28.29131	22.52978
	111.3946	55.39163	28.32533	22.74828
	107.7681	55.59216	28.06628	22.67123
	108.6934	54.88104	28.61862	22.18259
	108.6284	54.56741	28.20649	22.86113
	107.6436	55.10211	28.08365	22.51752
	107.5864	55.19817	28.44978	22.23562
Average	108.6389 sec	54.72402 sec	28.27418 sec	22.52891 sec

توضیحات کاملا مشابه حالت قبل

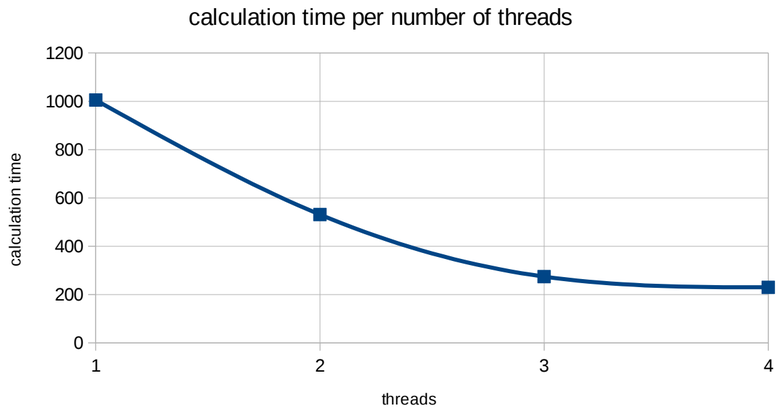
1	108.6389
2	54.72402
4	28.27418
8	22.52891



در اینجا مشخص است که به علت حجم بسیار بالای محاسبات افزایش نخ ها تا ۸ عدد، در کاهش زمان محاسبات بسیار موثر است

2048*2048	1 thread	2 threads	4 threads	8 threads
	988.1171	500.306	255.7622	230.2325
	987.254	574.6027	258.8329	228.3161
	994.3047	539.1982	280.4317	231.0264
	996.1087	623.8232	262.6073	228.4637
	1008.598	557.0056	286.2518	233.8396
	1035.404	505.7855	276.9679	227.5177
	1009.734	507.8955	290.4678	228.3634
	1003.344	501.3734	308.3353	230.5777
	1009.807	502.6862	260.3585	232.0117
	1019.915	500.5306	263.1249	229.9145
Average	1005.259 sec	531.3207 sec	274.314 sec	230.0263 sec

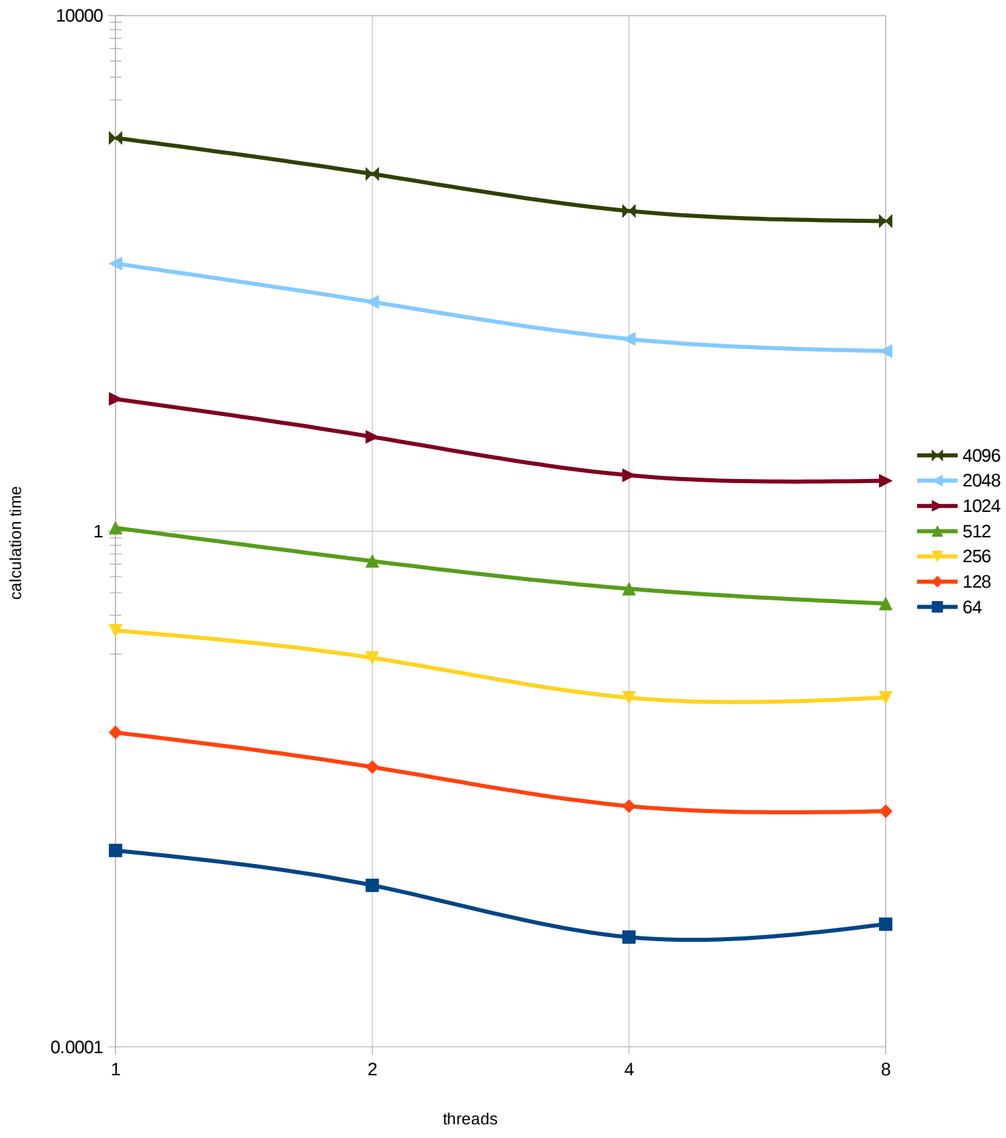
1	1005.259
2	531.3207
4	274.314
8	230.0263



تحلیل افزایش اندازه ماتریس ها

	64	128	256	512	1024	2048	4096
1	0.0033317	0.024119	0.142213	0.892888	9.557802	108.6389	1005.259
2	0.001788	0.013008	0.089262	0.482318	4.812754	54.72402	531.3207
4	0.000709	0.006656	0.04371	0.306387	2.361941	28.27418	274.314
8	0.000893	0.005825	0.044457	0.223148	2.18797	22.52891	230.0263

مقایسه کلی



در این قسمت می توان نمایی کلی از همه نتایج محاسبات را بدست آورد..

محاسباتی که در این فایل آورده شده اند به این گونه بوده که زمان اندازه گیری شده برابر تفاضل زمان ایجاد اولین نخ تا زمان پایان آخرین نخ می باشد.

همانطور که در نمودار قابل مشاهده است رابطه افزایش حجم ماتریس ها به زمان صرف شده لگاریتمی می باشد و تنها در حالت هایی که ماتریس ها کوچک باشند نمودار ها شکل طبیعی خود را از دست می دهند که این موضوع به دلیل کارایی منفی تعداد زیاد نخ ها است.

<u>نتایج کسب شده نشان می دهند:</u>

تا ثیر وجود ۸ نخ از این جهت زیاد نبوده که پردازنده استفاده شده ۴ هسته فیزیکی داشته و ۴ هسته دیگر منطقی بوده اند که این موضوع باعث می شود نتایج به نتایج ۴ نخ نزدیک باشد.

افزایش نخ ها از ۱ به ۲ و از ۲ به ۴ تقریبا باعث نصف شدن زمان محاسبه می گردند.

به ازای دو برابر شدن ابعاد ماتریس ها، زمان محاسبه حاصل ضرب آن ها ۱۰ برابر می شود.

با افزایش تعداد نخ ها در ماتریس های بزرگ تر کاهش زمانی از نصف کمی کمتر است. در حالی که در ماتریس های کوچک تر زمان محاسبات بیشتر از نصف کاهش می یابد. این موضوع نشان می دهد که با بزرگ شدن ماتریس ها **cache miss** به علت **context switch** بیشتر اتفاق می افتد که این در زمان محاسبات خود را نشان می دهد. (در این حالات **overhead** ناشی از **context switch** زیاد نیست – البته بجز حالت ۸ نخی-)

پس با زیاد شدن حجم ماتریس و همچنین با زیاد شدن تعداد نخ ها **cache miss** افزایش می افتد و نقش **cache** در کاهش زمان محاسبات کمرنگ تر می شود.