

Learning To Benchmark

By Ashot Vardanian,
Founder of [Unum.cloud](https://unum.cloud)



--

linkedin.com/in/ashvardanian

t.me/ashvardanian

ashvardanian.com

The Plan for Ultimate Google Benchmark Exploration

1. Nanosecond-resolution done right 
2. Bigger routines and daily programming
3. `copmare.py` the results and `perf`
4. Big data and performance addiction 

compare.py: Epyc vs Threadripper

Comparing results_epyc2x16core_o3.json to results_proripper64core_o3.json

Benchmark	Time	CPU	Time Old	Time New	CPU Old	CPU New
i32_addition	+0.1608	+0.1000	0	0	0	0
i32_addition_random	-0.2326	-0.2327	26	20	26	20
i32_addition_random_and_used	-0.2372	-0.2372	0	0	0	0
i32_addition_random/threads:8	-0.1951	-0.2241	505	407	3859	2994
i32_addition_random_and_used/threads:8	-0.3330	-0.3331	0	0	0	0
f64_sin_maclaurin	-0.2593	-0.2595	37	27	37	27
f64_sin_maclaurin_powless	-0.2192	-0.2194	3	2	3	2
f64_sin_maclaurin_with_fast_math	-0.2188	-0.2190	1	1	1	1
i64_division	-0.1667	-0.1671	4	4	4	4
i64_division_by_const	-0.2227	-0.2230	4	3	4	3
i64_division_by_constexpr	-0.2779	-0.2780	1	1	1	1
u64_population_count/min_time:10.000	-0.2244	-0.2246	2	2	2	2
u64_population_count_x86/min_time:10.000	-0.2493	-0.2495	0	0	0	0
sorting/3/0	-0.2134	-0.2142	288	227	292	229
sorting/3/1	-0.2075	-0.2078	12	9	12	9
sorting/4/0	-0.2329	-0.2338	298	229	302	231
sorting/4/1	-0.2440	-0.2442	19	14	19	14
upper_cost_of_branching	-0.2162	-0.2163	3	2	3	2
upper_cost_of_pausing	-0.2202	-0.2201	269	210	273	213
sorting_template<false>/3	-0.2043	-0.2049	288	229	291	232
sorting_template<true>/3	-0.2486	-0.2486	11	9	11	9
sorting_template<false>/4	-0.2140	-0.2140	291	228	294	231
sorting_template<true>/4	-0.2225	-0.2227	18	14	18	14
supersort/seq/1048576/min_time:10.000	-0.3429	-0.3430	12452005	8182061	12449871	8179546
supersort/seq/2097152/min_time:10.000	-0.4069	-0.4071	29070864	17241057	29069260	17234537
supersort/seq/16777216/min_time:10.000	-0.3118	-0.3121	235651679	162174550	235621280	162084395
supersort/par_unseq/1048576/min_time:10.000	-0.2426	-0.2412	2085363	1579550	2070193	1570895
supersort/par_unseq/2097152/min_time:10.000	-0.1928	-0.1908	3573183	2884248	3540504	2865039
supersort/par_unseq/16777216/min_time:10.000	-0.2037	-0.2022	37764082	30071609	37448853	29877778
supersort/par_unseq/1048576/min_time:10.000/real_time	-0.3054	-0.3072	1998767	1388438	1988567	1377725
supersort/par_unseq/2097152/min_time:10.000/real_time	-0.1279	-0.1318	3330092	2904142	3318336	2880815
supersort/par_unseq/16777216/min_time:10.000/real_time	-0.2890	-0.2887	42249399	30039769	42005130	29880237

compare.py: O1 vs O3

Comparing results_proripper64core_o1.json to results_proripper64core_o3.json						
Benchmark	Time	CPU	Time Old	Time New	CPU Old	CPU New
supersort/seq/1048576/min_time:10.000	-0.2860	-0.2861	11460205	8182061	11457282	8179546
supersort/seq/2097152/min_time:10.000	-0.2991	-0.2992	24598392	17241057	24591333	17234537
supersort/seq/16777216/min_time:10.000	-0.2654	-0.2656	220780034	162174550	220702858	162084395
supersort/seq/134217728/min_time:10.000	-0.1948	-0.1950	1820948340	1466143776	1820114719	1465222949
supersort/seq/1073741824/min_time:10.000	-0.0917	-0.0917	14728303563	13378114405	14721443008	13371574132
supersort/seq/4294967296/min_time:10.000	-0.1307	-0.1307	290213234104	252293336907	290076568824	252161121071
supersort/par_unseq/1048576/min_time:10.000	-0.0550	-0.0443	1671502	1579550	1643638	1570895
supersort/par_unseq/2097152/min_time:10.000	+0.0205	+0.0268	2826381	2884248	2790176	2865039
supersort/par_unseq/16777216/min_time:10.000	-0.0073	-0.0045	30291790	30071609	30013922	29877778
supersort/par_unseq/134217728/min_time:10.000	+0.0065	+0.0011	216143277	217542511	212383874	212612391
supersort/par_unseq/1073741824/min_time:10.000	+0.0378	+0.1316	1987026768	2062182500	1733090932	1961110135
supersort/par_unseq/4294967296/min_time:10.000	+0.0334	+0.0415	8757824467	9050303863	7980895174	8311968415
supersort/par_unseq/1048576/min_time:10.000/real_time	-0.1670	-0.1650	1666705	1388438	1650047	1377725
supersort/par_unseq/2097152/min_time:10.000/real_time	-0.0219	-0.0205	2969102	2904142	2941143	2880815
supersort/par_unseq/16777216/min_time:10.000/real_time	-0.0079	-0.0042	30279776	30039769	30004919	29880237
supersort/par_unseq/134217728/min_time:10.000/real_time	+0.0271	+0.0434	209718276	215402795	204316397	213179561
supersort/par_unseq/1073741824/min_time:10.000/real_time	-0.0649	-0.0337	2015077898	1884217994	1843638280	1781486333
supersort/par_unseq/4294967296/min_time:10.000/real_time	+0.0016	+0.0813	8923044812	8937212388	7657745746	8280547435

perf ing

```
sudo perf stat taskset 0xFFFFEFFFEEFFFEFFFEFFFEFFFEFFFEFF \
./release/main \
--benchmark_enable_random_interleaving=true \
--benchmark_filter=supersort
```

Results

```
23048674.55 msec task-clock                # 35.901 CPUs utilized
      6627669      context-switches        # 0.288 K/sec
      75843        cpu-migrations          # 0.003 K/sec
     119085703      page-faults            # 0.005 M/sec
91429892293048      cycles                  # 3.967 GHz                    (83.33%)
13895432483288      stalled-cycles-frontend # 15.20% frontend cycles idle (83.33%)
 3277370121317      stalled-cycles-backend  # 3.58% backend cycles idle   (83.33%)
16689799241313      instructions            # 0.18 insn per cycle
                                     # 0.83 stalled cycles per insn (83.33%)
 3413731599819      branches                # 148.110 M/sec                (83.33%)
  11861890556        branch-misses          # 0.35% of all branches       (83.34%)

642.008618457 seconds time elapsed

21779.611381000 seconds user
1244.984080000 seconds sy
```

Performance Becomes An Addiction

I am abusing that drug for 6 years now

- 879 GB/s Parallel Reductions in C++ & CUDA.
- Failing to Reach DDR4 Bandwidth.

Let's explore the hard cases.

Try Yourself

GitHub:

- [ashvardanian/BenchmarkingTutorial](#)
- [unum-cloud/ParallelReductions](#)