

www.phoronix-test-suite.com

new ep

AMD EPYC 7F32 8-Core testing with a ASRockRack EPYCD8 (P2.40 BIOS) and ASPEED on Debian 11 via the Phoronix Test Suite.

Automated Executive Summary

B had the most wins, coming in first place for 64% of the tests.

Based on the geometric mean of all complete results, the fastest (B) was 1.02x the speed of the slowest (A).

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

Apache Spark (Row Count: 1000000 - Partitions: 100 - Repartition Test Time) at 1.286x
Apache Spark (Row Count: 1000000 - Partitions: 100 - Broadcast Inner Join Test Time) at 1.209x
Apache Spark (Row Count: 1000000 - Partitions: 2000 - Repartition Test Time) at 1.194x
Apache Spark (Row Count: 20000000 - Partitions: 100 - Broadcast Inner Join Test Time) at 1.177x
Apache Spark (Row Count: 20000000 - Partitions: 100 - Inner Join Test Time) at 1.162x
Apache Spark (Row Count: 10000000 - Partitions: 100 - Inner Join Test Time) at 1.138x
Apache Spark (Row Count: 10000000 - Partitions: 500 - Broadcast Inner Join Test Time) at 1.133x
Apache Spark (Row Count: 1000000 - Partitions: 1000 - Broadcast Inner Join Test Time) at 1.114x
Apache Spark (Row Count: 40000000 - Partitions: 2000 - Group By Test Time) at 1.106x
Apache Spark (Row Count: 40000000 - Partitions: 1000 - Group By Test Time) at 1.104x.

Test Systems:

A

B

Processor: AMD EPYC 7F32 8-Core @ 3.70GHz (8 Cores / 16 Threads), Motherboard: ASRockRack EPYCD8 (P2.40 BIOS), Chipset: AMD Starship/Matisse, Memory: 28GB, Disk: Samsung SSD 970 EVO Plus 250GB, Graphics: ASPEED, Network: 2 x Intel I350

OS: Debian 11, Kernel: 5.10.0-10-amd64 (x86_64), Desktop: GNOME Shell 3.38.6, Display Server: X Server, Compiler: GCC 10.2.1 20210110, File-System: ext4, Screen Resolution: 1024x768

Kernel Notes: Transparent Huge Pages: always
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-Km9U7s/gcc-10-10.2.1/debian/tmp-nvptx/usr,amdgcn-amdhsa=/build/gcc-10-Km9U7s/gcc-10-10.2.1/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: acpi-cpufreq schedutil (Boost: Enabled) - CPU Microcode: 0x8301034
Java Notes: OpenJDK Runtime Environment (build 11.0.13+8-post-Debian-1deb11u1)
Python Notes: Python 3.9.2
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/swapgs barriers and __user pointer sanitization + spectre_v2: Mitigation of Full AMD retpoline IBPB: conditional IBRS_FW STIBP: conditional RSB filling + srbds: Not affected + tsx_async_abort: Not affected

	A	B
LAMMPS Molecular Dynamics Simulator - 20k Atoms (ns/day)	**6.409**	**6.481**
Normalized	98.89%	100%
LAMMPS Molecular Dynamics Simulator - Rhodopsin Protein (ns/day)	**6.21**	**6.347**
Normalized	97.84%	100%
SVT-AV1 - Preset 4 - Bosphorus 4K (FPS)	**1.457**	**1.453**
Normalized	100%	99.73%
SVT-AV1 - Preset 8 - Bosphorus 4K (FPS)	**26.519**	**27.869**
Normalized	95.16%	100%
SVT-AV1 - Preset 10 - Bosphorus 4K (FPS)	**50.919**	**54.454**
Normalized	93.51%	100%
SVT-AV1 - Preset 12 - Bosphorus 4K (FPS)	**71.075**	**77.384**
Normalized	91.85%	100%
SVT-AV1 - Preset 4 - Bosphorus 1080p (FPS)	**4.289**	**4.316**
Normalized	99.37%	100%
SVT-AV1 - Preset 8 - Bosphorus 1080p (FPS)	**73.239**	**74.347**
Normalized	98.51%	100%
SVT-AV1 - Preset 10 - Bosphorus 1080p (FPS)	**155.009**	**158.579**
Normalized	97.75%	100%
SVT-AV1 - Preset 12 - Bosphorus 1080p (FPS)	**236.896**	**245.433**
Normalized	96.52%	100%
Primesieve - 1e12 (sec)	**21.334**	**21.423**

new ep

Normalized	100%	99.58%
Primesieve - 1e13 (sec)	268.884	270.327
Normalized	100%	99.47%
Aircrack-ng (k/s)	28958	28872
Normalized	100%	99.7%
Node.js V8 Web Tooling Benchmark (runs/s)	8.78	8.76
Normalized	100%	99.77%
Apache Spark - 1000000 - 100 - S.S.B.T (sec)	4.08	3.76
Normalized	92.16%	100%
Apache Spark - 1000000 - 100 - C.P.B (sec)	186.644599348	187.21356049
Normalized	100%	99.7%
Apache Spark - 1000000 - 100 - C.P.B.U.D (sec)	11.382844748	11.460916257
Normalized	100%	99.32%
Apache Spark - 1000000 - 100 - Group By Test Time (sec)	4.87	4.62
Normalized	94.87%	100%
Apache Spark - 1000000 - 100 - R.T.T (sec)	2.34	1.82
Normalized	77.78%	100%
Apache Spark - 1000000 - 100 - I.J.T.T (sec)	2.12	2.08
Normalized	98.11%	100%
Apache Spark - 1000000 - 100 - B.I.J.T.T (sec)	2.20	1.82
Normalized	82.73%	100%
Apache Spark - 1000000 - 500 - S.S.B.T (sec)	4.297287719	4.11
Normalized	95.64%	100%
Apache Spark - 1000000 - 500 - C.P.B (sec)	188.339926254	188.797248018
Normalized	100%	99.76%
Apache Spark - 1000000 - 500 - C.P.B.U.D (sec)	11.486340669	11.481548594
Normalized	99.96%	100%
Apache Spark - 1000000 - 500 - Group By Test Time (sec)	5.08	5.21
Normalized	100%	97.5%
Apache Spark - 1000000 - 500 - R.T.T (sec)	2.21	2.04
Normalized	92.31%	100%
Apache Spark - 1000000 - 500 - I.J.T.T (sec)	2.314010652	2.12
Normalized	91.62%	100%
Apache Spark - 1000000 - 500 - B.I.J.T.T (sec)	1.95	2.07
Normalized	100%	94.2%
Apache Spark - 1000000 - 1000 - S.S.B.T (sec)	4.492061685	4.21
Normalized	93.72%	100%
Apache Spark - 1000000 - 1000 - C.P.B (sec)	187.439773683	186.710805845
Normalized	99.61%	100%
Apache Spark - 1000000 - 1000 - C.P.B.U.D (sec)	11.568129108	11.445969536
Normalized	98.94%	100%
Apache Spark - 1000000 - 1000 - Group By Test Time (sec)	5.44	5.63
Normalized	100%	96.63%
Apache Spark - 1000000 - 1000 - R.T.T (sec)	2.41	2.24
Normalized	92.95%	100%
Apache Spark - 1000000 - 1000 - I.J.T.T (sec)	2.77	2.57
Normalized	92.78%	100%
Apache Spark - 1000000 - 1000 - B.I.J.T.T (sec)	2.24	2.01
Normalized	89.73%	100%
Apache Spark - 1000000 - 2000 - S.S.B.T (sec)	4.93	4.60

Normalized	93.31%	100%
Apache Spark - 1000000 - 2000 - C.P.B (sec)	185.574874094	188.307131272
Normalized	100%	98.55%
Apache Spark - 1000000 - 2000 - C.P.B.U.D (sec)	11.548994067	11.408389073
Normalized	98.78%	100%
Apache Spark - 1000000 - 2000 - Group By Test Time (sec)	6.12	5.60
Normalized	91.5%	100%
Apache Spark - 1000000 - 2000 - R.T.T (sec)	2.89	2.42
Normalized	83.74%	100%
Apache Spark - 1000000 - 2000 - I.J.T.T (sec)	3.69	3.39
Normalized	91.87%	100%
Apache Spark - 1000000 - 2000 - B.I.J.T.T (sec)	2.47	2.36
Normalized	95.55%	100%
Apache Spark - 10000000 - 100 - S.5.B.T (sec)	17.828678109	16.70
Normalized	93.67%	100%
Apache Spark - 10000000 - 100 - C.P.B (sec)	188.819108164	188.041543346
Normalized	99.59%	100%
Apache Spark - 10000000 - 100 - C.P.B.U.D (sec)	11.372200844	11.482167871
Normalized	100%	99.04%
Apache Spark - 10000000 - 100 - Group By Test Time (sec)	9.833025159	9.36
Normalized	95.19%	100%
Apache Spark - 10000000 - 100 - R.T.T (sec)	11.208821814	10.990142921
Normalized	98.05%	100%
Apache Spark - 10000000 - 100 - I.J.T.T (sec)	14.996194688	13.18
Normalized	87.89%	100%
Apache Spark - 10000000 - 100 - B.I.J.T.T (sec)	15.173851215	14.35
Normalized	94.57%	100%
Apache Spark - 10000000 - 500 - S.5.B.T (sec)	17.684815049	16.26
Normalized	91.94%	100%
Apache Spark - 10000000 - 500 - C.P.B (sec)	189.325648896	189.343560947
Normalized	100%	99.99%
Apache Spark - 10000000 - 500 - C.P.B.U.D (sec)	11.39	11.590029387
Normalized	100%	98.27%
Apache Spark - 10000000 - 500 - Group By Test Time (sec)	9.57	9.25
Normalized	96.66%	100%
Apache Spark - 10000000 - 500 - R.T.T (sec)	10.671932464	10.46
Normalized	98.01%	100%
Apache Spark - 10000000 - 500 - I.J.T.T (sec)	12.795955438	12.27
Normalized	95.89%	100%
Apache Spark - 10000000 - 500 - B.I.J.T.T (sec)	13.546636341	11.96
Normalized	88.29%	100%
Apache Spark - 20000000 - 100 - S.5.B.T (sec)	32.50	31.74
Normalized	97.66%	100%
Apache Spark - 20000000 - 100 - C.P.B (sec)	191.628376235	191.584324668
Normalized	99.98%	100%
Apache Spark - 20000000 - 100 - C.P.B.U.D (sec)	11.588828559	11.45
Normalized	98.8%	100%

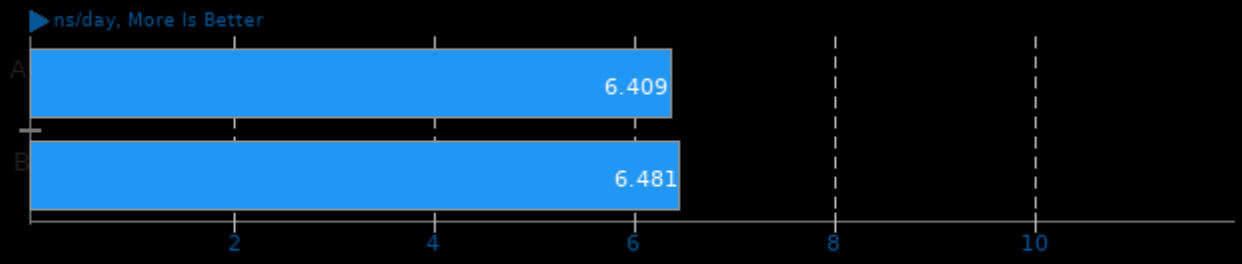
Apache Spark - 20000000 - 100 - Group By Test Time (sec)	14.43	14.53
Normalized	100%	99.31%
Apache Spark - 20000000 - 100 - R.T.T (sec)	21.320366883	22.098401525
Normalized	100%	96.48%
Apache Spark - 20000000 - 100 - I.J.T.T (sec)	26.939544404	31.300359261
Normalized	100%	86.07%
Apache Spark - 20000000 - 100 - B.I.J.T.T (sec)	25.970713725	30.57
Normalized	100%	84.95%
Apache Spark - 20000000 - 500 - S.5.B.T (sec)	29.83	28.238779779
Normalized	94.67%	100%
Apache Spark - 20000000 - 500 - C.P.B (sec)	186.935221902	187.715675799
Normalized	100%	99.58%
Apache Spark - 20000000 - 500 - C.P.B.U.D (sec)	11.596259096	11.44
Normalized	98.65%	100%
Apache Spark - 20000000 - 500 - Group By Test Time (sec)	13.851438186	14.045160005
Normalized	100%	98.62%
Apache Spark - 20000000 - 500 - R.T.T (sec)	20.465179648	19.741685367
Normalized	96.46%	100%
Apache Spark - 20000000 - 500 - I.J.T.T (sec)	24.79735074	23.660469254
Normalized	95.42%	100%
Apache Spark - 20000000 - 500 - B.I.J.T.T (sec)	24.261263426	24.305547585
Normalized	100%	99.82%
Apache Spark - 40000000 - 100 - S.5.B.T (sec)	55.496945258	54.07
Normalized	97.43%	100%
Apache Spark - 40000000 - 100 - C.P.B (sec)	188.990234527	188.419756443
Normalized	99.7%	100%
Apache Spark - 40000000 - 100 - C.P.B.U.D (sec)	11.46	11.51
Normalized	100%	99.57%
Apache Spark - 40000000 - 100 - Group By Test Time (sec)	34.841378524	34.17
Normalized	98.07%	100%
Apache Spark - 40000000 - 100 - R.T.T (sec)	40.60	40.27
Normalized	99.19%	100%
Apache Spark - 40000000 - 100 - I.J.T.T (sec)	45.281924112	46.980349758
Normalized	100%	96.38%
Apache Spark - 40000000 - 100 - B.I.J.T.T (sec)	48.938027776	50.449473523
Normalized	100%	97%
Apache Spark - 40000000 - 500 - S.5.B.T (sec)	58.856305942	53.69997012
Normalized	91.24%	100%
Apache Spark - 40000000 - 500 - C.P.B (sec)	191.487416434	189.911082384
Normalized	99.18%	100%
Apache Spark - 40000000 - 500 - C.P.B.U.D (sec)	11.478861584	11.43
Normalized	99.57%	100%
Apache Spark - 40000000 - 500 - Group By Test Time (sec)	30.284735623	30.10
Normalized	99.39%	100%
Apache Spark - 40000000 - 500 - R.T.T (sec)	40.4149037	37.09
Normalized	91.77%	100%
Apache Spark - 40000000 - 500 - I.J.T.T (sec)	49.623129949	46.10
Normalized	92.9%	100%

Apache Spark - 40000000 - 500 - B.I.J.T.T (sec)	51.874385394	51.514099262
Normalized	99.31%	100%
Apache Spark - 10000000 - 1000 - S.5.B.T (sec)	16.47	16.41
Normalized	99.64%	100%
Apache Spark - 10000000 - 1000 - C.P.B (sec)	188.63526141	189.044599158
Normalized	100%	99.78%
Apache Spark - 10000000 - 1000 - C.P.B.U.D (sec)	11.45	11.618367387
Normalized	100%	98.55%
Apache Spark - 10000000 - 1000 - Group By Test Time (sec)	9.393762531	9.17
Normalized	97.62%	100%
Apache Spark - 10000000 - 1000 - R.T.T (sec)	10.65	10.38
Normalized	97.46%	100%
Apache Spark - 10000000 - 1000 - I.J.T.T (sec)	12.72	13.21
Normalized	100%	96.29%
Apache Spark - 10000000 - 1000 - B.I.J.T.T (sec)	12.17	11.84
Normalized	97.29%	100%
Apache Spark - 10000000 - 2000 - S.5.B.T (sec)	16.85	16.35
Normalized	97.03%	100%
Apache Spark - 10000000 - 2000 - C.P.B (sec)	187.892174284	186.890288481
Normalized	99.47%	100%
Apache Spark - 10000000 - 2000 - C.P.B.U.D (sec)	11.367760678	11.484183994
Normalized	100%	98.99%
Apache Spark - 10000000 - 2000 - Group By Test Time (sec)	9.41	9.53
Normalized	100%	98.74%
Apache Spark - 10000000 - 2000 - R.T.T (sec)	10.647281489	10.42
Normalized	97.87%	100%
Apache Spark - 10000000 - 2000 - I.J.T.T (sec)	14.300095854	13.33
Normalized	93.22%	100%
Apache Spark - 10000000 - 2000 - B.I.J.T.T (sec)	13.315319658	12.15
Normalized	91.25%	100%
Apache Spark - 20000000 - 1000 - S.5.B.T (sec)	28.759559706	29.232604249
Normalized	100%	98.38%
Apache Spark - 20000000 - 1000 - C.P.B (sec)	189.055357783	188.352232
Normalized	99.63%	100%
Apache Spark - 20000000 - 1000 - C.P.B.U.D (sec)	11.499868446	11.559230601
Normalized	100%	99.49%
Apache Spark - 20000000 - 1000 - Group By Test Time (sec)	12.66	13.45
Normalized	100%	94.13%
Apache Spark - 20000000 - 1000 - R.T.T (sec)	19.568627792	19.45
Normalized	99.39%	100%
Apache Spark - 20000000 - 1000 - I.J.T.T (sec)	24.58	25.05
Normalized	100%	98.12%
Apache Spark - 20000000 - 1000 - B.I.J.T.T (sec)	24.95	25.11
Normalized	100%	99.36%
Apache Spark - 20000000 - 2000 - S.5.B.T (sec)	28.415568725	27.88
Normalized	98.12%	100%
Apache Spark - 20000000 - 2000 - C.P.B (sec)	192.318093435	189.7412676
Normalized	98.66%	100%
Apache Spark - 20000000 - 2000 - C.P.B.U.D (sec)	11.42	11.391436234
Normalized	99.75%	100%

Apache Spark - 20000000 - 2000 - Group By Test Time (sec)	12.976053344	13.303787109
Normalized	100%	97.54%
Apache Spark - 20000000 - 2000 - R.T.T (sec)	19.19919626	19.60
Normalized	100%	97.96%
Apache Spark - 20000000 - 2000 - I.J.T.T (sec)	23.879705608	24.018967172
Normalized	100%	99.42%
Apache Spark - 20000000 - 2000 - B.I.J.T.T (sec)	22.67	23.63
Normalized	100%	95.94%
Apache Spark - 40000000 - 1000 - S.5.B.T (sec)	53.659775144	55.737381601
Normalized	100%	96.27%
Apache Spark - 40000000 - 1000 - C.P.B (sec)	190.285451821	188.993989736
Normalized	99.32%	100%
Apache Spark - 40000000 - 1000 - C.P.B.U.D (sec)	11.453251657	11.45
Normalized	99.97%	100%
Apache Spark - 40000000 - 1000 - Group By Test Time (sec)	33.216112166	36.66
Normalized	100%	90.61%
Apache Spark - 40000000 - 1000 - R.T.T (sec)	40.100468392	36.92
Normalized	92.07%	100%
Apache Spark - 40000000 - 1000 - I.J.T.T (sec)	47.644353213	46.81
Normalized	98.25%	100%
Apache Spark - 40000000 - 1000 - B.I.J.T.T (sec)	47.98	47.09
Normalized	98.15%	100%
Apache Spark - 40000000 - 2000 - S.5.B.T (sec)	53.456292297	52.982574422
Normalized	99.11%	100%
Apache Spark - 40000000 - 2000 - C.P.B (sec)	187.364578192	188.785043326
Normalized	100%	99.25%
Apache Spark - 40000000 - 2000 - C.P.B.U.D (sec)	11.410779673	11.46
Normalized	100%	99.57%
Apache Spark - 40000000 - 2000 - Group By Test Time (sec)	31.124457471	28.135234308
Normalized	90.4%	100%
Apache Spark - 40000000 - 2000 - R.T.T (sec)	37.249541226	40.12
Normalized	100%	92.85%
Apache Spark - 40000000 - 2000 - I.J.T.T (sec)	46.082391472	44.71
Normalized	97.02%	100%
Apache Spark - 40000000 - 2000 - B.I.J.T.T (sec)	46.650330339	45.65
Normalized	97.86%	100%
ASTC Encoder - Fast (MT/s)	114.53	115.0968
Normalized	99.51%	100%
ASTC Encoder - Medium (MT/s)	38.3078	38.3327
Normalized	99.94%	100%
ASTC Encoder - Thorough (MT/s)	4.6014	4.6013
Normalized	100%	100%
ASTC Encoder - Exhaustive (MT/s)	0.503	0.5018
Normalized	100%	99.76%

LAMMPS Molecular Dynamics Simulator 23Jun2022

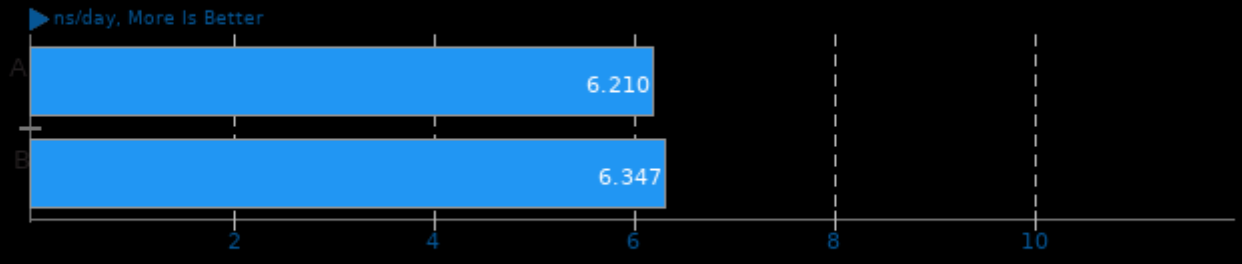
Model: 20k Atoms



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

LAMMPS Molecular Dynamics Simulator 23Jun2022

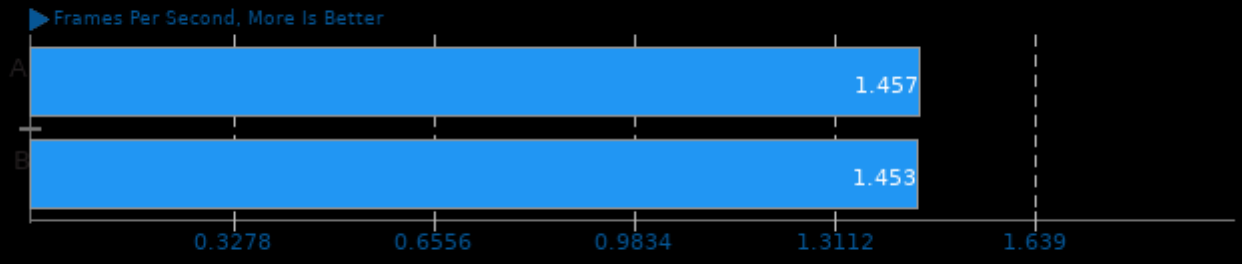
Model: Rhodopsin Protein



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

SVT-AV1 1.2

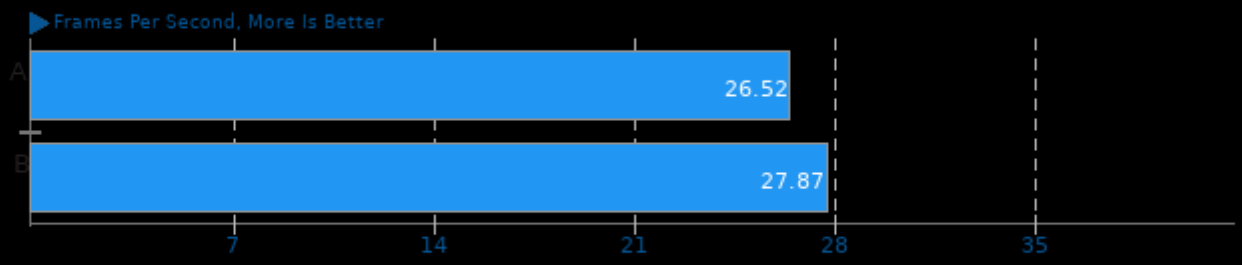
Encoder Mode: Preset 4 - Input: Bosphorus 4K



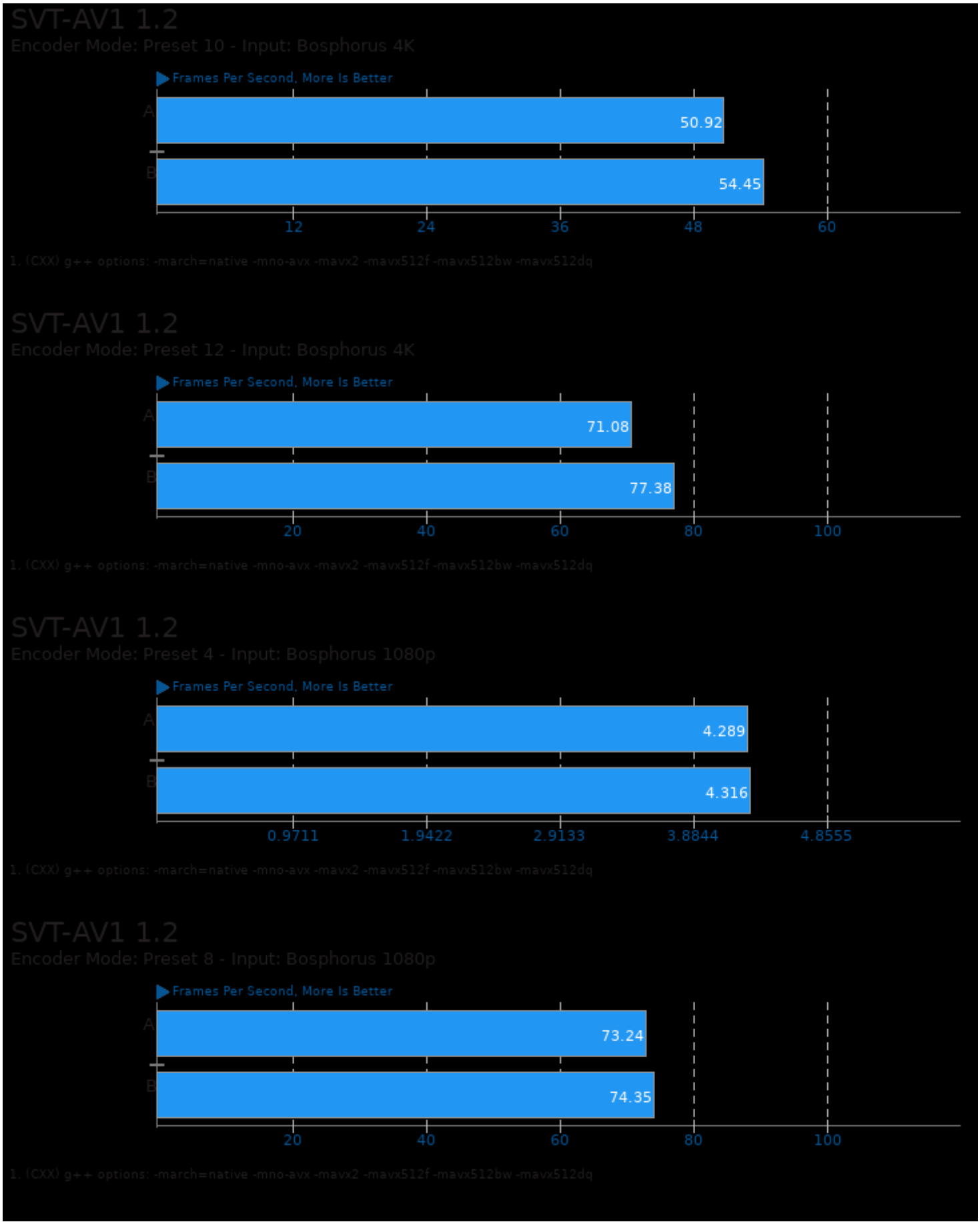
1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq

SVT-AV1 1.2

Encoder Mode: Preset 8 - Input: Bosphorus 4K

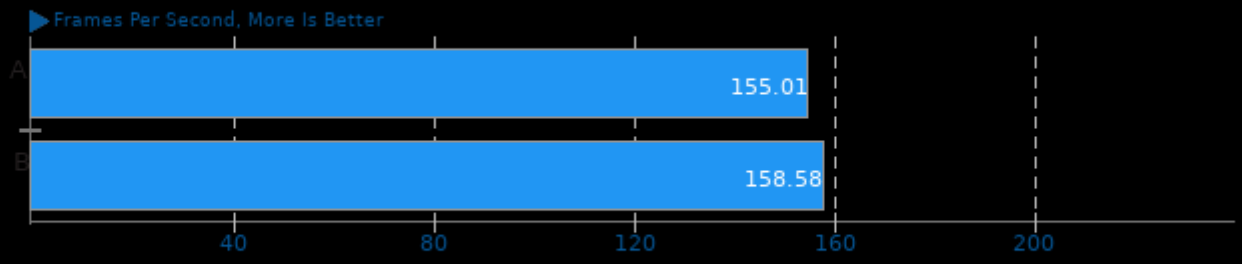


1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq


SVT-AV1 1.2
Encoder Mode: Preset 10 - Input: Bosphorus 4K
Frames Per Second, More Is Better
A
50.92
B
54.45
12
24
36
48
60
1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq
SVT-AV1 1.2
Encoder Mode: Preset 12 - Input: Bosphorus 4K
Frames Per Second, More Is Better
A
71.08
B
77.38
20
40
60
80
100
1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq
SVT-AV1 1.2
Encoder Mode: Preset 4 - Input: Bosphorus 1080p
Frames Per Second, More Is Better
A
4.289
B
4.316
0.9711
1.9422
2.9133
3.8844
4.8555
1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq
SVT-AV1 1.2
Encoder Mode: Preset 8 - Input: Bosphorus 1080p
Frames Per Second, More Is Better
A
73.24
B
74.35
20
40
60
80
100
1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq

SVT-AV1 1.2

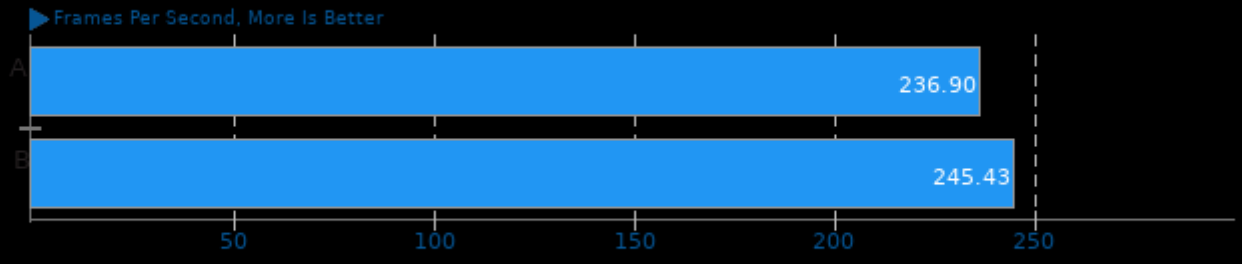
Encoder Mode: Preset 10 - Input: Bosphorus 1080p



1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq

SVT-AV1 1.2

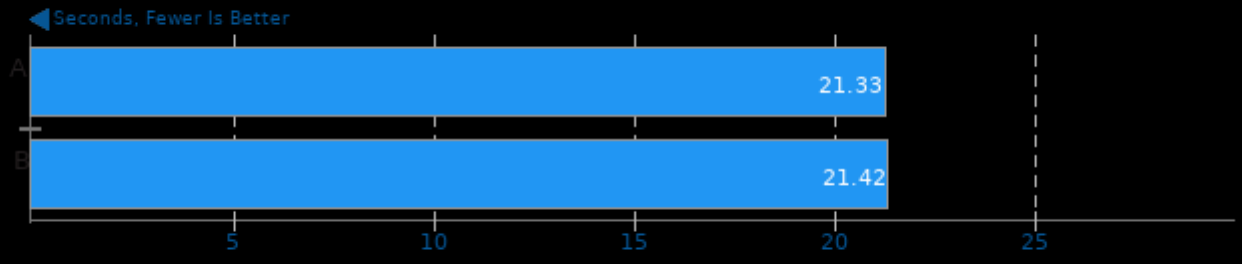
Encoder Mode: Preset 12 - Input: Bosphorus 1080p



1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq

Primesieve 8.0

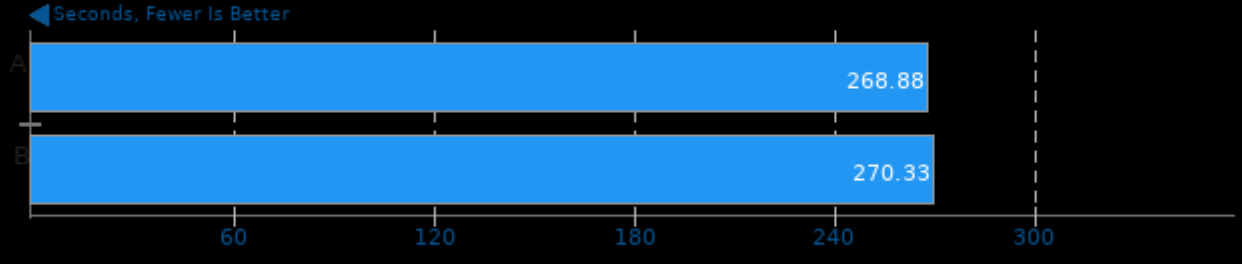
Length: 1e12



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

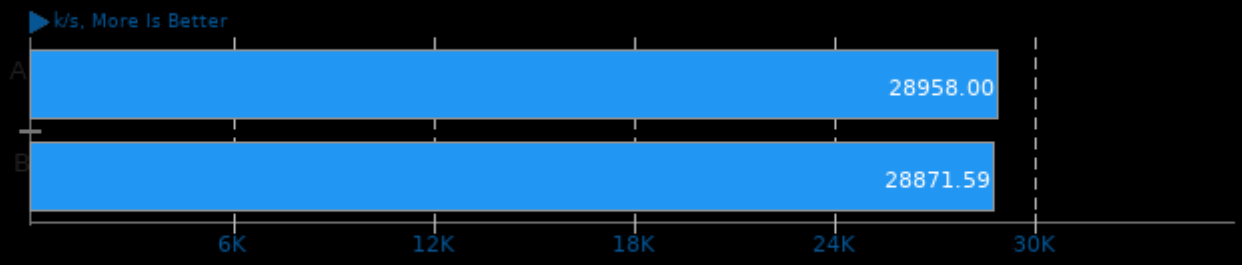
Primesieve 8.0

Length: 1e13



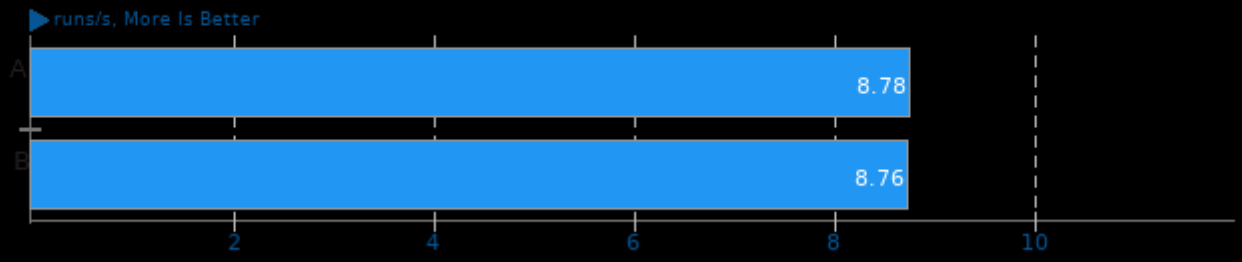
1. (CXX) g++ options: -O3 -lpthread

Aircrack-ng 1.7



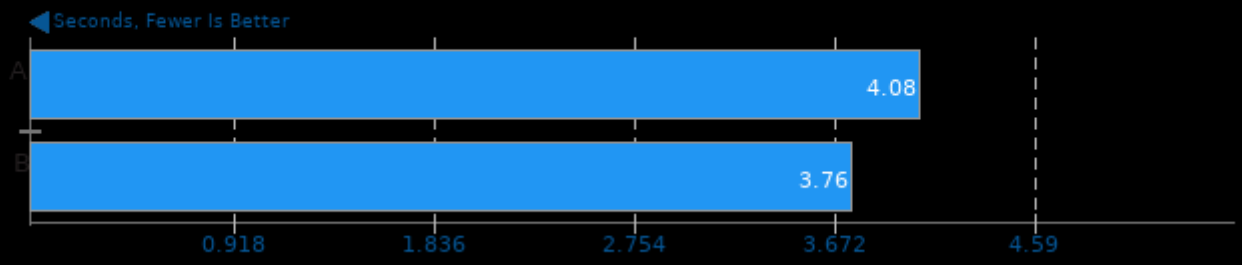
1. (CXX) g++ options: -std=gnu++17 -O3 -fvisibility=hidden -fcommon -rdynamic -lnl-3 -lnl-genl-3 -lpcre -lpthread -lz -lssl -lcrypto -lhwloc -ldl -lm -pthre

Node.js V8 Web Tooling Benchmark



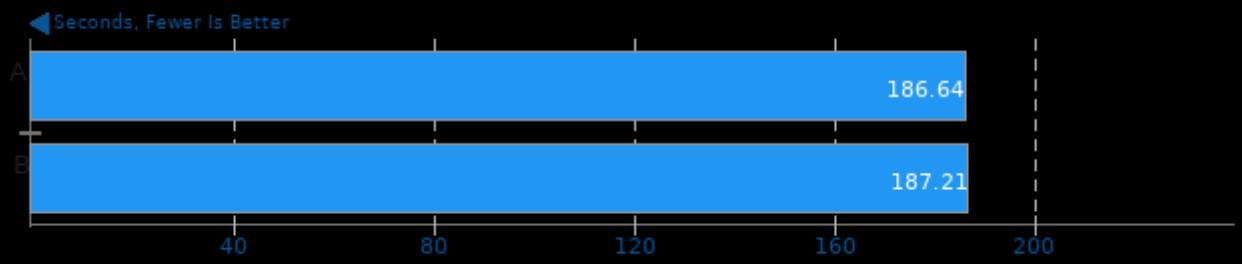
Apache Spark 3.3

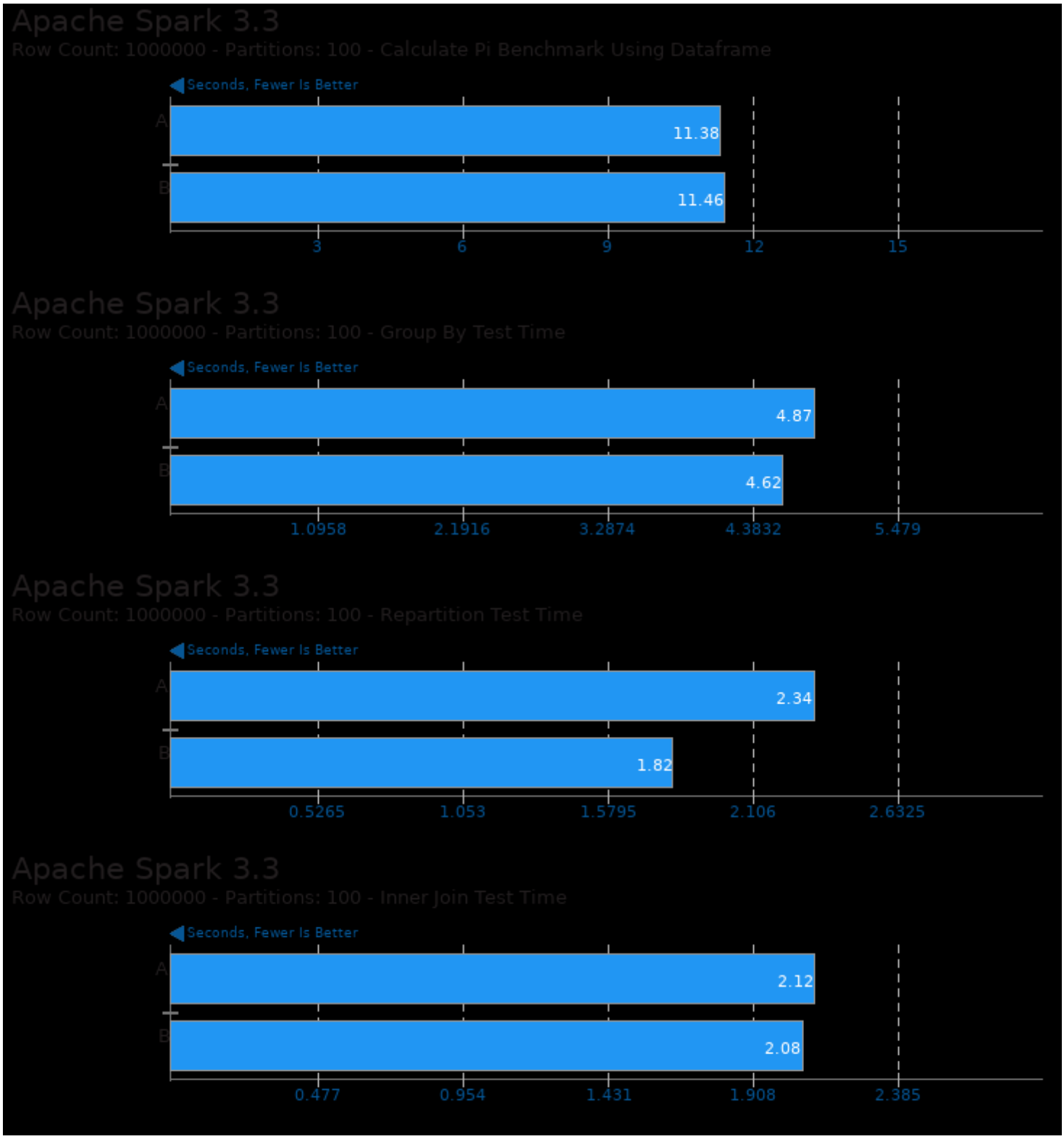
Row Count: 1000000 - Partitions: 100 - SHA-512 Benchmark Time

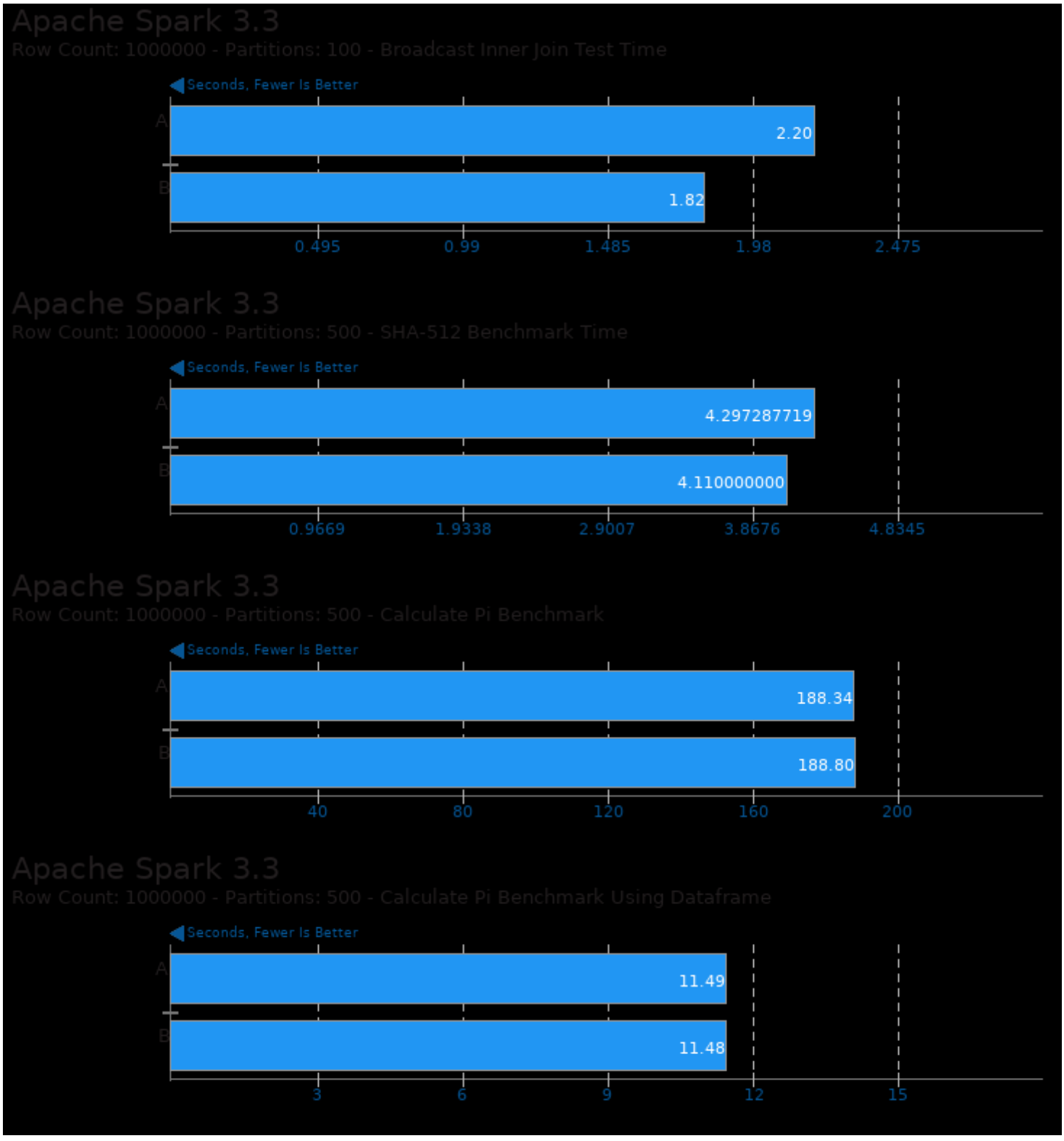


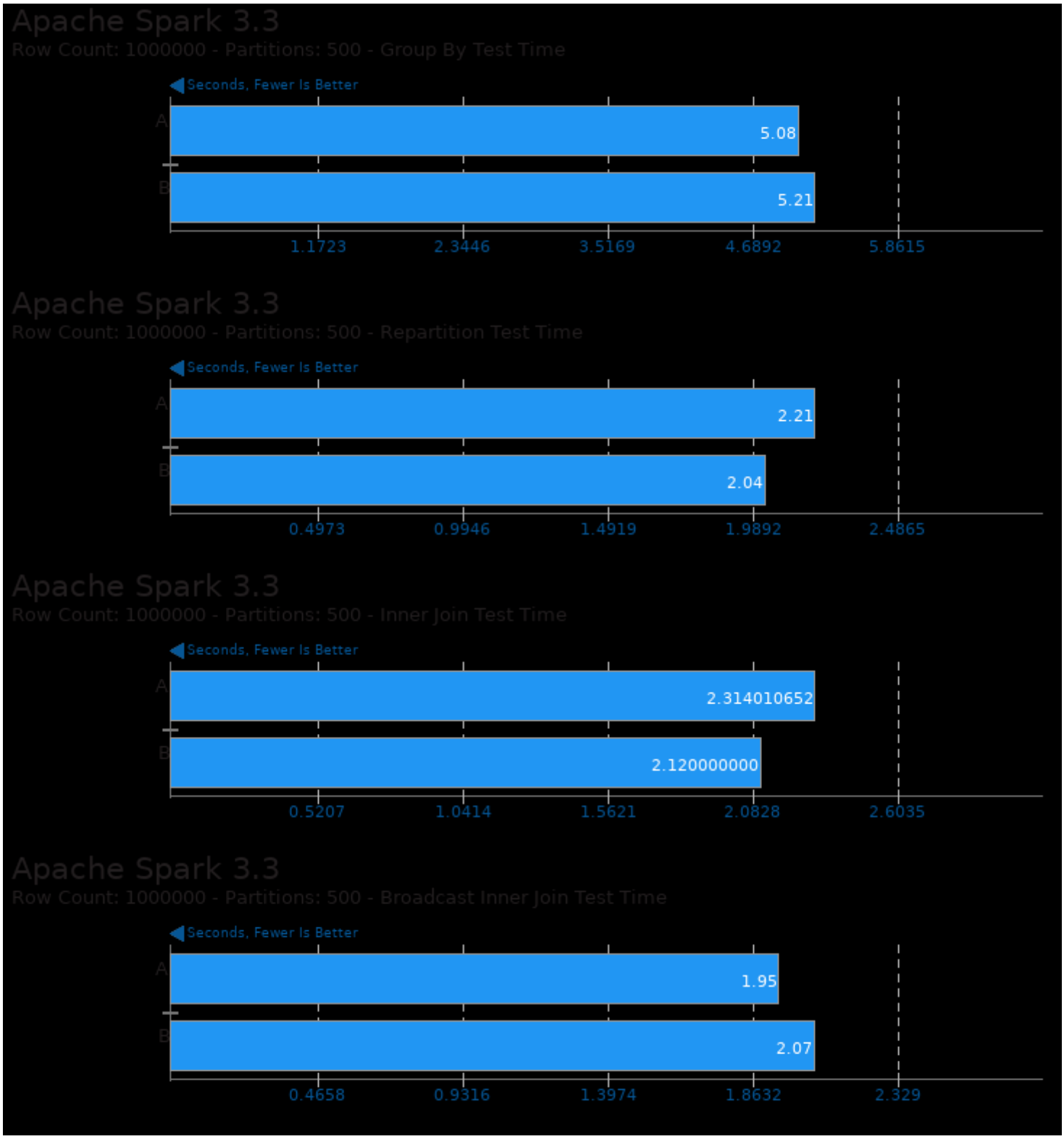
Apache Spark 3.3

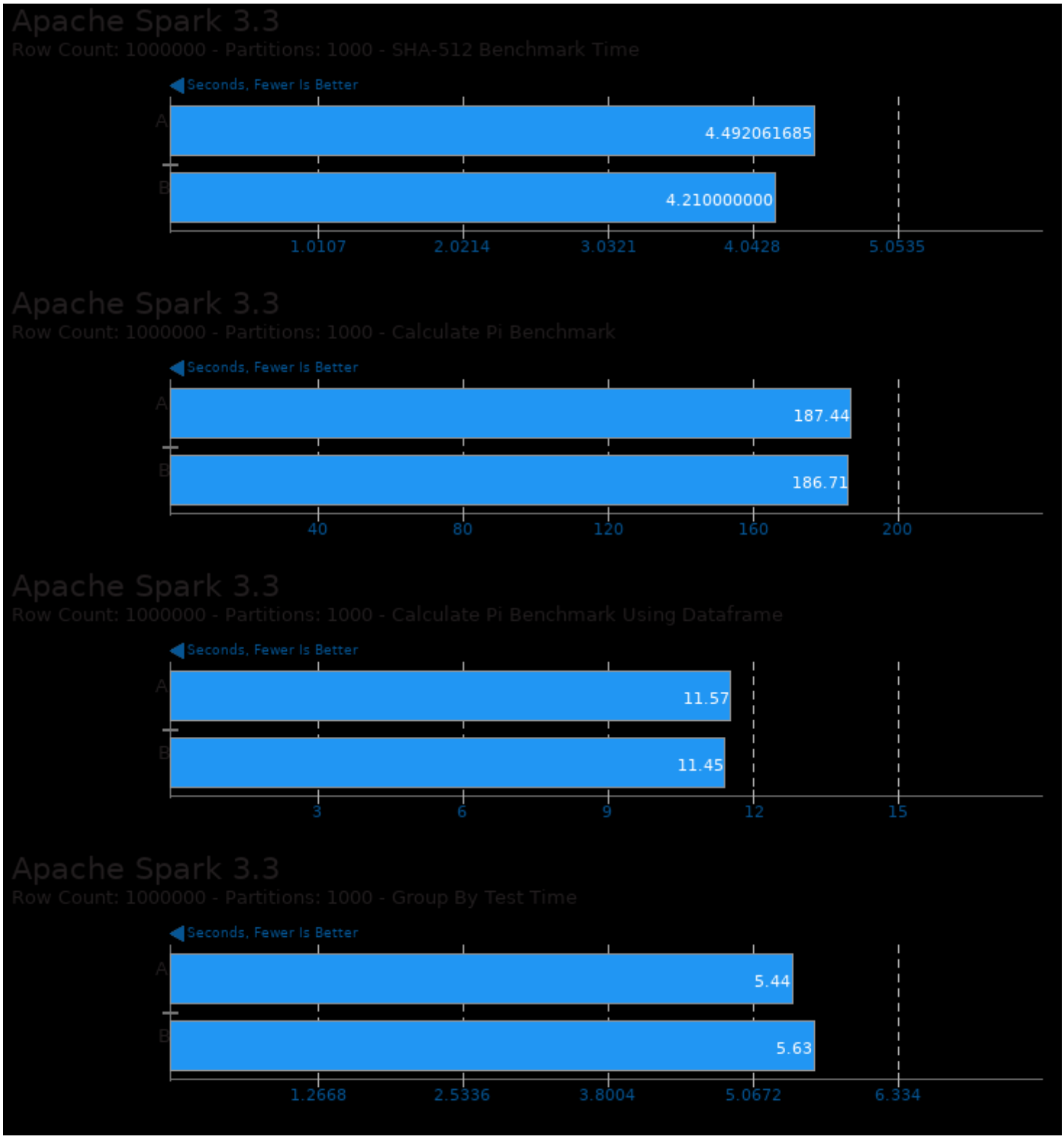
Row Count: 1000000 - Partitions: 100 - Calculate Pi Benchmark



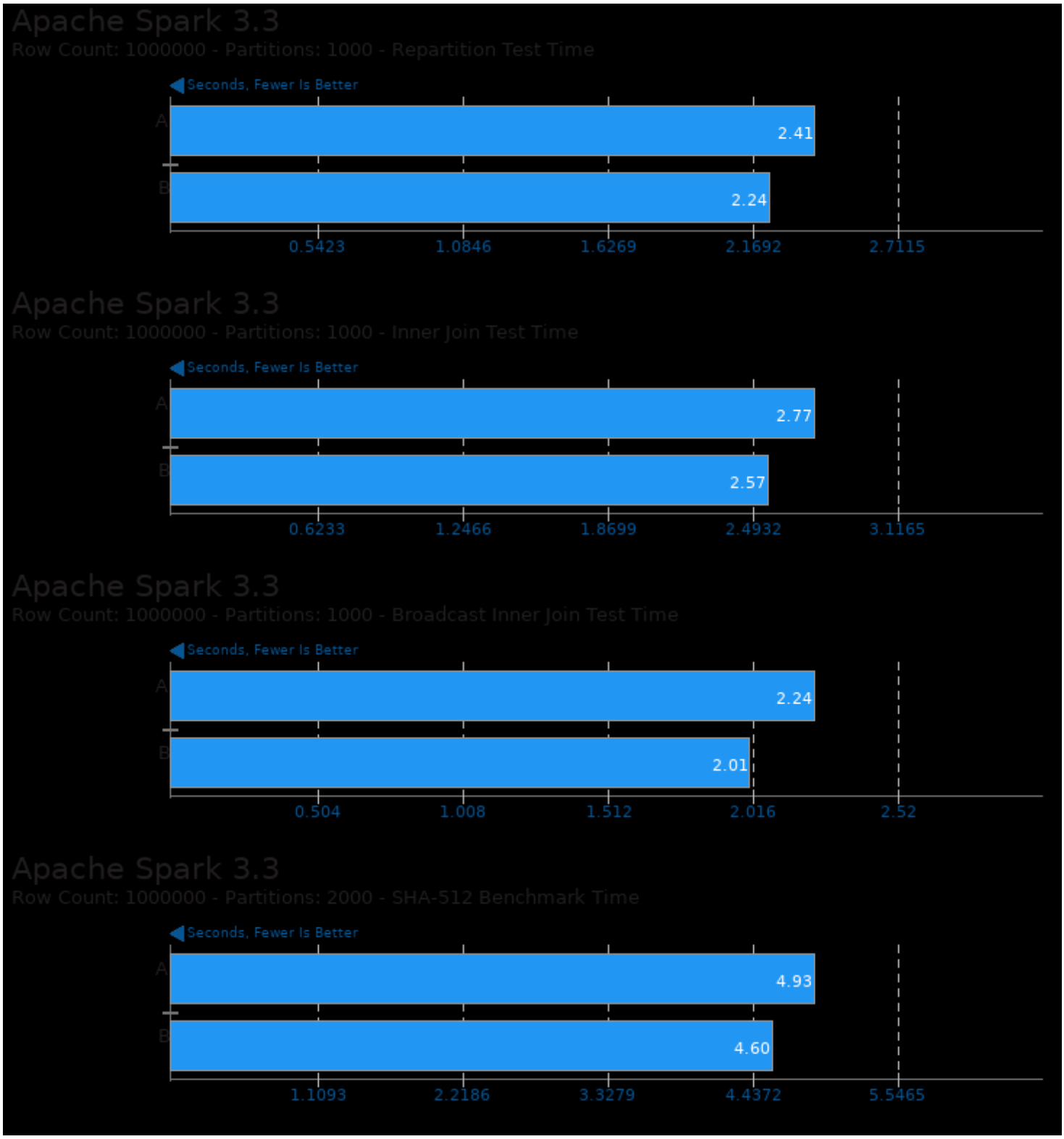

Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
B
11.38
11.46
3
6
9
12
15
Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
B
4.87
4.62
1.0958
2.1916
3.2874
4.3832
5.479
Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
B
2.34
1.82
0.5265
1.053
1.5795
2.106
2.6325
Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
B
2.12
2.08
0.477
0.954
1.431
1.908
2.385

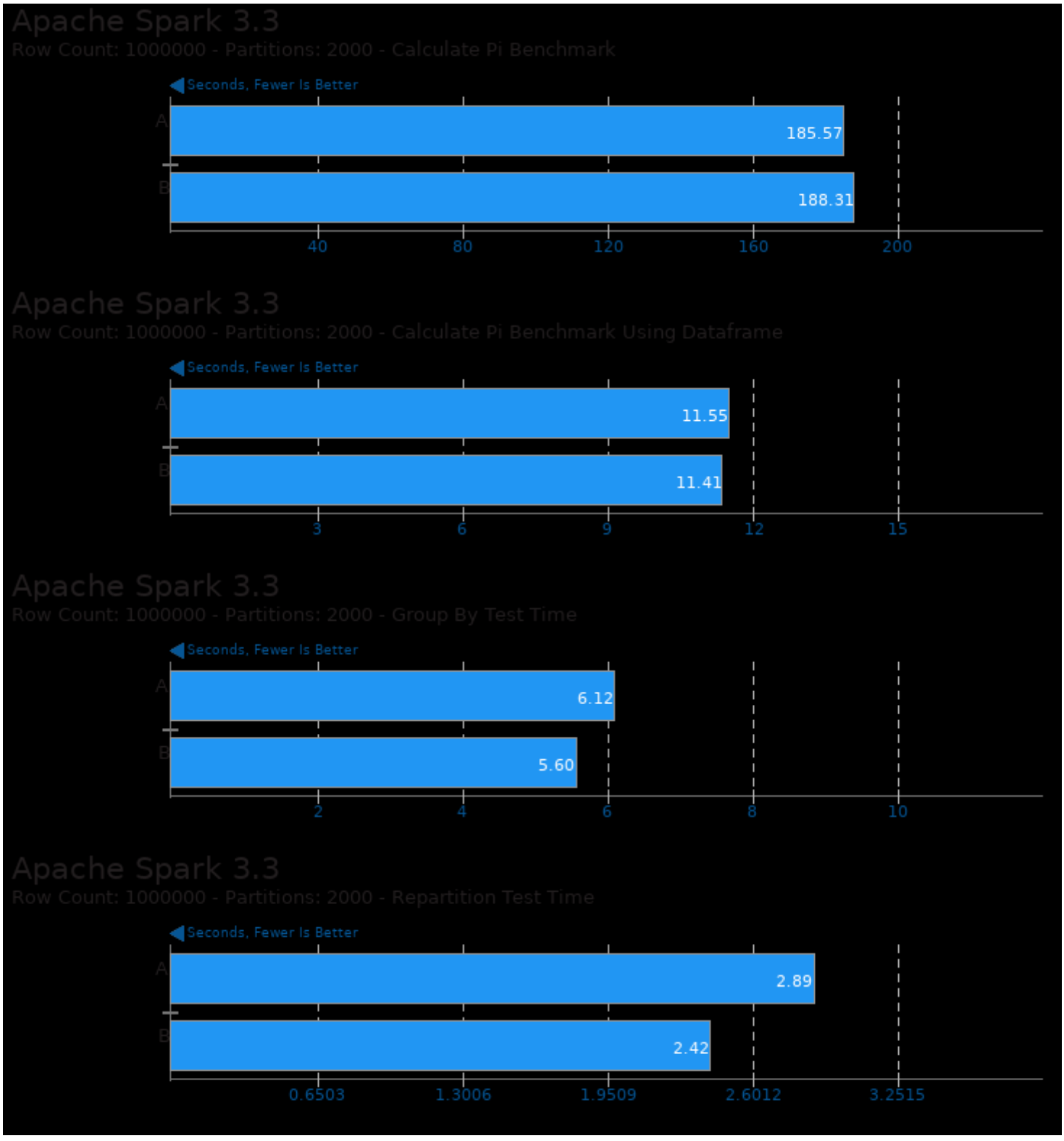

Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
2.20
B
1.82
0.495
0.99
1.485
1.98
2.475
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
4.297287719
B
4.110000000
0.9669
1.9338
2.9007
3.8676
4.8345
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
188.34
B
188.80
40
80
120
160
200
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.49
B
11.48
3
6
9
12
15

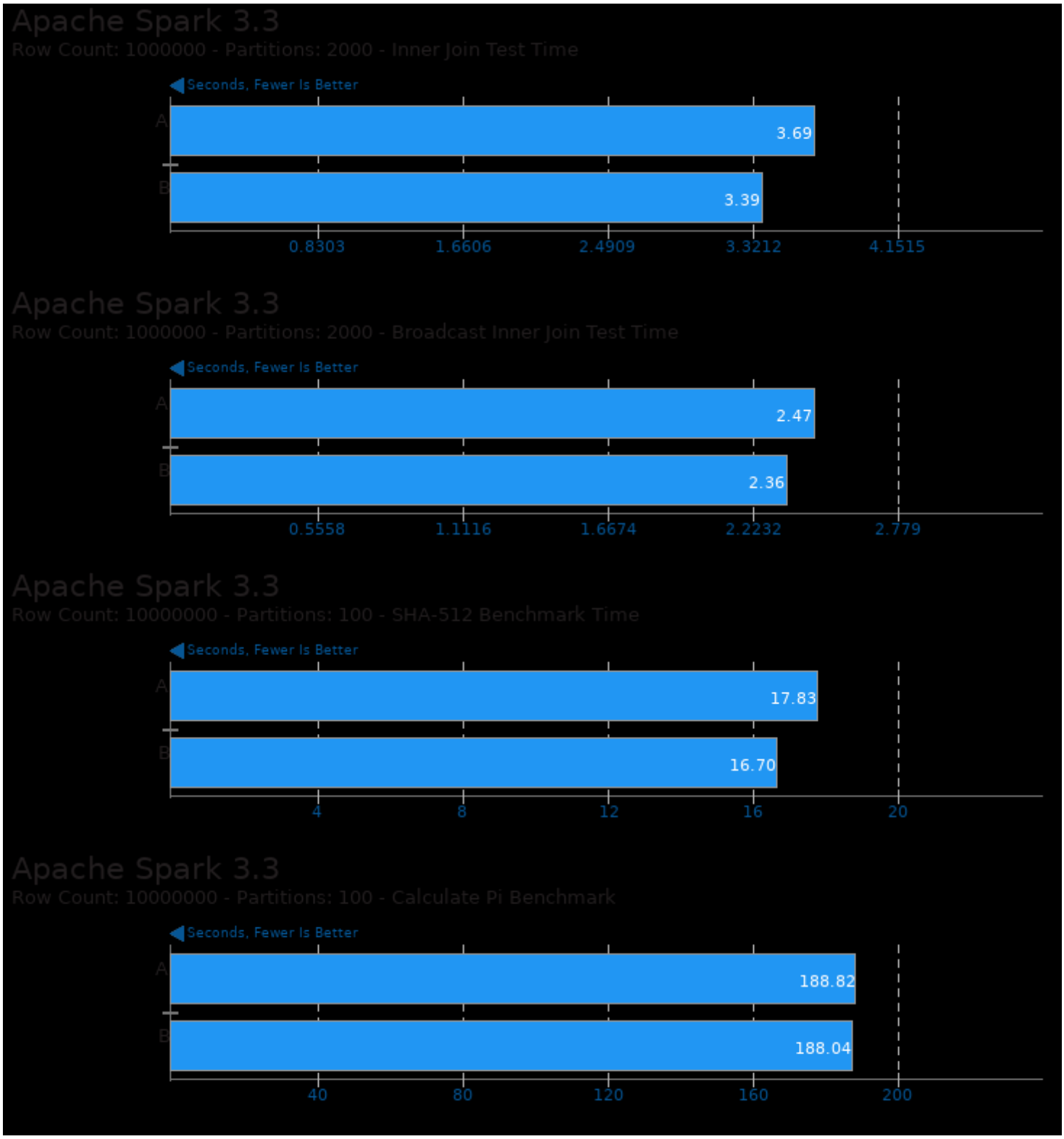

Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
5.08
B
5.21
1.1723
2.3446
3.5169
4.6892
5.8615
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
2.21
B
2.04
0.4973
0.9946
1.4919
1.9892
2.4865
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
2.314010652
B
2.120000000
0.5207
1.0414
1.5621
2.0828
2.6035
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
1.95
B
2.07
0.4658
0.9316
1.3974
1.8632
2.329


Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
4.492061685
B
4.210000000
1.0107
2.0214
3.0321
4.0428
5.0535
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
187.44
B
186.71
40
80
120
160
200
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.57
B
11.45
3
6
9
12
15
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
5.44
B
5.63
1.2668
2.5336
3.8004
5.0672
6.334

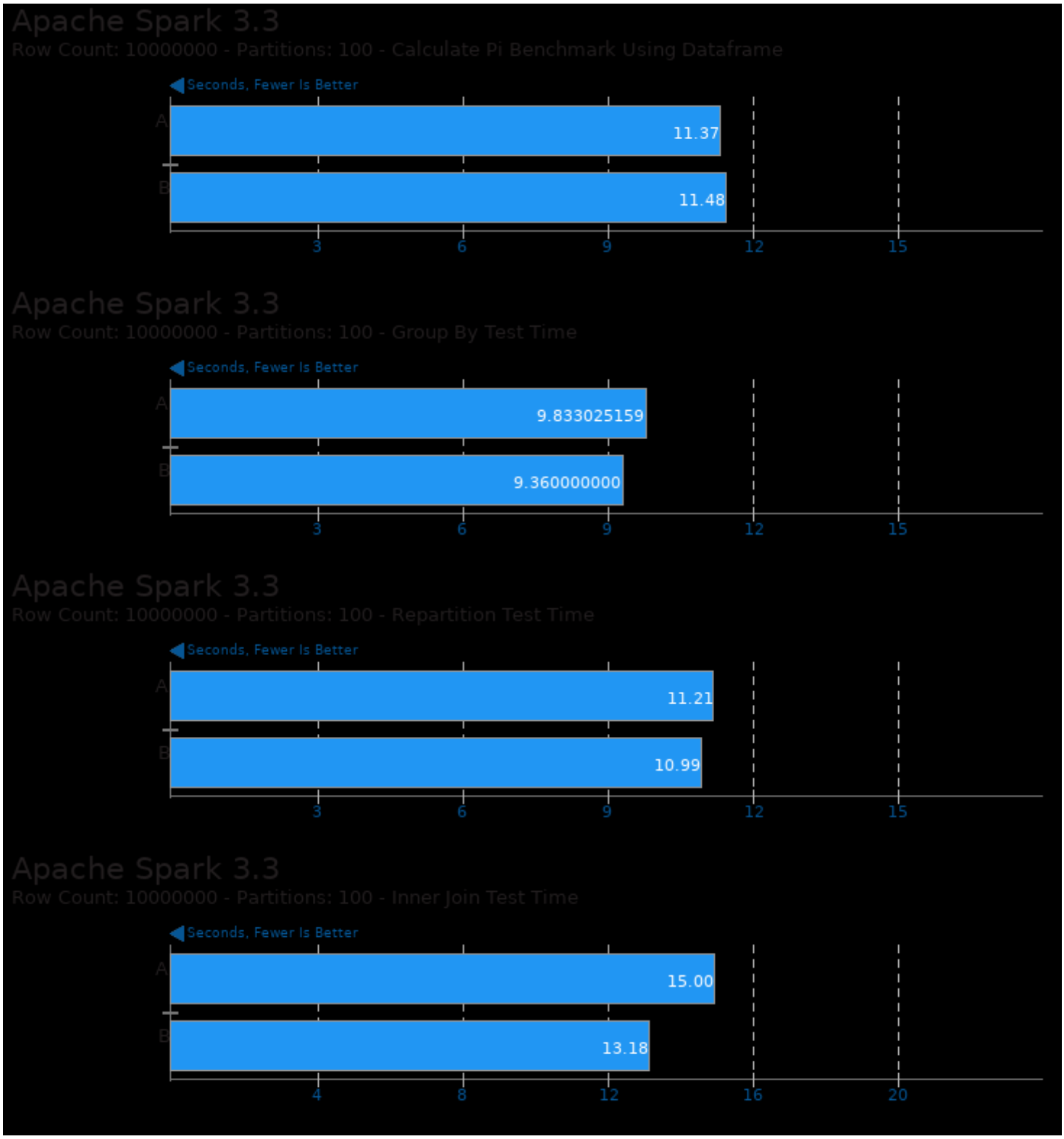
new ep

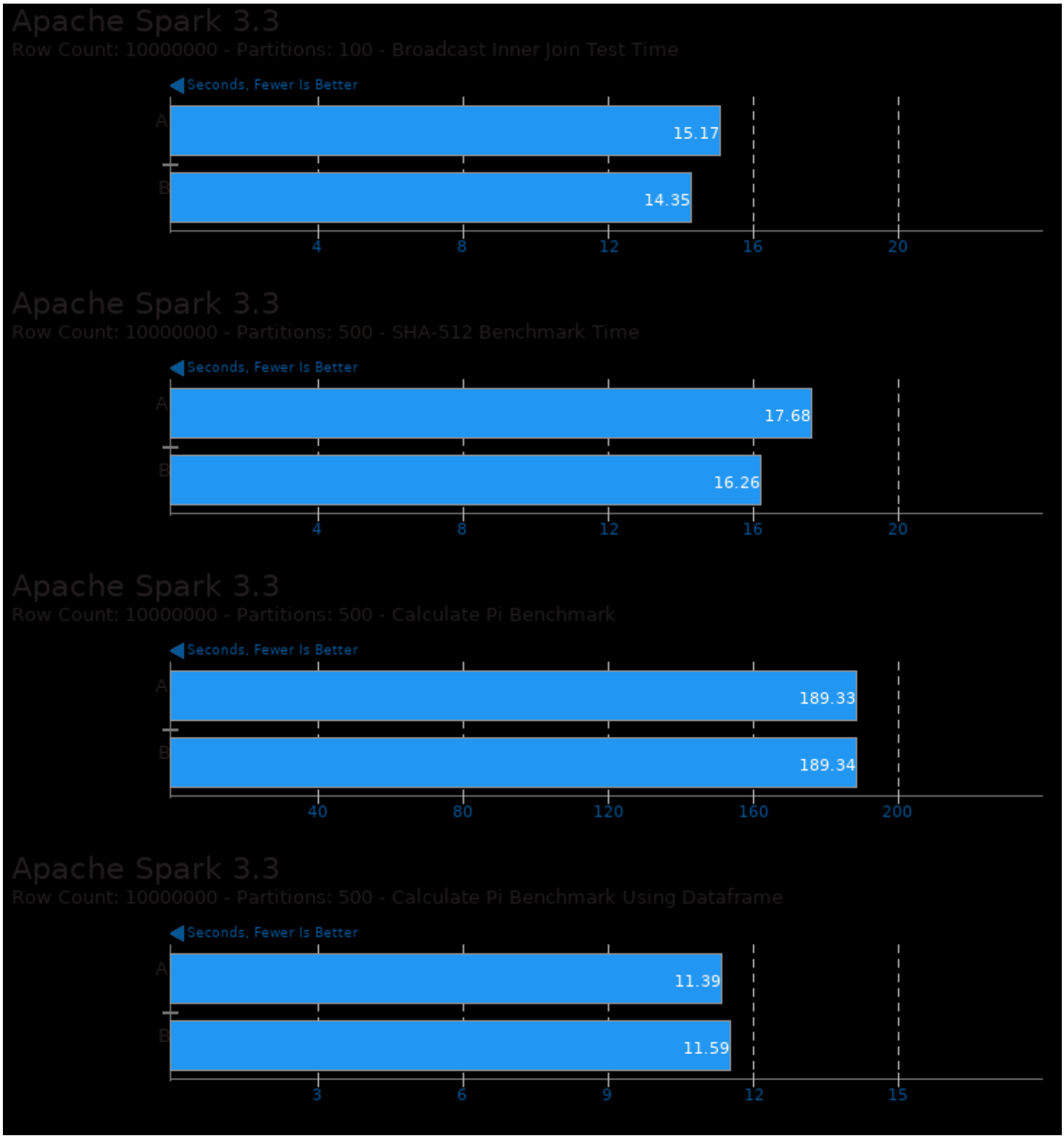

Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
B
2.41
2.24
0.5423
1.0846
1.6269
2.1692
2.7115
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
B
2.77
2.57
0.6233
1.2466
1.8699
2.4932
3.1165
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
B
2.24
2.01
0.504
1.008
1.512
2.016
2.52
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
B
4.93
4.60
1.1093
2.2186
3.3279
4.4372
5.5465

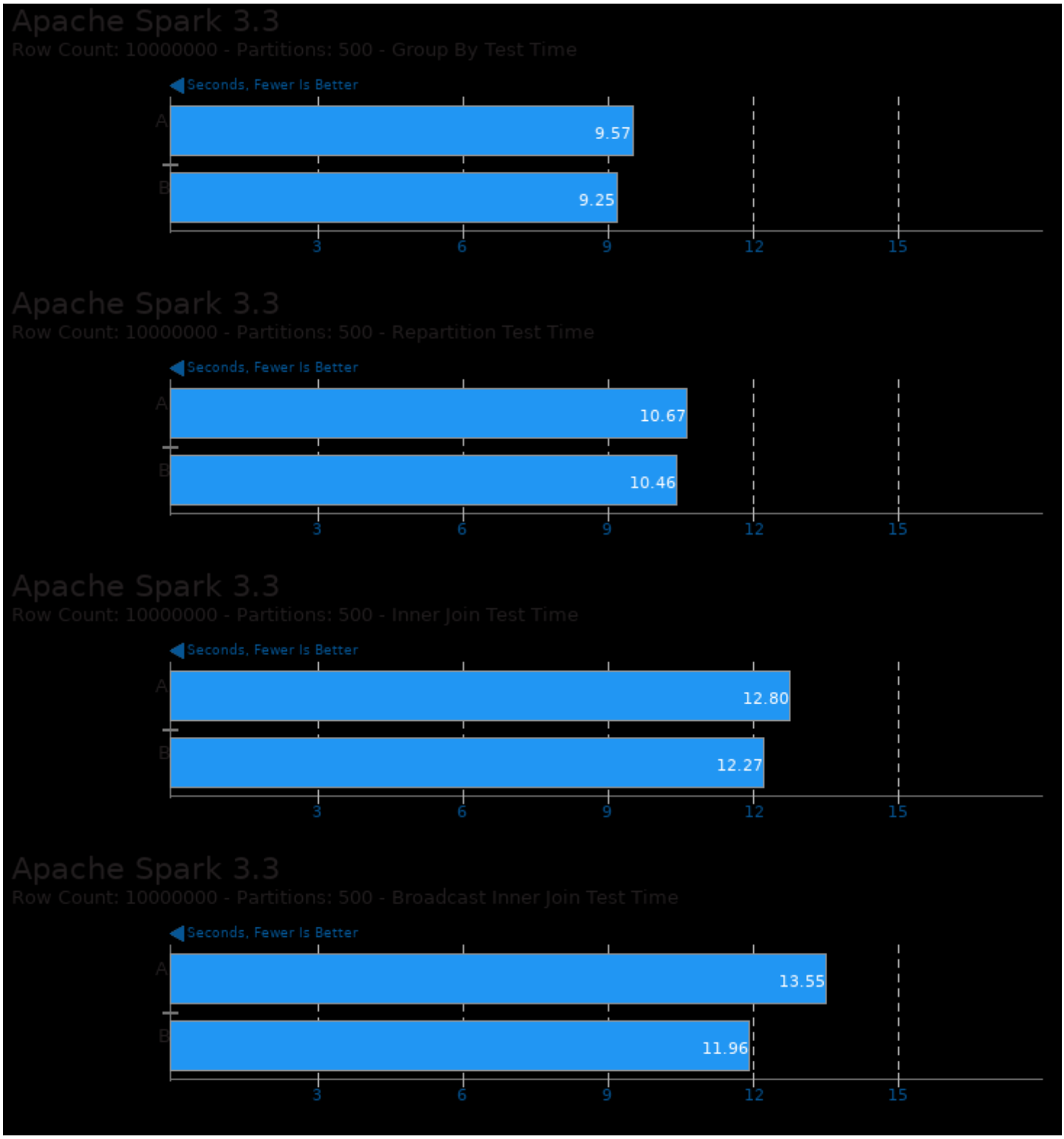

Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
185.57
B
188.31
40
80
120
160
200
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.55
B
11.41
3
6
9
12
15
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
6.12
B
5.60
2
4
6
8
10
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
2.89
B
2.42
0.6503
1.3006
1.9509
2.6012
3.2515

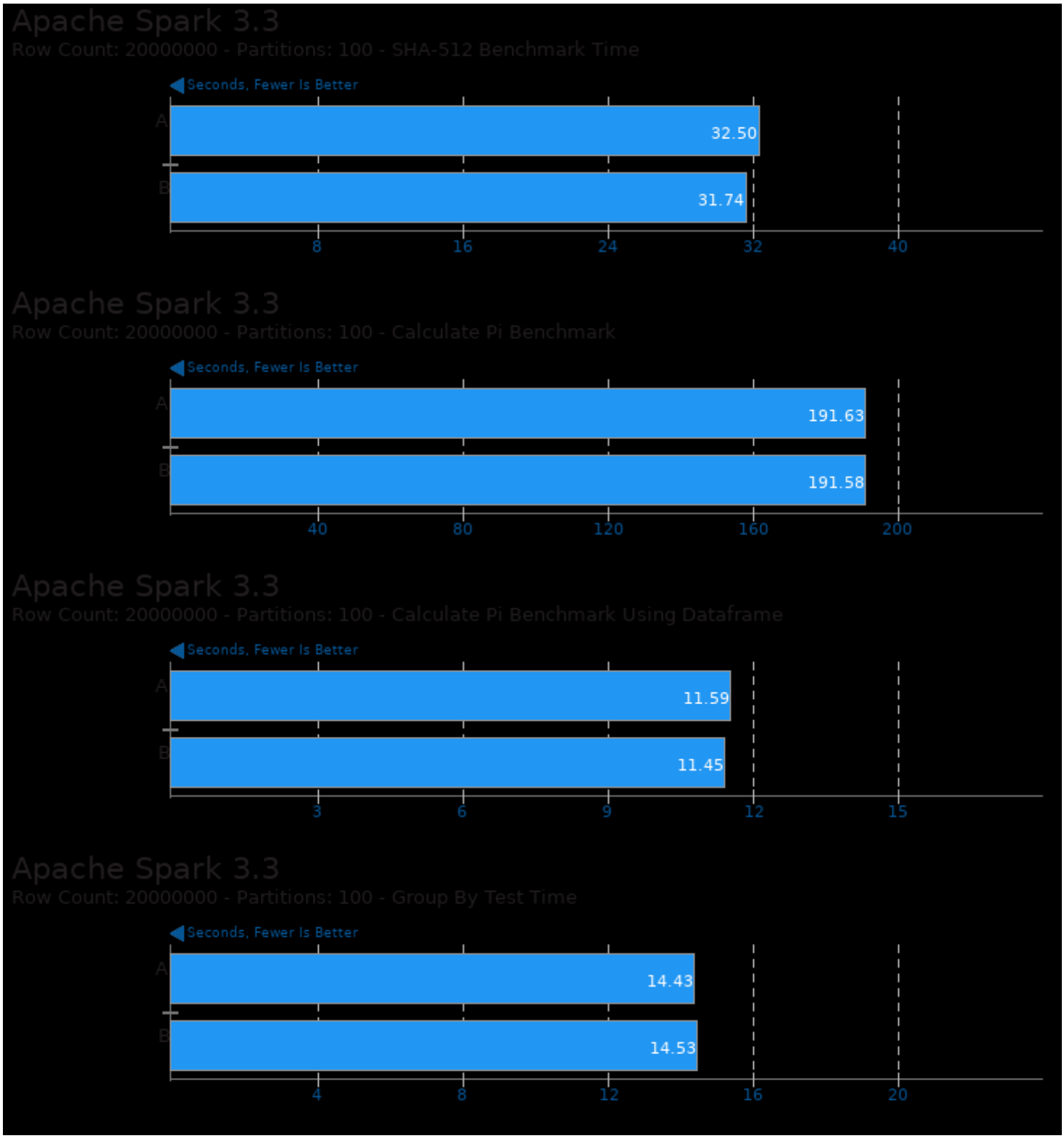

Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Inner Join Test Time
Seconds, Fewer Is Better
A
3.69
B
3.39
0.8303
1.6606
2.4909
3.3212
4.1515
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
2.47
B
2.36
0.5558
1.1116
1.6674
2.2232
2.779
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
17.83
B
16.70
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
188.82
B
188.04
40
80
120
160
200

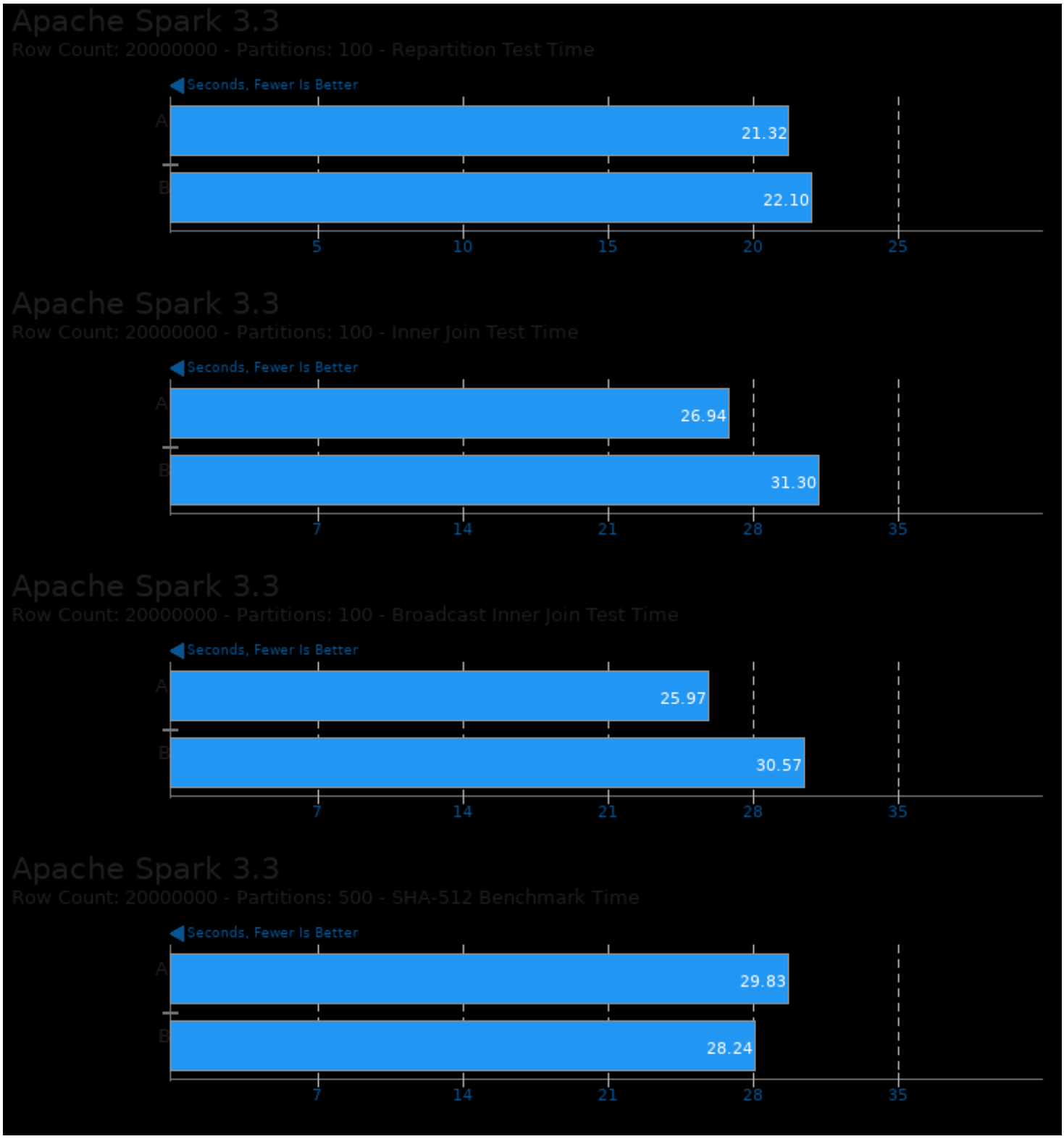
new ep

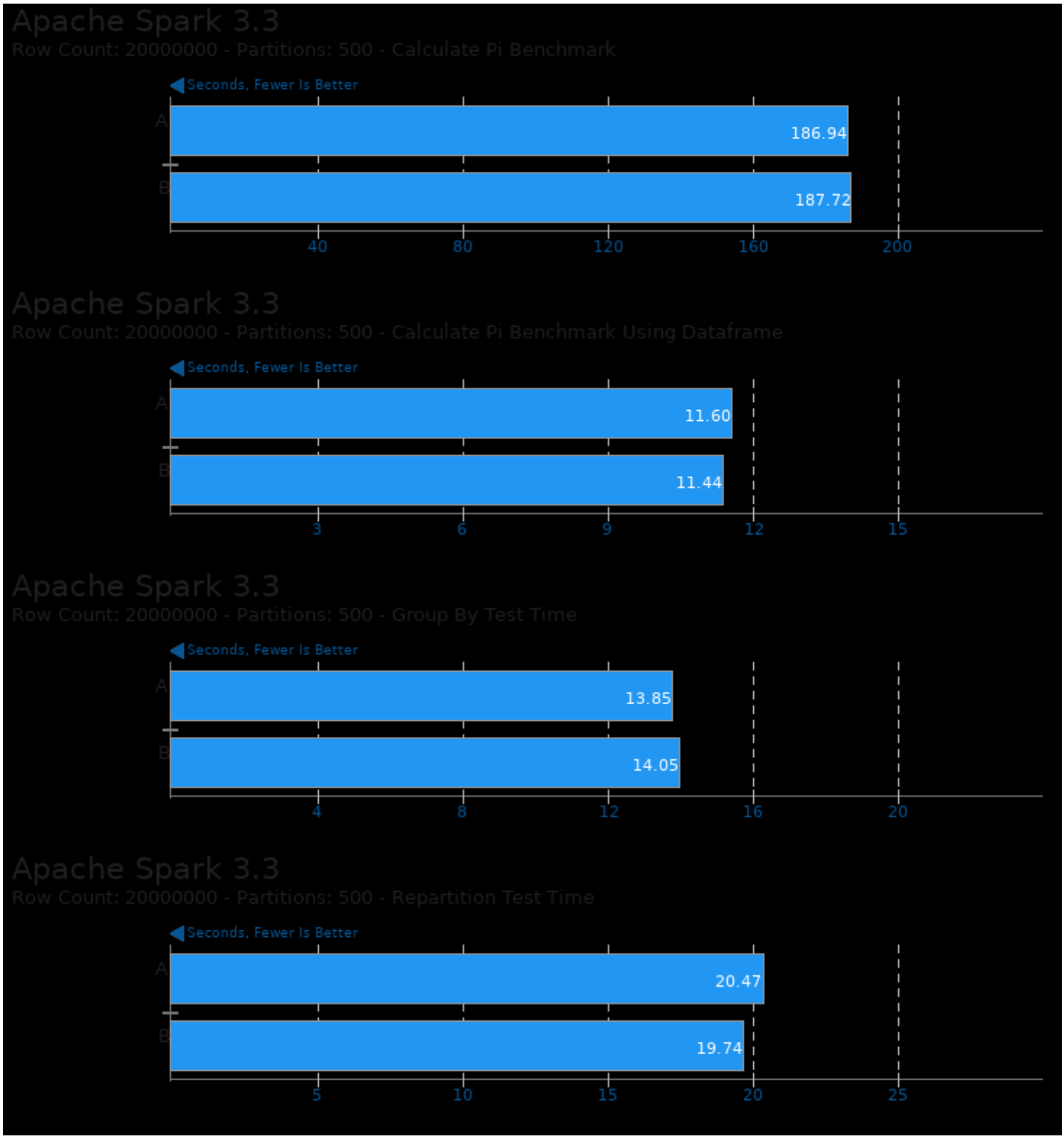

Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.37
B
11.48
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
9.833025159
B
9.360000000
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
11.21
B
10.99
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
15.00
B
13.18
4
8
12
16
20

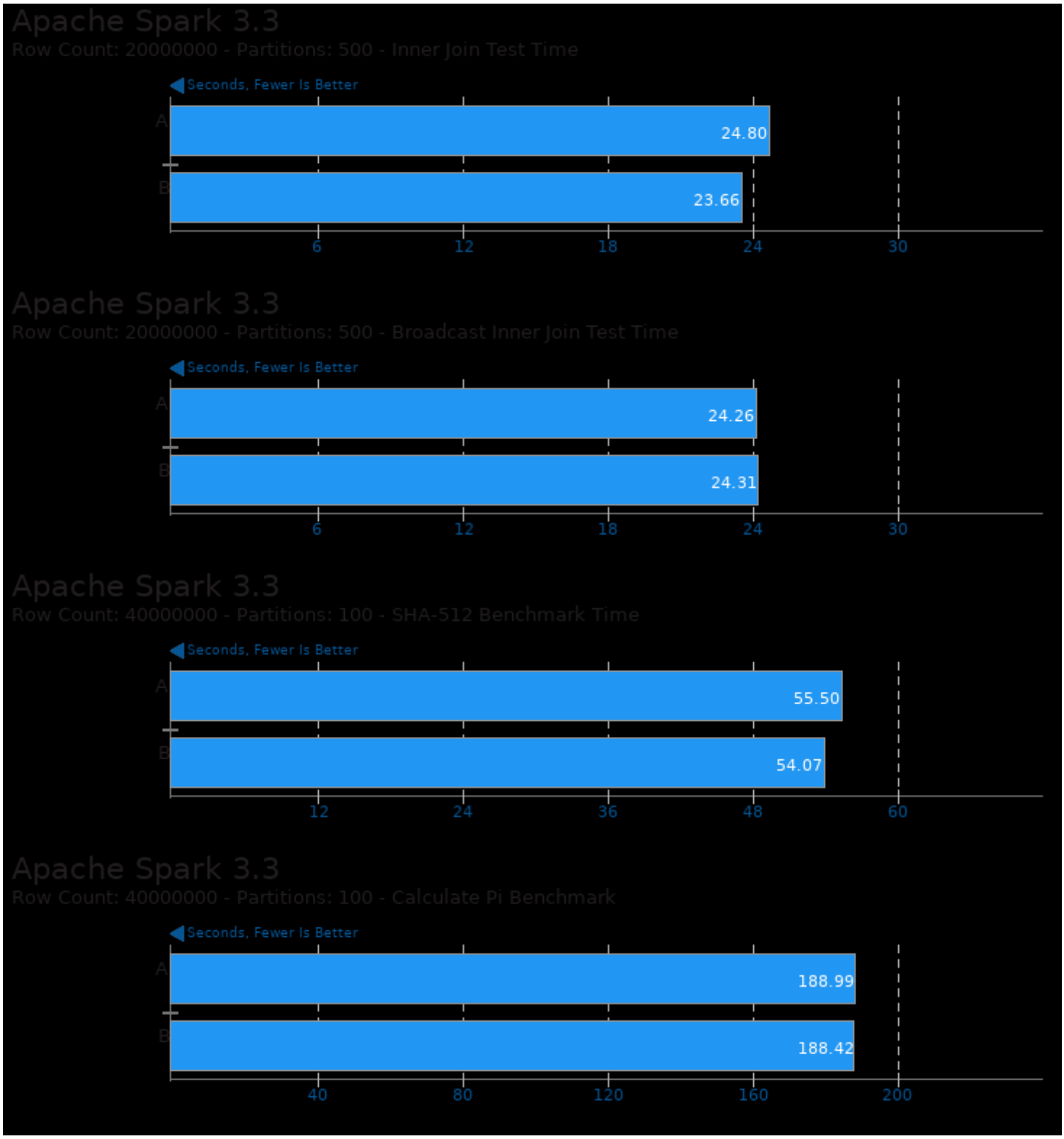

Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
15.17
B
14.35
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
17.68
B
16.26
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
189.33
B
189.34
40
80
120
160
200
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.39
B
11.59
3
6
9
12
15

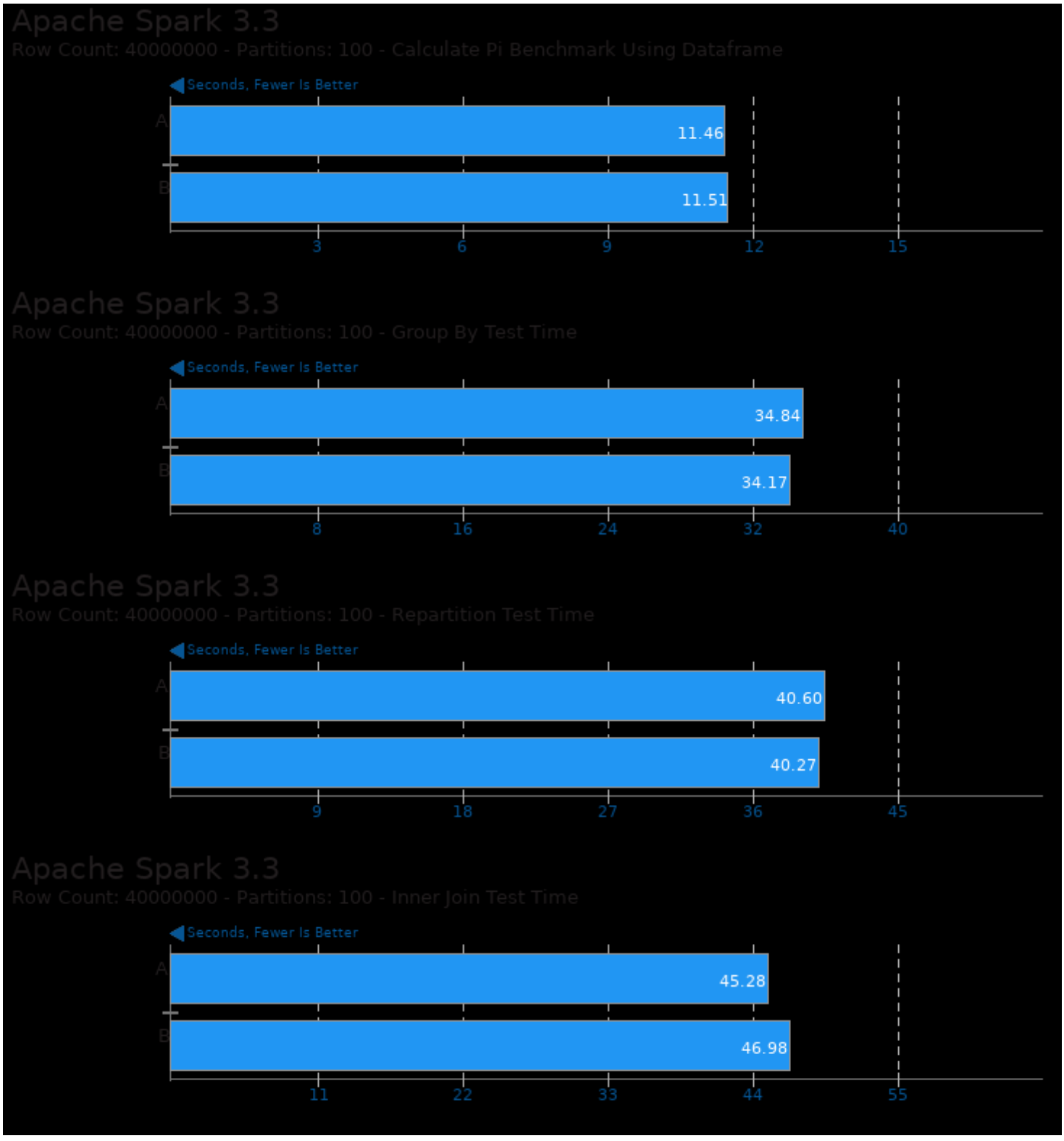

Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
9.57
B
9.25
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
10.67
B
10.46
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
12.80
B
12.27
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
13.55
B
11.96
3
6
9
12
15

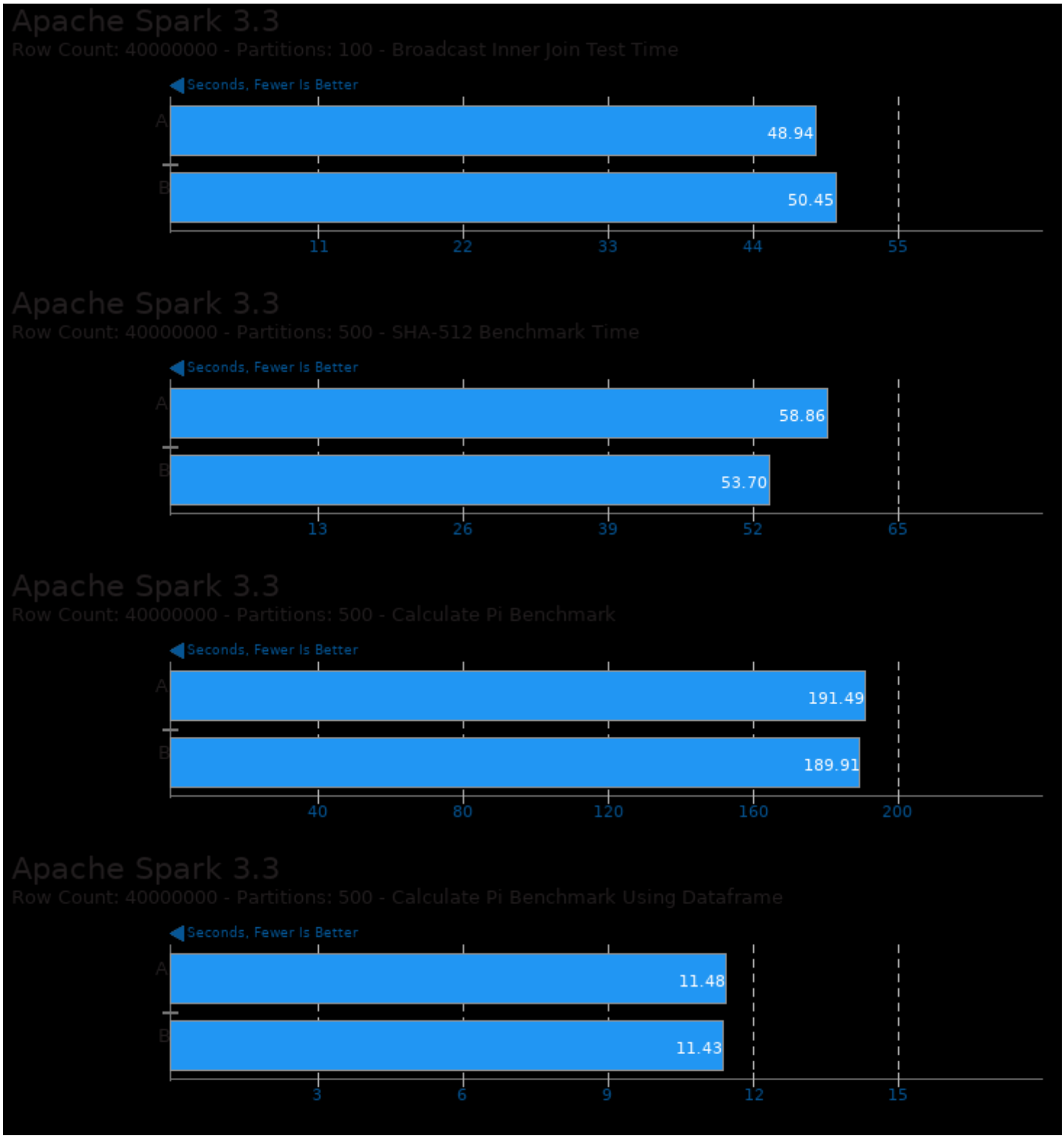

Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
32.50
B
31.74
8
16
24
32
40
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
191.63
B
191.58
40
80
120
160
200
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.59
B
11.45
3
6
9
12
15
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
14.43
B
14.53
4
8
12
16
20

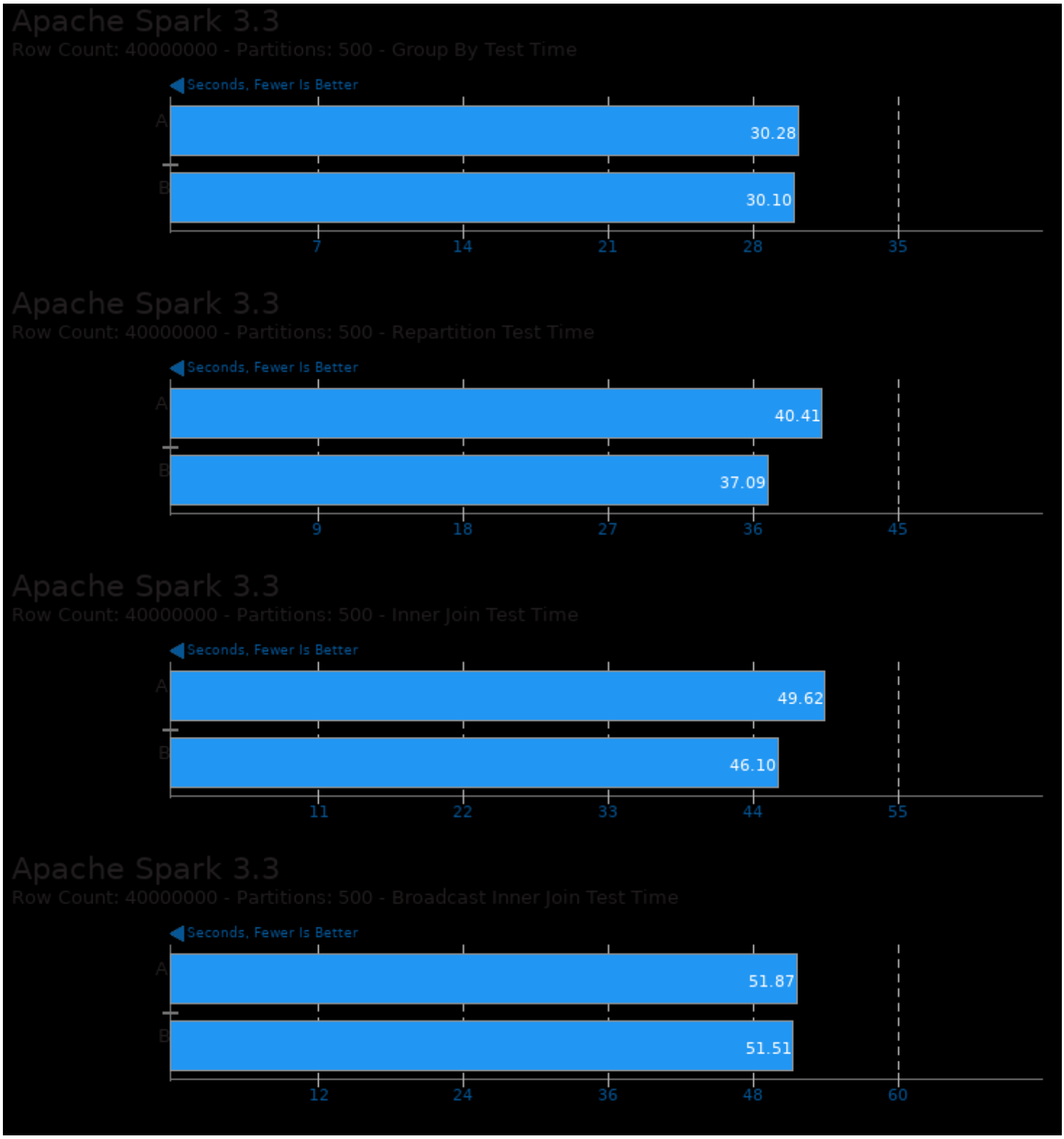

Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
21.32
B
22.10
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
26.94
B
31.30
7
14
21
28
35
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
25.97
B
30.57
7
14
21
28
35
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
29.83
B
28.24
7
14
21
28
35

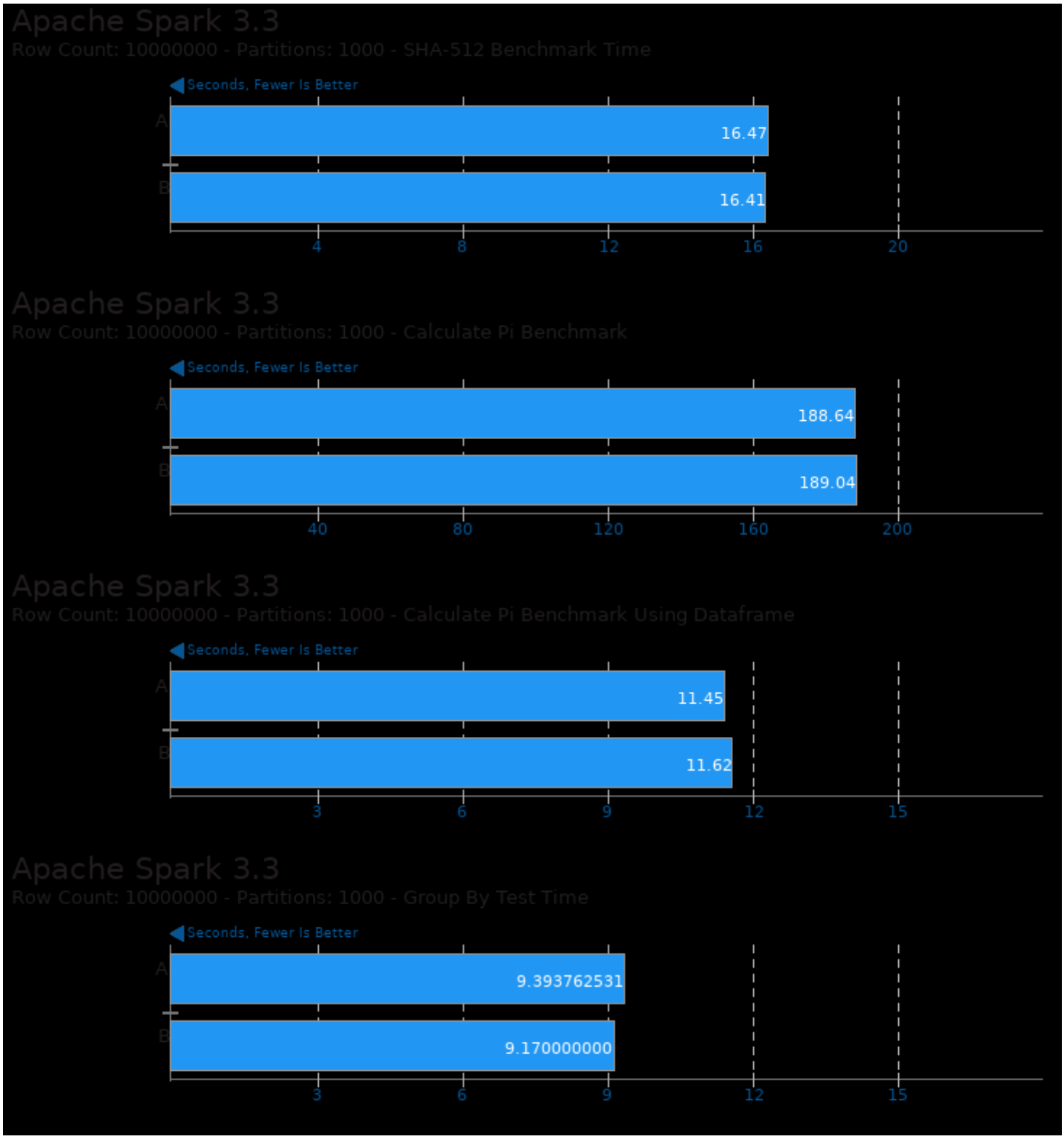

Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
186.94
B
187.72
40
80
120
160
200
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.60
B
11.44
3
6
9
12
15
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
13.85
B
14.05
4
8
12
16
20
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
20.47
B
19.74
5
10
15
20
25

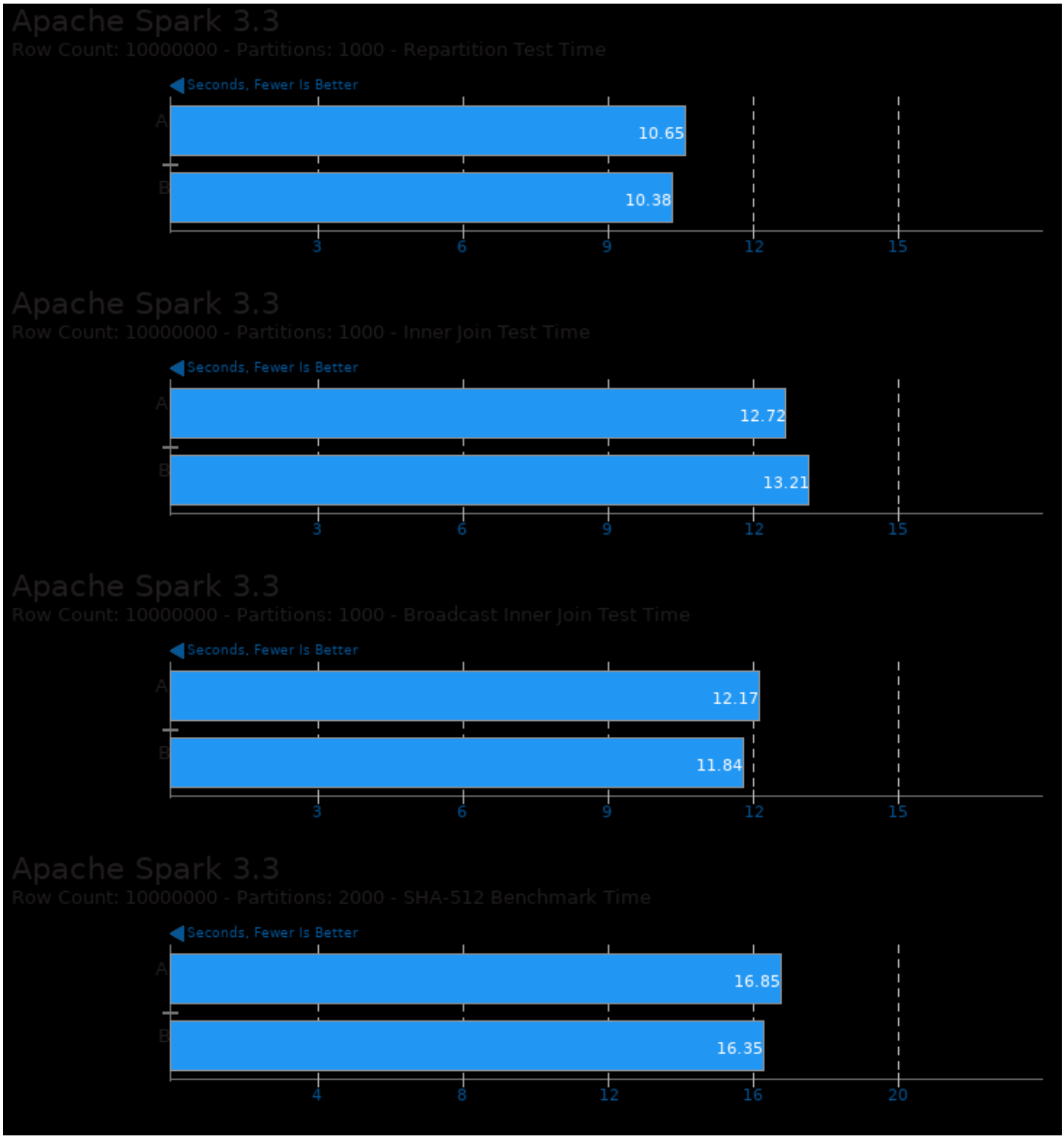

Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
24.80
B
23.66
6
12
18
24
30
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
24.26
B
24.31
6
12
18
24
30
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
55.50
B
54.07
12
24
36
48
60
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
188.99
B
188.42
40
80
120
160
200

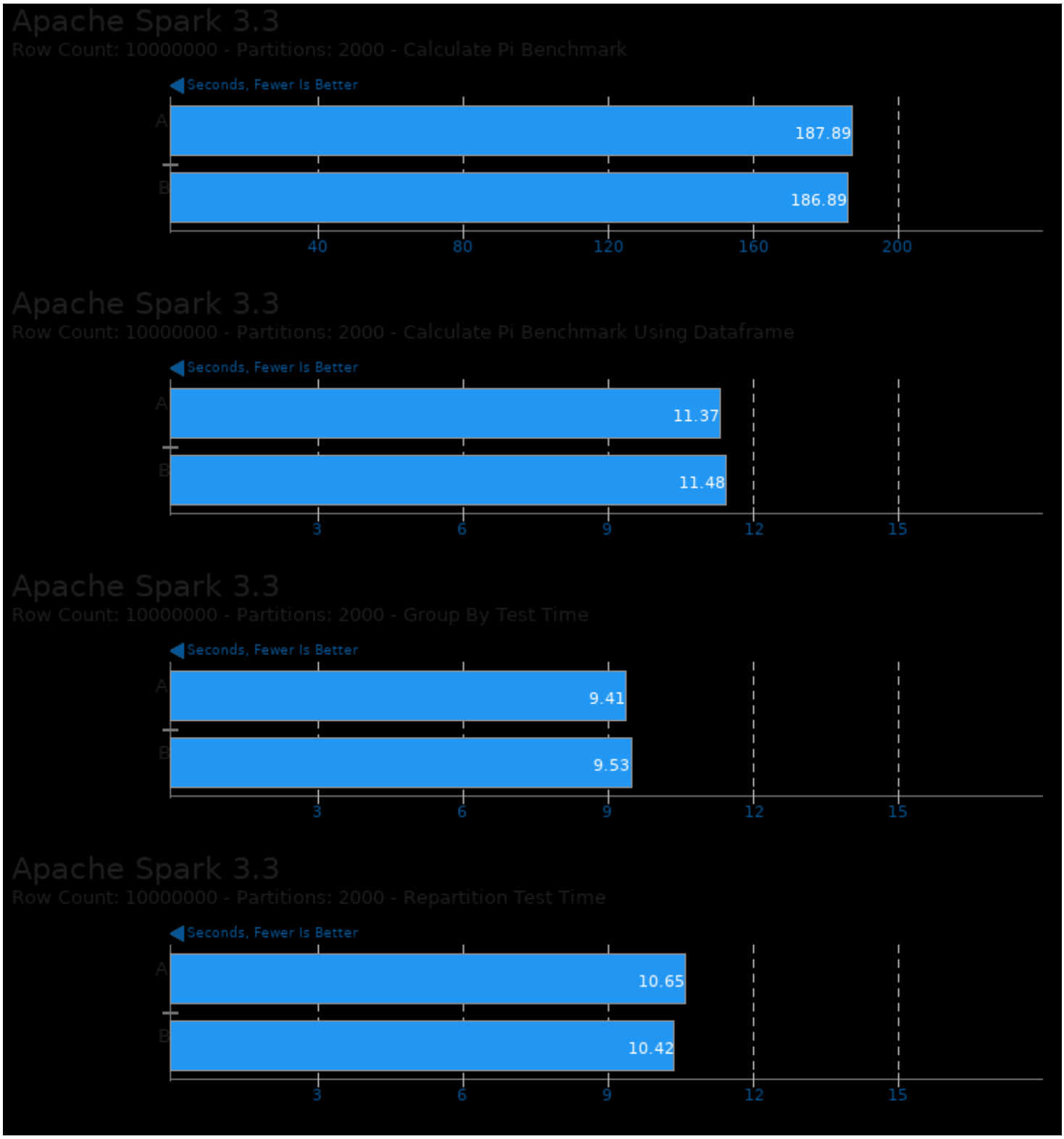

Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.46
B
11.51
3
6
9
12
15
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
34.84
B
34.17
8
16
24
32
40
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
40.60
B
40.27
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
45.28
B
46.98
11
22
33
44
55

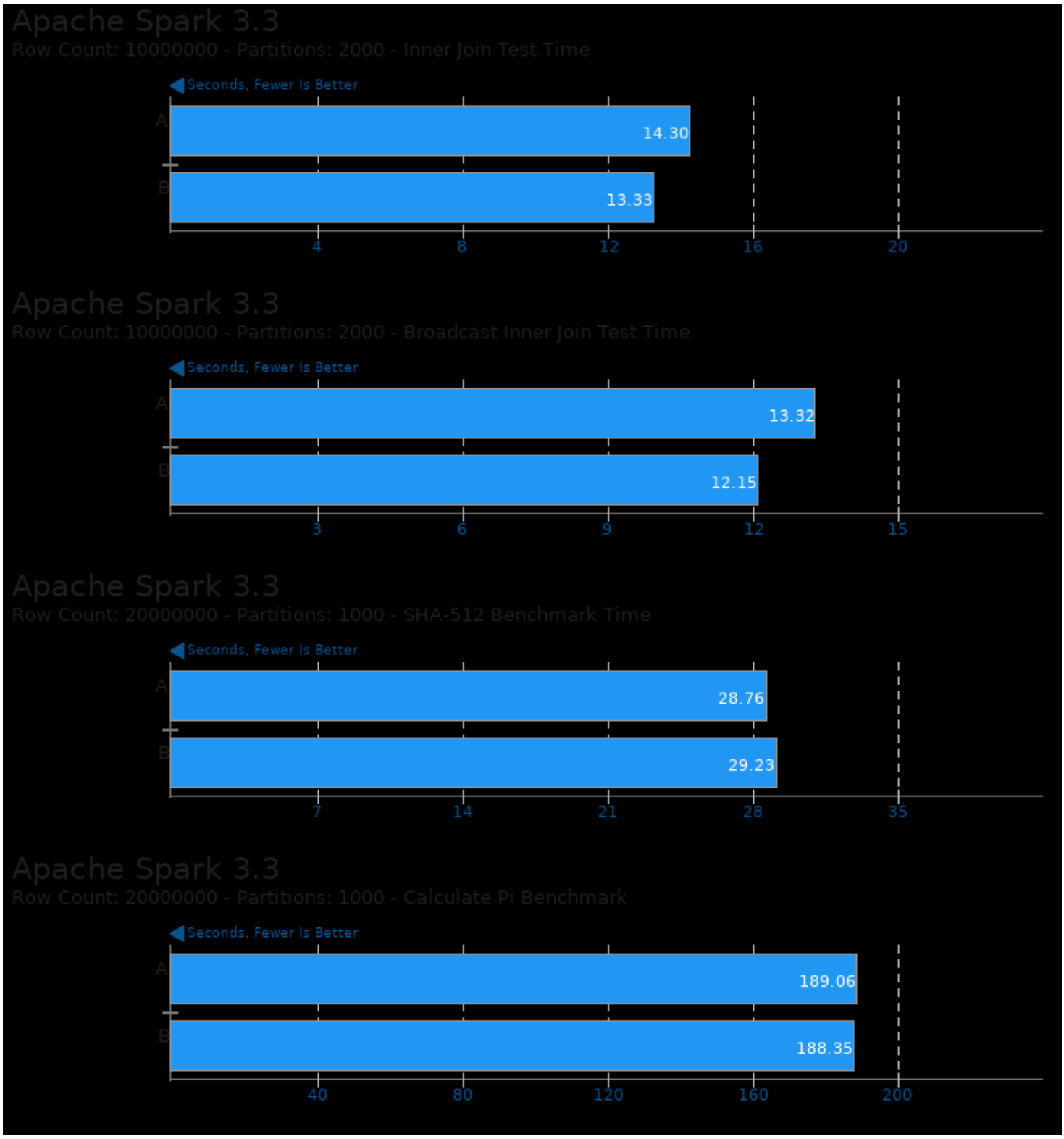

Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
48.94
B
50.45
11
22
33
44
55
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
58.86
B
53.70
13
26
39
52
65
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
191.49
B
189.91
40
80
120
160
200
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.48
B
11.43
3
6
9
12
15

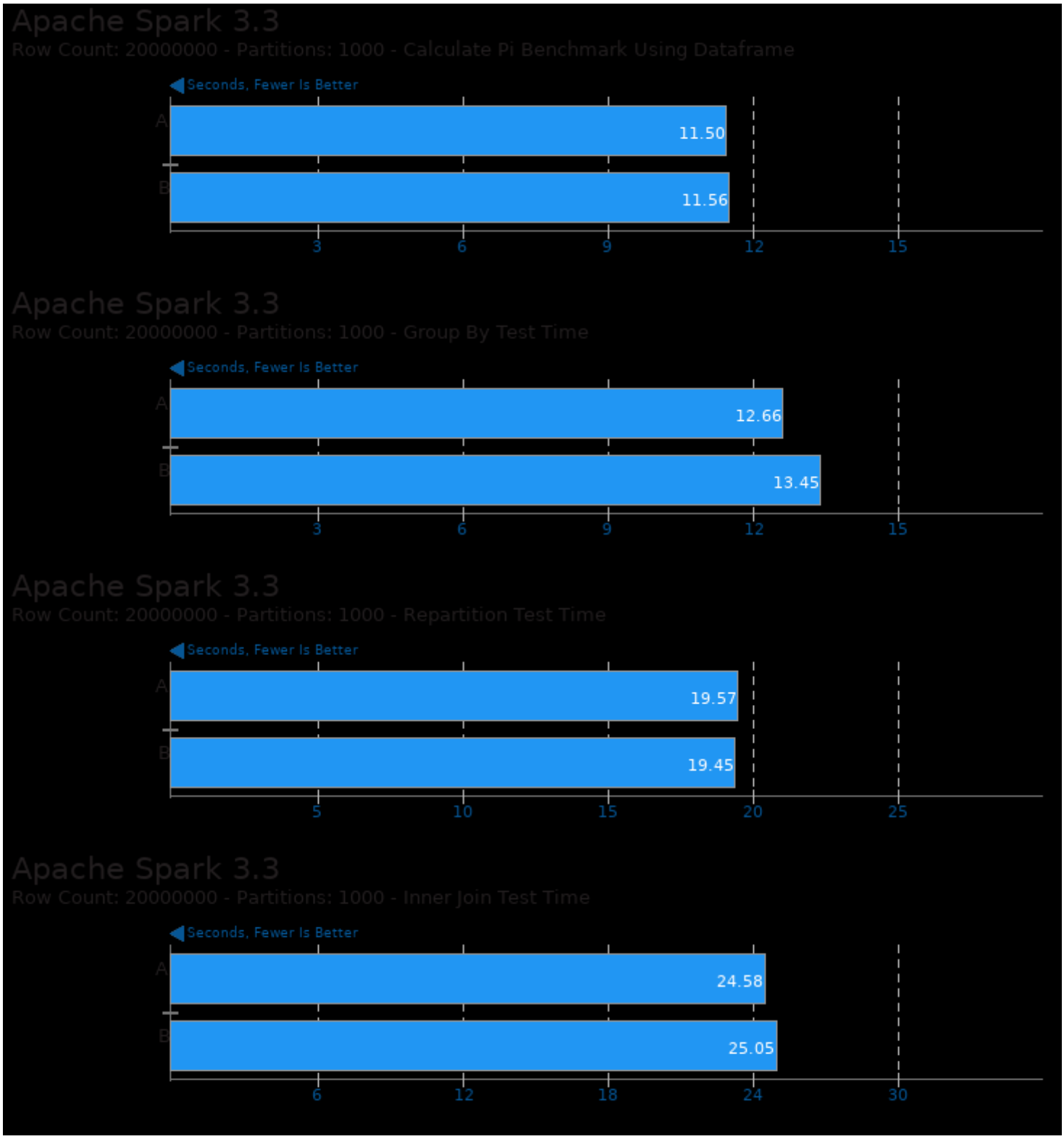

Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
30.28
B
30.10
7
14
21
28
35
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
40.41
B
37.09
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
49.62
B
46.10
11
22
33
44
55
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
51.87
B
51.51
12
24
36
48
60

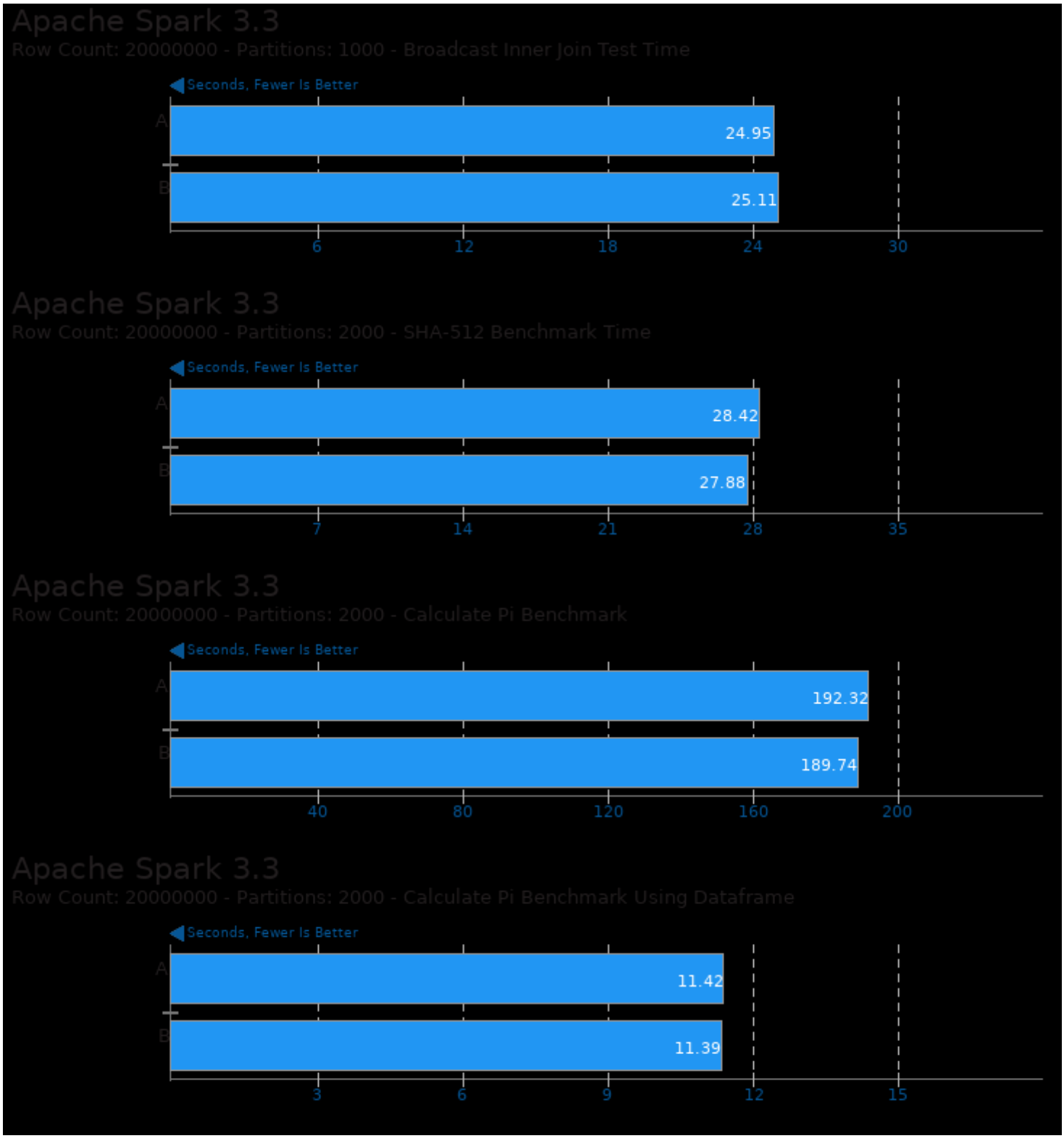

Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
16.47
B
16.41
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
188.64
B
189.04
40
80
120
160
200
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.45
B
11.62
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
9.393762531
B
9.170000000
3
6
9
12
15

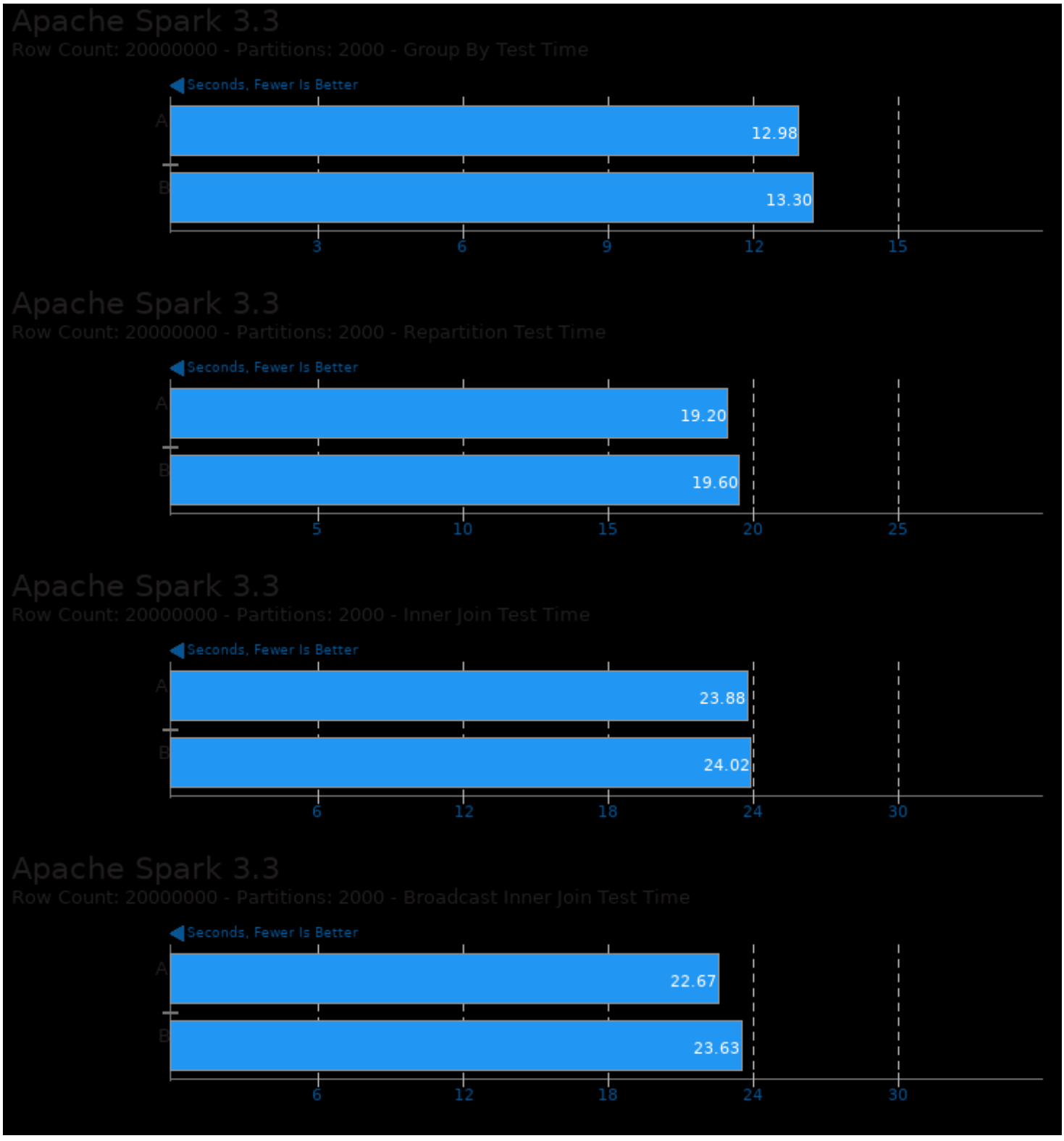

Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
10.65
B
10.38
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
12.72
B
13.21
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
12.17
B
11.84
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
16.85
B
16.35
4
8
12
16
20

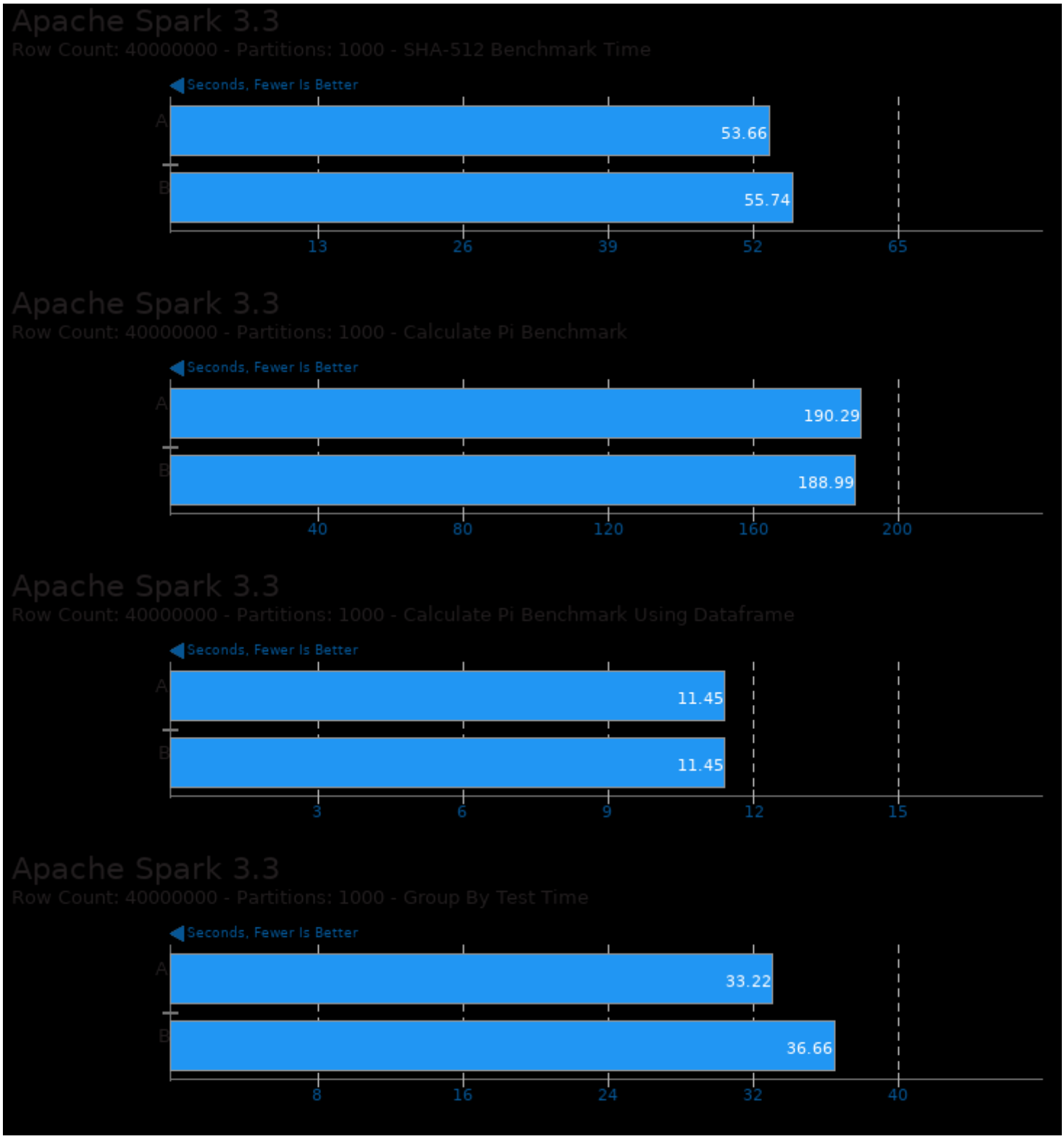

Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
187.89
B
186.89
40
80
120
160
200
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.37
B
11.48
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
9.41
B
9.53
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
10.65
B
10.42
3
6
9
12
15

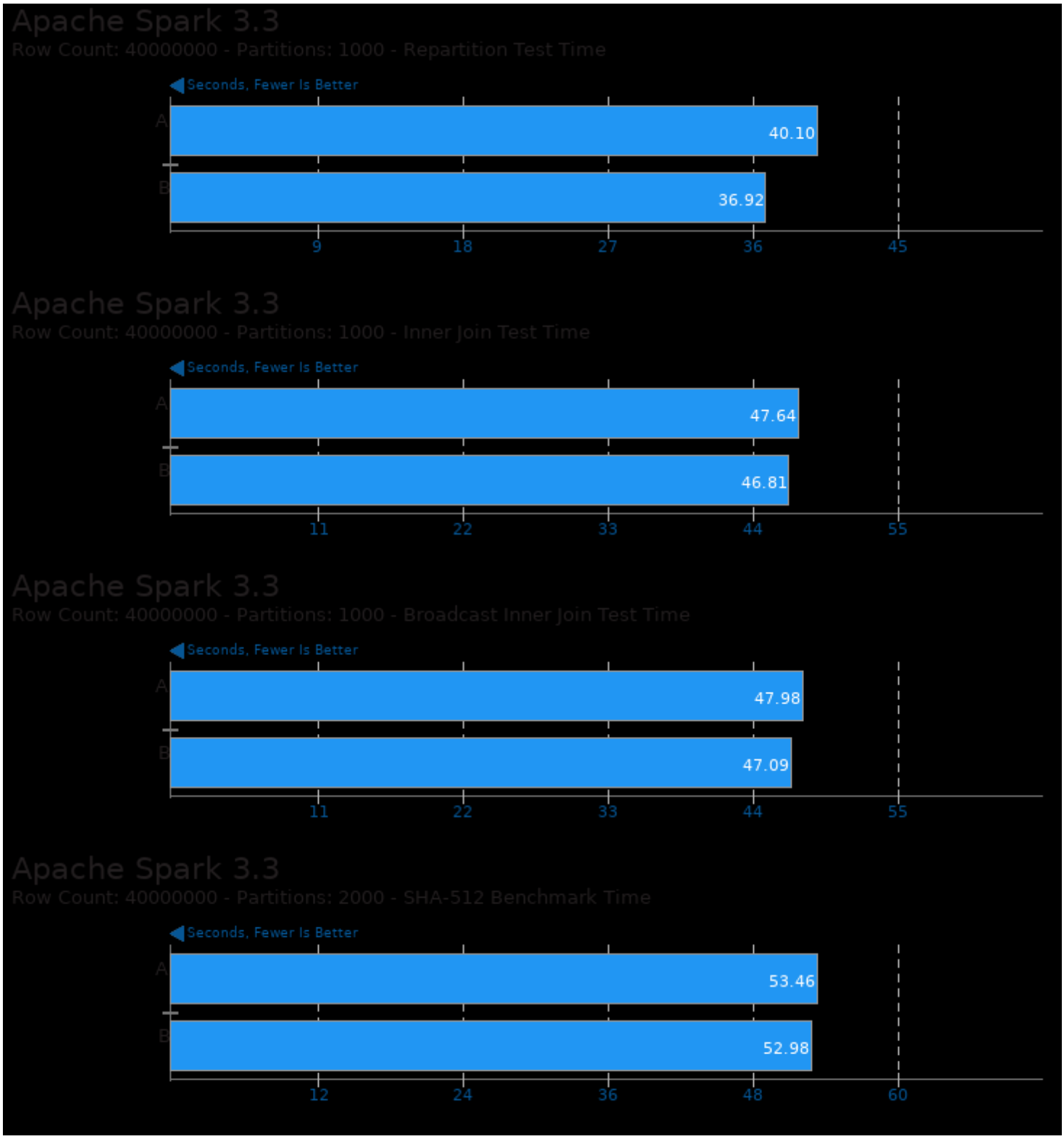

Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Inner Join Test Time
Seconds, Fewer Is Better
A
14.30
B
13.33
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
13.32
B
12.15
3
6
9
12
15
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
28.76
B
29.23
7
14
21
28
35
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
189.06
B
188.35
40
80
120
160
200

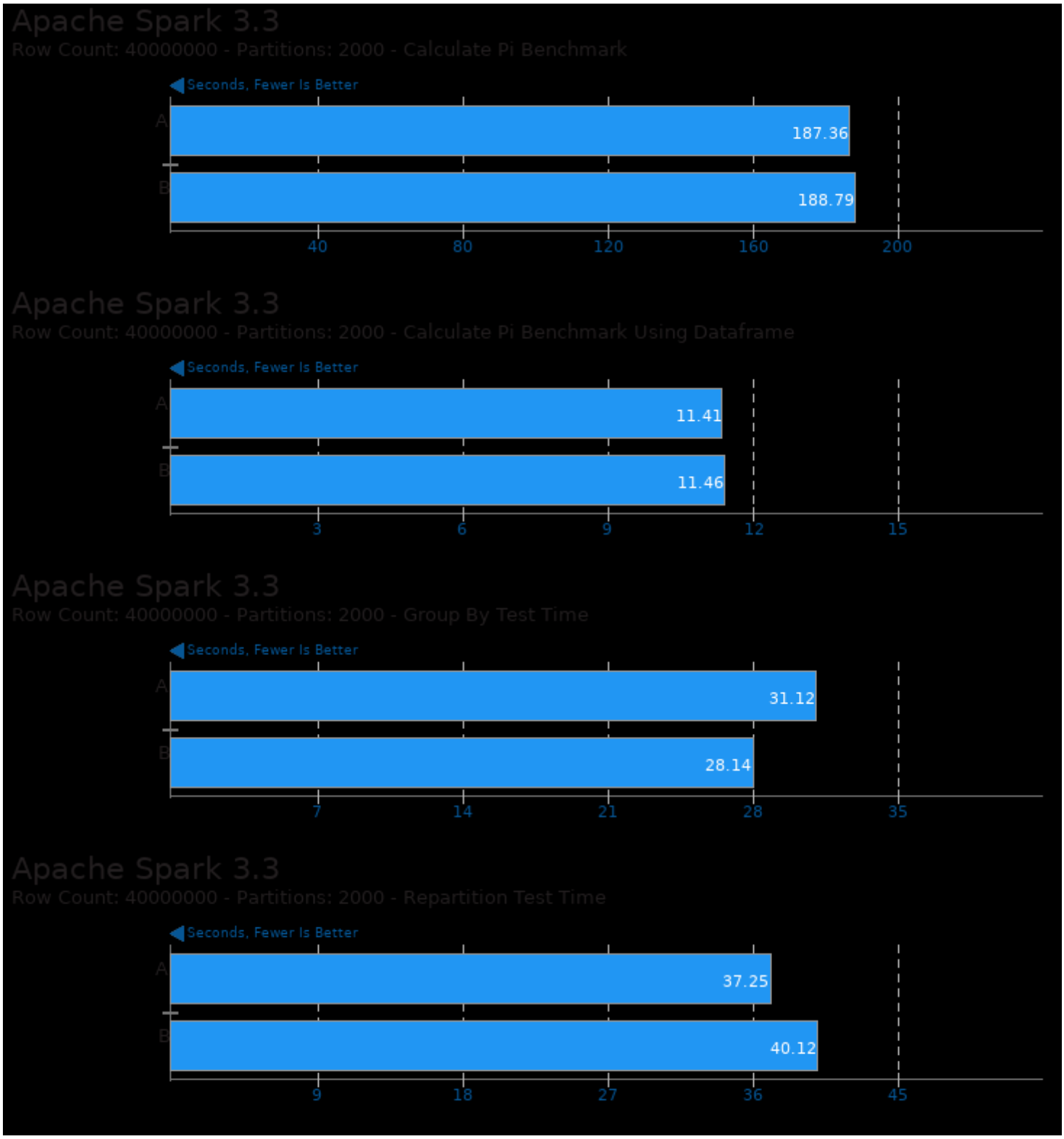

Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.50
B
11.56
3
6
9
12
15
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
12.66
B
13.45
3
6
9
12
15
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
19.57
B
19.45
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
24.58
B
25.05
6
12
18
24
30


Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
24.95
B
25.11
6
12
18
24
30
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
28.42
B
27.88
7
14
21
28
35
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
192.32
B
189.74
40
80
120
160
200
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.42
B
11.39
3
6
9
12
15


Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
12.98
B
13.30
3
6
9
12
15
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
19.20
B
19.60
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Inner Join Test Time
Seconds, Fewer Is Better
A
23.88
B
24.02
6
12
18
24
30
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
22.67
B
23.63
6
12
18
24
30

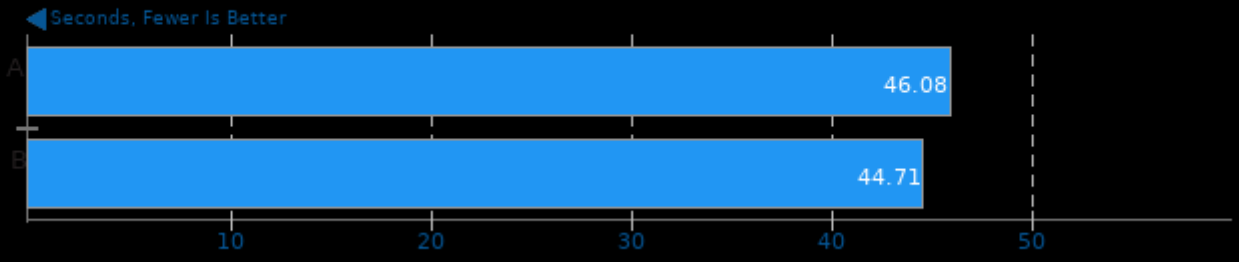

Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
53.66
B
55.74
13
26
39
52
65
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
190.29
B
188.99
40
80
120
160
200
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.45
B
11.45
3
6
9
12
15
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
33.22
B
36.66
8
16
24
32
40


Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
40.10
B
36.92
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
47.64
B
46.81
11
22
33
44
55
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
47.98
B
47.09
11
22
33
44
55
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
53.46
B
52.98
12
24
36
48
60


Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
187.36
B
188.79
40
80
120
160
200
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
11.41
B
11.46
3
6
9
12
15
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
31.12
B
28.14
7
14
21
28
35
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
37.25
B
40.12
9
18
27
36
45

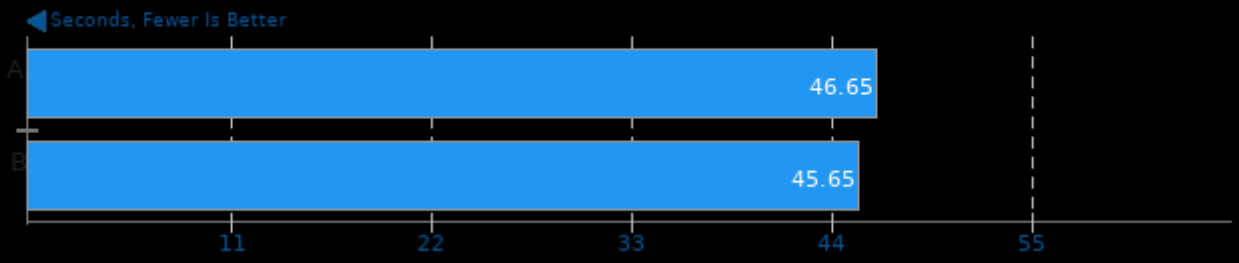
Apache Spark 3.3

Row Count: 40000000 - Partitions: 2000 - Inner Join Test Time



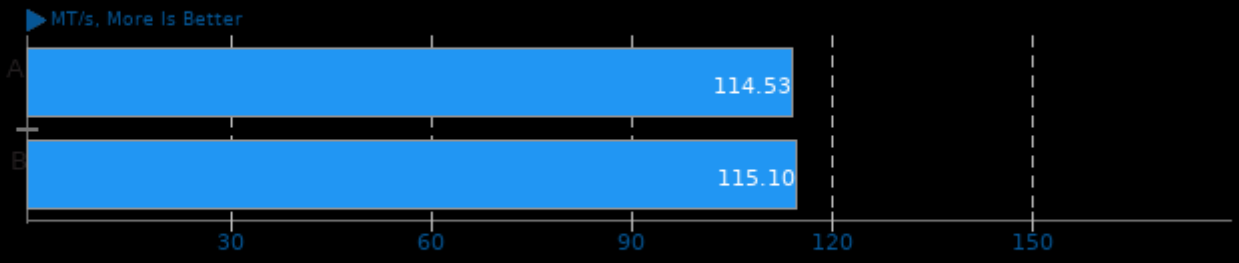
Apache Spark 3.3

Row Count: 40000000 - Partitions: 2000 - Broadcast Inner Join Test Time



ASTC Encoder 4.0

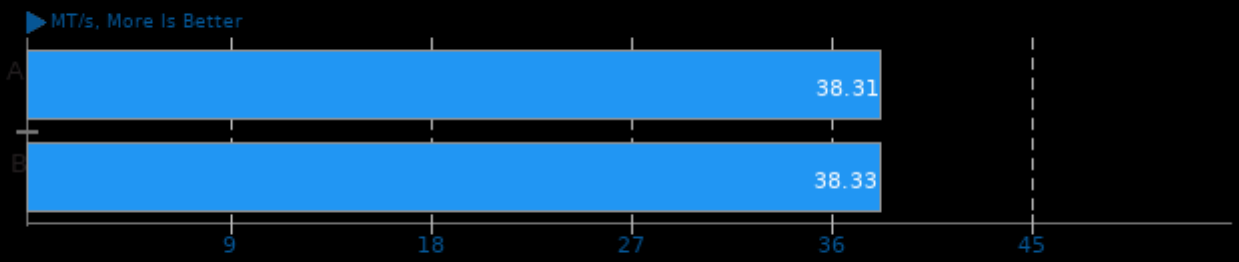
Preset: Fast



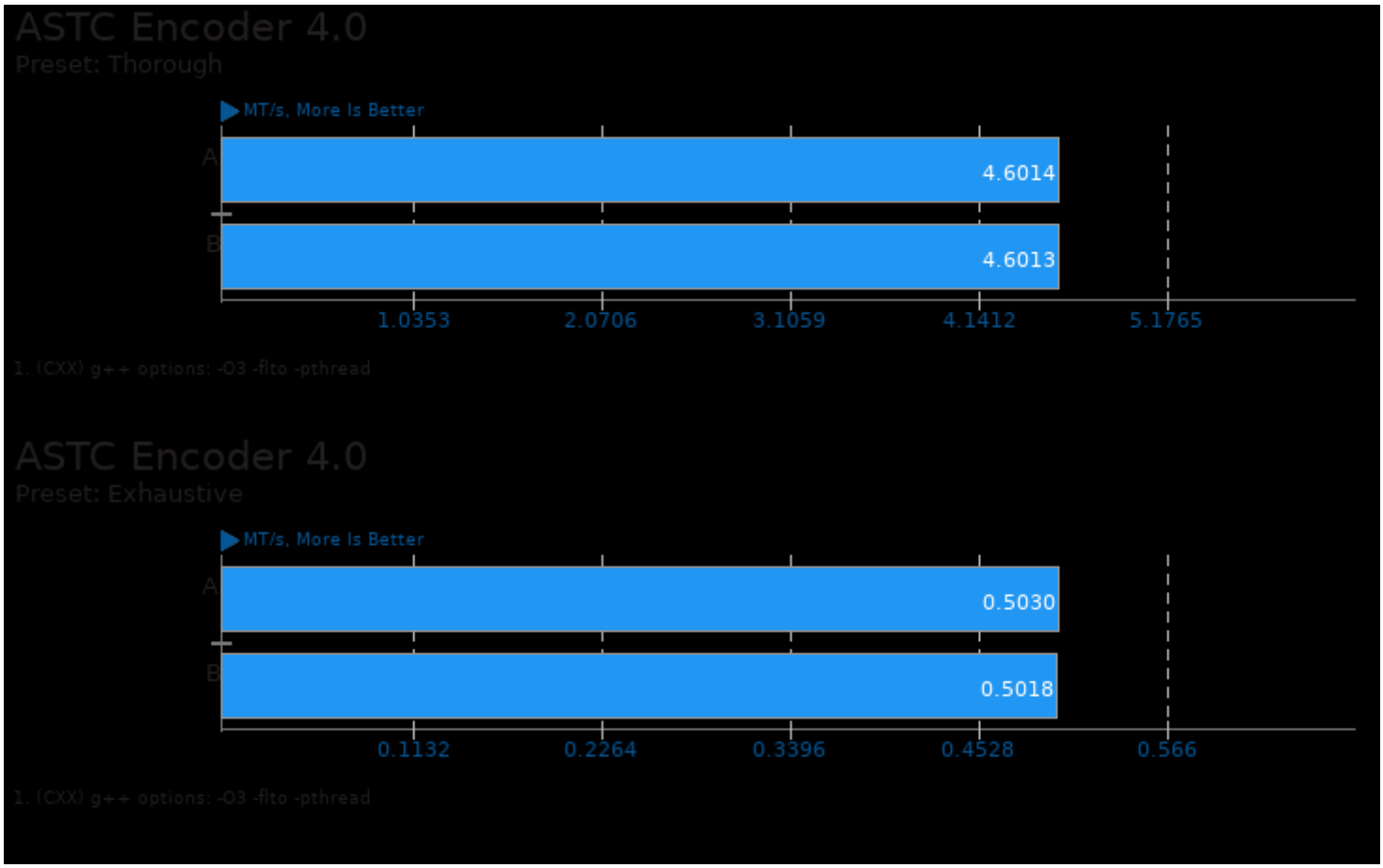
1. (CXX) g++ options: -O3 -flto -pthread

ASTC Encoder 4.0

Preset: Medium

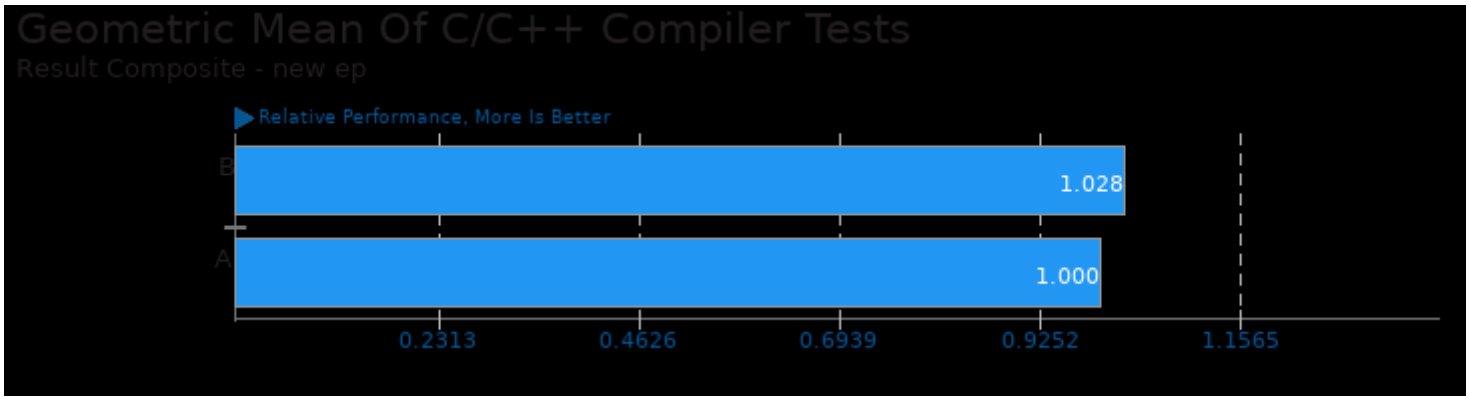


1. (CXX) g++ options: -O3 -flto -pthread

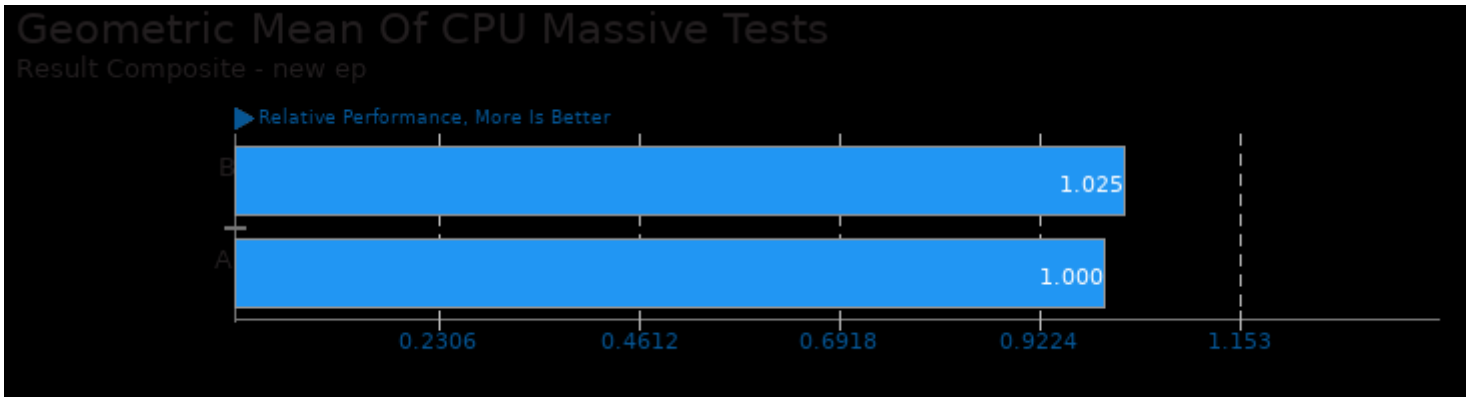

ASTC Encoder 4.0
Preset: Thorough
MT/s, More Is Better
A
4.6014
B
4.6013
1.0353
2.0706
3.1059
4.1412
5.1765
1. (CXX) g++ options: -O3 -flto -pthread
ASTC Encoder 4.0
Preset: Exhaustive
MT/s, More Is Better
A
0.5030
B
0.5018
0.1132
0.2264
0.3396
0.4528
0.566
1. (CXX) g++ options: -O3 -flto -pthread

new ep

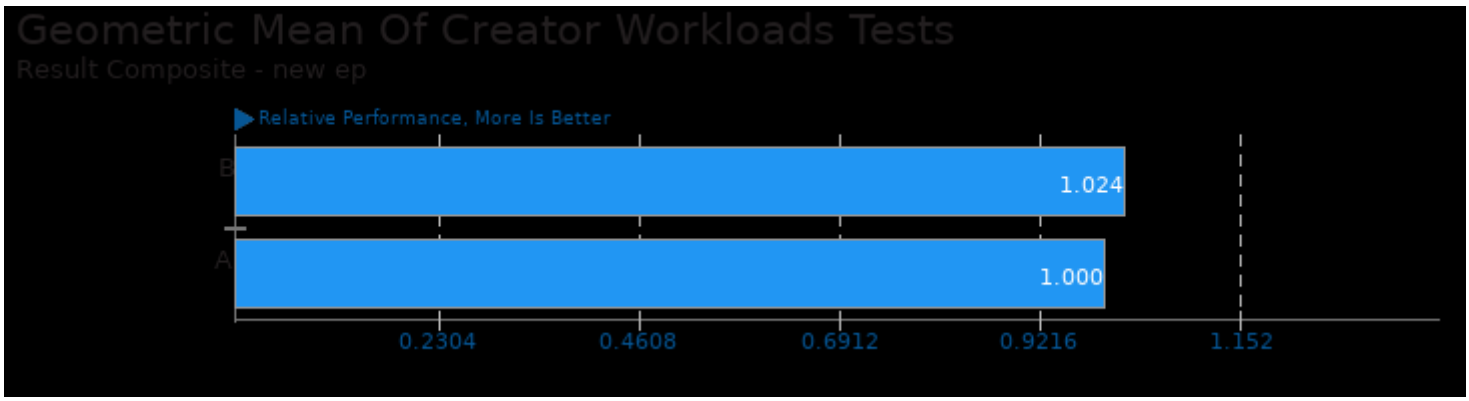
These geometric means are based upon test groupings / test suites for this result file.



Geometric mean based upon tests: pts/lammps, pts/aircrack-ng and pts/svt-av1

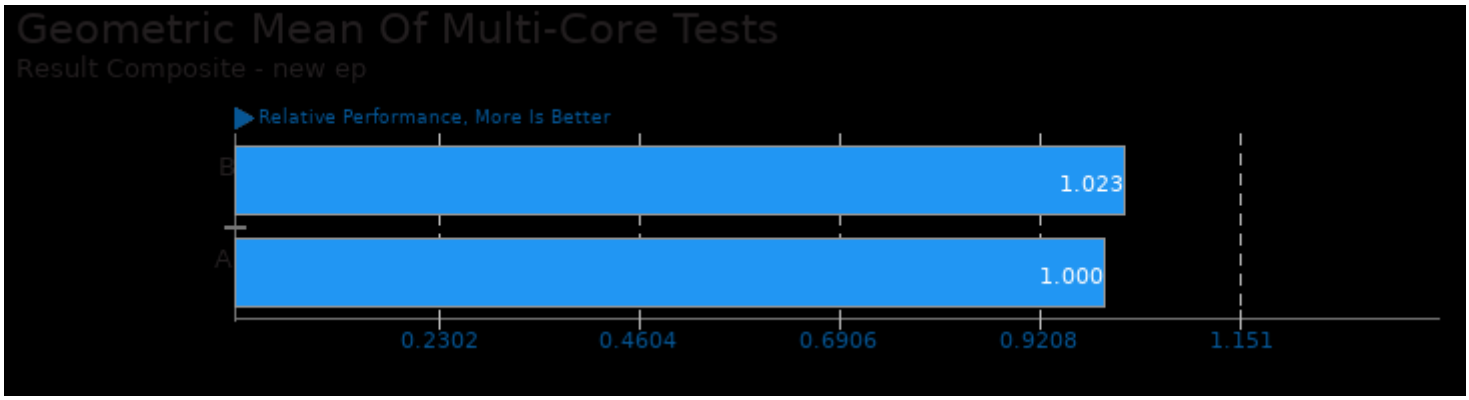


Geometric mean based upon tests: pts/svt-av1, pts/lammps and pts/primesieve



Geometric mean based upon tests: pts/svt-av1 and pts/astcenc

new ep



Geometric mean based upon tests: pts/aircrack-ng, pts/svt-av1, pts/primesieve and pts/lammps



Geometric mean based upon tests: pts/node-web-tooling and pts/spark

This file was automatically generated via the Phoronix Test Suite benchmarking software on Thursday, 28 March 2024 12:03.