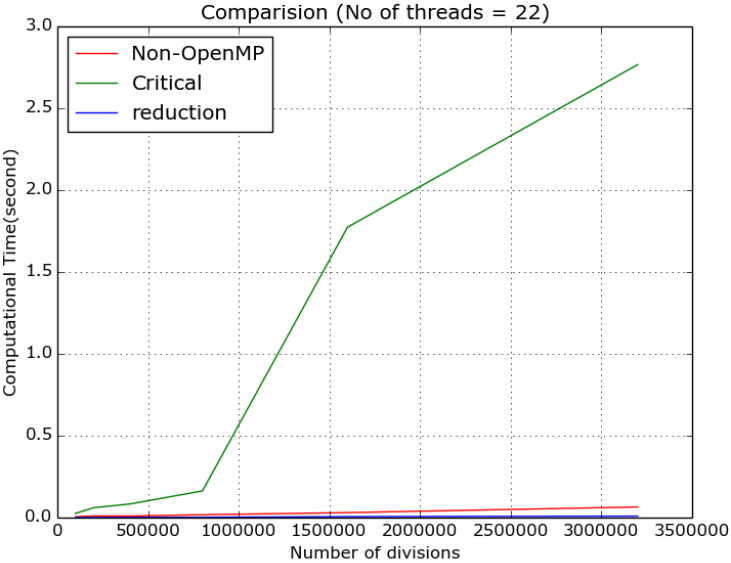
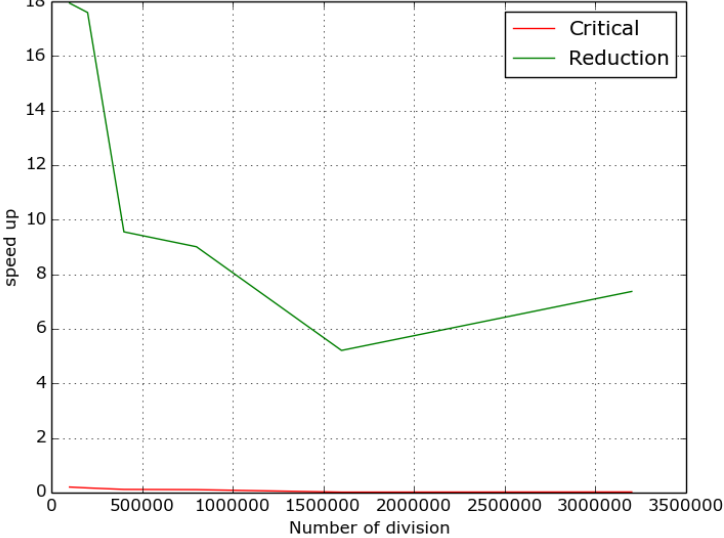


Comparison between three strategies (threads = 22) (Ixlogin5.lrz.de)					
Divisions	Serial Computational time(sec)	Critical		Reduction	
		Computational time(sec)	speedup	Computational time(sec)	speedup
100000	0.005184	0.025477	0.2034776	0.000289	17.93772
200000	0.010344	0.060391	0.1712838	0.000588	17.59184
400000	0.009737	0.083205	0.1170242	0.001019	9.555447
800000	0.017604	0.162887	0.1080749	0.001954	9.009212
1600000	0.030395	1.773891	0.01713465	0.005834	5.209976
3200000	0.064916	2.766577	0.02346437	0.008807	7.370955

Speedup comparison between reduction and critical(threads = 22)



Weak Scaling analysis executed on linux cluster (lxlogin5.lrz.de)								
Division	Non-OpenMP computational time(sec)	Number of threads	Reduction			Critical		
			Computation time(sec)	speedup	efficiency	Computation time(sec)	speedup	efficiency
1000000	0.028113	1	0.023605	1.190976***	1.190976***	0.031736	0.8858394	0.8858394
2000000	0.054428	2	0.017801	3.057581***	1.528791***	0.196787	0.2765833	0.1382917
4000000	0.098105	4	0.019972	4.912127***	1.228032***	0.406059	0.2416028	0.06040071
8000000	0.18249	8	0.030294	6.023965	0.7529956	1.38105	0.1321386	0.01651732
16000000	0.309869	16	0.027731	11.1741	0.6983813	3.798564	0.0815753	0.005098456
32000000	0.617224	32	0.043052	14.33671	0.4480222	29.2469	0.02110391	0.0006594973
*** May be because of compiler auto vectorization, reduction is very fast								

Strong Scaling analysis executed on linux cluster (lxlogin5.lrz.de)

Division	Non-OpenMP computational time(sec)	Number of threads	Reduction			Critical		
			Computation time(sec)	speedup	efficiency	Computation time(sec)	speedup	efficiency
8000000	0.184917	2	0.075032	2.464508 ***	1.232254 ***	1.173571	0.1575678	0.0787839
8000000	0.193327	4	0.053113	3.639919	0.9099797	0.716224	0.2699253	0.06748133
8000000	0.14995	6	0.037547	3.993661	0.6656102	1.107087	0.1354455	0.02257426
8000000	0.185198	8	0.030455	6.081038	0.7601297	1.29742	0.1427433	0.01784291
8000000	0.172461	10	0.029259	5.894289	0.5894289	1.398849	0.1232878	0.01232878
8000000	0.174408	12	0.019541	8.925234	0.7437695	1.066402	0.1635481	0.01362901
8000000	0.147945	14	0.028893	5.120444	0.365746	1.871784	0.07903957	0.005645683
8000000	0.157037	16	0.025469	6.165809	0.3853631	1.551801	0.1011966	0.006324788
8000000	0.195152	18	0.022613	8.63008	0.4794489	5.364394	0.03637913	0.002021063
8000000	0.187772	20	0.017199	10.91761	0.5458806	1.57815	0.1189824	0.005949118
8000000	0.189968	22	0.018282	10.39099	0.4723175	1.742825	0.109	0.004954547
8000000	0.146895	24	0.019942	7.366112	0.3069213	1.75978	0.0834735	0.003478063
8000000	0.147535	26	0.020226	7.294324	0.2805509	6.586375	0.02240003	0.0008615396
8000000	0.150761	28	0.016104	9.361711	0.3343468	8.270326	0.01822915	0.000651041
8000000	0.152732	30	0.019134	7.982231	0.2660744	7.55638	0.02021232	0.0006737441
8000000	0.152901	32	0.018375	8.321143	0.2600357	7.377036	0.02072662	0.0006477068
8000000	0.148192	34	0.016215	9.139192	0.2687998	6.170425	0.0240165	0.0007063676
*** May be because of compiler auto vectorization, reduction is very fast								