

www.phoronix-test-suite.com

8400 jan

Intel Core i5-8400 testing with a MSI Z370M MORTAR (MS-7B54) v1.0 (1.80 BIOS) and MSI Intel UHD 630 CFL GT2 3GB on Ubuntu 20.04 via the Phoronix Test Suite.

Automated Executive Summary

A had the most wins, coming in first place for 45% of the tests.

Based on the geometric mean of all complete results, the fastest (A) was 1.003x the speed of the slowest (C). B was 0.998x the speed of A and C was 0.999x the speed of B.

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

srsRAN (Test: 4G PHY_DL_Test 100 PRB SISO 64-QAM) at 1.018x
srsRAN (Test: 4G PHY_DL_Test 100 PRB SISO 256-QAM) at 1.016x
Kvazaar (Video Input: Bosphorus 4K - Video Preset: Medium) at 1.015x
srsRAN (Test: 4G PHY_DL_Test 100 PRB MIMO 256-QAM) at 1.014x
Kvazaar (Video Input: Bosphorus 4K - Video Preset: Very Fast) at 1.013x
Timed Node.js Compilation (Time To Compile) at 1.013x
srsRAN (Test: 4G PHY_DL_Test 100 PRB MIMO 64-QAM) at 1.012x
Kvazaar (Video Input: Bosphorus 1080p - Video Preset: Ultra Fast) at 1.012x
Kvazaar (Video Input: Bosphorus 1080p - Video Preset: Very Fast) at 1.009x

srsRAN (Test: 5G PHY_DL_NR Test 52 PRB SISO 64-QAM) at 1.009x.

Test Systems:

A

B

C

Processor: Intel Core i5-8400 @ 4.00GHz (6 Cores), Motherboard: MSI Z370M MORTAR (MS-7B54) v1.0 (1.80 BIOS), Chipset: Intel 8th Gen Core, Memory: 8GB, Disk: 512GB INTEL SSDPEKNW512G8, Graphics: MSI Intel UHD 630 CFL GT2 3GB (1050MHz), Audio: Realtek ALC892, Monitor: DELL S2409W, Network: Intel I219-V

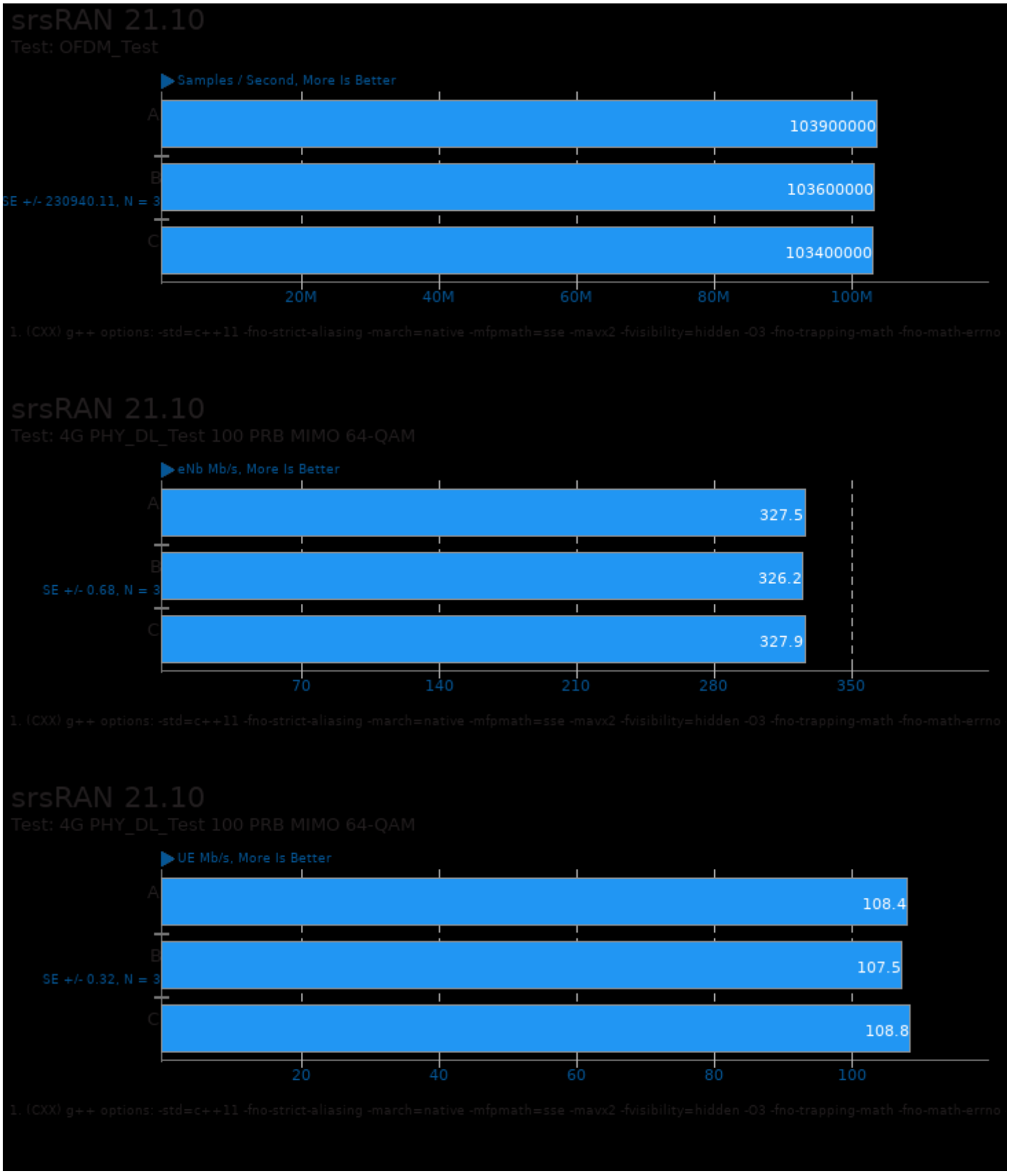
OS: Ubuntu 20.04, Kernel: 5.11.0-41-generic (x86_64), Desktop: GNOME Shell 3.36.4, Display Server: X Server 1.20.9, OpenGL: 4.6 Mesa 20.0.8, Compiler: GCC 9.3.0, File-System: ext4, Screen Resolution: 1920x1080

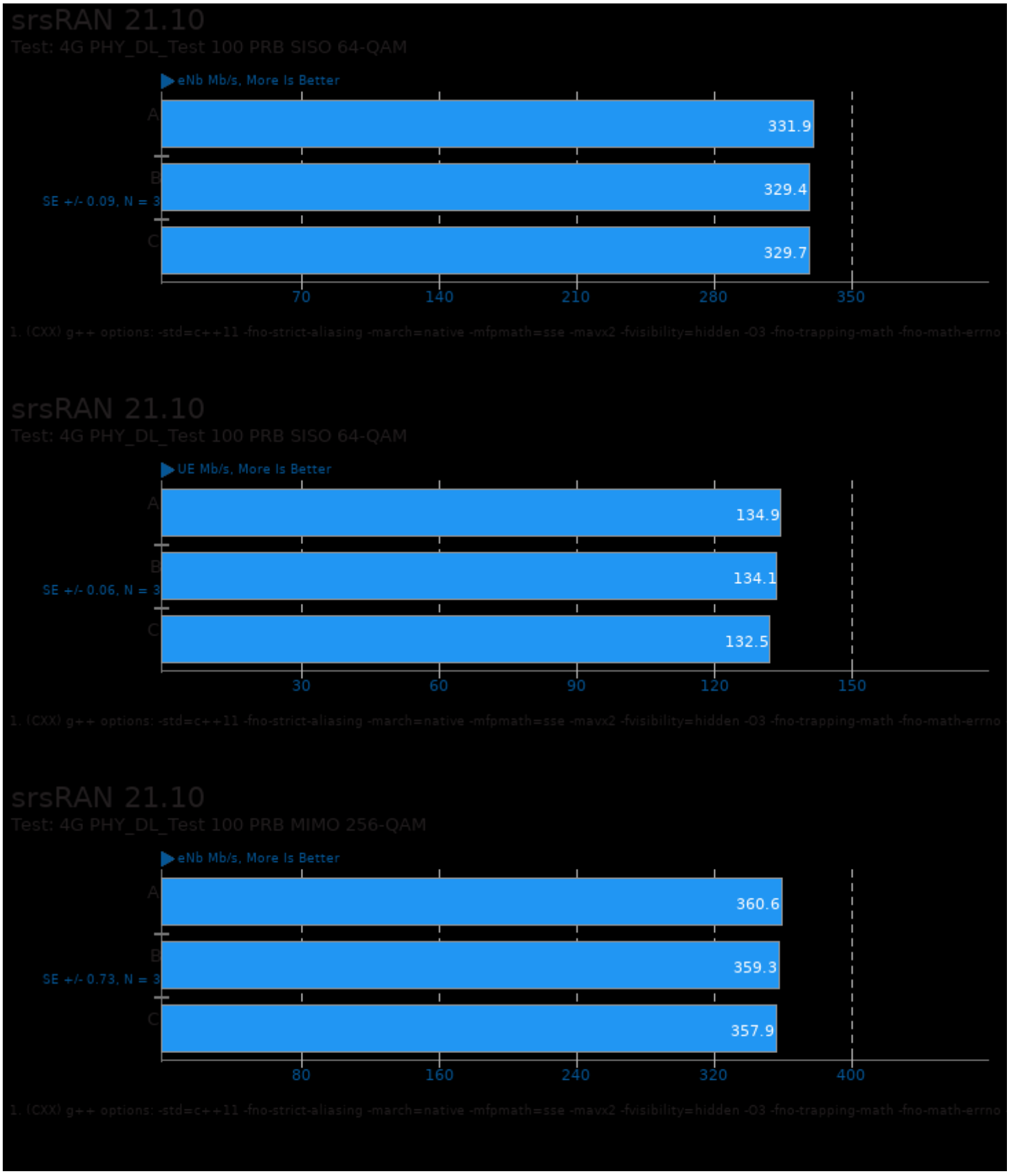
Kernel Notes: Transparent Huge Pages: madvise
Compiler Notes: --build=x86_64-linux-gnu --disable-vtable-verify --disable-werror --enable-checking=release --enable-clocale=gnu --enable-default-pie --enable-gnu-unique-object --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,gm2 --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-multiarch --enable-multilib --enable-nls --enable-objc-gc=auto --enable-offload-targets=nvptx-none=/build/gcc-9-HskZEa/gcc-9-9.3.0/debian/tmp-nvptx/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-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 powersave (EPP: balance_performance) - CPU Microcode: 0xea - Thermald 1.9.1
Python Notes: Python 3.8.10
Security Notes: itlb_multihit: KVM: Mitigation of VMX disabled + l1tf: Mitigation of PTE Inversion; VMX: conditional cache flushes SMT disabled + mds: Mitigation of Clear buffers; SMT disabled + meltdown: Mitigation of PTI + spec_store_bypass: Mitigation of SSB disabled via prctl and seccomp + spectre_v1: Mitigation of usercopy/swapgs barriers and __user pointer sanitization + spectre_v2: Mitigation of Full generic retpoline IBPB: conditional IBRS_FW STIBP: disabled RSB filling + srbds: Mitigation of Microcode + tsx_async_abort: Not affected

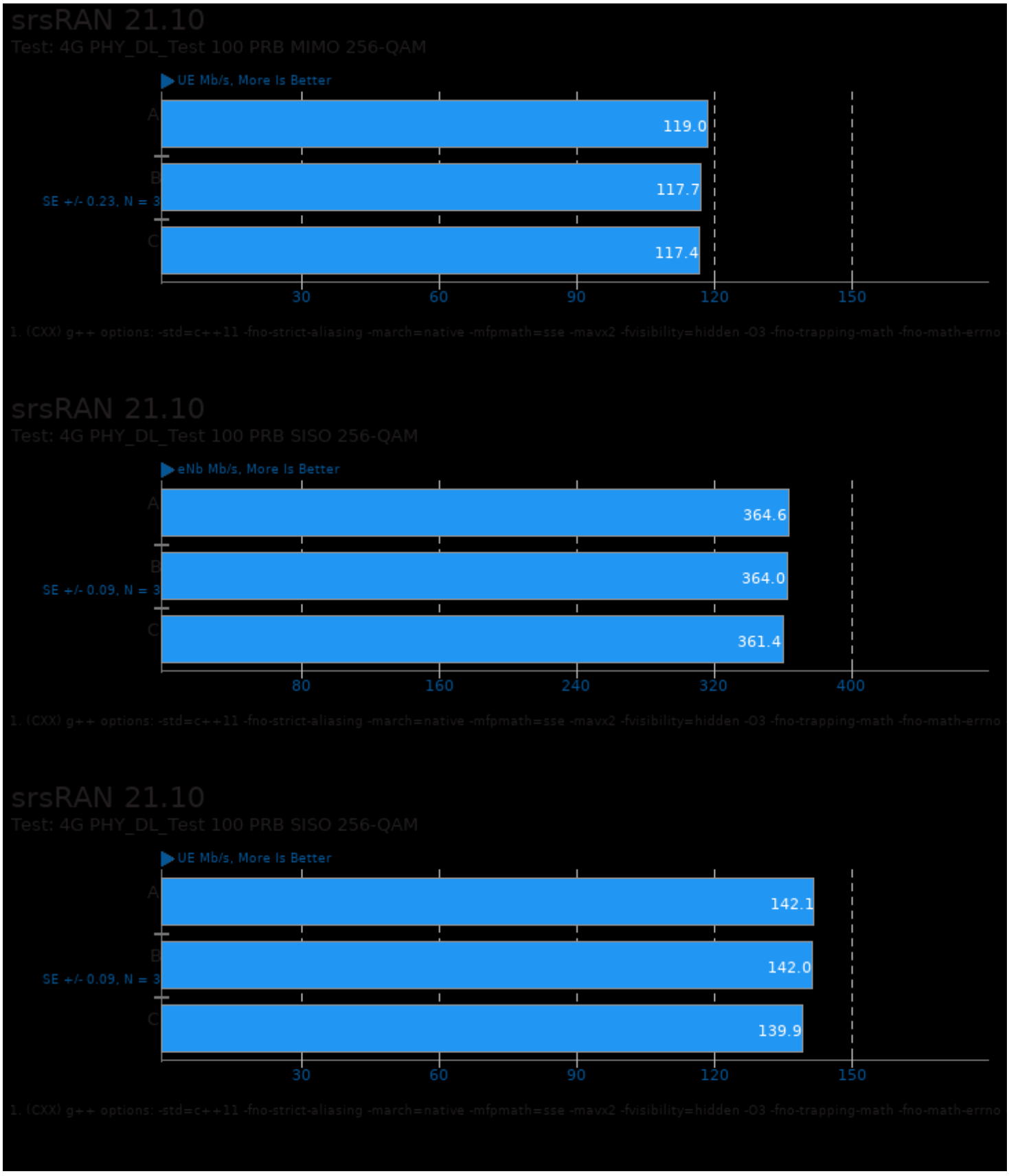
	A	B	C
srsRAN - OFDM_Test (Samples / Second)	**103900000**	103600000	**103400000**
Normalized	100%	99.71%	99.52%
Standard Deviation		0.4%	
srsRAN - 4.P.1.P.M.6.Q (eNb Mb/s)	327.5	**326.2**	**327.9**
Normalized	99.88%	99.48%	100%
Standard Deviation		0.4%	
srsRAN - 4.P.1.P.M.6.Q (UE Mb/s)	108.4	**107.5**	**108.8**
Normalized	99.63%	98.81%	100%
Standard Deviation		0.5%	
srsRAN - 4.P.1.P.S.6.Q (eNb Mb/s)	**331.9**	**329.4**	329.7
Normalized	100%	99.25%	99.34%
Standard Deviation		0%	
srsRAN - 4.P.1.P.S.6.Q (UE Mb/s)	**134.9**	134.1	**132.5**
Normalized	100%	99.41%	98.22%
Standard Deviation		0.1%	
srsRAN - 4.P.1.P.M.2.Q (eNb Mb/s)	**360.6**	359.3	**357.9**

Normalized	100%	99.64%	99.25%
Standard Deviation		0.4%	
srsRAN - 4.P.1.P.M.2.Q (UE Mb/s)	**119**	117.7	**117.4**
Normalized	100%	98.91%	98.66%
Standard Deviation		0.3%	
srsRAN - 4.P.1.P.S.2.Q (eNb Mb/s)	**364.6**	364.0	**361.4**
Normalized	100%	99.84%	99.12%
Standard Deviation		0%	
srsRAN - 4.P.1.P.S.2.Q (UE Mb/s)	**142.1**	142.0	**139.9**
Normalized	100%	99.93%	98.45%
Standard Deviation		0.1%	
srsRAN - 5.P.T.5.P.S.6.Q (eNb Mb/s)	**111**	**110.0**	110.8
Normalized	100%	99.1%	99.82%
Standard Deviation		0.6%	
srsRAN - 5.P.T.5.P.S.6.Q (UE Mb/s)	**51.3**	**51.3**	**51.2**
Normalized	100%	100%	99.81%
Standard Deviation		0.1%	
Kvazaar - Bosphorus 4K - Slow (FPS)	3.85	3.85	3.85
Standard Deviation		0.3%	
Kvazaar - Bosphorus 4K - Medium (FPS)	3.93	**3.88**	**3.94**
Normalized	99.75%	98.48%	100%
Standard Deviation		0.5%	
Kvazaar - Bosphorus 1080p - Slow (FPS)	20.61	**20.67**	**20.6**
Normalized	99.71%	100%	99.66%
Standard Deviation		0.3%	
Kvazaar - Bosphorus 1080p - Medium (FPS)	21.25	**21.27**	**21.23**
Normalized	99.91%	100%	99.81%
Standard Deviation		0.1%	
Kvazaar - Bosphorus 4K - Very Fast (FPS)	9.1	**9.09**	**9.21**
Normalized	98.81%	98.7%	100%
Standard Deviation		0.4%	
Kvazaar - Bosphorus 4K - Ultra Fast (FPS)	16.02	**15.95**	**16.06**
Normalized	99.75%	99.32%	100%
Standard Deviation		0.1%	
Kvazaar - Bosphorus 1080p - Very Fast (FPS)	42.47	**42.80**	**42.41**
Normalized	99.23%	100%	99.09%
Standard Deviation		0.6%	
Kvazaar - Bosphorus 1080p - Ultra Fast	**70.84**	**71.66**	70.89
Normalized	98.86%	100%	98.93%
Standard Deviation		0.2%	
Timed Node.js Compilation - Time To Compile (sec)	1186	**1185**	**1201**
Normalized	99.97%	100%	98.7%
Standard Deviation		0%	
OpenJPEG - N.C.P.M (ms)	55223	**55162**	**55252**
Normalized	99.89%	100%	99.84%
Standard Deviation		0.1%	
ASTC Encoder - Medium (sec)	**6.312**	6.3276	**6.3341**
Normalized	100%	99.75%	99.65%
Standard Deviation		0.1%	
ASTC Encoder - Thorough (sec)	12.7894	**12.7870**	**12.7971**
Normalized	99.98%	100%	99.92%
Standard Deviation		0%	
ASTC Encoder - Exhaustive (sec)	116.2304	**116.2472**	**116.1785**
Normalized	99.96%	99.94%	100%

Standard Deviation	0%

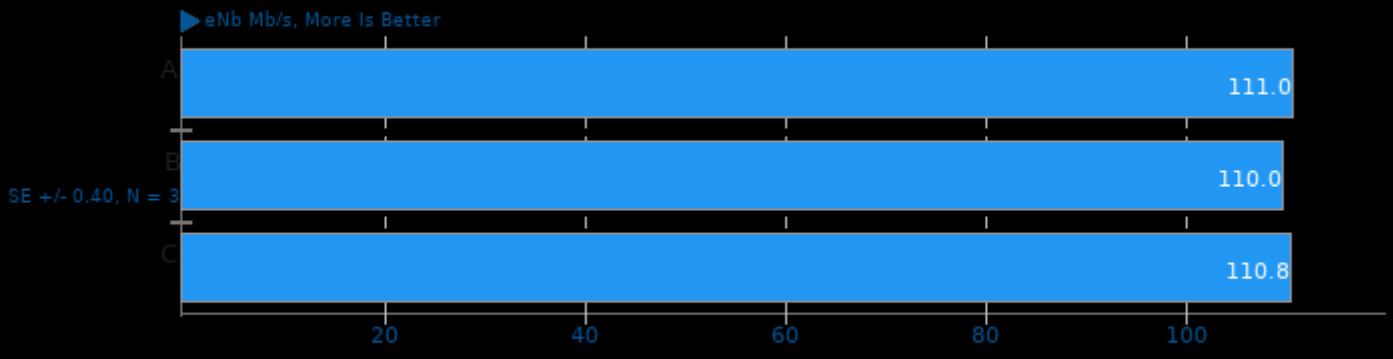

srsRAN 21.10
Test: OFDM_Test
Samples / Second, More Is Better
A
B
C
103900000
103600000
103400000
SE +/- 230940.11, N = 3
20M
40M
60M
80M
100M
1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno
srsRAN 21.10
Test: 4G PHY_DL_Test 100 PRB MIMO 64-QAM
eNb Mb/s, More Is Better
A
B
C
327.5
326.2
327.9
SE +/- 0.68, N = 3
70
140
210
280
350
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 21.10
Test: 4G PHY_DL_Test 100 PRB MIMO 64-QAM
UE Mb/s, More Is Better
A
B
C
108.4
107.5
108.8
SE +/- 0.32, N = 3
20
40
60
80
100
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 21.10
Test: 4G PHY_DL_Test 100 PRB SISO 64-QAM
eNb Mb/s, More Is Better
A
B
C
331.9
329.4
329.7
SE +/- 0.09, N = 3
70
140
210
280
350
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 21.10
Test: 4G PHY_DL_Test 100 PRB SISO 64-QAM
UE Mb/s, More Is Better
A
B
C
134.9
134.1
132.5
SE +/- 0.06, N = 3
30
60
90
120
150
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 21.10
Test: 4G PHY_DL_Test 100 PRB MIMO 256-QAM
eNb Mb/s, More Is Better
A
B
C
360.6
359.3
357.9
SE +/- 0.73, N = 3
80
160
240
320
400
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 21.10
Test: 4G PHY_DL_Test 100 PRB MIMO 256-QAM
UE Mb/s, More Is Better
A
B
C
119.0
117.7
117.4
SE +/- 0.23, N = 3
30
60
90
120
150
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 21.10
Test: 4G PHY_DL_Test 100 PRB SISO 256-QAM
eNb Mb/s, More Is Better
A
B
C
364.6
364.0
361.4
SE +/- 0.09, N = 3
80
160
240
320
400
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 21.10
Test: 4G PHY_DL_Test 100 PRB SISO 256-QAM
UE Mb/s, More Is Better
A
B
C
142.1
142.0
139.9
SE +/- 0.09, N = 3
30
60
90
120
150
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 21.10

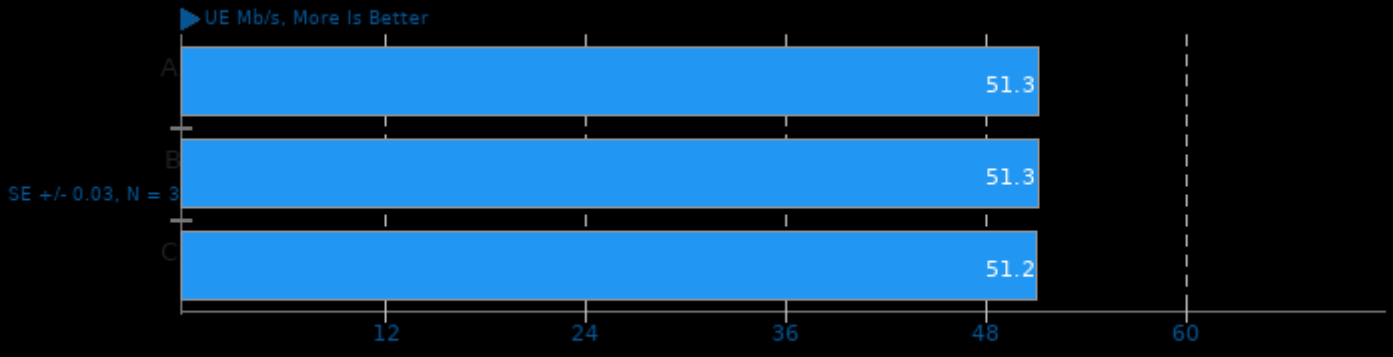
Test: 5G PHY_DL_NR Test 52 PRB SISO 64-QAM



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 21.10

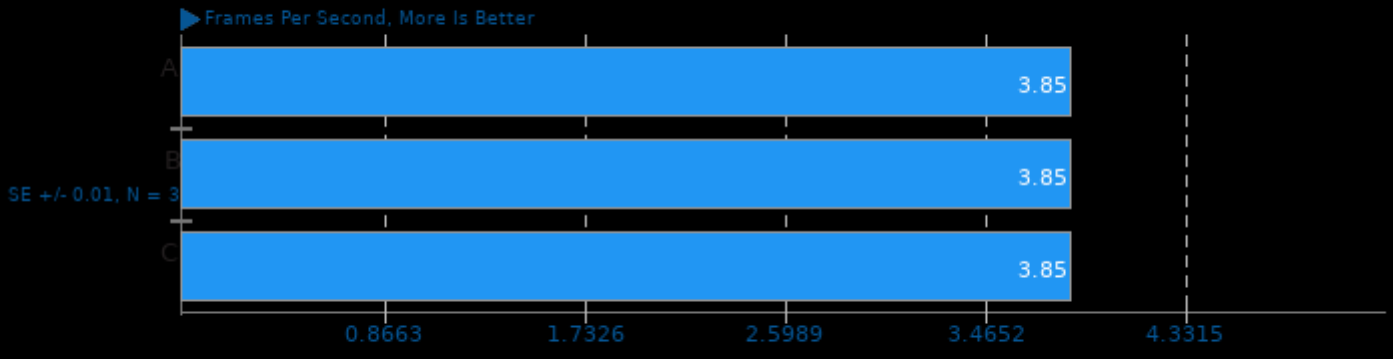
Test: 5G PHY_DL_NR Test 52 PRB SISO 64-QAM



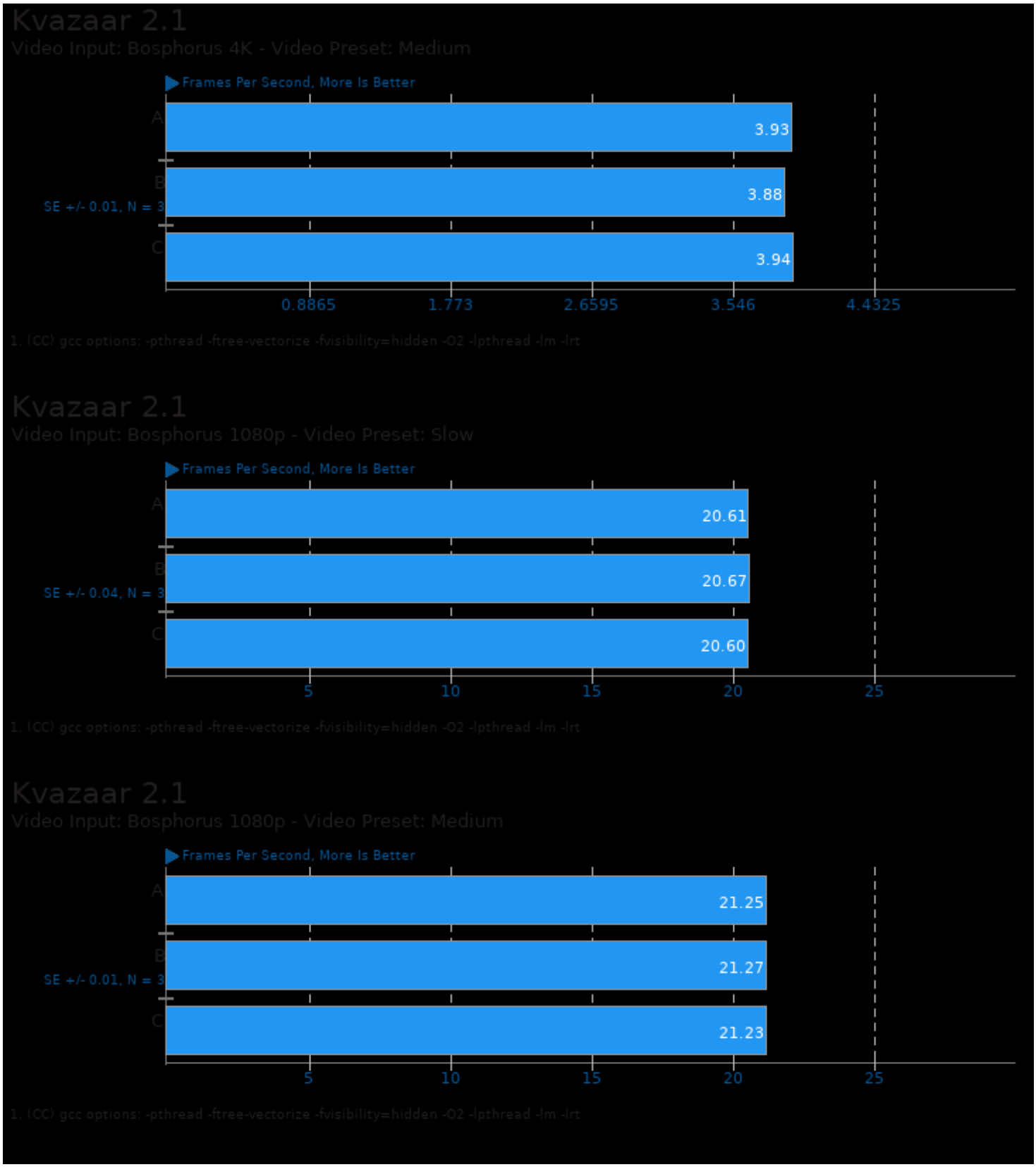
1. (CXX) g++ options: -std=c++11 -fno-strict-aliasing -march=native -mfpmath=sse -mavx2 -fvisibility=hidden -O3 -fno-trapping-math -fno-math-errno

Kvazaar 2.1

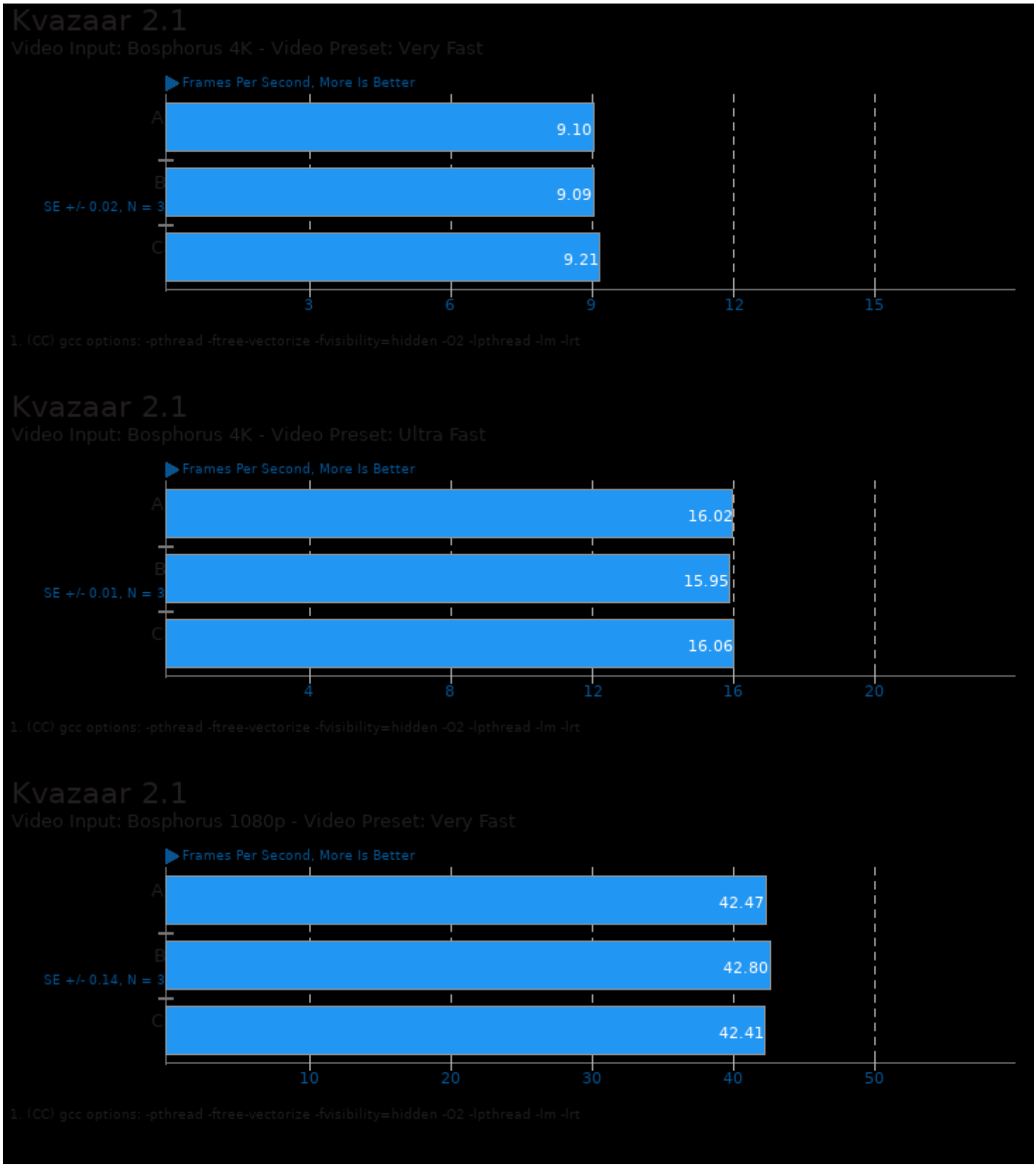
Video Input: Bosphorus 4K - Video Preset: Slow



1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt

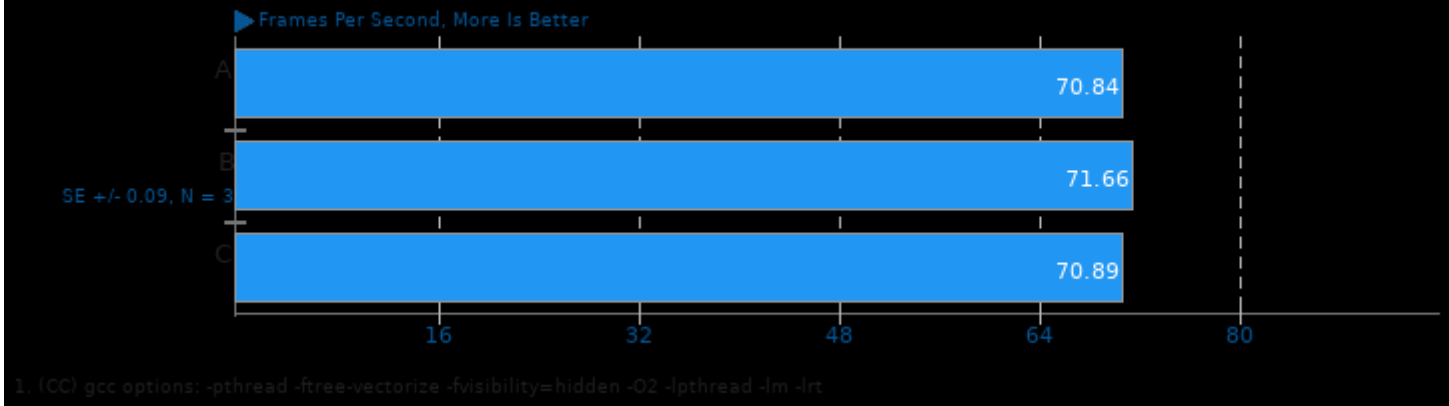

Kvazaar 2.1
Video Input: Bosphorus 4K - Video Preset: Medium
Frames Per Second, More Is Better
A
3.93
B
3.88
C
3.94
SE +/- 0.01, N = 3
0.8865
1.773
2.6595
3.546
4.4325
1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt
Kvazaar 2.1
Video Input: Bosphorus 1080p - Video Preset: Slow
Frames Per Second, More Is Better
A
20.61
B
20.67
C
20.60
SE +/- 0.04, N = 3
5
10
15
20
25
1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt
Kvazaar 2.1
Video Input: Bosphorus 1080p - Video Preset: Medium
Frames Per Second, More Is Better
A
21.25
B
21.27
C
21.23
SE +/- 0.01, N = 3
5
10
15
20
25
1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt

8400 jan


Kvazaar 2.1
Video Input: Bosphorus 4K - Video Preset: Very Fast
Frames Per Second, More Is Better
A
9.10
B
9.09
SE +/- 0.02, N = 3
C
9.21
3
6
9
12
15
1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt
Kvazaar 2.1
Video Input: Bosphorus 4K - Video Preset: Ultra Fast
Frames Per Second, More Is Better
A
16.02
B
15.95
SE +/- 0.01, N = 3
C
16.06
4
8
12
16
20
1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt
Kvazaar 2.1
Video Input: Bosphorus 1080p - Video Preset: Very Fast
Frames Per Second, More Is Better
A
42.47
B
42.80
SE +/- 0.14, N = 3
C
42.41
10
20
30
40
50
1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt

Kvazaar 2.1

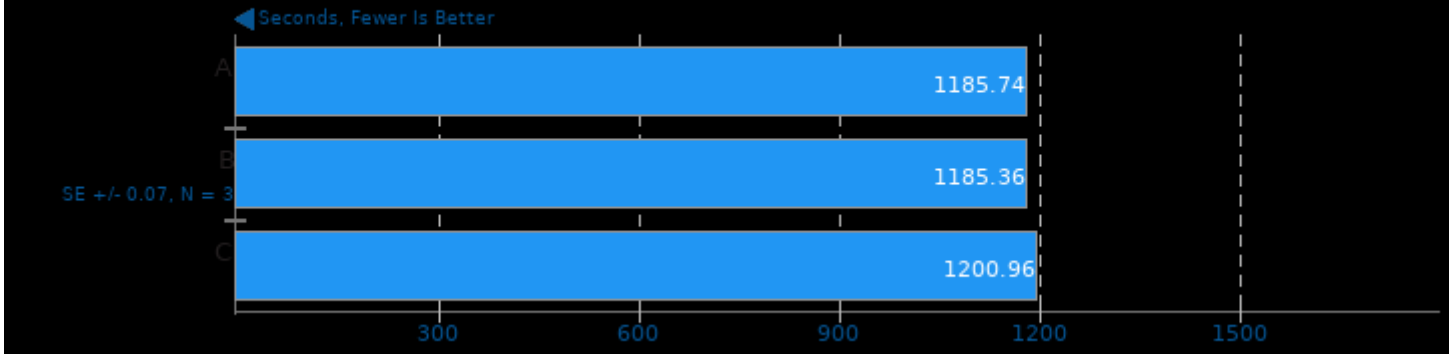
Video Input: Bosphorus 1080p - Video Preset: Ultra Fast



1. (CC) gcc options: -pthread -ftree-vectorize -fvisibility=hidden -O2 -lpthread -lm -lrt

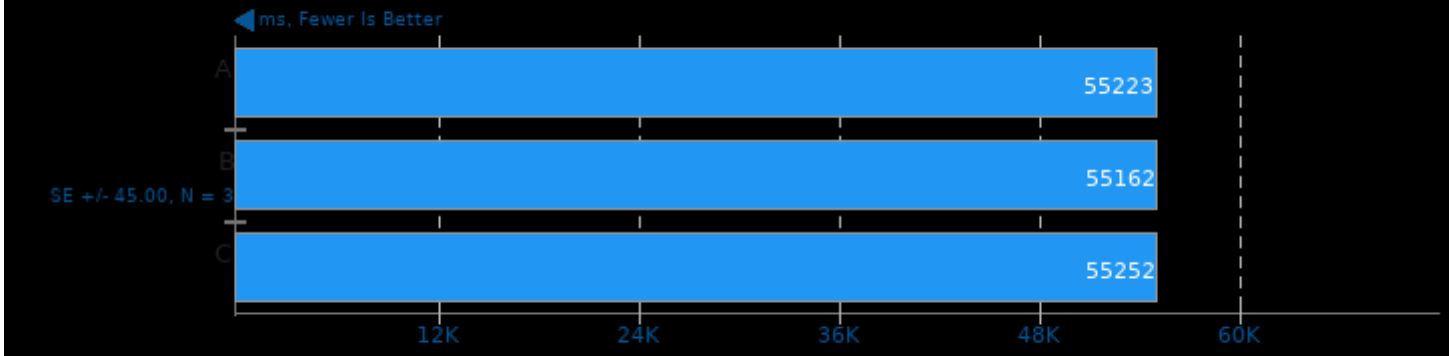
Timed Node.js Compilation 17.3

Time To Compile

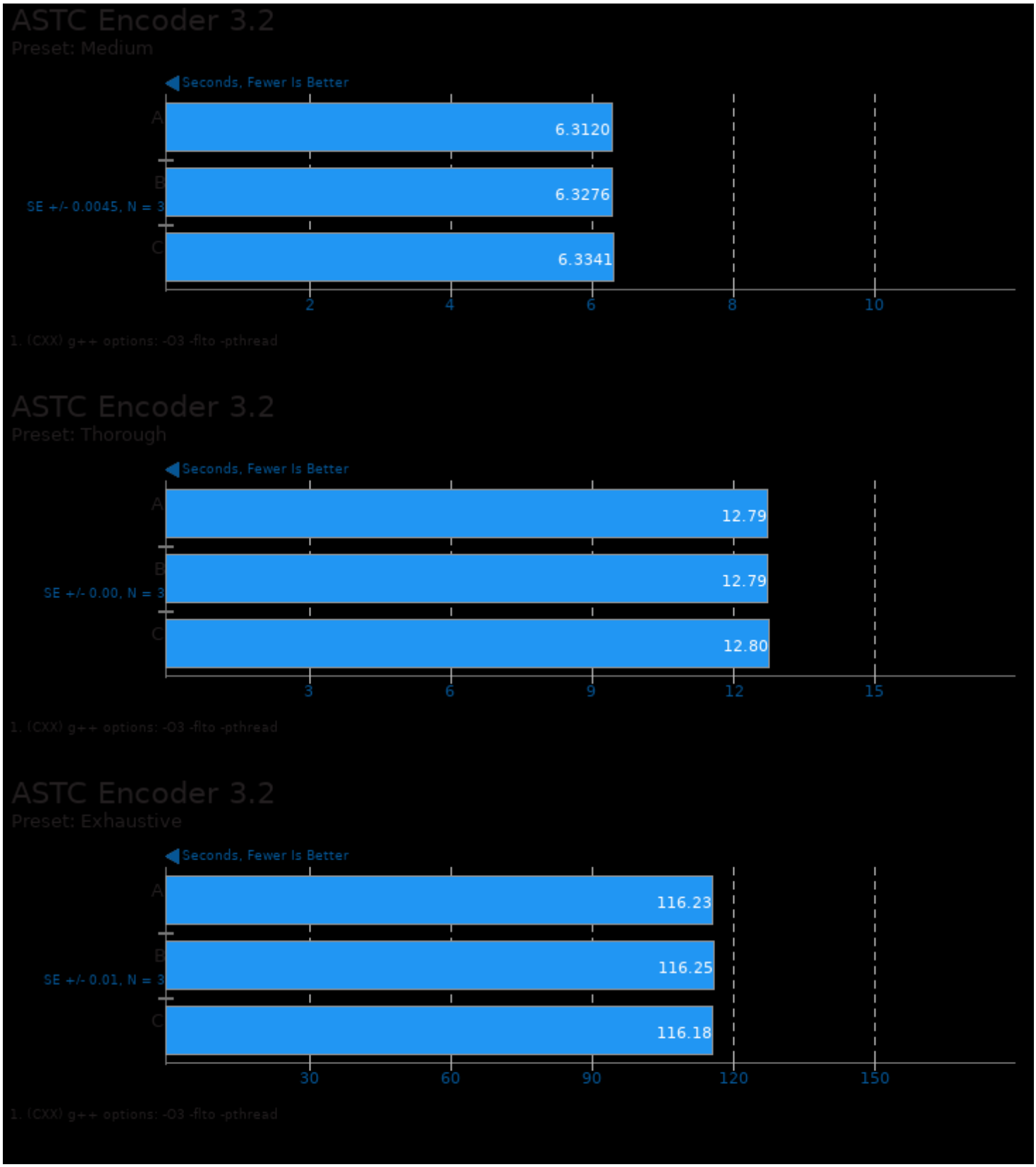


OpenJPEG 2.4

Encode: NASA Curiosity Panorama M34



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


ASTC Encoder 3.2
Preset: Medium
Seconds, Fewer Is Better
A
6.3120
B
6.3276
C
6.3341
SE +/- 0.0045, N = 3
2
4
6
8
10
1. (CXX) g++ options: -O3 -flto -pthread
ASTC Encoder 3.2
Preset: Thorough
Seconds, Fewer Is Better
A
12.79
B
12.79
C
12.80
SE +/- 0.00, N = 3
3
6
9
12
15
1. (CXX) g++ options: -O3 -flto -pthread
ASTC Encoder 3.2
Preset: Exhaustive
Seconds, Fewer Is Better
A
116.23
B
116.25
C
116.18
SE +/- 0.01, N = 3
30
60
90
120
150
1. (CXX) g++ options: -O3 -flto -pthread

8400 jan

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



Geometric mean based upon tests: pts/kvazaar, pts/openjpeg and pts/astcenc



Geometric mean based upon tests: pts/kvazaar and pts/build-nodejs

This file was automatically generated via the Phoronix Test Suite benchmarking software on Friday, 29 March 2024 01:32.