CS211: High Performance Computing Project 1

Parallel Sieve of Eratosthenes for Finding All Prime Numbers within 10^{10}

Sieve0	Sieve1	Sieve2
33.38	16.70	15.55

Tab. 1 Performance of Sieve0, Sieve2 and Sieve3 with 32 Cores in 1 Node

Tab. 1 shows the execution time of Sieve0, Siev1 and Sieve2 with 32 cores in 1 node when sieving primes under 10^{10}.

bb \ b	10^5	2×10^5	3×10^5	4×10^5	5×10^5	6×10^5	7×10^5	8×10^5	9×10^5	10^6
700	6.68	6.53	6.50	6.51	6.52	6.51	6.53	6.54	6.54	6.57
800	6.54	6.47	6.43	6.44	7.51	6.45	7.15	7.86	7.17	7.45
900	7.93	6.40	6.34	6.37	6.32	6.34	6.37	6.36	6.40	6.38
1000	6.34	6.34	6.33	6.30	6.30	6.29	6.28	6.29	6.33	6.31
2000	/	5.93	/	/	5.98	5.84	5.95	5.99	6.01	6.01
3000	5.85	5.79	5.87	5.85	5.85	5.80	5.86	5.85	5.89	5.83
4000	5.76	5.77	5.78	5.77	5.78	5.81	5.81	5.82	5.83	5.79
5000	5.86	5.82	5.78	5.82	5.82	5.81	5.83	5.80	5.79	5.80
6000	5.90	5.84	5.86	5.85	5.86	5.85	5.86	5.90	5.84	5.85
7000	/	/	/	5.95	5.85	5.86	/	5.89	5.88	5.85
8000	/	/	/	/	/	/	/	5.91	5.91	5.93
9000	/	/	/	/	/	/	/	5.92	5.96	5.95

Tab. 2 Performance of Sieve3 with 32 cores in 1 node

Tab. 2 shows different combinations of b[1] and bb[2] to find the best performance of Sieve3 with 32 cores in 1 node.

bb \ b	10^5	10^6
2	13.88	13.84
4	12.42	11.24
8	11.31	9.40
32	9.36	8.61
128	7.86	7.92
256	7.29	7.24
512	6.83	6.78
1024	6.32	6.33

[1] b stands for segment size

[2] bb stands for the number of primes we deal with in one segment together

2048	5.95	6.05
4096	5.75	5.84
8192	6.08	5.90

Tab. 3 Performance of Sieve3 with 32 cores in 1 node

bb \ b	10^5	10^6
2	7.02	6.94
4	6.24	6.21
8	5.66	5.64
32	4.72	4.72
128	3.96	3.97
256	3.65	3.66
512	3.41	3.35
1024	3.16	3.13
2048	3.02	2.94
4096	2.99	2.89
8192	3.00	2.98

Tab. 4 Performance of Sieve3 with 64 cores in 2 nodes

bb \ b	1000	5000	10^5	10^6	1.1×10^6	1.2×10^6	1.3×10^6
4000	8.70	6.18	5.76	5.79	5.94	6.17	6.29

Tab. 5 Performance of Sieve3 with different Segment Size and same Prime Group Size with 32 cores in 1 node

Tab. 2, Tab. 3, Tab. 4, and Tab. 5 shows that when b is around 10^5 to 10^6 and bb is around 4000 to 5000, the performance is better than other situations.

Therefore, we choose b = 10^6 and bb = 4000 to test Sieve3 and Sieve0, Sieve1, Sieve 2 with different cores and see their performance.

Cores	Sieve0	Sieve1	Sieve2	Sieve3
1	353.35	184.03	156.93	105.36
32	33.53	16.79	15.55	5.71
64	17.57	8.97	7.78	2.88
128	8.76	4.52	3.88	1.46
256	4.52	2.30	1.90	0.74

Tab. 6 Performance of Sieve0, Sieve 1, Sieve 2, Sieve 3 with 1, 32, 64, 128, 256 cores

Also, we can calculate speedup and efficiency from Tab. 6.

Cores	Sieve0	Sieve1	Sieve2	Sieve3
32	10.54	10.96	10.09	18.45
64	20.11	20.52	20.17	36.58
128	40.34	40.71	40.45	72.16
256	78.17	80.01	82.59	142.38

Tab.7 Speedup of Siev0, Sieve1, Sieve2, Sieve3 with 1, 32, 128, 256 cores

Cores	Sieve0	Sieve1	Sieve2	Sieve3
32	0.329	0.343	0.315	0.577
64	0.314	0.321	0.315	0.572
128	0.315	0.318	0.316	0.564
256	0.305	0.313	0.323	0.556

Tab.8 Efficiency of Siev0, Sieve1, Sieve2, Sieve3 with 1, 32, 128, 256 cores

We can draw Fig. 1 from Tab. 6, Fig. 2 from Tab. 7 and Fig. 3 from Tab. 8.

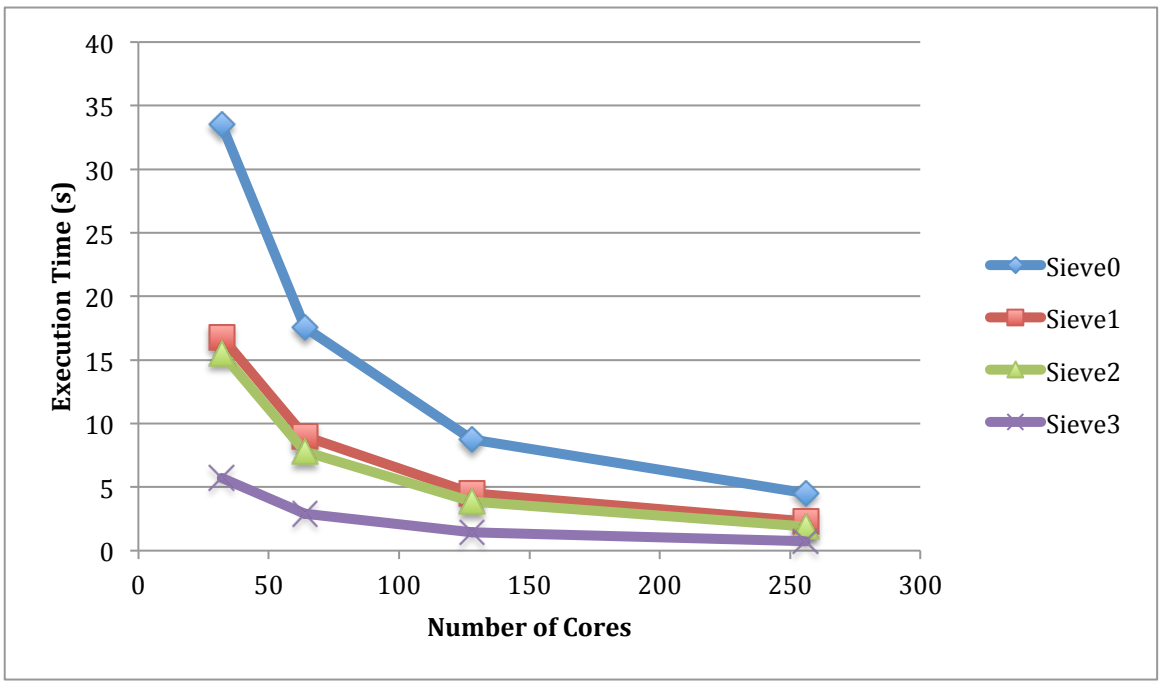

Fig. 1 Performance of Sieve0, Sieve 1, Sieve 2, Sieve 3 with 32, 64, 128, 256 cores

We can see that the decent rate of execution time tends to slow and the differences of execution time between each version become lower and slower while the number of cores increases.

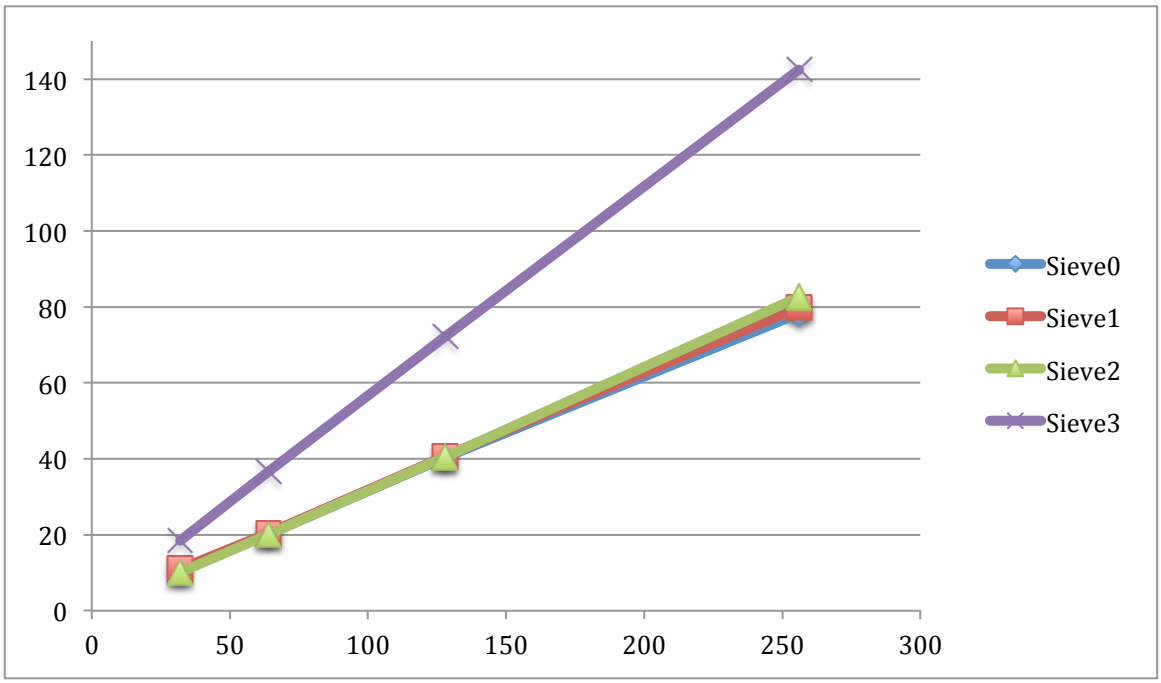

Fig. 2 Speedup of Sieve1, Sieve2, Sieve3 with 32, 64, 128, 256 cores

From Fig. 2 we see that the speedup is nearly 2 but will slightly decrease with the number of cores increasing, which satisfies Amdahl's Law.

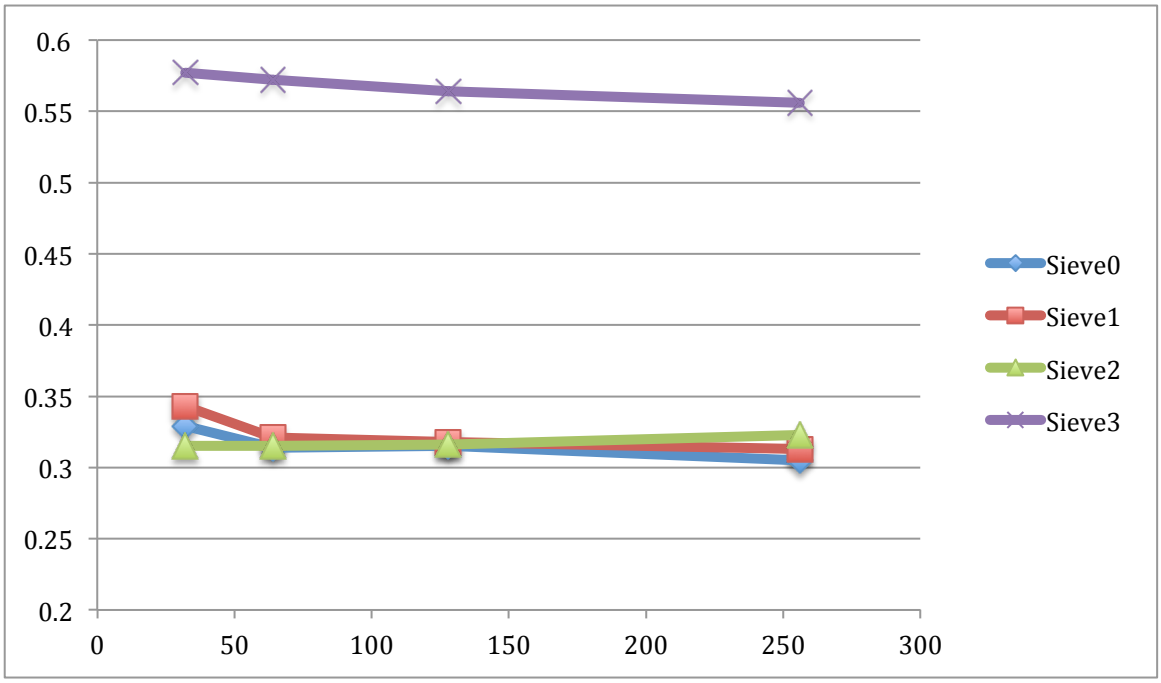

Fig. 3 Efficiency of Sieve0, Sieve 1, Sieve2, Sieve 3 with 32, 64, 128, 256 cores

From Fig. 3 we can see that the efficiency decreases if we increase the number of cores. Obviously, the efficiency of Sieve3 is much higher than the other three versions, which means reducing cache miss can improve efficiency of parallel program.

Cores	Sieve0	Sieve1	Sieve2	Sieve3
1	455052511	455052511	455052511	455052511
32	455052511	455052511	455052511	455052511
64	455052511	455052511	455052511	455052511
128	455052511	455052511	455052511	455052511
256	455052511	455052511	455052511	455052511

Tab.9 Number of Primes within 10^{10} finding by Sieve0, Sieve1, Sieve2 and Sieve3 with 1, 32, 128, 256 cores

Tab. 9 shows the number of primes within 10^{10} finding by Sieve0, Sieve1, Sieve2 and Sieve3. The correct result is 455,052,511; therefore, we verify the correctness of Sieve0, Sieve1, Sieve2 and Sieve3.