



g15-memory-density

Intel Core i7-11800H testing with a Dell G15 (model 5511, 1.7.0 BIOS, DDR4-3200AA) and GeForce RTX 3060 Laptop GPU 6 GB on Ubuntu 20.04.3 via the Phoronix Test Suite.

Automated Executive Summary

x08 had the most wins, coming in first place for 73% of the tests.

Based on the geometric mean of all complete results, the fastest (x08) was 1.058x the speed of the slowest (x16).

The results with the greatest spread from best to worst included:

Zstd Compression (Compression Level: 8 - Compression Speed) at 1.498x

Algebraic Multi-Grid Benchmark at 1.382x

Stream (Type: Scale) at 1.38x

Stream (Type: Triad) at 1.356x

Stream (Type: Add) at 1.349x

Y-Cruncher (Calculating 500M Pi Digits) at 1.303x

Zstd Compression (Compression Level: 19 - Compression Speed) at 1.302x

RAMspeed SMP (Type: Scale - Benchmark: Integer) at 1.296x

Stream (Type: Copy) at 1.273x

Zstd Compression (Compression Level: 19, Long Mode - Compression Speed) at 1.272x.

Test Systems:

x16

x08

Processor: Intel Core i7-11800H @ 4.60GHz (8 Cores / 16 Threads), Motherboard: Dell 0YC2KJ (1.7.0 BIOS), Chipset: Intel Device 43ef, Memory: 16GB, Disk: Kioxia KBG40ZNS1T02 NVMe 1024GB, Graphics: NVIDIA GeForce RTX 3060 Laptop GPU 6GB, Audio: Intel Device 43c8, Network: Realtek Device 2600 + Intel Device 43f0

OS: Ubuntu 20.04, Kernel: 5.14.0-1007-oem (x86_64), Desktop: GNOME Shell 3.36.9, Display Server: X Server 1.20.11, Display Driver: NVIDIA 470.82.00, OpenGL: 4.6.0, OpenCL: OpenCL 3.0 + OpenCL 3.0 CUDA 11.4.153, Vulkan: 1.2.145, Compiler: GCC 10.3.0 + Clang 10.0.0-4ubuntu1, File-System: ext4, Screen Resolution: 1920x1080

Kernel Notes: Transparent Huge Pages: madvise

Compiler Notes: --build=x86_64-linux-gnu --disable-vtable-verify --disable-werror --enable-bootstrap --enable-checking=release --enable-clocale=gnu --enable-default-pie --enable-gnu-unique-object --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,m2 --enable-libphobos-checking=release --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-link-mutex --enable-multiarch --enable-multilib --enable-nls --enable-objc-gc=auto --enable-offload-targets=nvptx-none=/build/gcc-10-S4I5Pr/gcc-10-10.3.0/debian/tmp-nvptx/usr,amdgc- amdhsa=/build/gcc-10-S4I5Pr/gcc-10-10.3.0/debian/tmp-gcn/usr,hsa --enable-plugin --enable-shared --enable-threads=posix --host=x86_64-linux-gnu --program-prefix=x86_64-linux-gnu- --target=x86_64-linux-gnu --with-abi=m64 --with-arch-32=i686 --with-build-config=bootstrap-lto-lean --with-default-libstdcxx-abi=new --with-gcc-major-version-only --with-multilib-list=m32,m64,mx32 --with-target-system-zlib=auto --with-tune=generic --without-cuda-driver -v

Processor Notes: Scaling Governor: intel_pstate performance - CPU Microcode: 0x2c - Thermald 1.9.1

OpenCL Notes: GPU Compute Cores: 3840

Python Notes: Python 3.8.10

Security Notes: itlb_multihit: Not affected + l1tf: Not affected + mds: Not affected + meltdown: Not affected + spec_store_bypass: Mitigation of SSB disabled via prctl and seccomp + spectre_v1: Mitigation of usercopy/swaps barriers and __user pointer sanitization + spectre_v2: Mitigation of Enhanced IBRS IBPB: conditional RSB filling + srbds: Not affected + tsx_async_abort: Not affected

	x16	x08
RAMspeed SMP - Add - Integer (MB/s)	26469	32449
Normalized	81.57%	100%
Standard Deviation	0.1%	0.3%
RAMspeed SMP - Copy - Integer (MB/s)	22757	28909
Normalized	78.72%	100%
Standard Deviation	0.6%	1%
RAMspeed SMP - Scale - Integer (MB/s)	22172	28745
Normalized	77.14%	100%
Standard Deviation	0.1%	0.1%
RAMspeed SMP - Triad - Integer (MB/s)	26269	32530
Normalized	80.75%	100%
Standard Deviation	0.3%	0.9%
RAMspeed SMP - Average - Integer (MB/s)	24319	29591
Normalized	82.18%	100%
Standard Deviation	0.5%	1.8%
RAMspeed SMP - Add - Floating Point (MB/s)	26061	30800
Normalized	84.61%	100%
Standard Deviation	0.2%	0.8%
RAMspeed SMP - Copy - Floating Point (MB/s)	22829	27332
Normalized	83.52%	100%
Standard Deviation	0.1%	0.2%
RAMspeed SMP - Scale - Floating Point (MB/s)	21835	25963
Normalized	84.1%	100%

	Standard Deviation	0.2%	2.3%
RAMspeed SMP - Triad - Floating Point (MB/s)		26197	30606
	Normalized	85.59%	100%
	Standard Deviation	0.1%	0.1%
RAMspeed SMP - Average - Floating Point (MB/s)		24071	28739
	Normalized	83.76%	100%
	Standard Deviation	0.7%	0%
Stream - Copy (MB/s)		28240	35956
	Normalized	78.54%	100%
	Standard Deviation	0.4%	1.3%
Stream - Scale (MB/s)		15358	21187
	Normalized	72.49%	100%
	Standard Deviation	0.3%	1.4%
Stream - Triad (MB/s)		16392	22220
	Normalized	73.77%	100%
	Standard Deviation	0.4%	1.4%
Stream - Add (MB/s)		16473	22230
	Normalized	74.1%	100%
	Standard Deviation	0.3%	1.2%
Tinymembench - Standard Memcpy (MB/s)		20238	21396
	Normalized	94.59%	100%
	Standard Deviation	0.3%	2.4%
Tinymembench - Standard Memset (MB/s)		39452	39898
	Normalized	98.88%	100%
	Standard Deviation	0.4%	1.9%
MBW - Memory Copy - 128 MiB (MiB/s)		15689	16848
	Normalized	93.12%	100%
	Standard Deviation	0.4%	0.2%
MBW - Memory Copy - 512 MiB (MiB/s)		14884	16189
	Normalized	91.94%	100%
	Standard Deviation	0.4%	0.8%
MBW - Memory Copy - 1024 MiB (MiB/s)		14649	15857
	Normalized	92.38%	100%
	Standard Deviation	0.2%	0.9%
MBW - Memory Copy - 4096 MiB (MiB/s)		14647	15880
	Normalized	92.24%	100%
	Standard Deviation	0.3%	1.6%
MBW - M.C.F.B.S - 128 MiB (MiB/s)		12619	14225
	Normalized	88.7%	100%
	Standard Deviation	2.4%	0.1%
MBW - M.C.F.B.S - 512 MiB (MiB/s)		12051	13591
	Normalized	88.67%	100%
	Standard Deviation	0.4%	0.4%
MBW - M.C.F.B.S - 1024 MiB (MiB/s)		12037	13357
	Normalized	90.11%	100%
	Standard Deviation	0%	0.1%
MBW - M.C.F.B.S - 4096 MiB (MiB/s)		12107	13369
	Normalized	90.56%	100%
	Standard Deviation	0.7%	1%
t-test1 - 1 (sec)		12.860	12.717
	Normalized	98.89%	100%
	Standard Deviation	0.5%	1.8%
t-test1 - 2 (sec)		4.474	4.397
	Normalized	98.28%	100%
	Standard Deviation	1.7%	0.8%

C-Blosc - blosclz (MB/s)	23270	24997
Normalized	93.09%	100%
Standard Deviation	0.7%	0.7%
HPC Challenge - G-HPL (GFLOPS)	50.69073	51.49380
Normalized	98.44%	100%
Standard Deviation	1.5%	1.7%
Algebraic Multi-Grid Benchmark (Figure Of Merit)	185269100	256019367
Normalized	72.37%	100%
Standard Deviation	0.1%	2.3%
simdjson - Kostya (GB/s)	3.46	3.50
Normalized	98.86%	100%
Standard Deviation	1%	0.4%
simdjson - LargeRand (GB/s)	1.20	1.21
Normalized	99.17%	100%
Standard Deviation	0.5%	0.5%
simdjson - PartialTweets (GB/s)	5.09	5.04
Normalized	100%	99.02%
Standard Deviation	1.6%	0.7%
simdjson - DistinctUserID (GB/s)	5.65	5.67
Normalized	99.65%	100%
Standard Deviation	0.1%	0.2%
CacheBench - Read (MB/s)	4375	4375
Normalized	99.99%	100%
Standard Deviation	0%	0%
CacheBench - Write (MB/s)	31306	31235
Normalized	100%	99.77%
Standard Deviation	0.1%	0.4%
CacheBench - R.M.W (MB/s)	49307	49302
Normalized	100%	99.99%
Standard Deviation	0%	0%
Zstd Compression - 3 - Compression Speed (MB/s)	3329	3620
Normalized	91.94%	100%
Standard Deviation	0.7%	2.4%
Zstd Compression - 3 - D.S (MB/s)	4292	4396
Normalized	97.65%	100%
Standard Deviation	0.3%	0.6%
Zstd Compression - 8 - Compression Speed (MB/s)	266.3	398.8
Normalized	66.78%	100%
Standard Deviation	2.3%	2.1%
Zstd Compression - 8 - D.S (MB/s)	4408	4504
Normalized	97.88%	100%
Standard Deviation	0.6%	0.6%
Zstd Compression - 19 - Compression Speed (MB/s)	26.8	34.9
Normalized	76.79%	100%
Standard Deviation	1%	2.1%
Zstd Compression - 19 - D.S (MB/s)	3738	3801
Normalized	98.35%	100%
Standard Deviation	0.3%	0.1%
Zstd Compression - 3, Long Mode - Compression Speed (MB/s)	1348	1385
Normalized	97.33%	100%
Standard Deviation	1%	0.5%
Zstd Compression - 3, Long Mode - D.S (MB/s)	4596	4716
Normalized	97.45%	100%
Standard Deviation	0.6%	0.7%

Zstd Compression - 8, Long Mode - Compression Speed (MB/s)	760.3	907.5
Normalized	83.78%	100%
Standard Deviation	2.8%	0.9%
Zstd Compression - 8, Long Mode - D.S (MB/s)	4691	4796
Normalized	97.8%	100%
Standard Deviation	0.5%	0.6%
Zstd Compression - 19, Long Mode - Compression Speed (MB/s)	24.6	31.3
Normalized	78.59%	100%
Standard Deviation	0.8%	1.1%
Zstd Compression - 19, Long Mode - D.S (MB/s)	3846	3916
Normalized	98.21%	100%
Standard Deviation	0.8%	0.6%
srsRAN - OFDM_Test (Samples / Second)	162666667	163733333
Normalized	99.35%	100%
Standard Deviation	0.7%	2.4%
srsRAN - 4.P.1.P.M.6.Q (eNb Mb/s)	401.8	401.5
Normalized	100%	99.93%
Standard Deviation	0.9%	0.5%
srsRAN - 4.P.1.P.M.6.Q (UE Mb/s)	164.6	164.2
Normalized	100%	99.76%
Standard Deviation	0.8%	0%
srsRAN - 4.P.1.P.S.6.Q (eNb Mb/s)	393.6	401.2
Normalized	98.11%	100%
Standard Deviation	0.6%	0.3%
srsRAN - 4.P.1.P.S.6.Q (UE Mb/s)	239.4	240.3
Normalized	99.63%	100%
Standard Deviation	0.1%	0.7%
srsRAN - 4.P.1.P.M.2.Q (eNb Mb/s)	442.4	445.0
Normalized	99.42%	100%
Standard Deviation	0.3%	0.4%
srsRAN - 4.P.1.P.M.2.Q (UE Mb/s)	180.9	182.4
Normalized	99.18%	100%
Standard Deviation	0.8%	0.2%
srsRAN - 4.P.1.P.S.2.Q (eNb Mb/s)	436.4	441.3
Normalized	98.89%	100%
Standard Deviation	2.5%	0.8%
srsRAN - 4.P.1.P.S.2.Q (UE Mb/s)	282.6	284.9
Normalized	99.19%	100%
Standard Deviation	1.4%	0.4%
srsRAN - 5.P.T.5.P.S.6.Q (eNb Mb/s)	158.8	158.3
Normalized	100%	99.69%
Standard Deviation	0.5%	0.8%
srsRAN - 5.P.T.5.P.S.6.Q (UE Mb/s)	100	99.8
Normalized	100%	99.8%
Standard Deviation		0.3%
srsRAN - 5.P.T.2.P.S.2.Q (eNb Mb/s)	182.1	181.6
Normalized	100%	99.73%
Standard Deviation	0.3%	0.6%
srsRAN - 5.P.T.2.P.S.2.Q (UE Mb/s)	137.1	135.9
Normalized	100%	99.12%
Standard Deviation	0.5%	0.7%
ArrayFire - BLAS CPU (GFLOPS)	953.851	949.270
Normalized	100%	99.52%

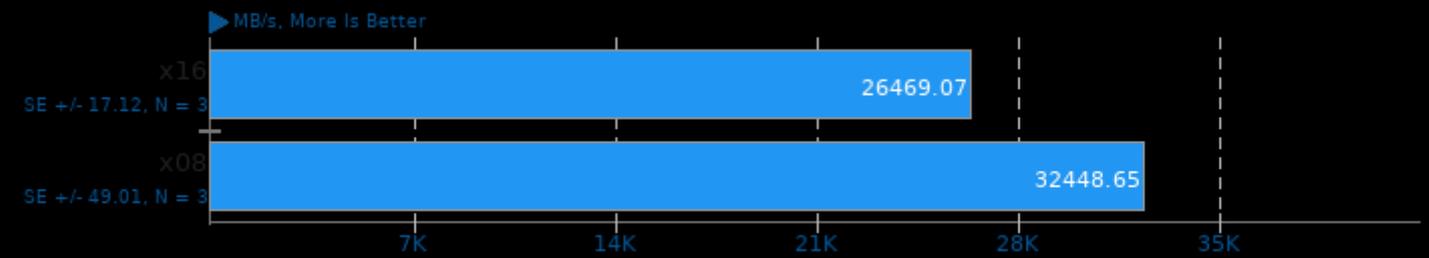
	Standard Deviation	0.2%	0.7%
ACES DGEMM - S.F.P.R (GFLOP/s)		3.407828	3.487483
	Normalized	97.72%	100%
	Standard Deviation	0.3%	0.8%
Timed Apache Compilation - Time To Compile (sec)		18.039	17.949
	Normalized	99.5%	100%
	Standard Deviation	0.6%	0.6%
Timed Clash Compilation - Time To Compile (sec)		2.174	2.160
	Normalized	99.36%	100%
	Standard Deviation	2.2%	2.4%
Timed FFmpeg Compilation - Time To Compile (sec)		56.401	56.608
	Normalized	100%	99.63%
	Standard Deviation	0.5%	0.7%
Timed GCC Compilation - Time To Compile (sec)		922.136	917.451
	Normalized	99.49%	100%
	Standard Deviation	0.4%	0.4%
Timed GDB GNU Debugger Compilation - Time To Compile (sec)		59.537	58.671
	Normalized	98.55%	100%
	Standard Deviation	0.3%	0.1%
Timed Godot Game Engine Compilation - Time To Compile (sec)		133.750	131.469
	Normalized	98.29%	100%
	Standard Deviation	0.3%	0.1%
Timed ImageMagick Compilation - Time To Compile		27.898	26.498
	Normalized	94.98%	100%
	Standard Deviation	0.9%	0.9%
Timed Linux Kernel Compilation - Time To Compile		89.092	89.336
	Normalized	100%	99.73%
	Standard Deviation	0.5%	0.6%
Timed LLVM Compilation - Ninja (sec)		719.732	696.733
	Normalized	96.8%	100%
	Standard Deviation	0.4%	0.2%
Timed LLVM Compilation - Unix Makefiles (sec)		738.988	719.136
	Normalized	97.31%	100%
	Standard Deviation	0.4%	0.5%
Timed Mesa Compilation - Time To Compile (sec)		50.010	48.470
	Normalized	96.92%	100%
	Standard Deviation	0.2%	0.4%
Timed MPlayer Compilation - Time To Compile (sec)		35.319	35.251
	Normalized	99.81%	100%
	Standard Deviation	0.5%	0%
Timed Node.js Compilation - Time To Compile (sec)		447.324	435.066
	Normalized	97.26%	100%
	Standard Deviation	0.2%	0.3%
Timed PHP Compilation - Time To Compile (sec)		53.992	53.299
	Normalized	98.72%	100%
	Standard Deviation	0.9%	1%
Build2 - Time To Compile (sec)		134.833	130.759
	Normalized	96.98%	100%
	Standard Deviation	0.9%	0.4%
Y-Cruncher - C.5.P.D (sec)		28.928	22.196
	Normalized	76.73%	100%
	Standard Deviation	0.1%	0.6%
Timed Eigen Compilation - Time To Compile (sec)		55.460	55.418

	Normalized	99.92%	100%
	Standard Deviation	0.4%	0.3%
Timed Erlang/OTP Compilation - Time To Compile		95.871	95.712
	Normalized	99.83%	100%
	Standard Deviation	0%	0.3%
Timed Wasmer Compilation - Time To Compile (sec)		59.297	58.977
	Normalized	99.46%	100%
	Standard Deviation	1.1%	0.6%
Node.js V8 Web Tooling Benchmark (runs/s)		16.13	16.42
	Normalized	98.23%	100%
	Standard Deviation	1%	1%
Cryptsetup - PBKDF2-sha512 (Iterations/sec)		2023037	2036075
	Normalized	99.36%	100%
	Standard Deviation	0.7%	0.2%
Cryptsetup - PBKDF2-whirlpool (Iterations/sec)		827056	833968
	Normalized	99.17%	100%
	Standard Deviation	1.4%	0.1%
Cryptsetup - A.X.2.E (MiB/s)		5517	5477
	Normalized	100%	99.27%
	Standard Deviation	0.7%	0.9%
Cryptsetup - A.X.2.D (MiB/s)		5547	5478
	Normalized	100%	98.75%
	Standard Deviation	0.2%	0.7%
Cryptsetup - S.X.2.E (MiB/s)		756.5	754.2
	Normalized	100%	99.7%
	Standard Deviation	0.9%	1.1%
Cryptsetup - S.X.2.D (MiB/s)		712.3	709.5
	Normalized	100%	99.61%
	Standard Deviation	0.4%	1.2%
Cryptsetup - T.X.2.E (MiB/s)		476.5	472.6
	Normalized	100%	99.18%
	Standard Deviation	0.7%	0.6%
Cryptsetup - T.X.2.D (MiB/s)		474.8	472.0
	Normalized	100%	99.41%
	Standard Deviation	0.5%	0.8%
Cryptsetup - A.X.5.E (MiB/s)		4884	4863
	Normalized	100%	99.58%
	Standard Deviation	0.6%	1%
Cryptsetup - A.X.5.D (MiB/s)		4878	4838
	Normalized	100%	99.18%
	Standard Deviation	0.8%	0.9%
Cryptsetup - S.X.5.E (MiB/s)		760.2	755.5
	Normalized	100%	99.38%
	Standard Deviation	0.1%	1.2%
Cryptsetup - S.X.5.D (MiB/s)		714.4	710.1
	Normalized	100%	99.4%
	Standard Deviation	0.1%	1.2%
Cryptsetup - T.X.5.E (MiB/s)		477.4	475.8
	Normalized	100%	99.66%
	Standard Deviation	0.4%	0.1%
Cryptsetup - T.X.5.D (MiB/s)		475.4	472.4
	Normalized	100%	99.37%
	Standard Deviation	0.3%	0.9%
SQLite Speedtest - Timed Time - Size 1,000 (sec)		47.052	47.063
	Normalized	100%	99.98%

	Standard Deviation	1.5%	0.2%
PyBench - T.F.A.T.T (Milliseconds)		720	714
	Normalized	99.17%	100%
	Standard Deviation	0.4%	0.1%
PyPerformance - go (Milliseconds)		189	186
	Normalized	98.41%	100%
PyPerformance - 2to3 (Milliseconds)		237	238
	Normalized	100%	99.58%
PyPerformance - chaos (Milliseconds)		77.3	76.8
	Normalized	99.35%	100%
	Standard Deviation	0.7%	0.3%
PyPerformance - float (Milliseconds)		79.8	79.9
	Normalized	100%	99.87%
	Standard Deviation	0.2%	0.2%
PyPerformance - nbody (Milliseconds)		83.6	82.9
	Normalized	99.16%	100%
	Standard Deviation	0.1%	0.6%
PyPerformance - pathlib (Milliseconds)		13.6	13.5
	Normalized	99.26%	100%
	Standard Deviation	0.7%	1.1%
PyPerformance - raytrace (Milliseconds)		354	354
	Standard Deviation		0.7%
PyPerformance - json_loads (Milliseconds)		17.2	17.1
	Normalized	99.42%	100%
	Standard Deviation	0.3%	1.2%
PyPerformance - crypto_pyaes (Milliseconds)		78.1	77.8
	Normalized	99.62%	100%
	Standard Deviation	0.7%	0.2%
PyPerformance - regex_compile (Milliseconds)		126	125
	Normalized	99.21%	100%
	Standard Deviation	0.9%	0.5%
PyPerformance - python_startup (Milliseconds)		6.37	6.37
	Standard Deviation	0.6%	0.2%
PyPerformance - django_template (Milliseconds)		35.9	35.6
	Normalized	99.16%	100%
	Standard Deviation	0.2%	0.3%
PyPerformance - pickle_pure_python (Milliseconds)		317	315
	Normalized	99.37%	100%
	Standard Deviation	0.2%	0.5%
Git - T.T.C.C.G.C (sec)		44.351	44.892
	Normalized	100%	98.79%
	Standard Deviation	0.3%	1.1%

RAMspeed SMP 3.5.0

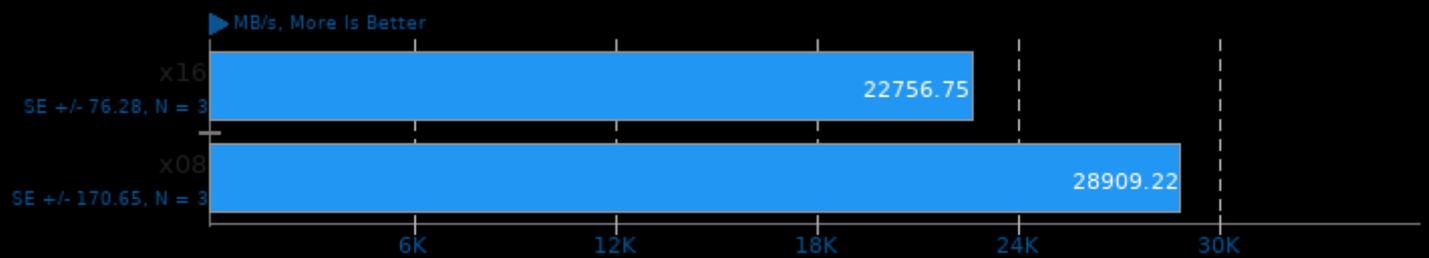
Type: Add - Benchmark: Integer



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

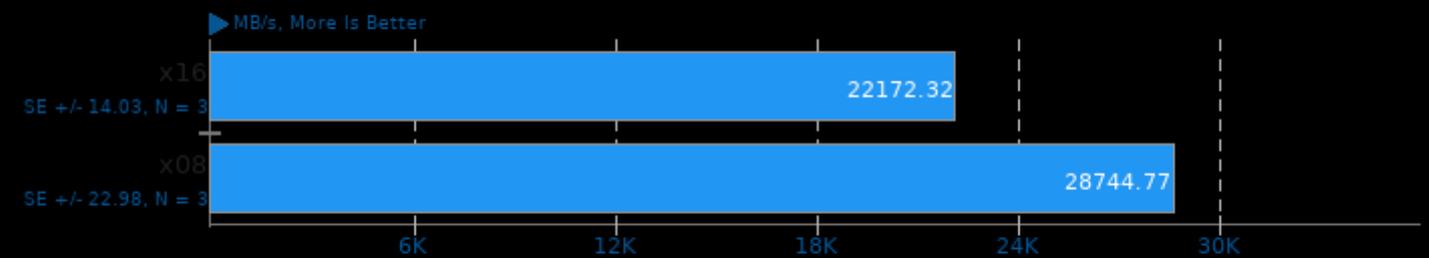
Type: Copy - Benchmark: Integer



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

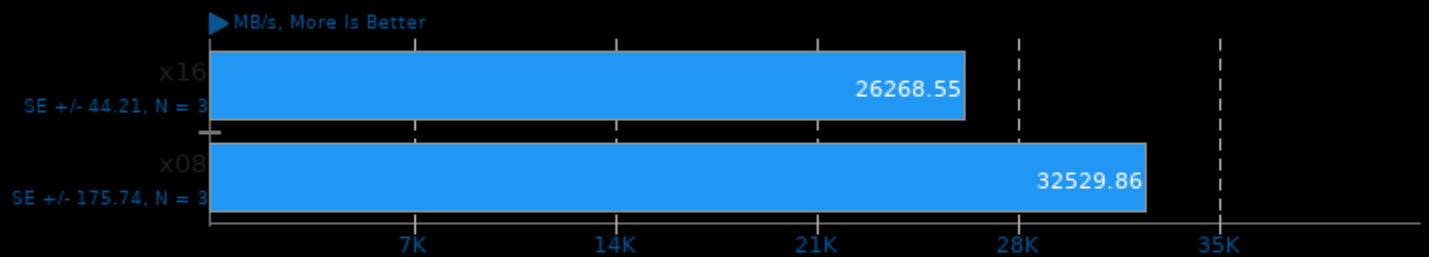
Type: Scale - Benchmark: Integer



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

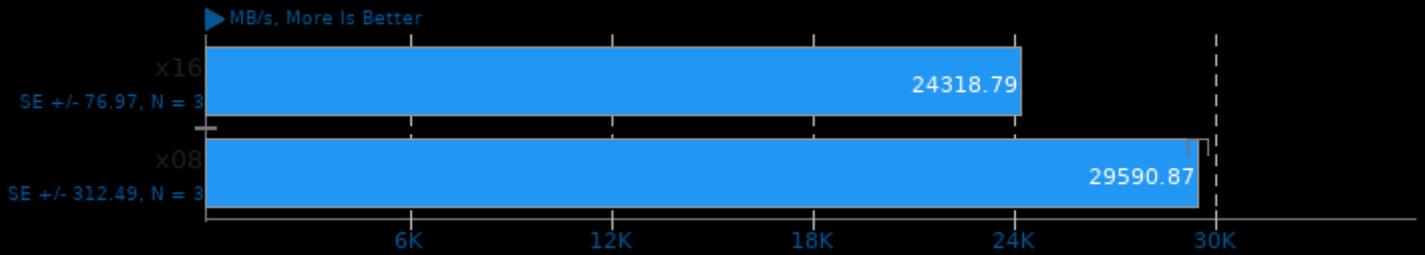
Type: Triad - Benchmark: Integer



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

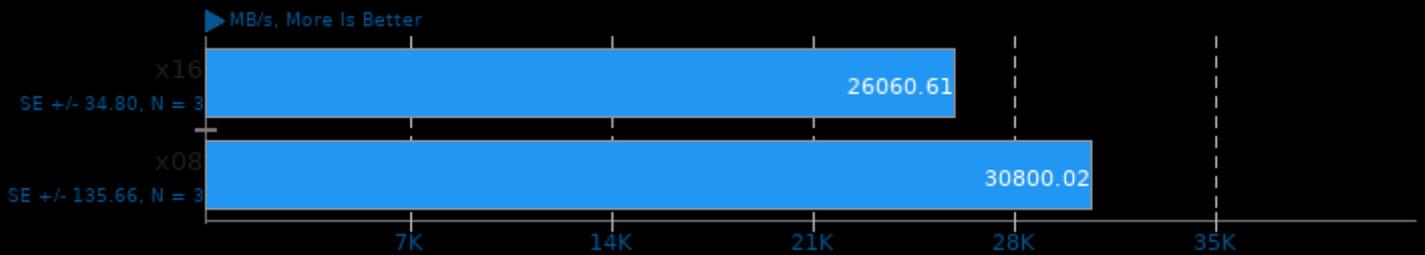
Type: Average - Benchmark: Integer



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

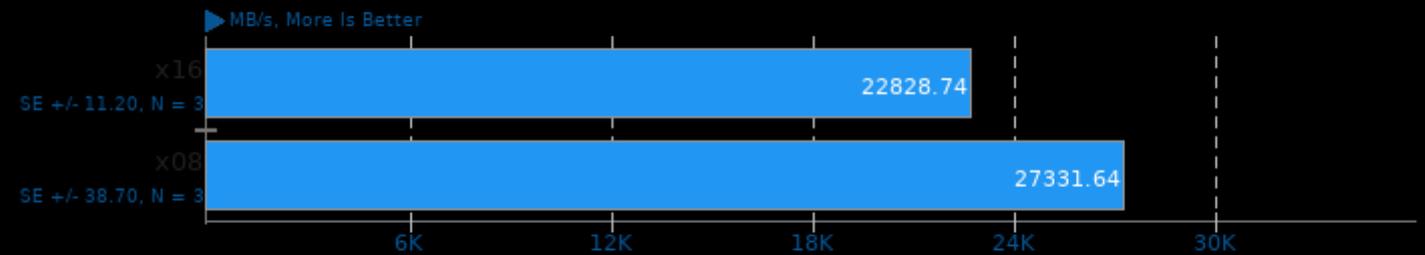
Type: Add - Benchmark: Floating Point



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

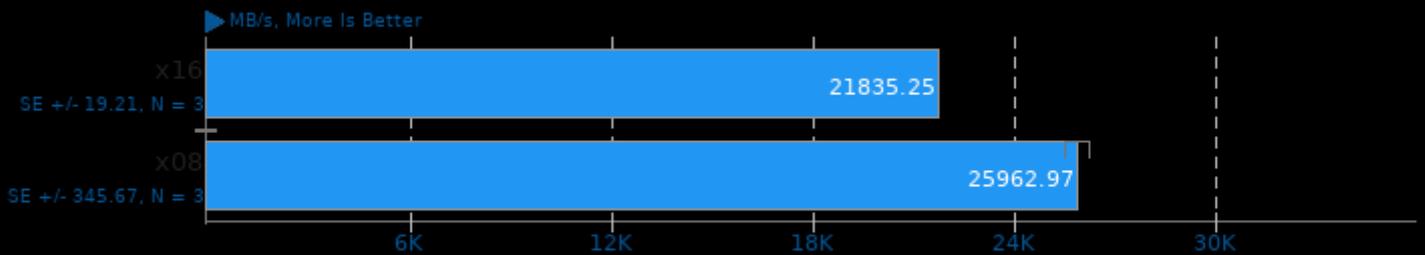
Type: Copy - Benchmark: Floating Point



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

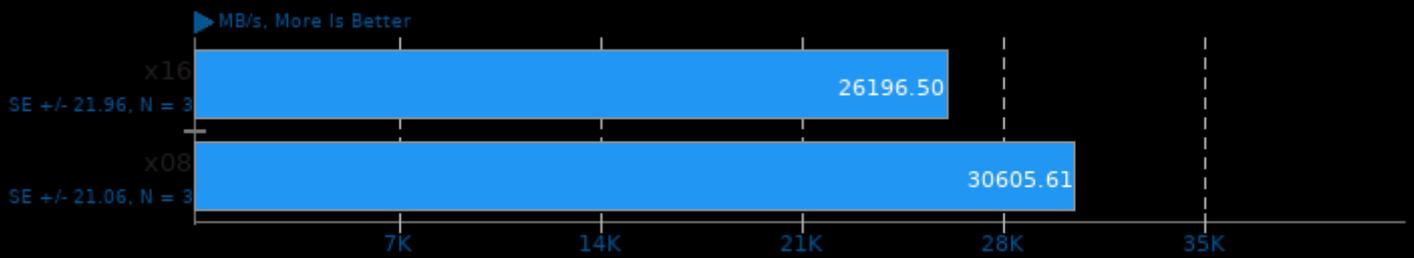
Type: Scale - Benchmark: Floating Point



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

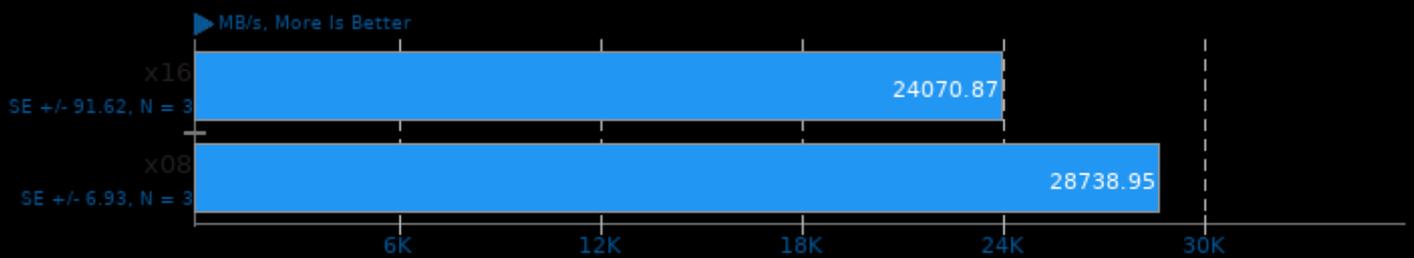
Type: Triad - Benchmark: Floating Point



1. (CC) gcc options: -O3 -march=native

RAMspeed SMP 3.5.0

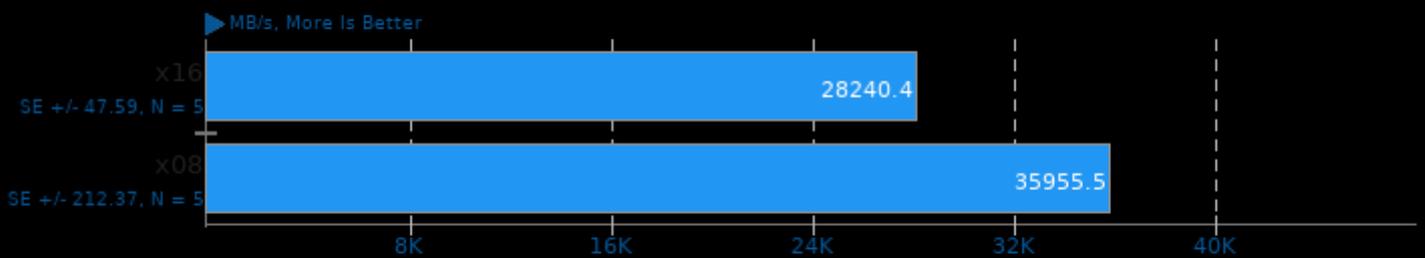
Type: Average - Benchmark: Floating Point



1. (CC) gcc options: -O3 -march=native

Stream 2013-01-17

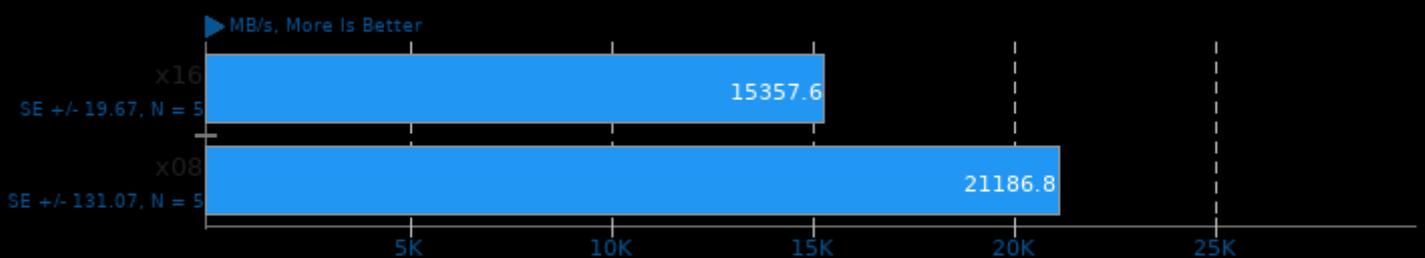
Type: Copy



1. (CC) gcc options: -O3 -march=native -fopenmp

Stream 2013-01-17

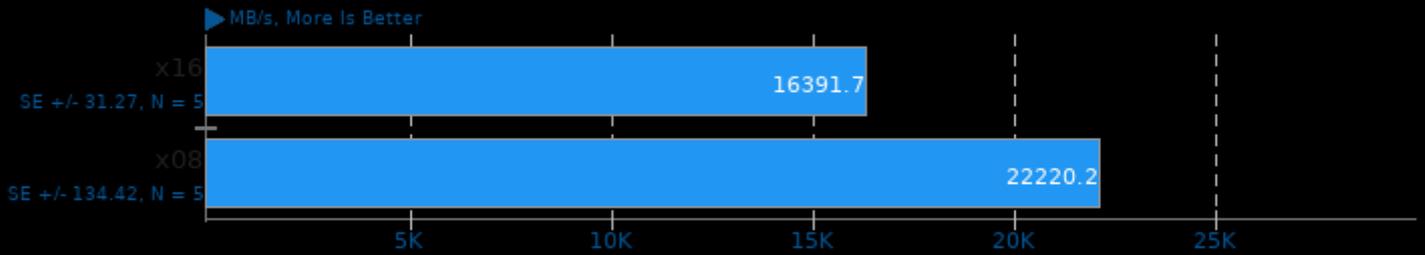
Type: Scale



1. (CC) gcc options: -O3 -march=native -fopenmp

Stream 2013-01-17

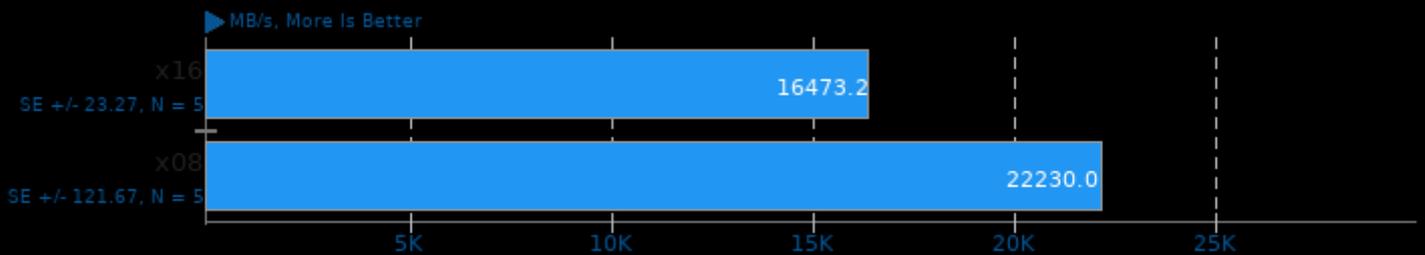
Type: Triad



1. (CC) gcc options: -O3 -march=native -fopenmp

Stream 2013-01-17

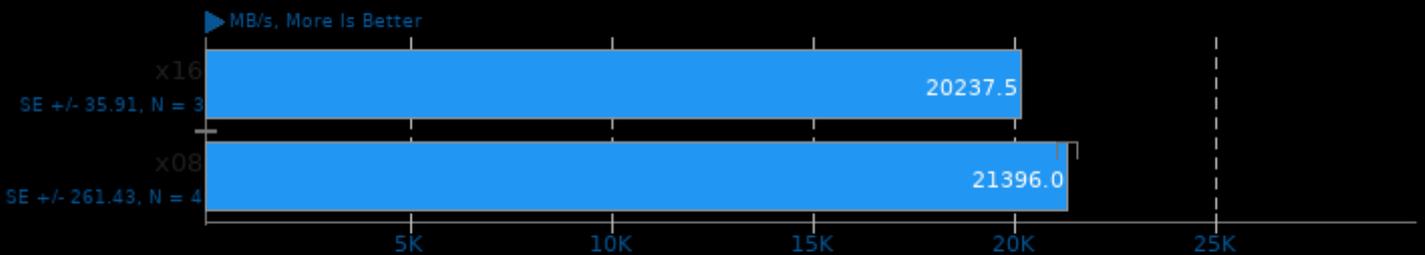
Type: Add



1. (CC) gcc options: -O3 -march=native -fopenmp

Tinymembench 2018-05-28

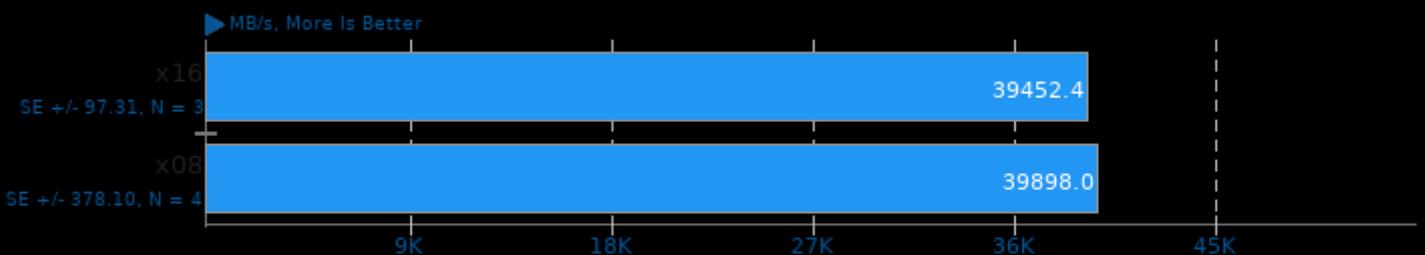
Standard Memcpy



1. (CC) gcc options: -O2 -lm

Tinymembench 2018-05-28

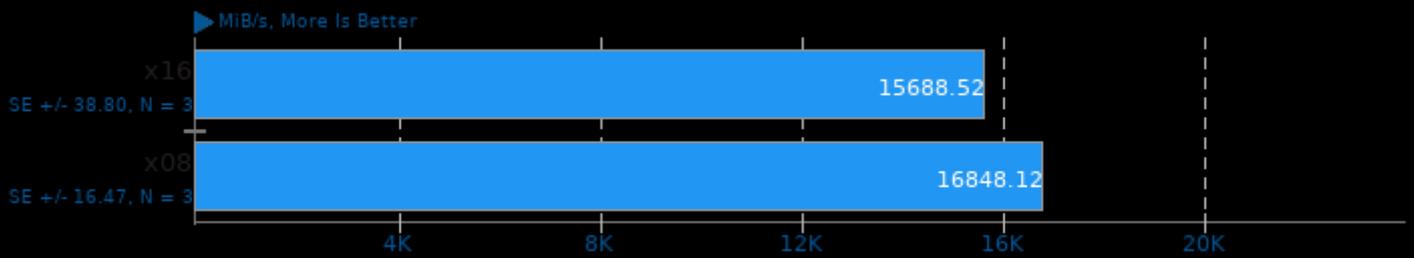
Standard Memset



1. (CC) gcc options: -O2 -lm

MBW 2018-09-08

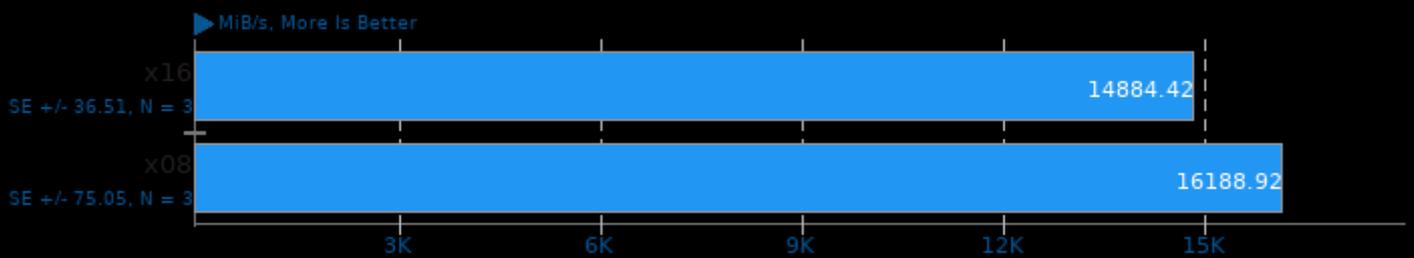
Test: Memory Copy - Array Size: 128 MiB



1. (CC) gcc options: -O3 -march=native

MBW 2018-09-08

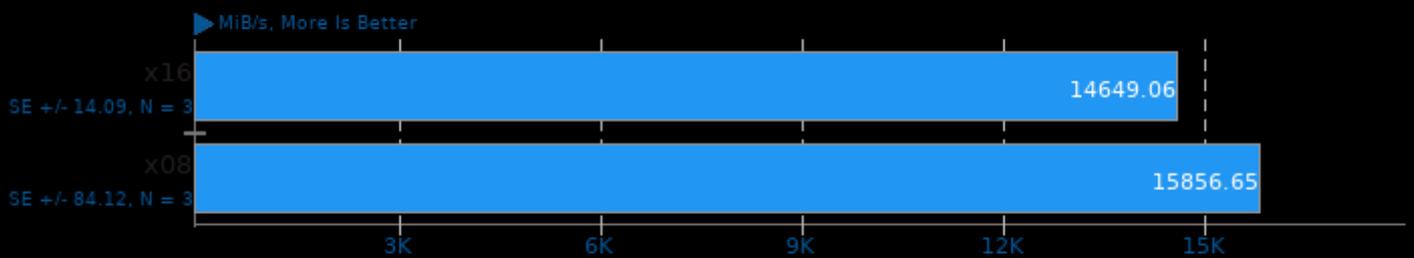
Test: Memory Copy - Array Size: 512 MiB



1. (CC) gcc options: -O3 -march=native

MBW 2018-09-08

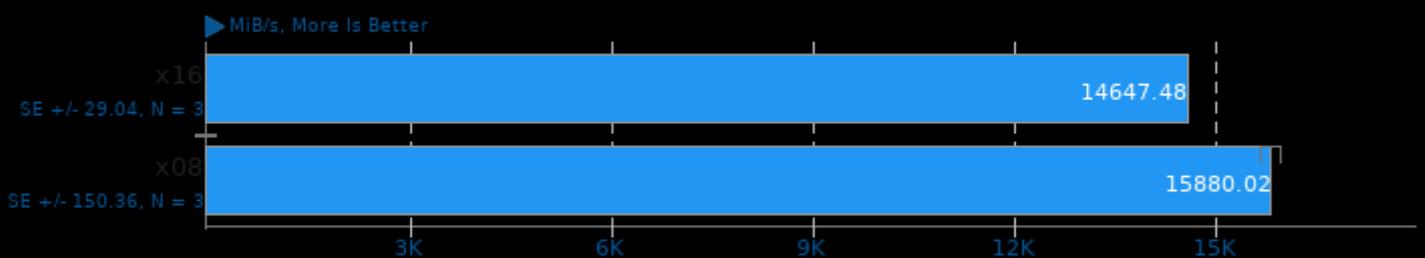
Test: Memory Copy - Array Size: 1024 MiB



1. (CC) gcc options: -O3 -march=native

MBW 2018-09-08

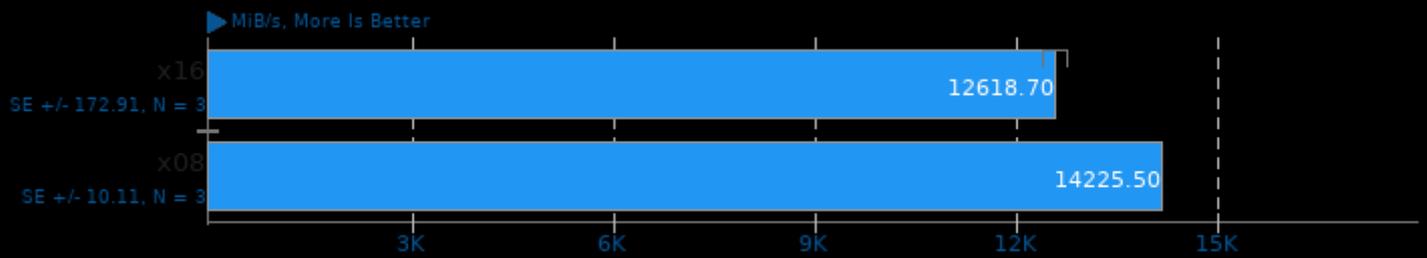
Test: Memory Copy - Array Size: 4096 MiB



1. (CC) gcc options: -O3 -march=native

MBW 2018-09-08

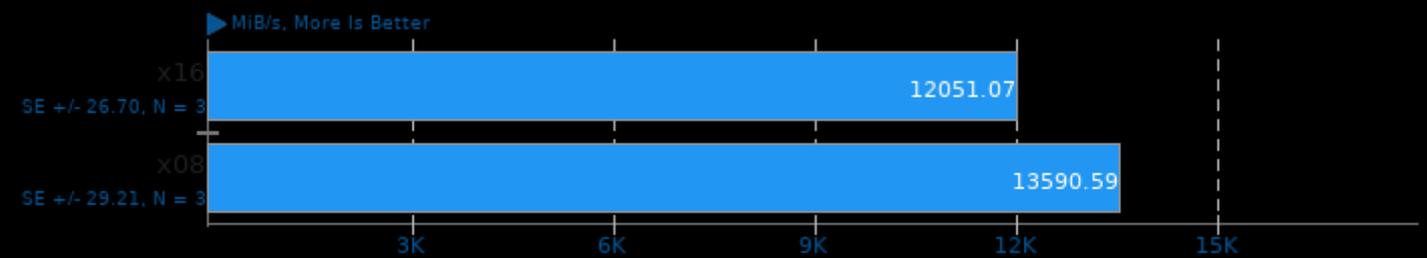
Test: Memory Copy, Fixed Block Size - Array Size: 128 MiB



1. (CC) gcc options: -O3 -march=native

MBW 2018-09-08

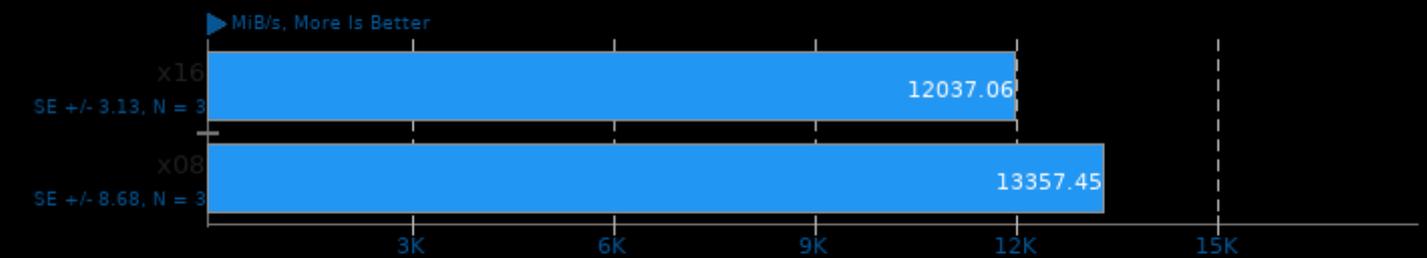
Test: Memory Copy, Fixed Block Size - Array Size: 512 MiB



1. (CC) gcc options: -O3 -march=native

MBW 2018-09-08

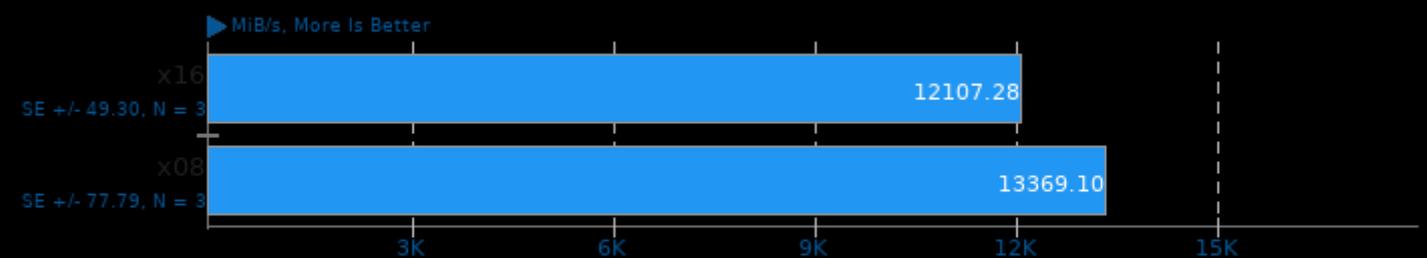
Test: Memory Copy, Fixed Block Size - Array Size: 1024 MiB



1. (CC) gcc options: -O3 -march=native

MBW 2018-09-08

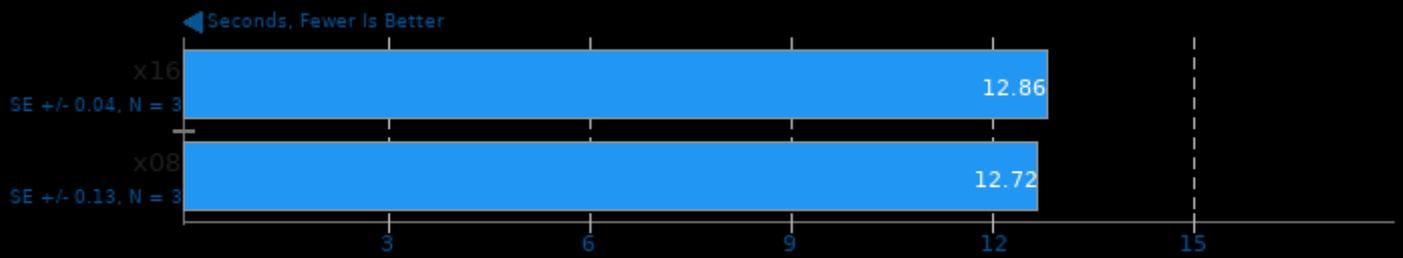
Test: Memory Copy, Fixed Block Size - Array Size: 4096 MiB



1. (CC) gcc options: -O3 -march=native

t-test1 2017-01-13

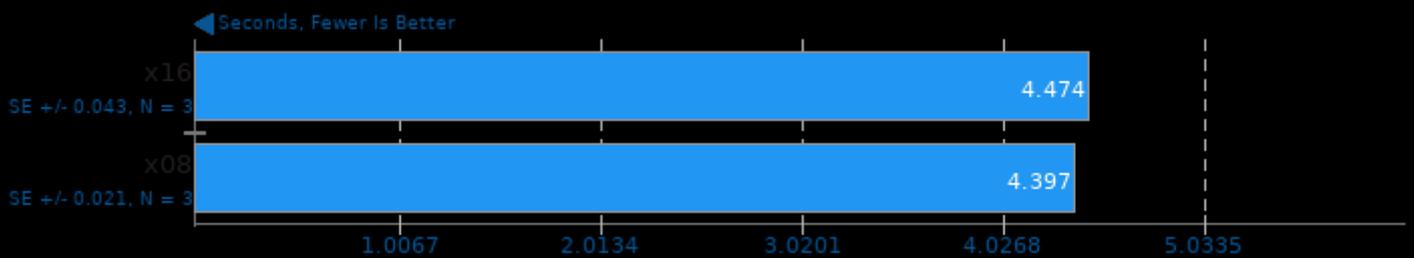
Threads: 1



1. (CC) gcc options: -pthread

t-test1 2017-01-13

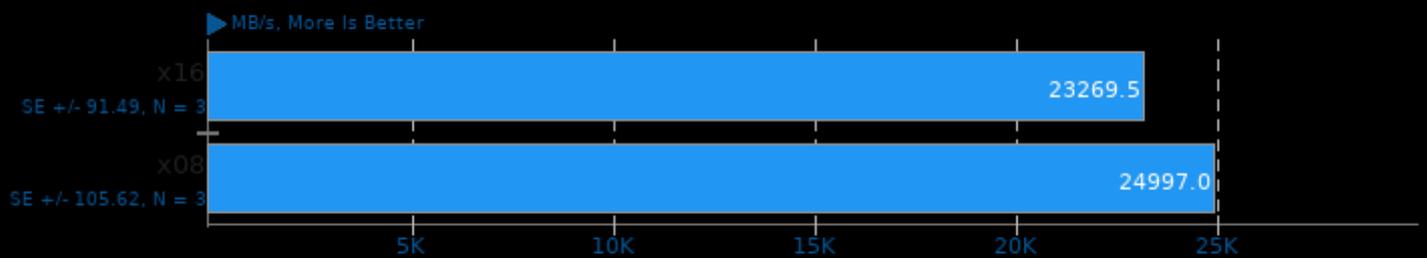
Threads: 2



1. (CC) gcc options: -pthread

C-Blosc 2.0

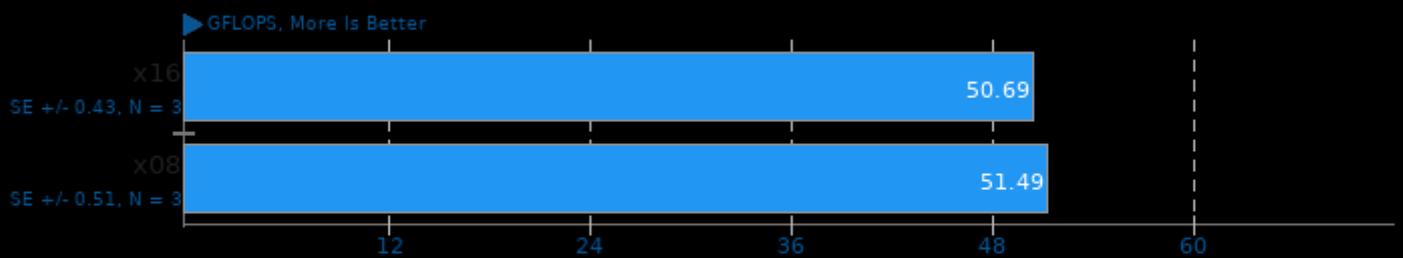
Compressor: blosclz



1. (CC) gcc options: -std=gnu99 -O3 -pthread -lrt -lm

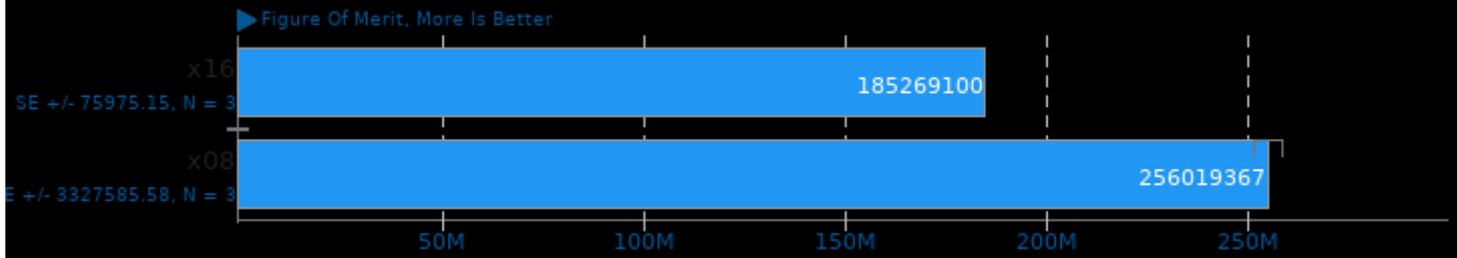
HPC Challenge 1.5.0

Test / Class: G-HPL



1. (CC) gcc options: -lblas -lm -pthread -lmpi -fomit-frame-pointer -funroll-loops
2. ATLAS + Open MPI 4.0.3

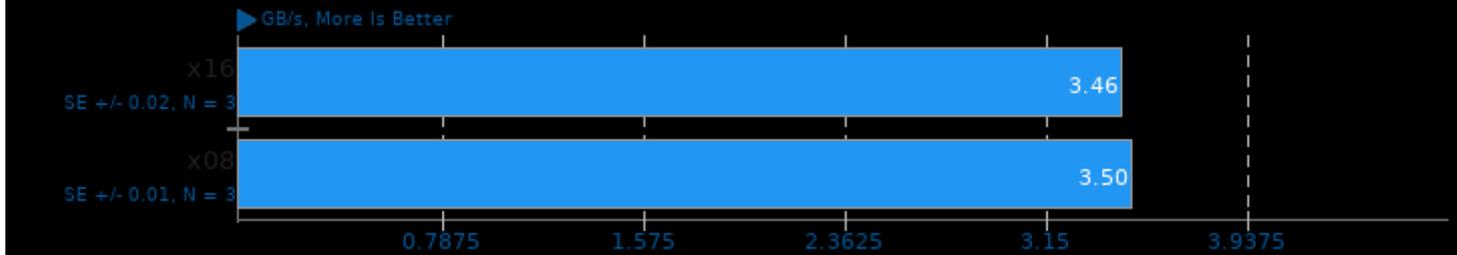
Algebraic Multi-Grid Benchmark 1.2



1. (CC) gcc options: -lparcsr_ls -lparcsr_mv -lseq_mv -llj_mv -lkrylov -lHYPRE_utilities -lm -fopenmp -pthread -lmpi

simdjson 1.0

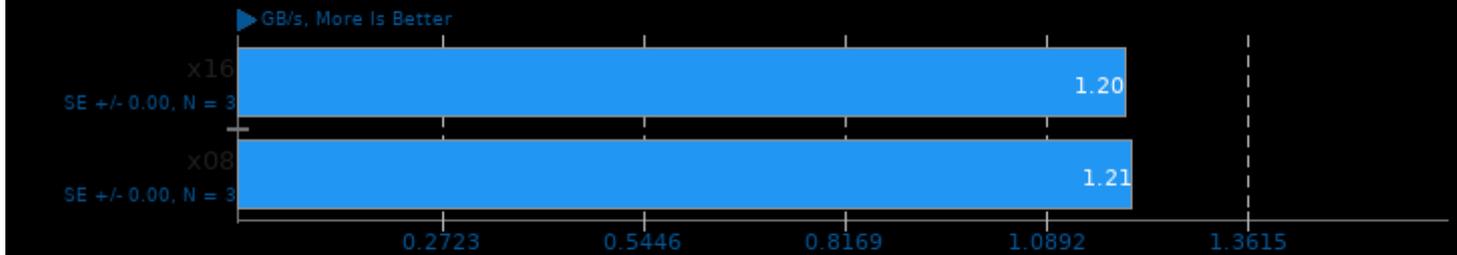
Throughput Test: Kostya



1. (CXX) g++ options: -O3 -pthread

simdjson 1.0

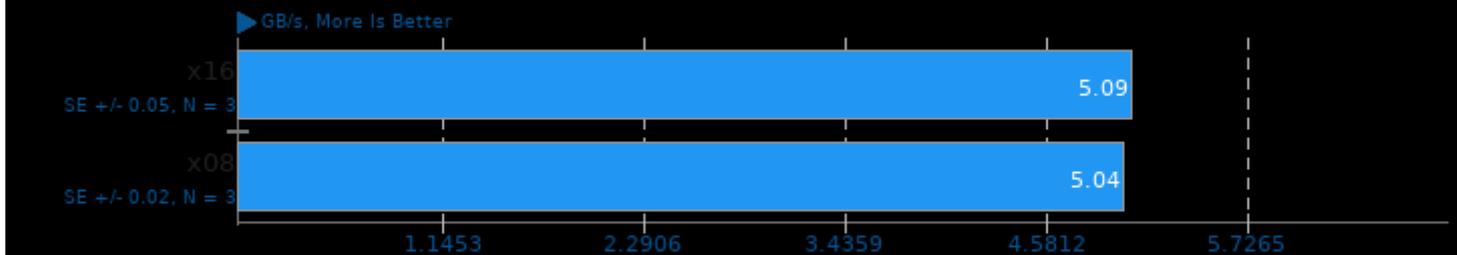
Throughput Test: LargeRandom



1. (CXX) g++ options: -O3 -pthread

simdjson 1.0

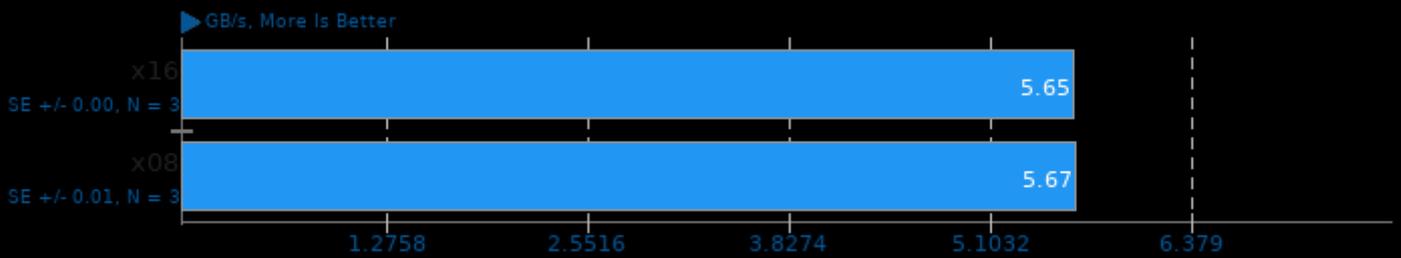
Throughput Test: PartialTweets



1. (CXX) g++ options: -O3 -pthread

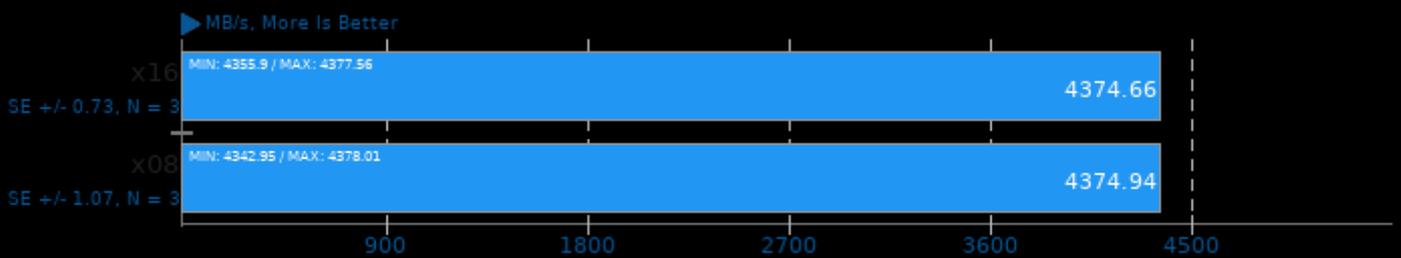
simdjson 1.0

Throughput Test: DistinctUserID



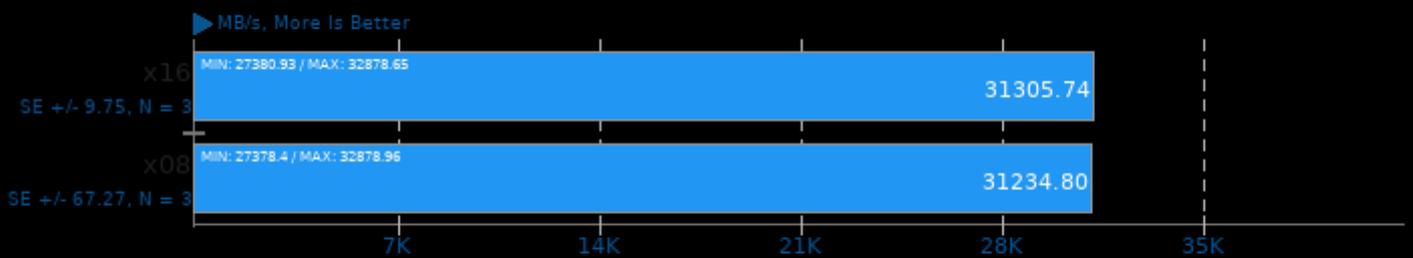
CacheBench

Test: Read



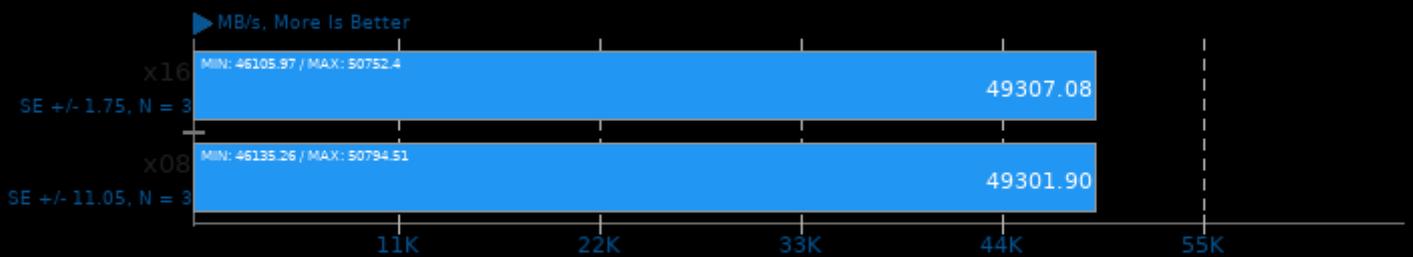
CacheBench

Test: Write



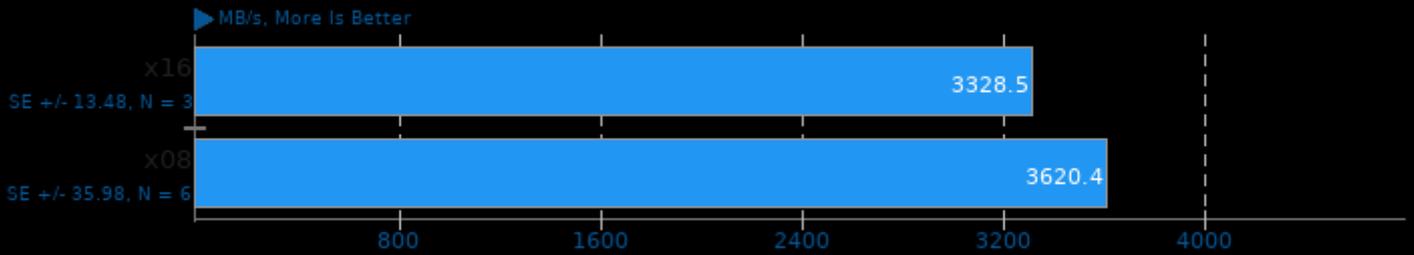
CacheBench

Test: Read / Modify / Write



Zstd Compression 1.5.0

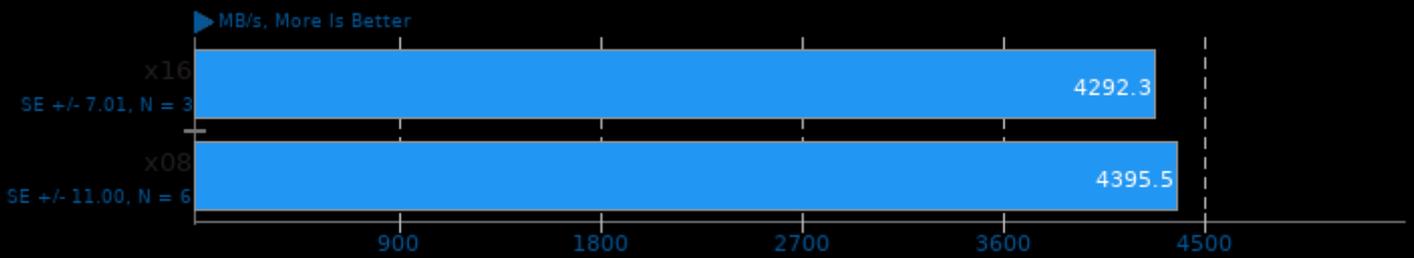
Compression Level: 3 - Compression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

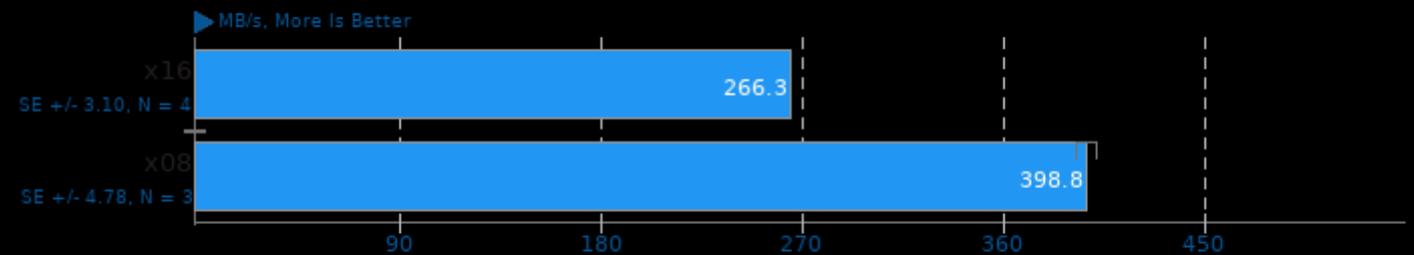
Compression Level: 3 - Decompression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

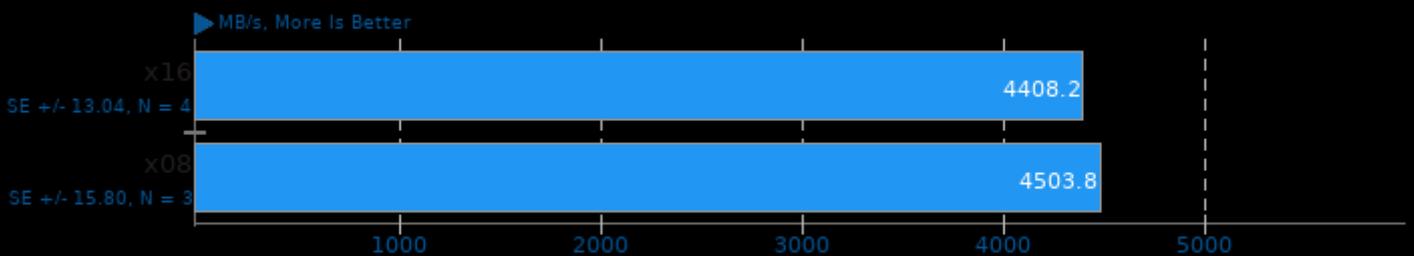
Compression Level: 8 - Compression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

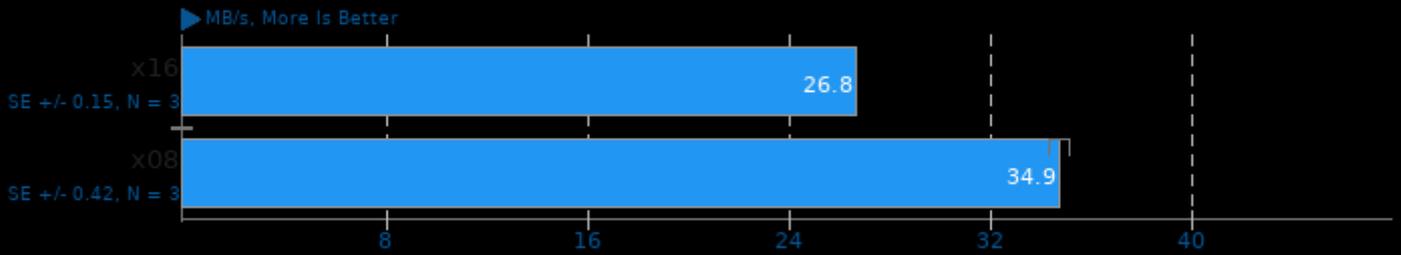
Compression Level: 8 - Decompression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

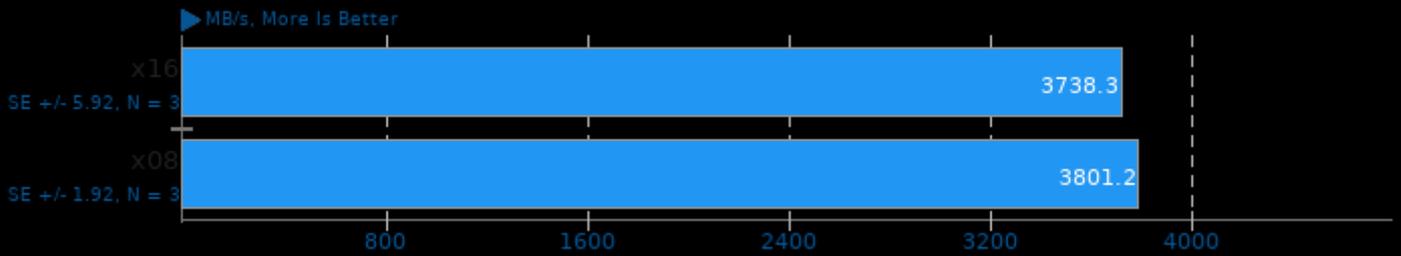
Compression Level: 19 - Compression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

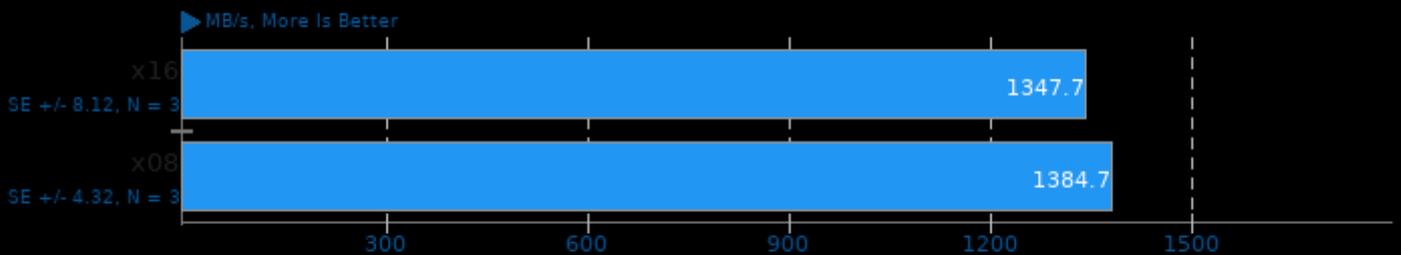
Compression Level: 19 - Decompression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

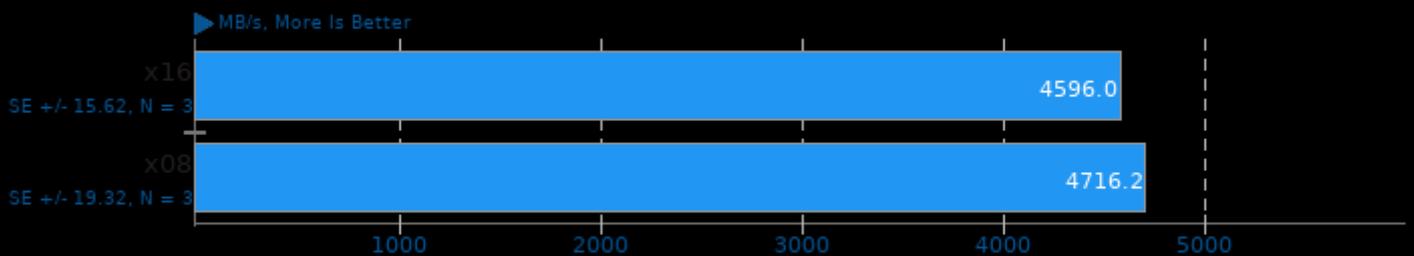
Compression Level: 3, Long Mode - Compression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

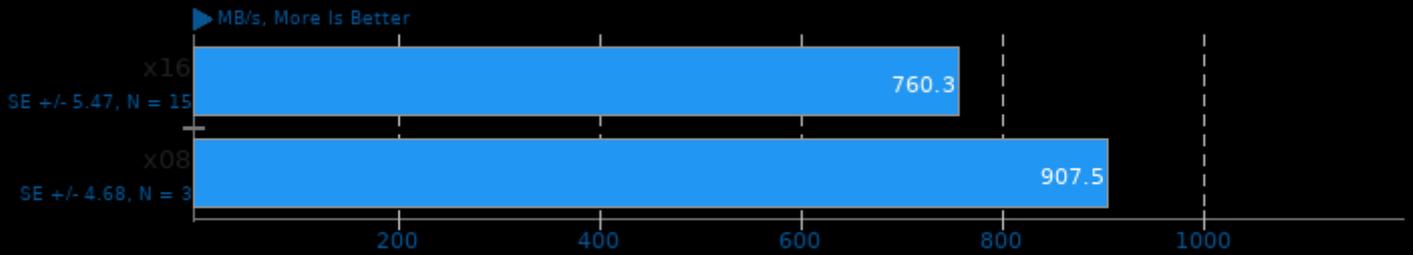
Compression Level: 3, Long Mode - Decompression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

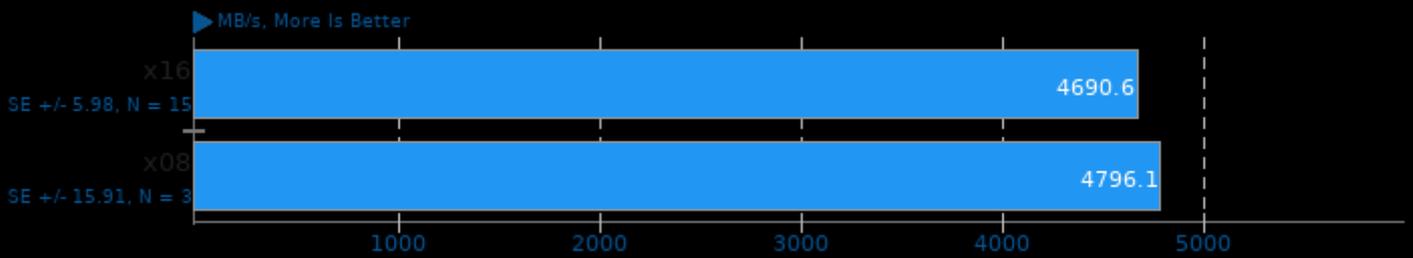
Compression Level: 8, Long Mode - Compression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

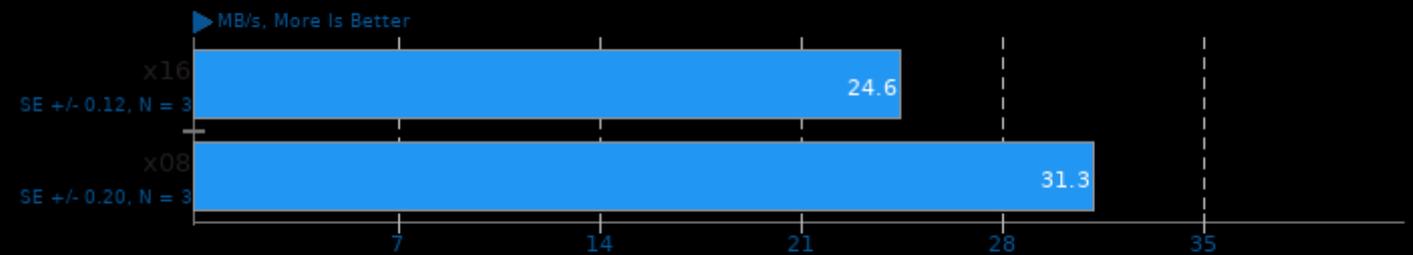
Compression Level: 8, Long Mode - Decompression Speed



1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

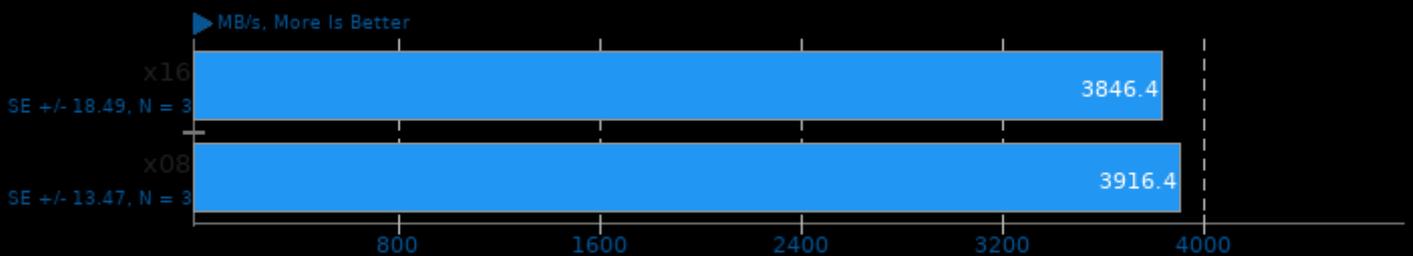
Compression Level: 19, Long Mode - Compression Speed



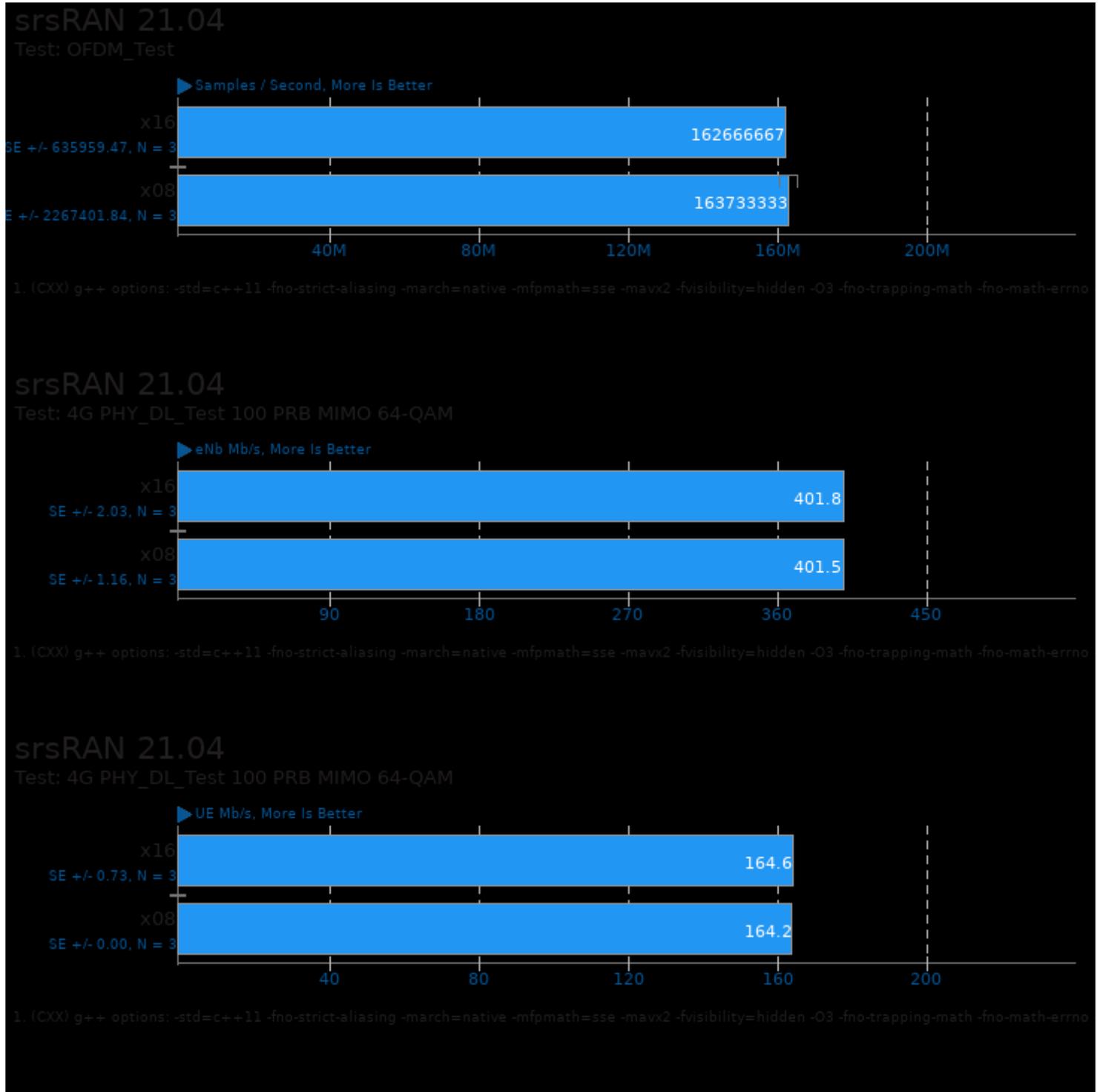
1. (CC) gcc options: -O3 -pthread -lz -llzma

Zstd Compression 1.5.0

Compression Level: 19, Long Mode - Decompression Speed

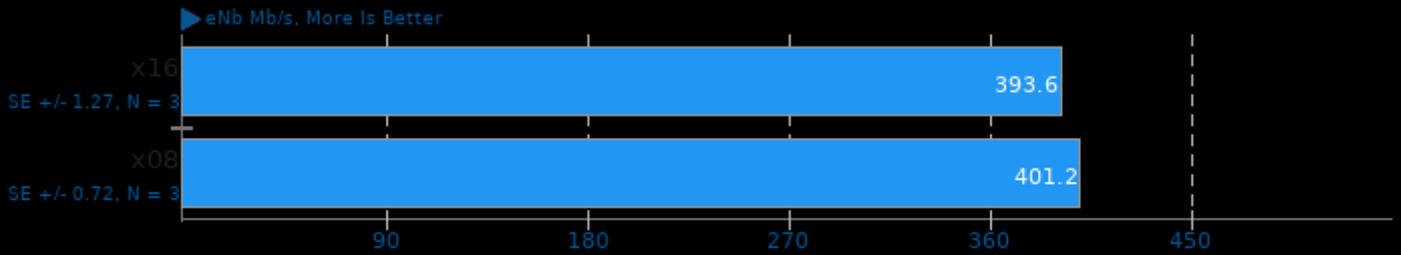


1. (CC) gcc options: -O3 -pthread -lz -llzma



srsRAN 21.04

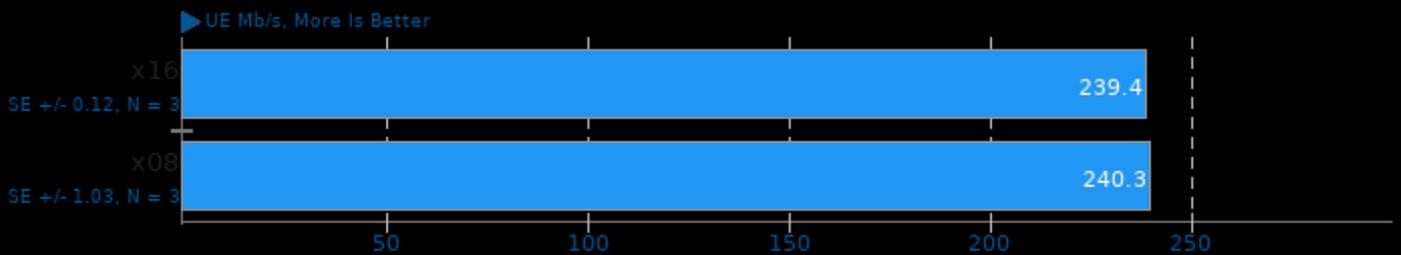
Test: 4G PHY_DL_Test 100 PRB SISO 64-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

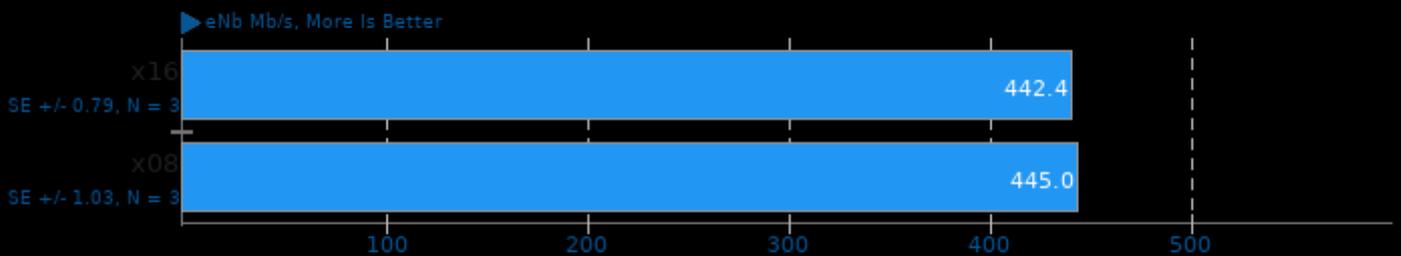
Test: 4G PHY_DL_Test 100 PRB SISO 64-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

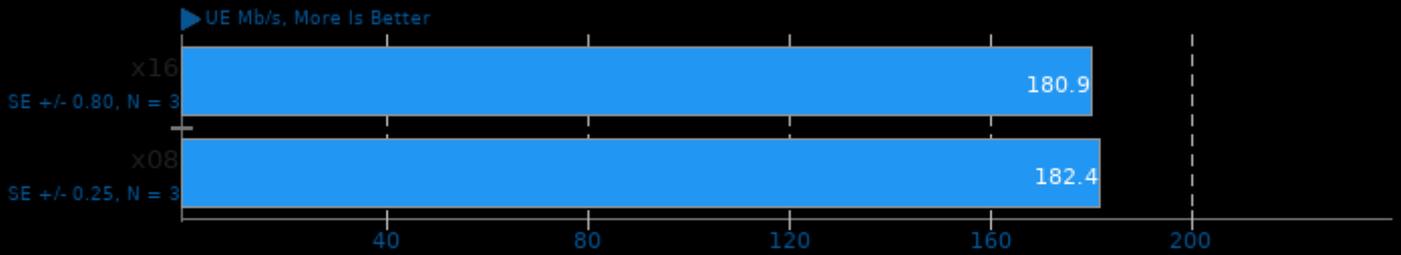
Test: 4G PHY_DL_Test 100 PRB MIMO 256-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

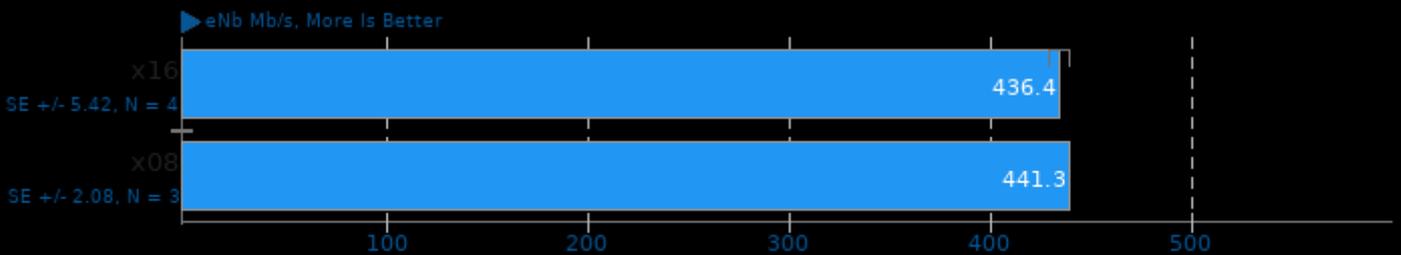
Test: 4G PHY_DL_Test 100 PRB MIMO 256-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

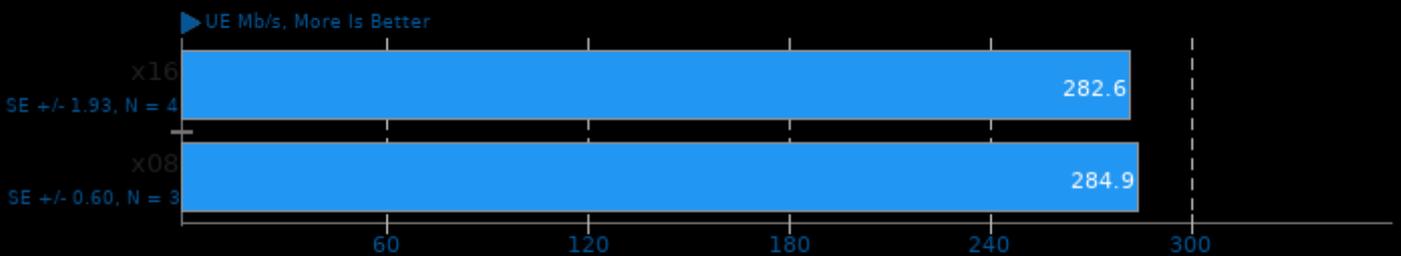
Test: 4G PHY_DL_Test 100 PRB SISO 256-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

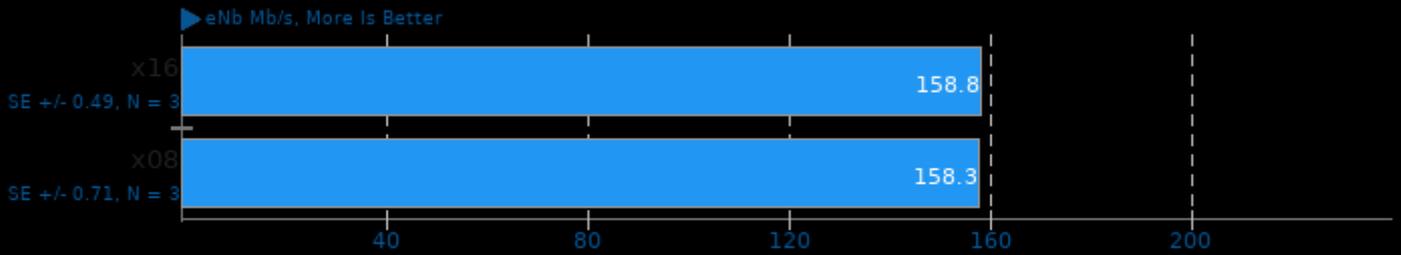
Test: 4G PHY_DL_Test 100 PRB SISO 256-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

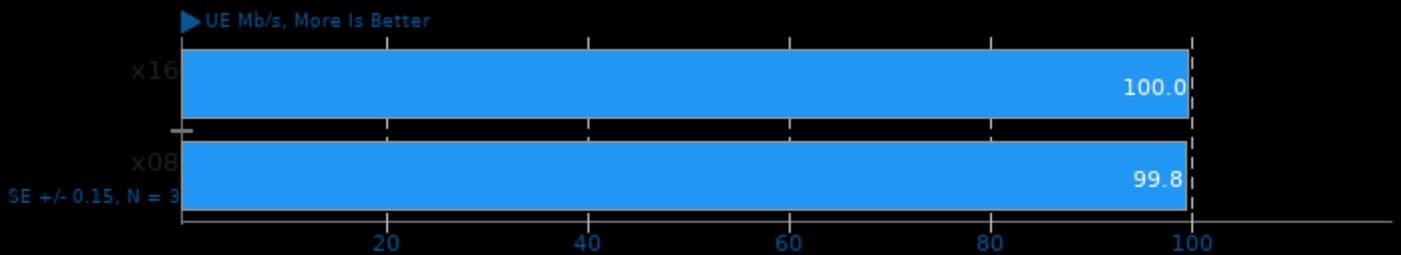
Test: 5G PHY_DL_NR Test 52 PRB SISO 64-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

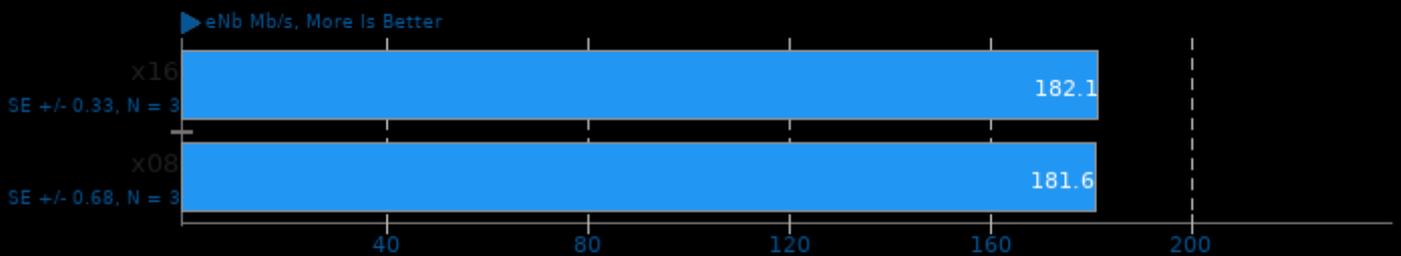
Test: 5G PHY_DL_NR Test 52 PRB SISO 64-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

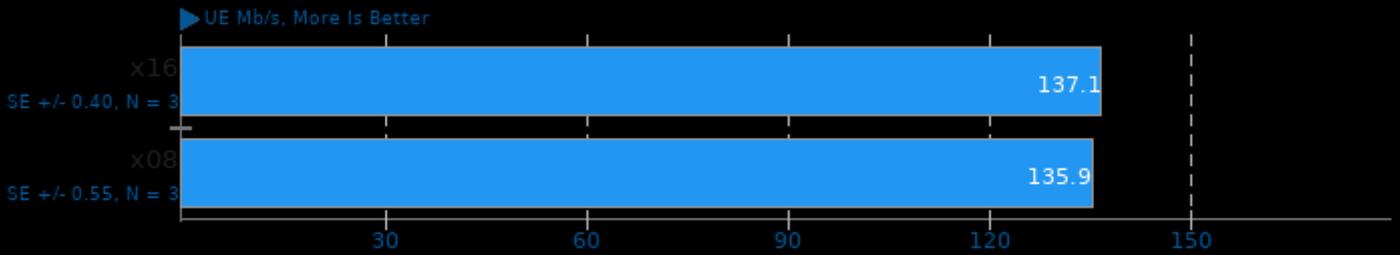
Test: 5G PHY_DL_NR Test 270 PRB SISO 256-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

srsRAN 21.04

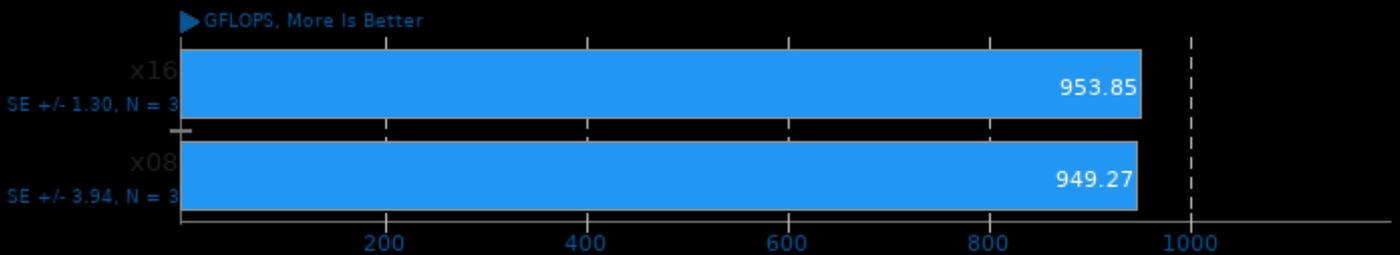
Test: 5G PHY_DL_NR Test 270 PRB SISO 256-QAM



1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

ArrayFire 3.7

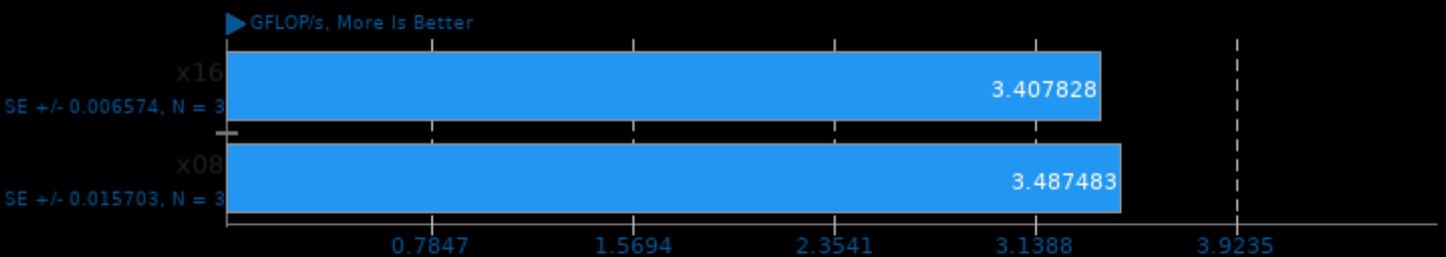
Test: BLAS CPU



1. (CXX) g++ options: -rdynamic

ACES DGEMM 1.0

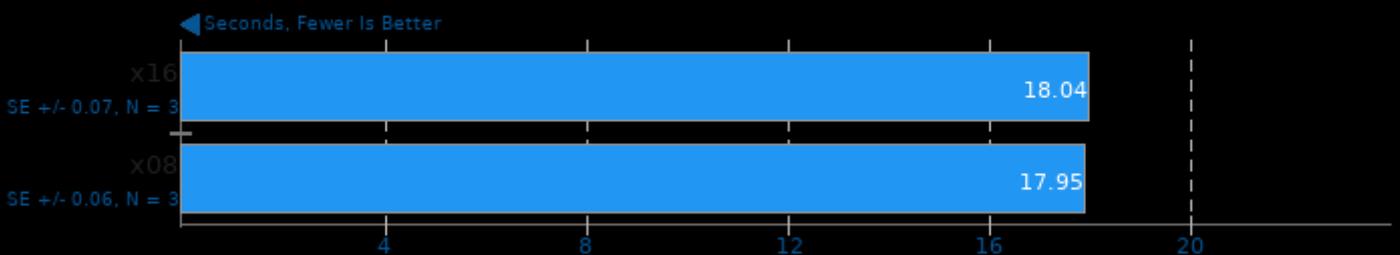
Sustained Floating-Point Rate



1. (CC) gcc options: -O3 -march=native -fopenmp

Timed Apache Compilation 2.4.41

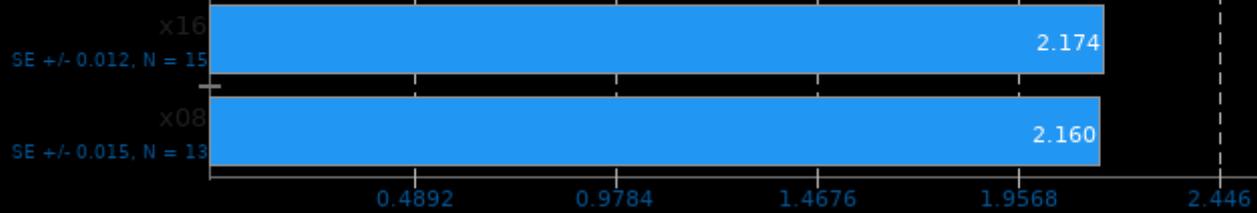
Time To Compile



Timed Clash Compilation

Time To Compile

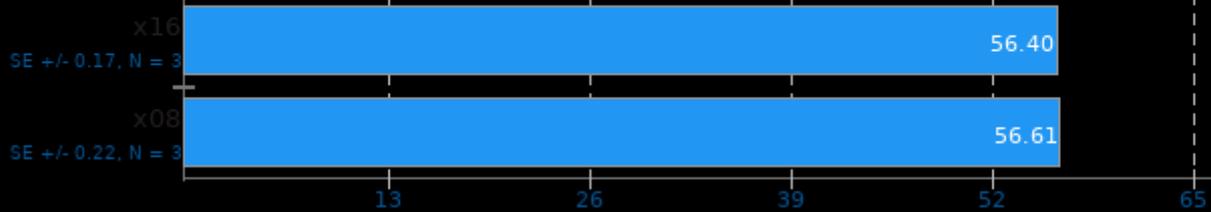
Seconds, Fewer Is Better



Timed FFmpeg Compilation 4.4

Time To Compile

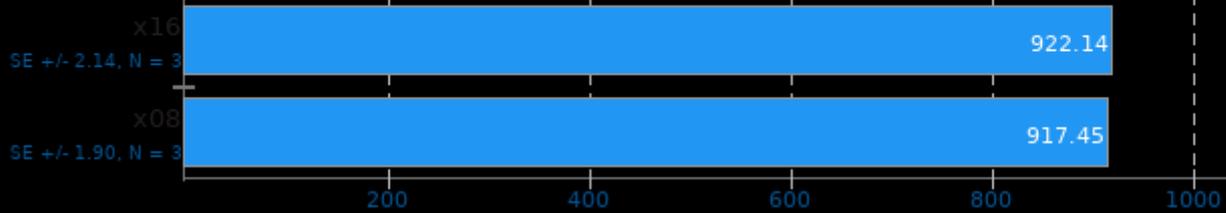
Seconds, Fewer Is Better



Timed GCC Compilation 11.2.0

Time To Compile

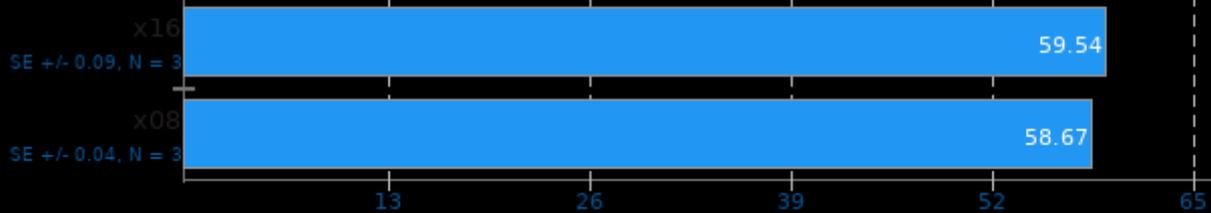
Seconds, Fewer Is Better



Timed GDB GNU Debugger Compilation 10.2

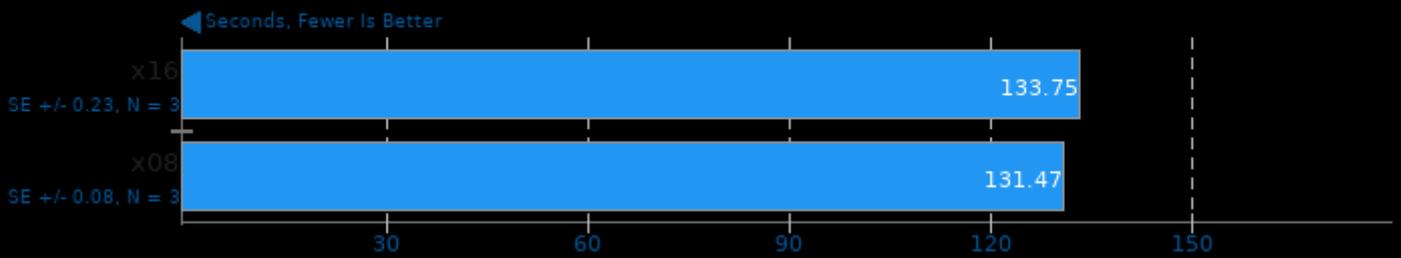
Time To Compile

Seconds, Fewer Is Better



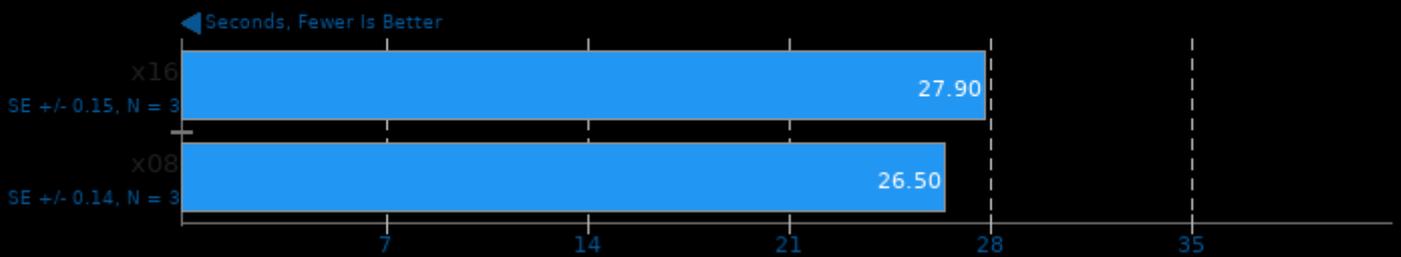
Timed Godot Game Engine Compilation 3.2.3

Time To Compile



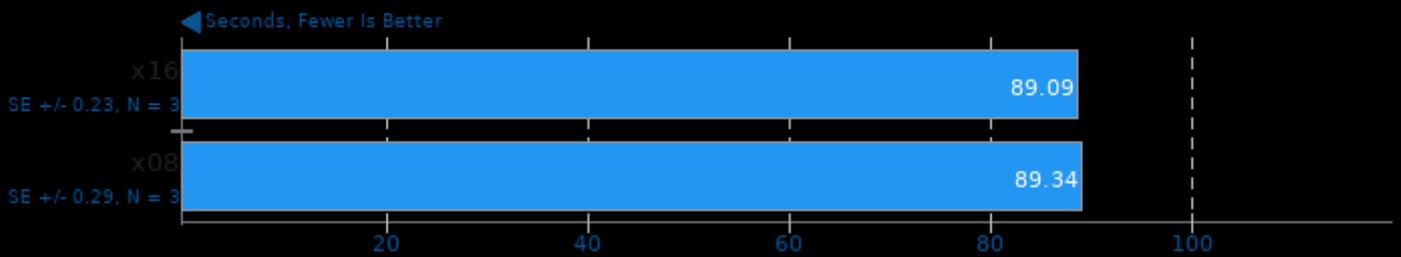
Timed ImageMagick Compilation 6.9.0

Time To Compile



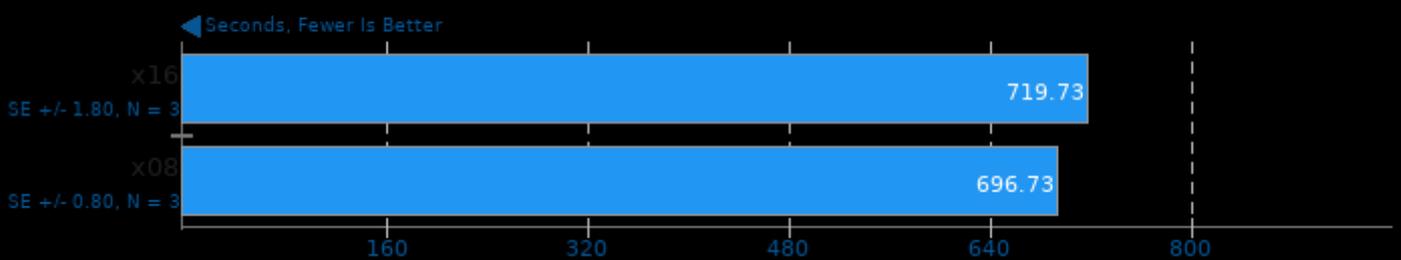
Timed Linux Kernel Compilation 5.14

Time To Compile



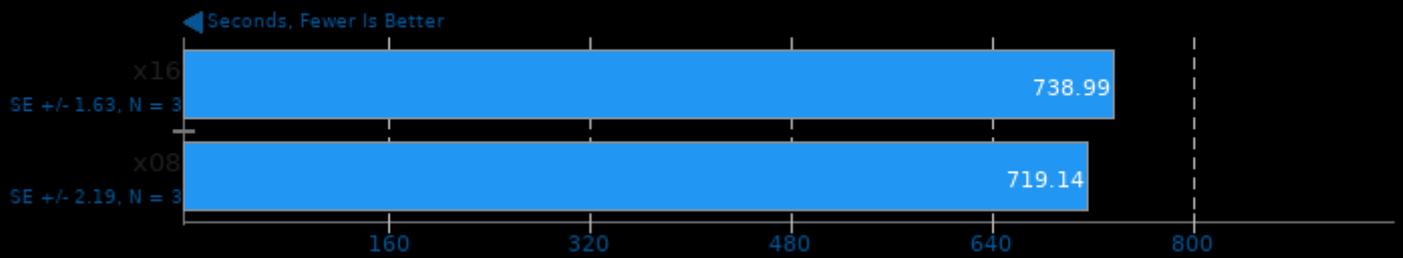
Timed LLVM Compilation 13.0

Build System: Ninja



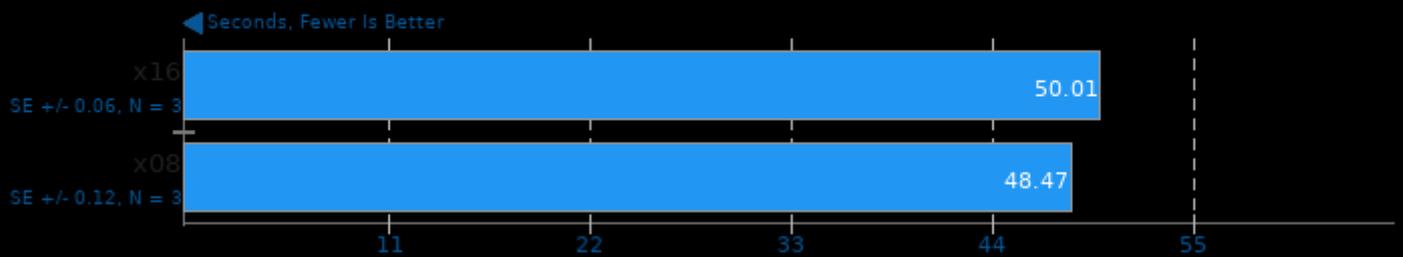
Timed LLVM Compilation 13.0

Build System: Unix Makefiles



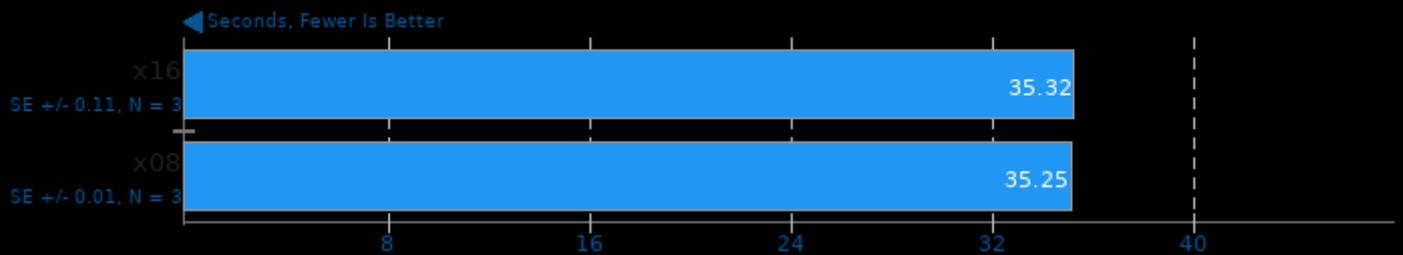
Timed Mesa Compilation 21.0

Time To Compile



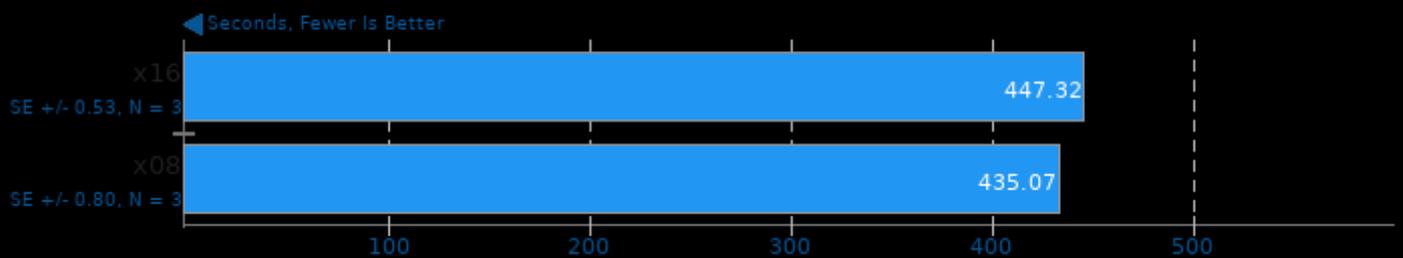
Timed MPlayer Compilation 1.4

Time To Compile



Timed Node.js Compilation 15.11

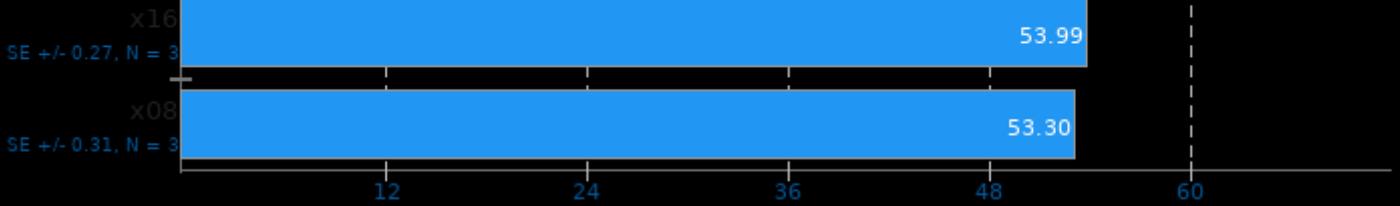
Time To Compile



Timed PHP Compilation 7.4.2

Time To Compile

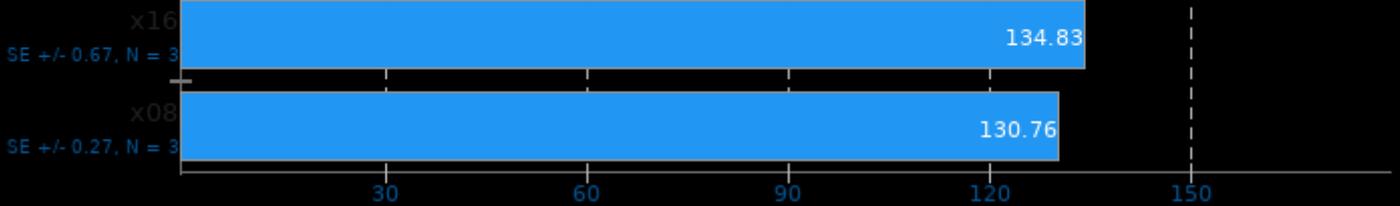
Seconds, Fewer Is Better



Build2 0.13

Time To Compile

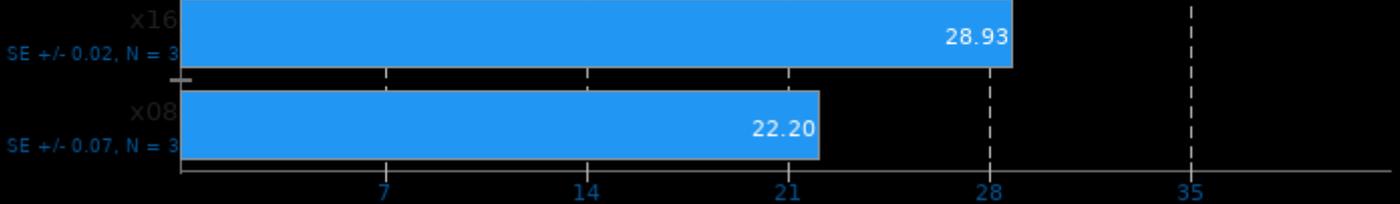
Seconds, Fewer Is Better



Y-Cruncher 0.7.8.9503

Calculating 500M Pi Digits

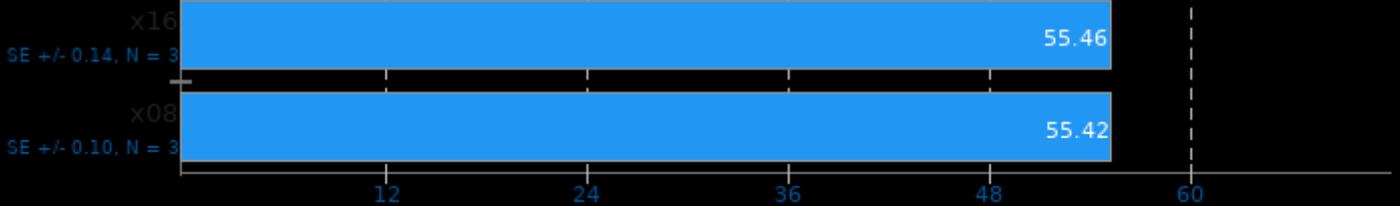
Seconds, Fewer Is Better



Timed Eigen Compilation 3.3.9

Time To Compile

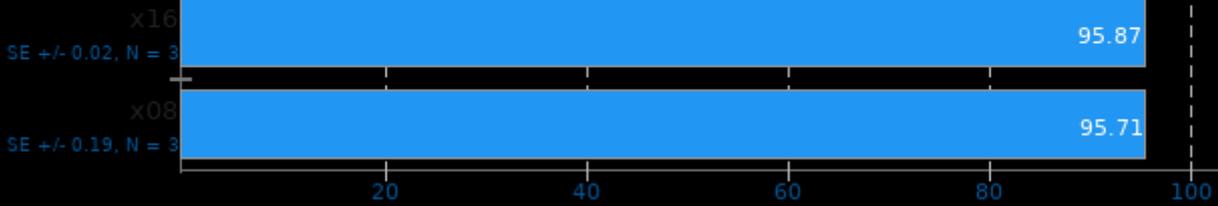
Seconds, Fewer Is Better



Timed Erlang/OTP Compilation 23.2

Time To Compile

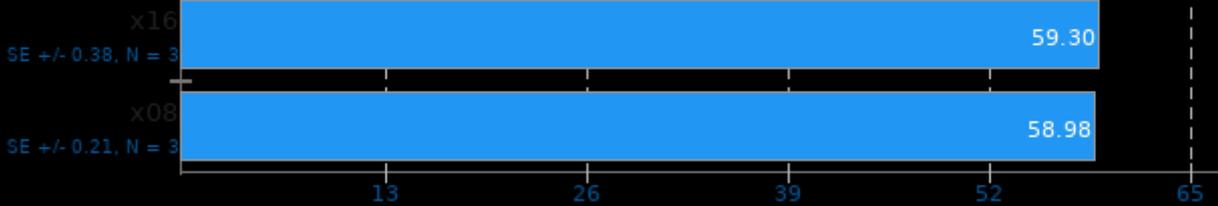
Seconds, Fewer Is Better



Timed Wasmer Compilation 1.0.2

Time To Compile

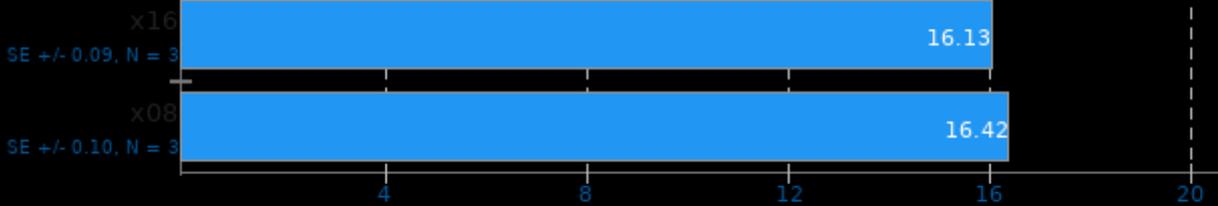
Seconds, Fewer Is Better



1. (C) gcc options: -m64 -pie -nodefaultlibs -ldl -lgcc_s -lutil -lrt -lpthread -lm -lc

Node.js V8 Web Tooling Benchmark

runs/s, More Is Better

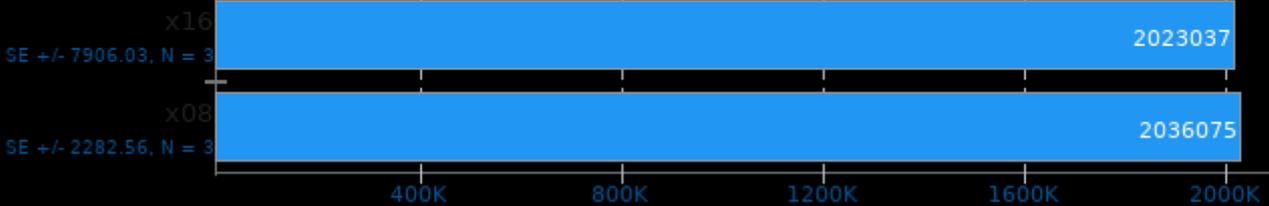


1. Nodejs
v10.19.0

Cryptsetup

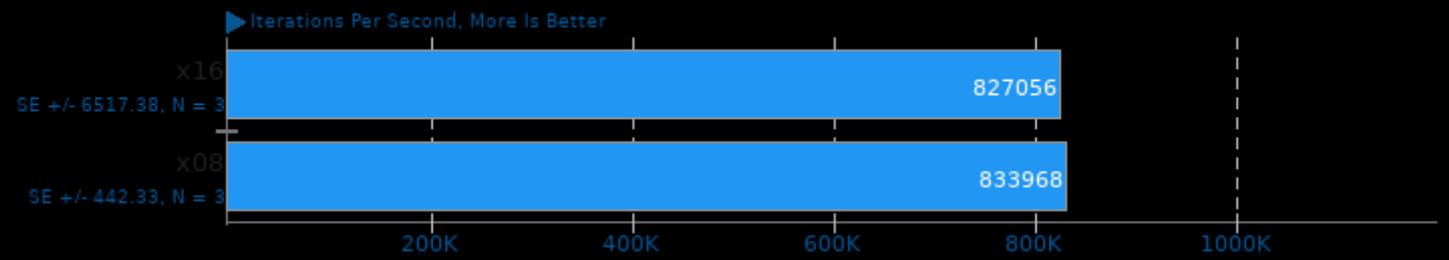
PBKDF2-sha512

Iterations Per Second, More Is Better



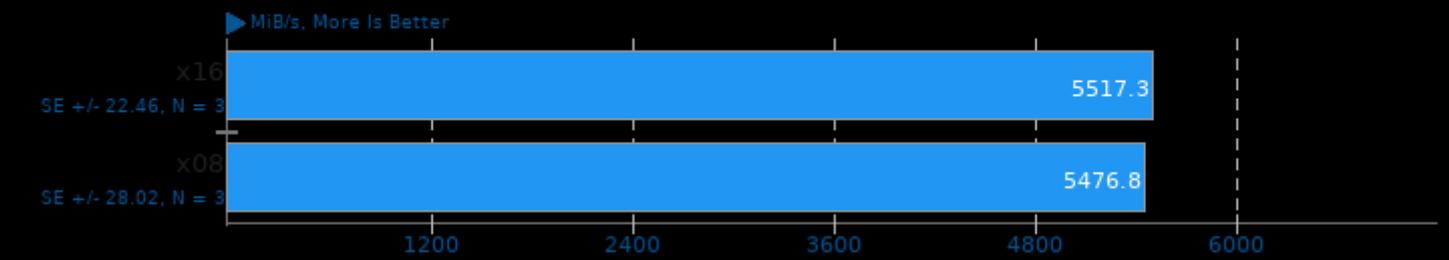
Cryptsetup

PBKDF2-whirlpool



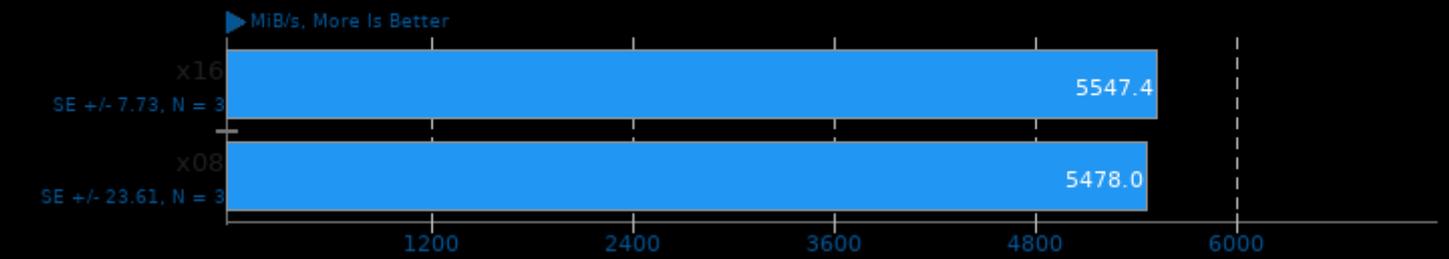
Cryptsetup

AES-XTS 256b Encryption



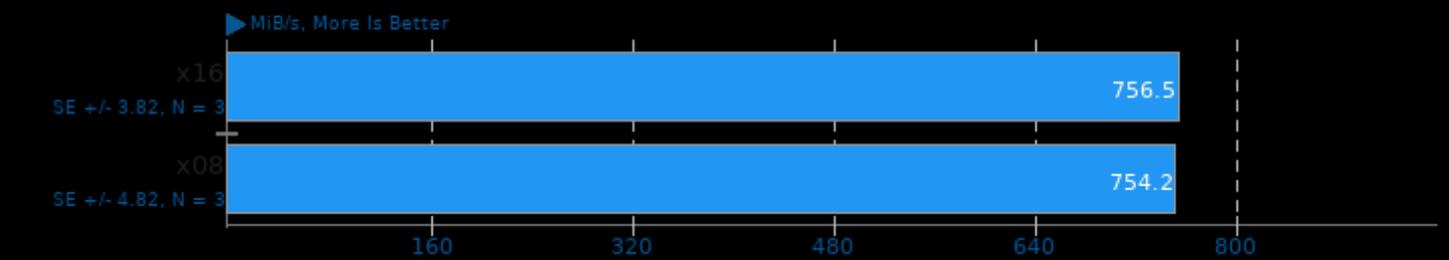
Cryptsetup

AES-XTS 256b Decryption



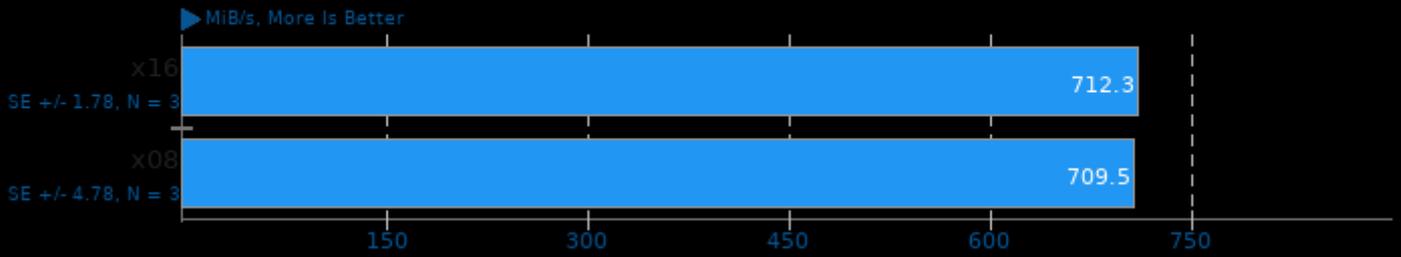
Cryptsetup

Serpent-XTS 256b Encryption



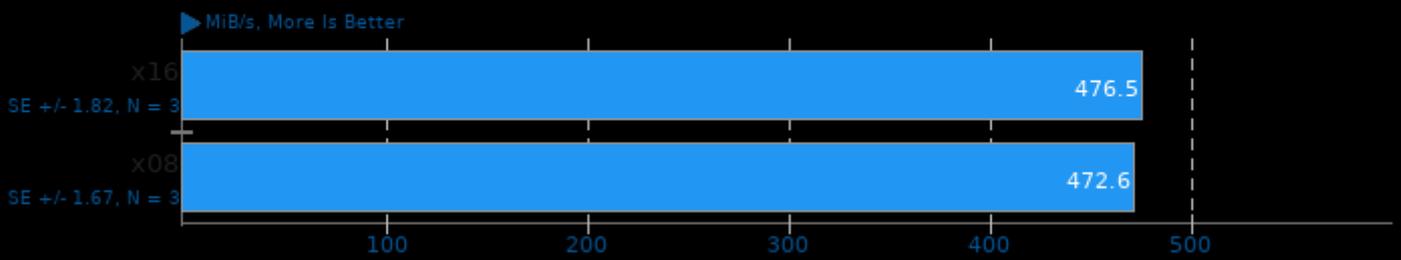
Cryptsetup

Serpent-XTS 256b Decryption



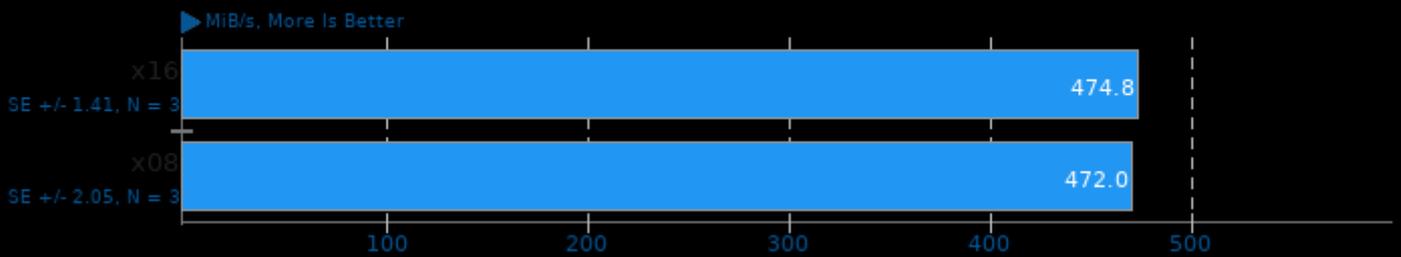
Cryptsetup

Twofish-XTS 256b Encryption



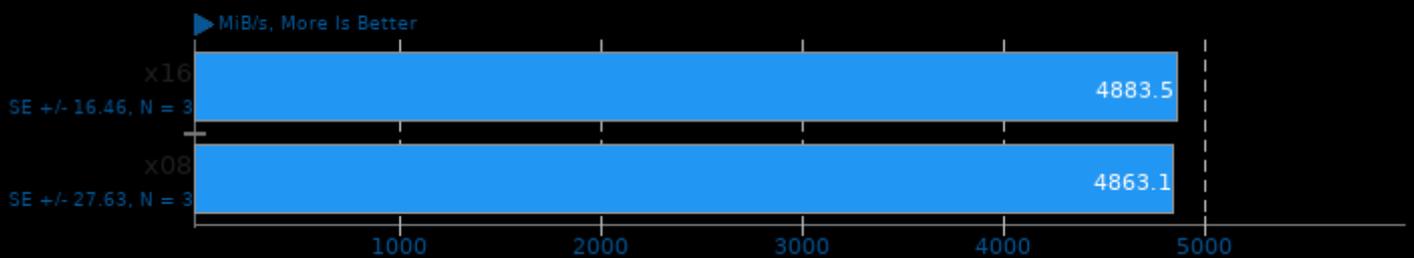
Cryptsetup

Twofish-XTS 256b Decryption



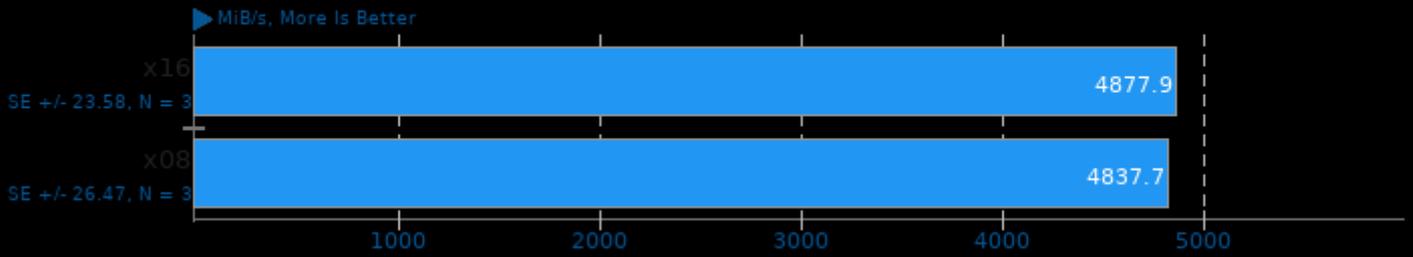
Cryptsetup

AES-XTS 512b Encryption



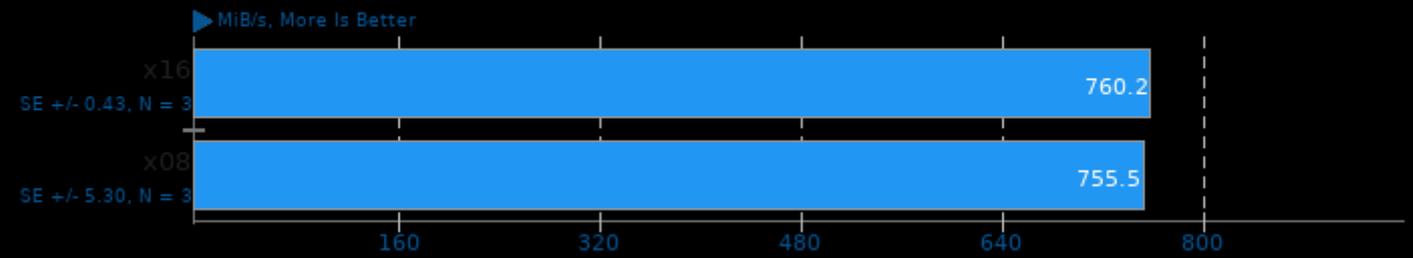
Cryptsetup

AES-XTS 512b Decryption



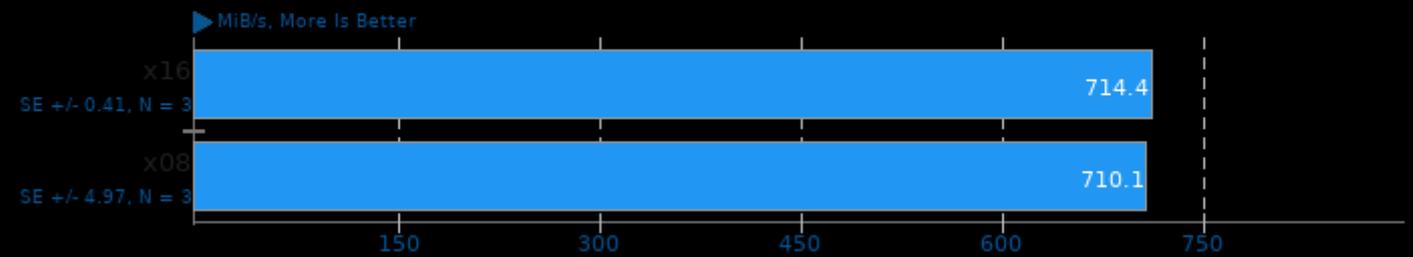
Cryptsetup

Serpent-XTS 512b Encryption



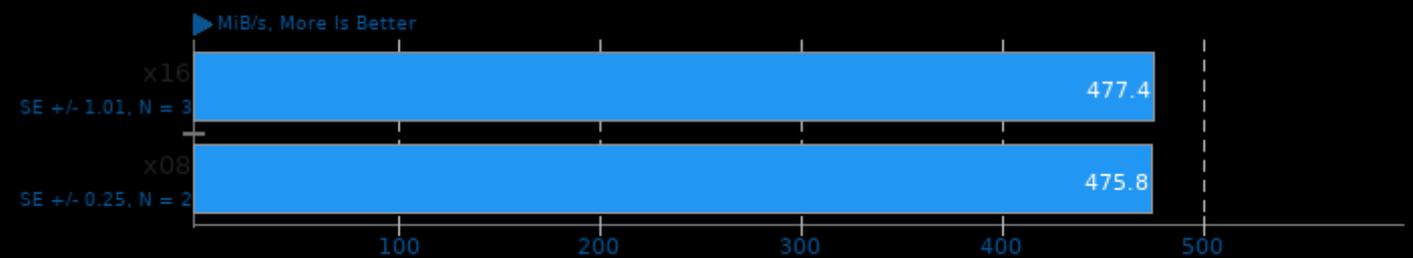
Cryptsetup

Serpent-XTS 512b Decryption



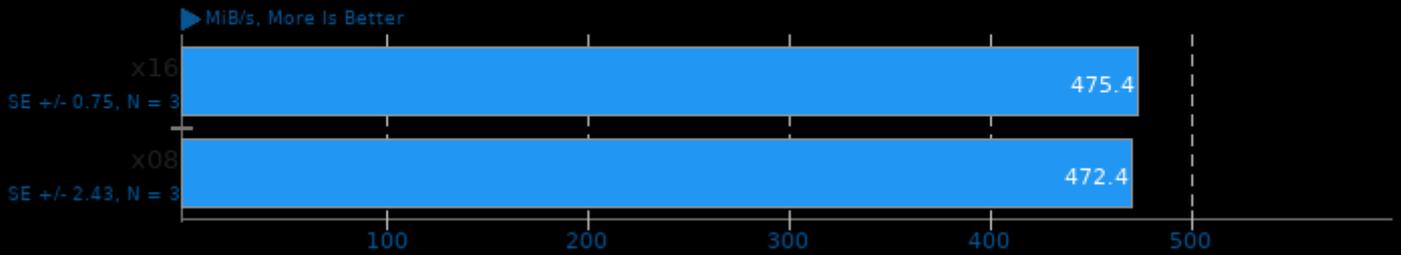
Cryptsetup

Twofish-XTS 512b Encryption



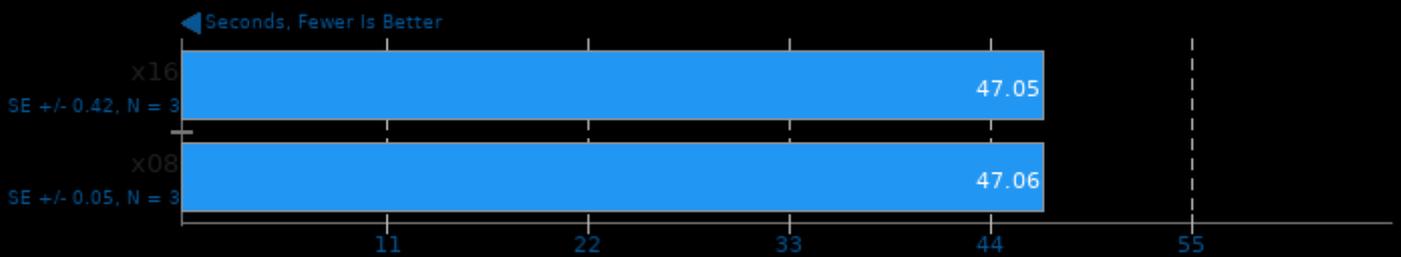
Cryptsetup

Twofish-XTS 512b Decryption



SQLite Speedtest 3.30

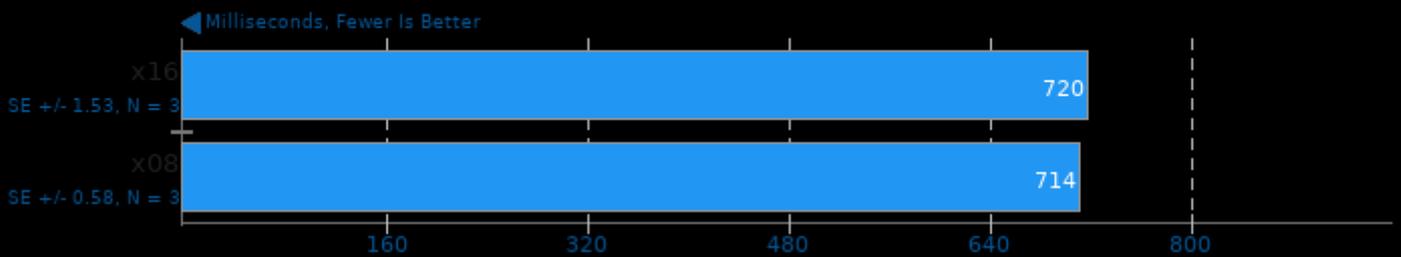
Timed Time - Size 1,000



1. (CC) gcc options: -O2 -ldl -lz -lpthread

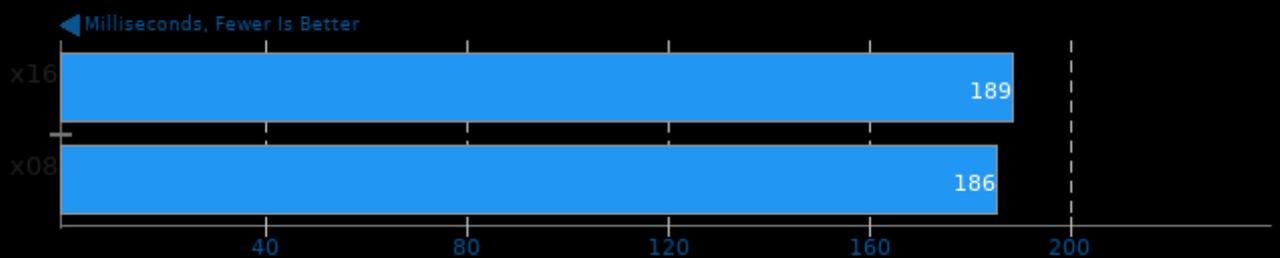
PyBench 2018-02-16

Total For Average Test Times



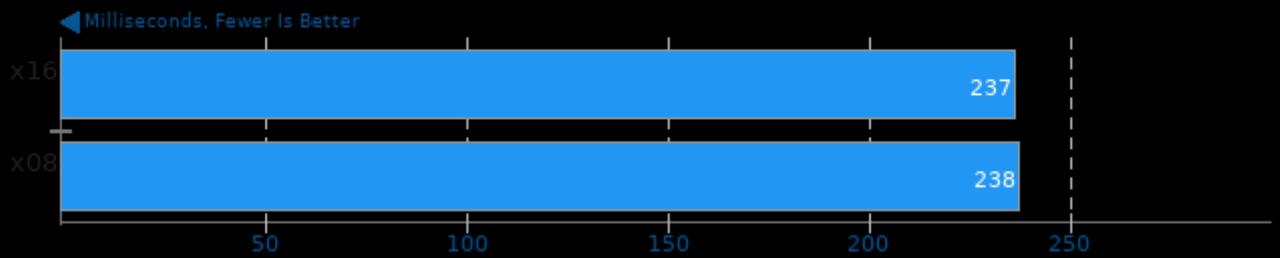
PyPerformance 1.0.0

Benchmark: go



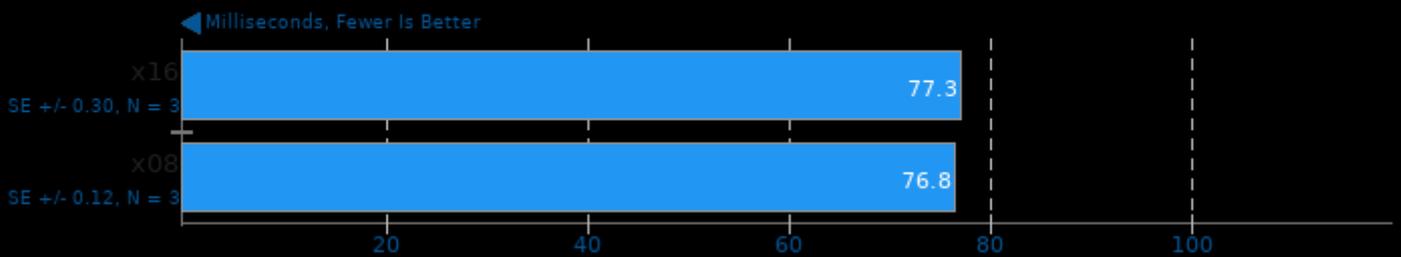
PyPerformance 1.0.0

Benchmark: 2to3



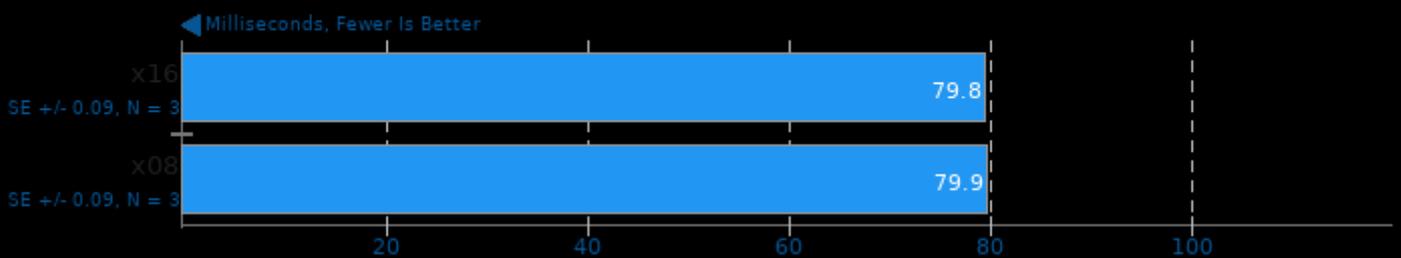
PyPerformance 1.0.0

Benchmark: chaos



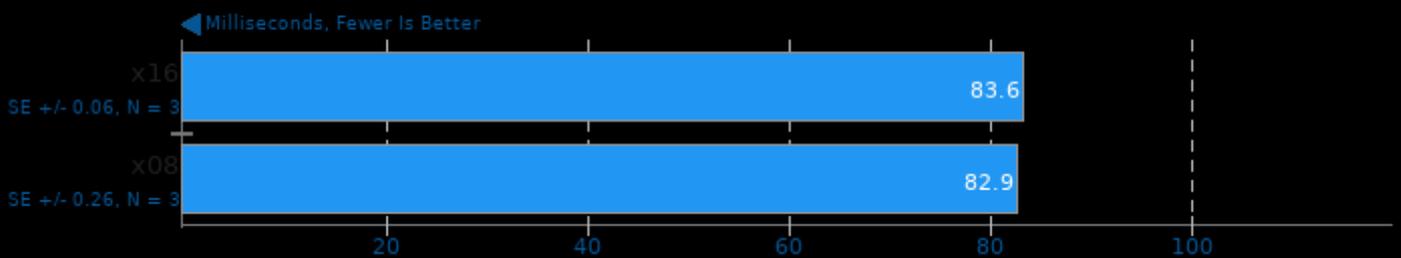
PyPerformance 1.0.0

Benchmark: float



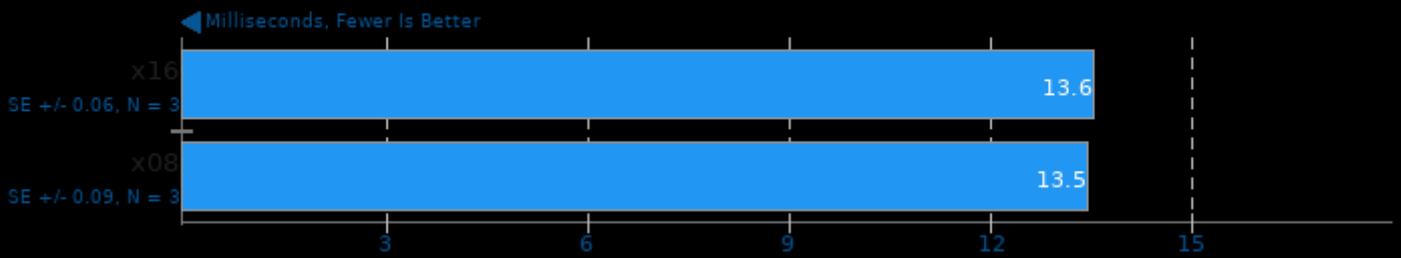
PyPerformance 1.0.0

Benchmark: nbody



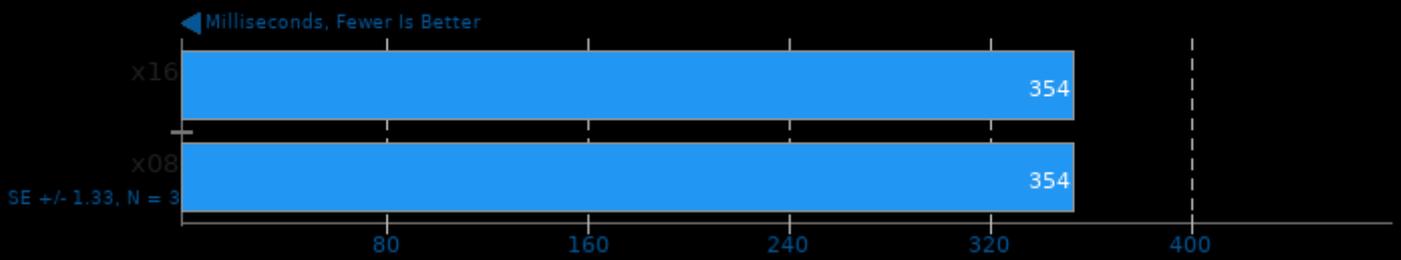
PyPerformance 1.0.0

Benchmark: pathlib



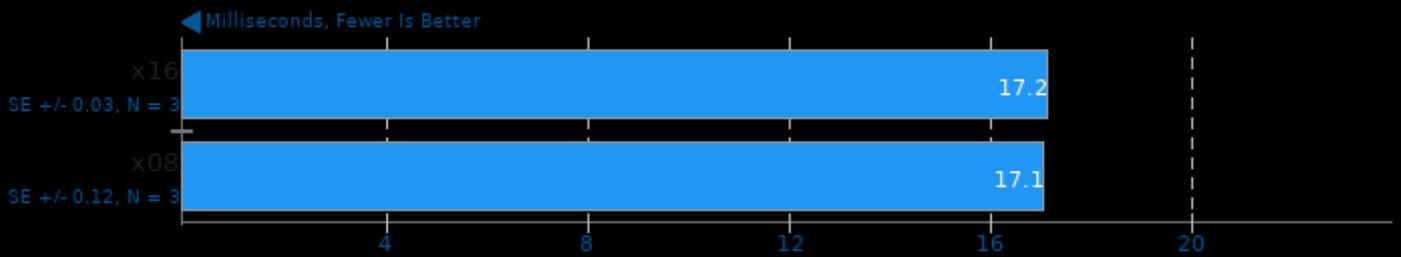
PyPerformance 1.0.0

Benchmark: raytrace



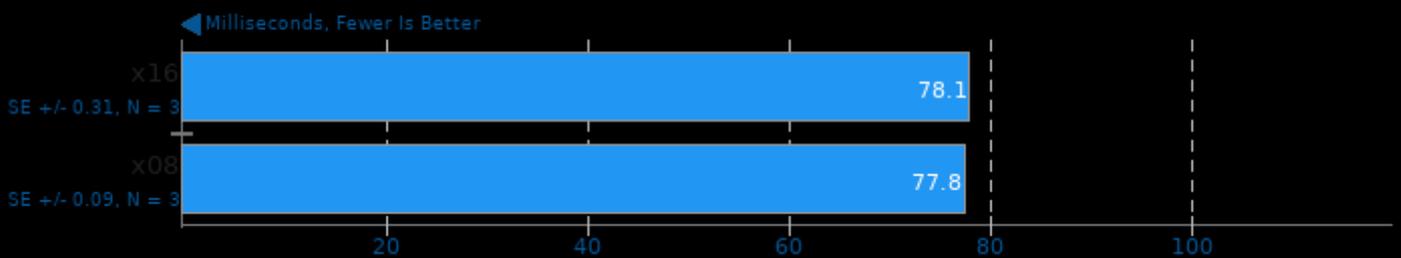
PyPerformance 1.0.0

Benchmark: json_loads



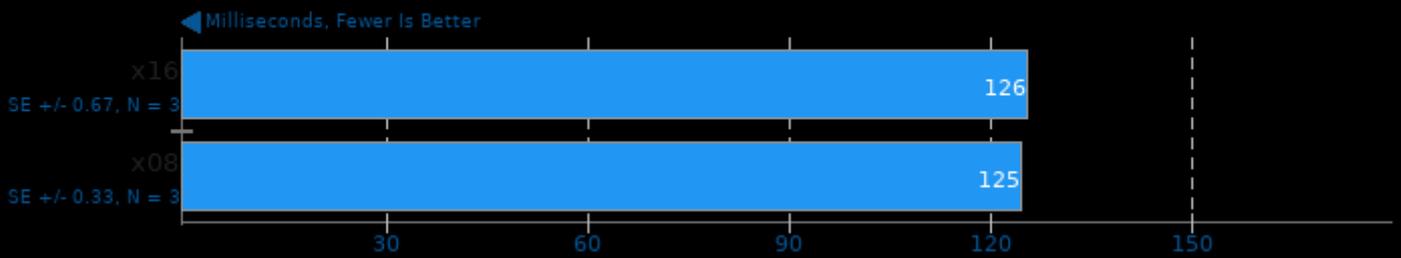
PyPerformance 1.0.0

Benchmark: crypto_pyaes



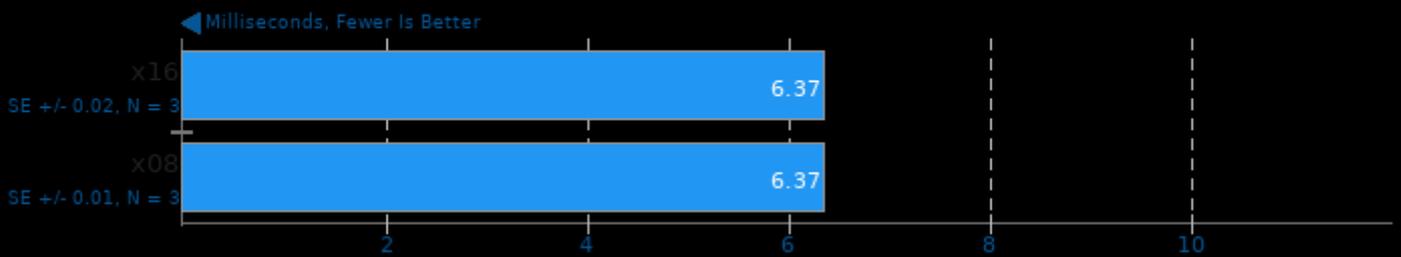
PyPerformance 1.0.0

Benchmark: regex_compile



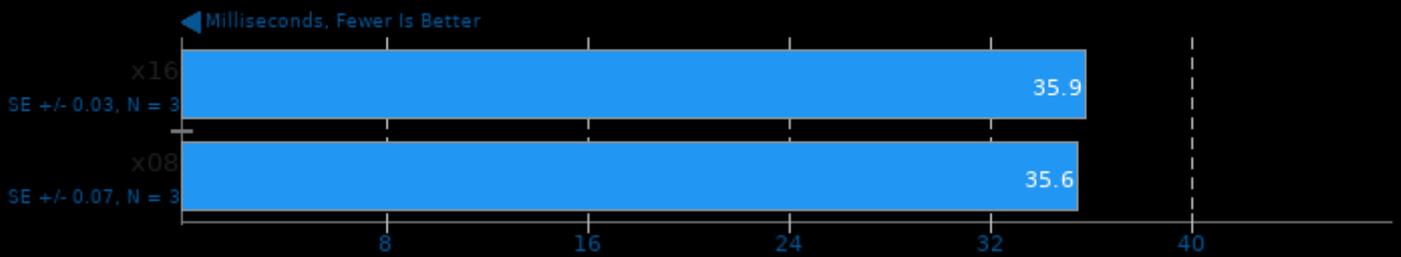
PyPerformance 1.0.0

Benchmark: python_startup



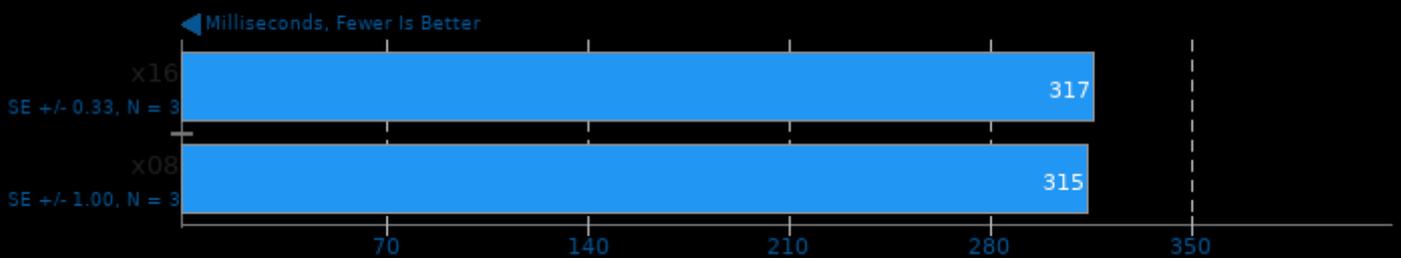
PyPerformance 1.0.0

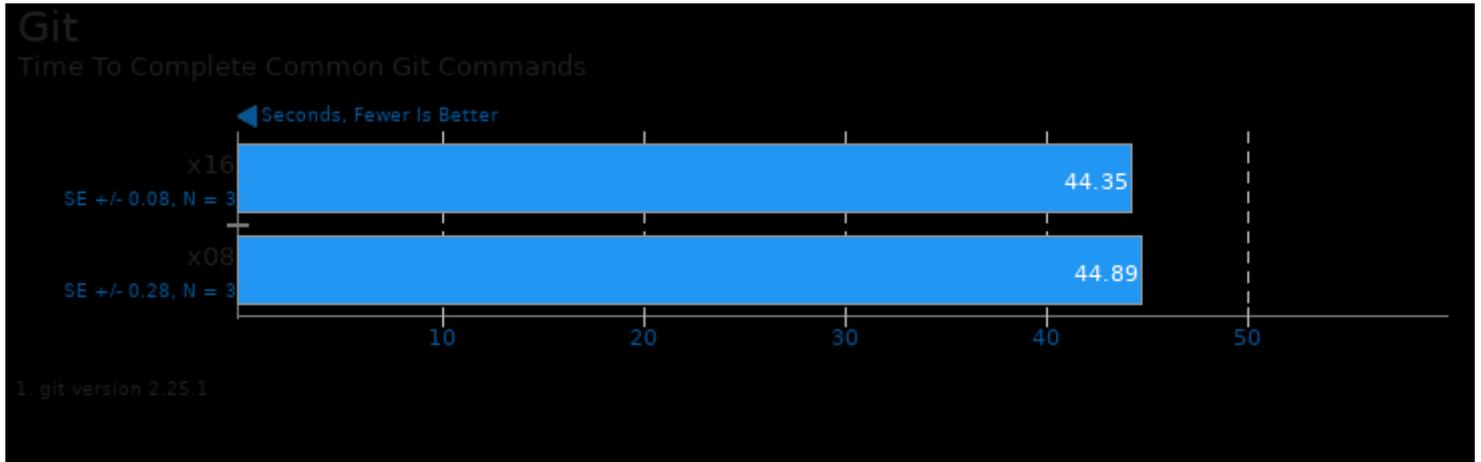
Benchmark: django_template



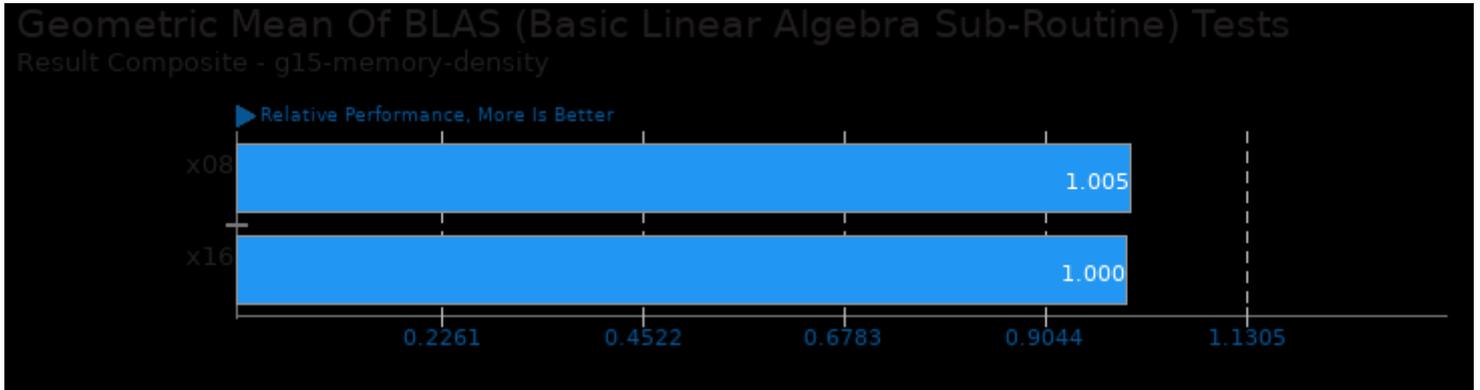
PyPerformance 1.0.0

Benchmark: pickle_pure_python

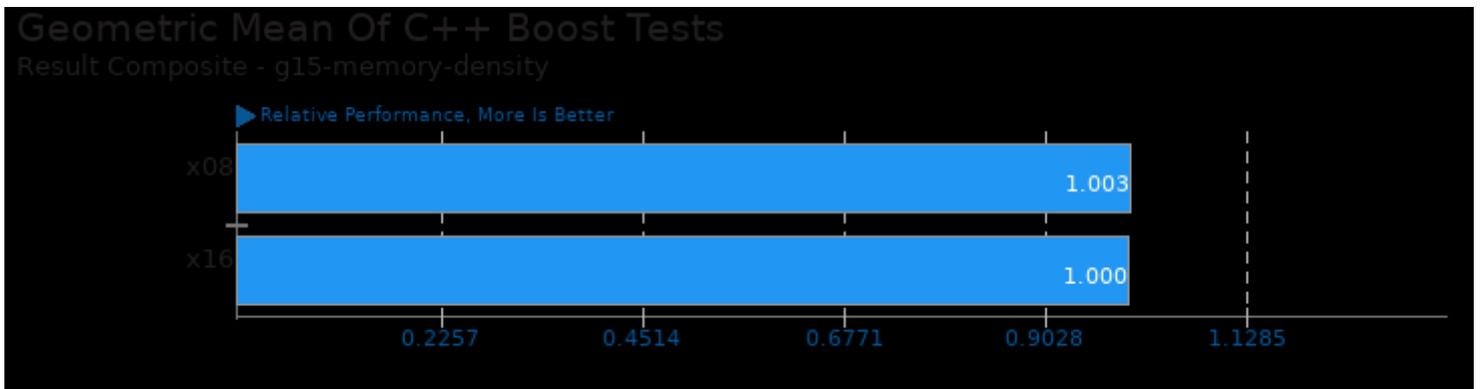




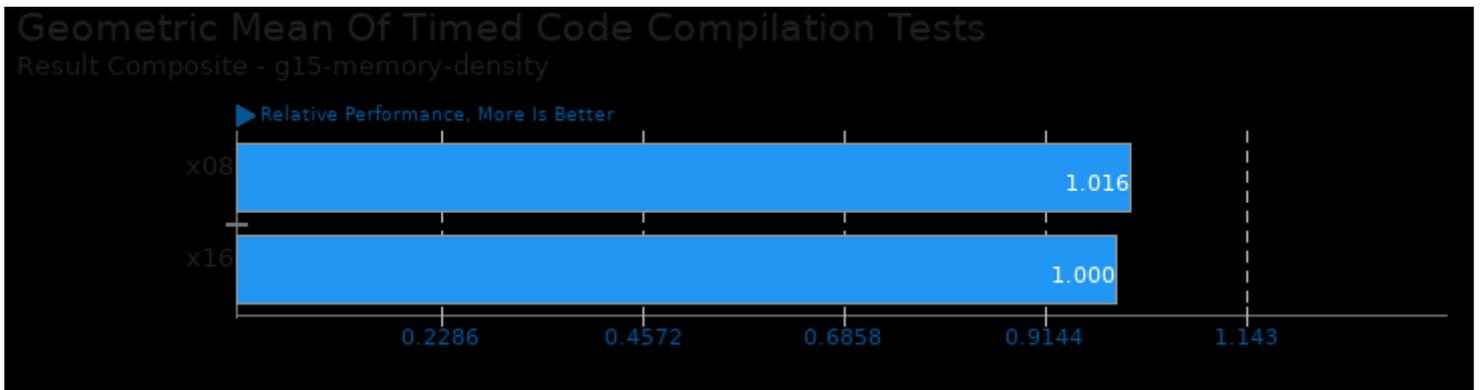
These geometric means are based upon test groupings / test suites for this result file.



Geometric mean based upon tests: pts/hpcc and pts/arrayfire



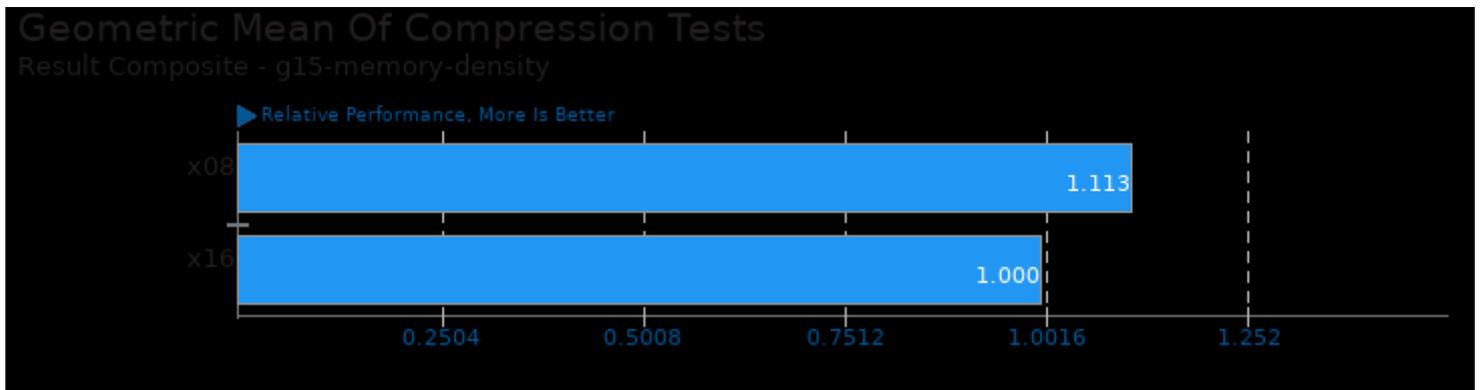
Geometric mean based upon tests: pts/srsran and pts/arrayfire



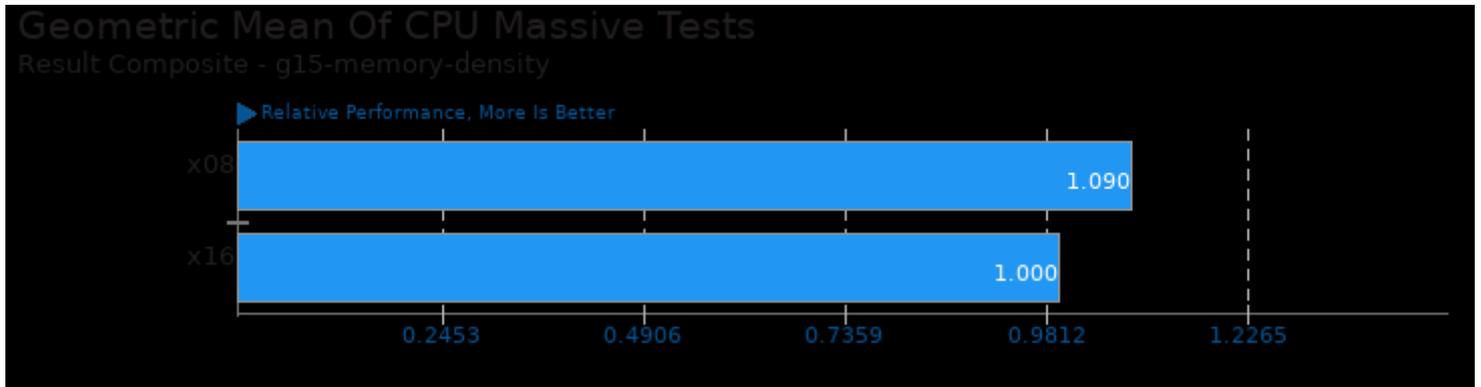
Geometric mean based upon tests: pts/build-apache, pts/build-php, pts/build-eigen, pts/build-linux-kernel, pts/build-imagemagick, pts/build-gcc, pts/build-gdb, pts/build-llvm, pts/build-ffmpeg, pts/build-mplayer, pts/build2, pts/build-godot, pts/build-erlang, pts/build-wasmer, pts/build-nodejs and pts/build-mesa



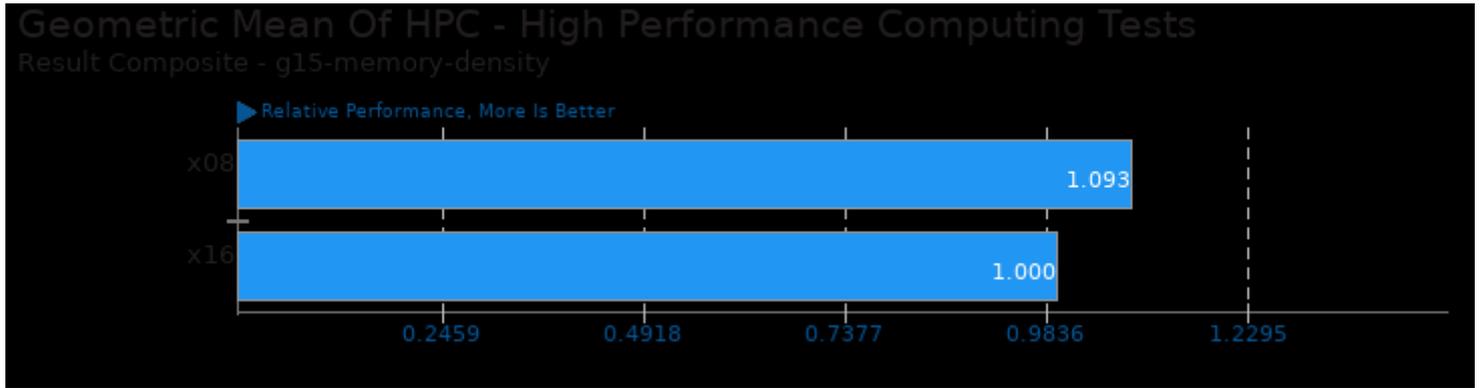
Geometric mean based upon tests: pts/build-php, pts/build-imagemagick, pts/build-llvm, pts/sqlite-speedtest, pts/compress-zstd, pts/build-gdb, pts/build-ffmpeg, pts/build-apache and pts/build-mplayer



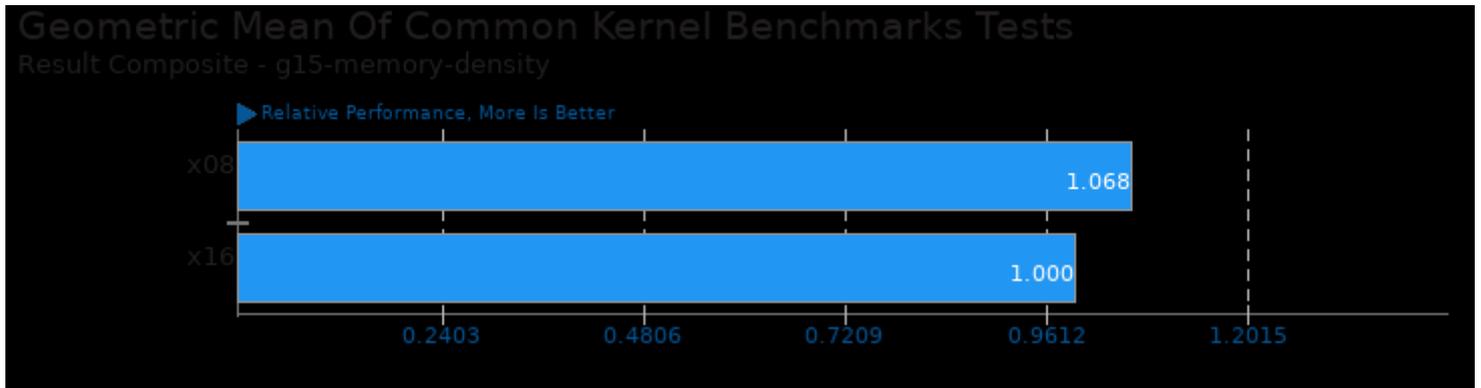
Geometric mean based upon tests: pts/compress-zstd and pts/blosc



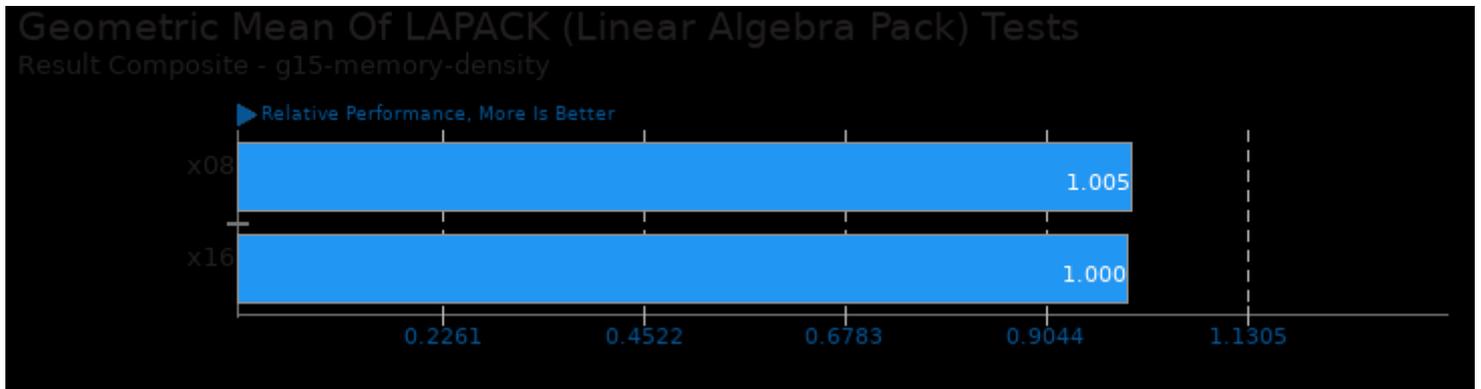
Geometric mean based upon tests: pts/build-apache, pts/build-gcc, pts/build-llvm, pts/build-linux-kernel, pts/build-php, pts/cachebench, pts/compress-zstd, pts/hpcc, pts/mbw, pts/ramspeed, pts/stream, pts/t-test1, pts/tinymembench and system/cryptsetup



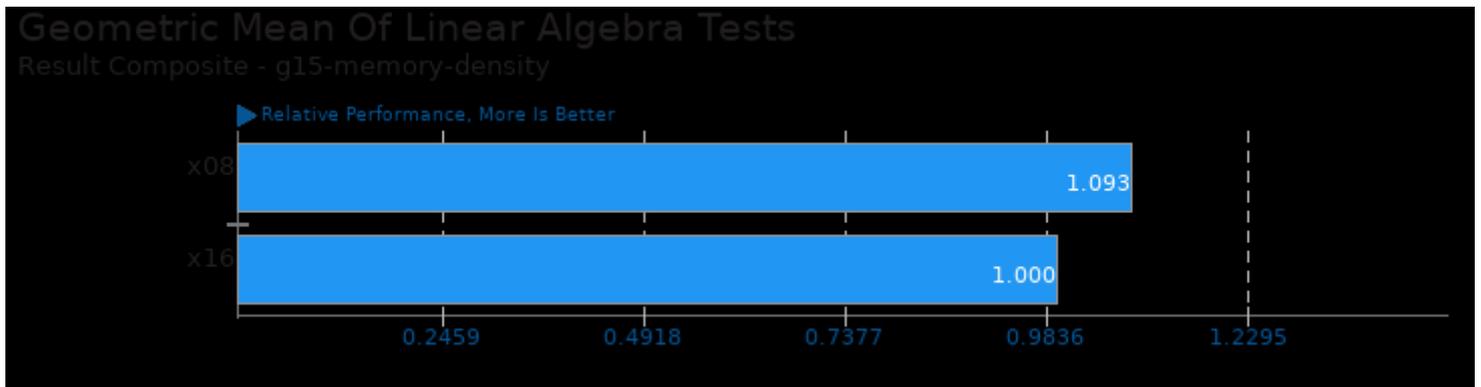
Geometric mean based upon tests: pts/hpcc, pts/mt-dgemm, pts/arrayfire and pts/amg



Geometric mean based upon tests: pts/sqlite-speedtest, pts/tinymembench, pts/mbw and pts/t-test1



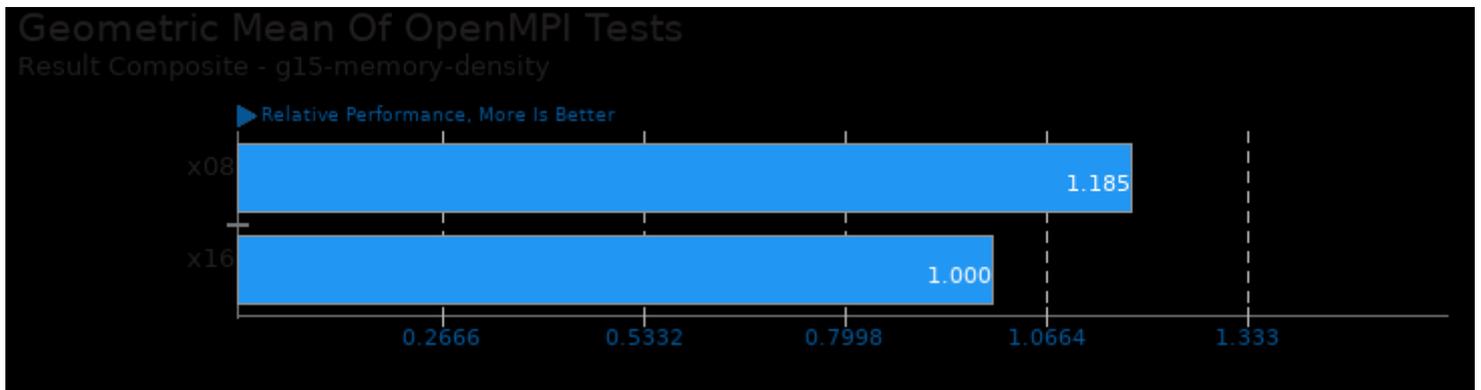
Geometric mean based upon tests: pts/hpcc and pts/arrayfire



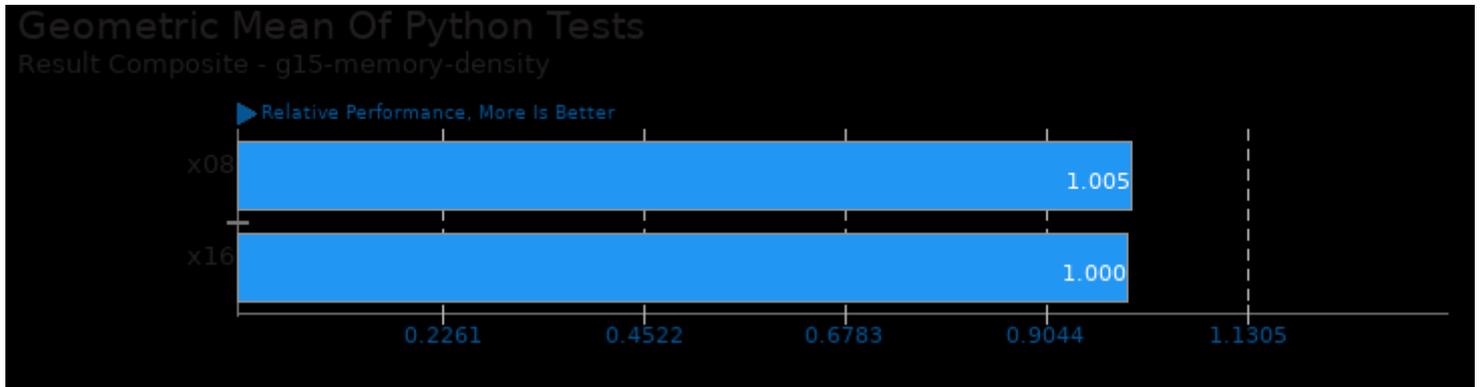
Geometric mean based upon tests: pts/mt-dgemm, pts/arrayfire, pts/amg and pts/hpcc



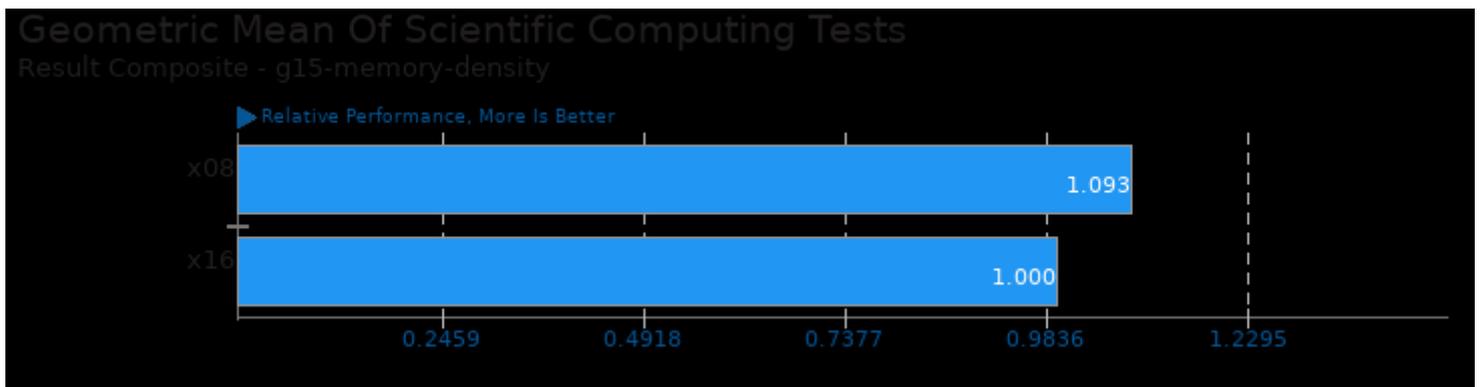
Geometric mean based upon tests: pts/ramspeed, pts/stream, pts/t-test1, pts/cachebench, pts/tinymembench and pts/mbw



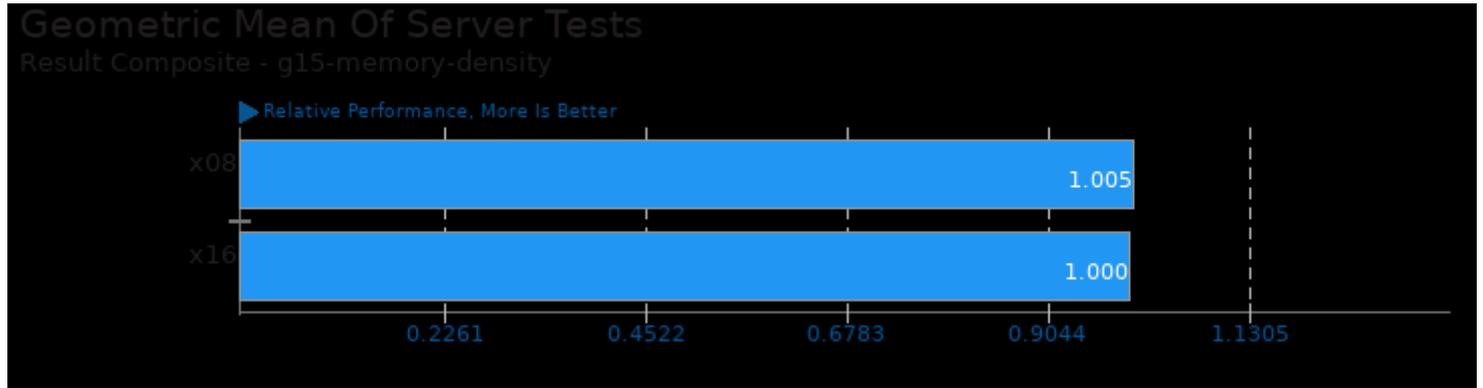
Geometric mean based upon tests: pts/hpcc and pts/amg



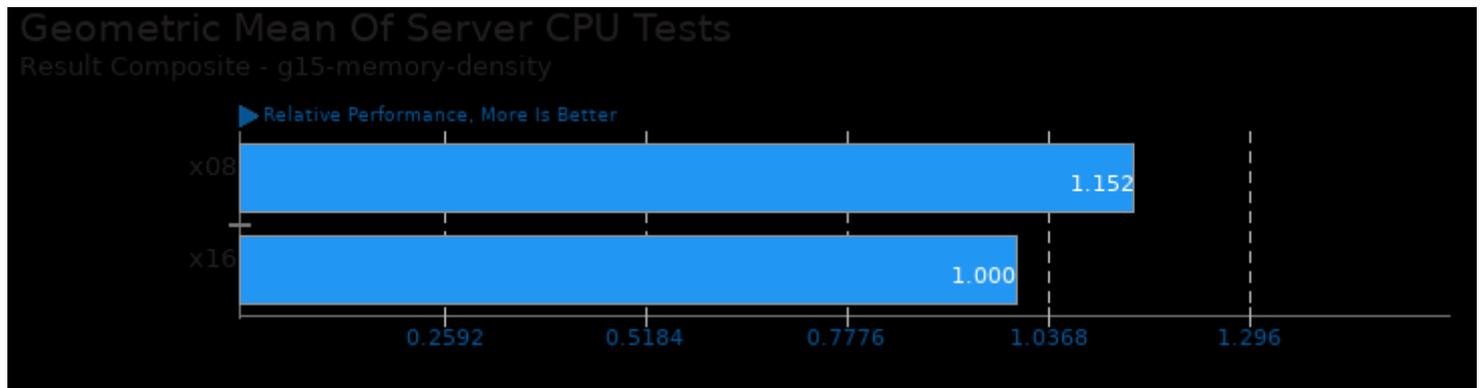
Geometric mean based upon tests: pts/pybench and pts/pyperformance



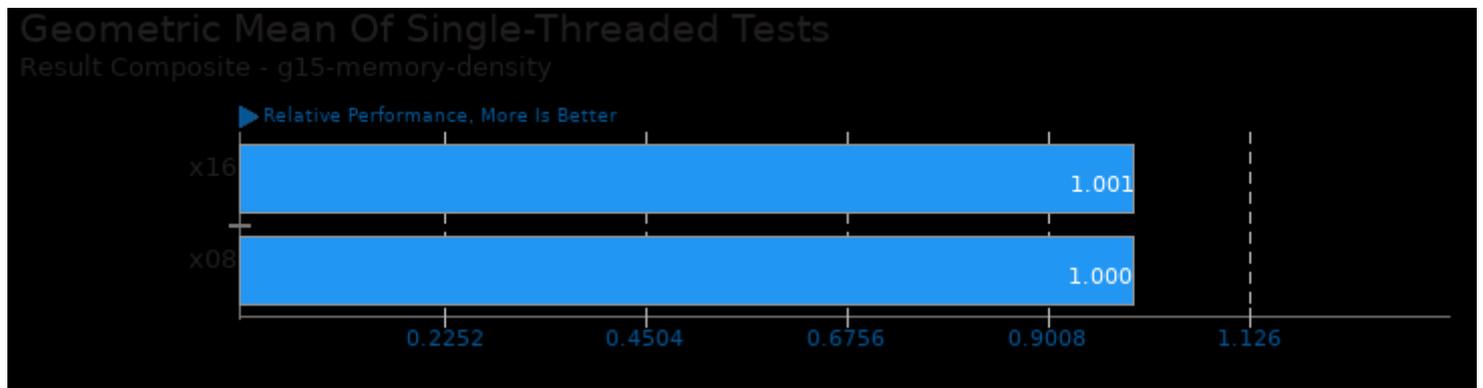
Geometric mean based upon tests: pts/mt-dgemm, pts/arrayfire, pts/amg and pts/hpcc



Geometric mean based upon tests: pts/simdjson, pts/node-web-tooling and pts/sqlite-speedtest



Geometric mean based upon tests: pts/build-gcc, pts/build-linux-kernel, pts/build-php, pts/build-llvm, pts/compress-zstd, pts/pybench, pts/ramspeed and pts/stream



Geometric mean based upon tests: pts/cachebench, pts/pybench and pts/git

This file was automatically generated via the Phoronix Test Suite benchmarking software on Friday, 29 March 2024 06:18.