

www.phoronix-test-suite.com

spark svt

AMD EPYC 7551 32-Core testing with a GIGABYTE MZ31-AR0-00 v01010101 (F10 BIOS) and ASPEED on Debian 11 via the Phoronix Test Suite.

Automated Executive Summary

C had the most wins, coming in first place for 30% of the tests.

Based on the geometric mean of all complete results, the fastest (C) was 1.01x the speed of the slowest (B). A was 0.995x the speed of C, D was 0.995x the speed of A, B was 0.999x the speed of D.

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

Apache Spark (Row Count: 1000000 - Partitions: 100 - Repartition Test Time) at 2.052x
Apache Spark (Row Count: 1000000 - Partitions: 1000 - Repartition Test Time) at 1.74x
Apache Spark (Row Count: 1000000 - Partitions: 2000 - Broadcast Inner Join Test Time) at 1.694x
Apache Spark (Row Count: 1000000 - Partitions: 100 - Broadcast Inner Join Test Time) at 1.512x
Apache Spark (Row Count: 1000000 - Partitions: 500 - Broadcast Inner Join Test Time) at 1.437x
Apache Spark (Row Count: 1000000 - Partitions: 500 - Repartition Test Time) at 1.432x
Apache Spark (Row Count: 1000000 - Partitions: 100 - Inner Join Test Time) at 1.423x
Apache Spark (Row Count: 10000000 - Partitions: 100 - Repartition Test Time) at 1.358x
Apache Spark (Row Count: 1000000 - Partitions: 1000 - Broadcast Inner Join Test Time) at 1.344x

spark svt

Apache Spark (Row Count: 40000000 - Partitions: 100 - SHA-512 Benchmark Time) at 1.316x.

Test Systems:

A

B

C

D

Processor: AMD EPYC 7551 32-Core @ 2.00GHz (32 Cores / 64 Threads), Motherboard: GIGABYTE MZ31-AR0-00 v01010101 (F10 BIOS), Chipset: AMD 17h, Memory: 8 x 4 GB DDR4-2667MT/s 9ASF51272PZ-2G6E1, Disk: Samsung SSD 960 EVO 500GB + 31GB SanDisk 3.2Gen1, Graphics: ASPEED, Network: Realtek RTL8111/8168/8411 + 2 x Broadcom NetXtreme II BCM57810 10

OS: Debian 11, Kernel: 5.10.0-9-amd64 (x86_64), Compiler: GCC 10.2.1 20210110, File-System: ext4, Screen Resolution: 1024x768

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: 0x8001227
Java Notes: OpenJDK Runtime Environment (build 11.0.12+7-post-Debian-2)
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 STIBP: disabled RSB filling + srbds: Not affected + tsx_async_abort: Not affected

	A	B	C	D
Apache Spark - 40000000 - 2000 - B.I.J.T.T (sec)	**38.32**	39.54	39.50	**40.69**
Normalized	100%	96.91%	97.01%	94.18%
Apache Spark - 40000000 - 2000 - I.J.T.T (sec)	**37.30**	**42.99**	41.55	40.05
Normalized	100%	86.76%	89.77%	93.13%
Apache Spark - 40000000 - 2000 - R.T.T (sec)	**34.29**	36.69	34.33	**37.23**
Normalized	100%	93.46%	99.88%	92.1%

spark svt

Apache Spark - 40000000 - 2000 - Group By Test Time (sec)	45.70	**46.88**	**43.42**	45.96
Normalized	95.01%	92.62%	100%	94.47%
Apache Spark - 40000000 - 2000 - C.P.B.U.D (sec)	6.03	**6.41**	6.09	**6.02**
Normalized	99.83%	93.92%	98.85%	100%
Apache Spark - 40000000 - 2000 - C.P.B (sec)	**80.61**	80.46	80.01	**79.77**
Normalized	98.96%	99.14%	99.7%	100%
Apache Spark - 40000000 - 2000 - S.5.B.T (sec)	63.54	**68.38**	65.00	**63.32**
Normalized	99.65%	92.6%	97.42%	100%
Apache Spark - 40000000 - 500 - B.I.J.T.T (sec)	**35.72**	**36.89**	36.39	36.09
Normalized	100%	96.83%	98.16%	98.97%
Apache Spark - 40000000 - 500 - I.J.T.T (sec)	**35.66**	35.86	**37.65**	36.94
Normalized	100%	99.44%	94.71%	96.53%
Apache Spark - 40000000 - 500 - R.T.T (sec)	**35.70**	**34.16**	35.64	34.35
Normalized	95.69%	100%	95.85%	99.45%
Apache Spark - 40000000 - 500 - Group By Test Time (sec)	**46.76**	45.47	**44.61**	45.73
Normalized	95.4%	98.11%	100%	97.55%
Apache Spark - 40000000 - 500 - C.P.B.U.D (sec)	6.07	**6.26**	6.18	**6.04**
Normalized	99.51%	96.49%	97.73%	100%
Apache Spark - 40000000 - 500 - C.P.B (sec)	80.69	**81.20**	80.43	**80.07**
Normalized	99.23%	98.61%	99.55%	100%
Apache Spark - 40000000 - 500 - S.5.B.T (sec)	**63.08**	65.06	66.57	**70.38**
Normalized	100%	96.96%	94.76%	89.63%
Apache Spark - 40000000 - 100 - B.I.J.T.T (sec)	**39.41**	**35.04**	37.29	36.87
Normalized	88.91%	100%	93.97%	95.04%
Apache Spark - 40000000 - 100 - I.J.T.T (sec)	**38.22**	36.85	**36.68**	37.70
Normalized	95.97%	99.54%	100%	97.29%
Apache Spark - 40000000 - 100 - R.T.T (sec)	**37.61**	**33.43**	34.68	36.69
Normalized	88.89%	100%	96.4%	91.11%
Apache Spark - 40000000 - 100 - Group By Test Time (sec)	42.58	**42.59**	**38.98**	40.39
Normalized	91.55%	91.52%	100%	96.51%
Apache Spark - 40000000 - 100 - C.P.B.U.D (sec)	6.06	6.09	**5.90**	**6.11**
Normalized	97.36%	96.88%	100%	96.56%
Apache Spark - 40000000 - 100 - C.P.B (sec)	80.39	**80.88**	**79.71**	79.72
Normalized	99.15%	98.55%	100%	99.99%

spark svt

Apache Spark - 40000000 - 100 - S.5.B.T (sec)	79.48	61.73	60.38	71.77
Normalized	75.97%	97.81%	100%	84.13%
Apache Spark - 40000000 - 1000 - B.I.J.T.T (sec)	37.36	36.63	36.12	35.58
Normalized	95.24%	97.13%	98.5%	100%
Apache Spark - 40000000 - 1000 - I.J.T.T (sec)	37.92	37.54	36.42	36.82
Normalized	96.04%	97.02%	100%	98.91%
Apache Spark - 40000000 - 1000 - R.T.T (sec)	35.82	35.27	34.77	34.94
Normalized	97.07%	98.58%	100%	99.51%
Apache Spark - 40000000 - 1000 - Group By Test Time (sec)	45.87	43.70	41.48	42.19
Normalized	90.43%	94.92%	100%	98.32%
Apache Spark - 40000000 - 1000 - C.P.B.U.D (sec)	6.33	6.04	6.24	6.26
Normalized	95.42%	100%	96.79%	96.49%
Apache Spark - 40000000 - 1000 - C.P.B (sec)	80.96	80.79	79.91	80.39
Normalized	98.7%	98.91%	100%	99.4%
Apache Spark - 40000000 - 1000 - S.5.B.T (sec)	71.67	63.03	63.36	65.19
Normalized	87.94%	100%	99.48%	96.69%
Apache Spark - 20000000 - 2000 - B.I.J.T.T (sec)	21.60	22.24	21.42	22.39
Normalized	99.17%	96.31%	100%	95.67%
Apache Spark - 20000000 - 2000 - I.J.T.T (sec)	23.02	22.88	22.15	23.34
Normalized	96.22%	96.81%	100%	94.9%
Apache Spark - 20000000 - 2000 - R.T.T (sec)	23.43	20.30	20.31	22.03
Normalized	86.64%	100%	99.95%	92.15%
Apache Spark - 20000000 - 2000 - Group By Test Time (sec)	23.47	23.71	22.79	22.55
Normalized	96.08%	95.11%	98.95%	100%
Apache Spark - 20000000 - 2000 - C.P.B.U.D (sec)	6.10	6.01	5.92	6.08
Normalized	97.05%	98.5%	100%	97.37%
Apache Spark - 20000000 - 2000 - C.P.B (sec)	80.42	79.81	78.65	79.09
Normalized	97.8%	98.55%	100%	99.44%
Apache Spark - 20000000 - 2000 - S.5.B.T (sec)	39.28	39.19	36.55	37.41
Normalized	93.05%	93.26%	100%	97.7%
Apache Spark - 20000000 - 1000 - B.I.J.T.T (sec)	20.17	19.47	19.45	19.77
Normalized	96.43%	99.9%	100%	98.38%
Apache Spark - 20000000 - 1000 - I.J.T.T (sec)	21.88	21.32	21.06	21.83
Normalized	96.25%	98.78%	100%	96.47%

spark svt

Apache Spark - 20000000 - 1000 - R.T.T (sec)	**22.42**	23.28	22.84	**24.90**
Normalized	100%	96.31%	98.16%	90.04%
Apache Spark - 20000000 - 1000 - Group By Test Time (sec)	21.57	22.62	**23.78**	**19.23**
Normalized	89.15%	85.01%	80.87%	100%
Apache Spark - 20000000 - 1000 - C.P.B.U.D (sec)	**6.04**	6.09	6.11	**6.27**
Normalized	100%	99.18%	98.85%	96.33%
Apache Spark - 20000000 - 1000 - C.P.B (sec)	**80.72**	79.73	79.44	**79.04**
Normalized	97.92%	99.13%	99.5%	100%
Apache Spark - 20000000 - 1000 - S.5.B.T (sec)	38.63	**41.15**	**35.63**	36.37
Normalized	92.23%	86.59%	100%	97.97%
Apache Spark - 20000000 - 100 - B.I.J.T.T (sec)	19.70	**19.17**	**21.90**	20.34
Normalized	97.31%	100%	87.53%	94.25%
Apache Spark - 20000000 - 100 - I.J.T.T (sec)	21.76	**20.00**	**22.90**	22.59
Normalized	91.91%	100%	87.34%	88.53%
Apache Spark - 20000000 - 100 - R.T.T (sec)	**23.16**	20.08	21.41	**19.57**
Normalized	84.5%	97.46%	91.41%	100%
Apache Spark - 20000000 - 100 - Group By Test Time (sec)	**21.05**	20.29	**18.53**	18.98
Normalized	88.03%	91.33%	100%	97.63%
Apache Spark - 20000000 - 100 - C.P.B.U.D (sec)	**6.03**	**6.03**	**5.97**	6.02
Normalized	99%	99%	100%	99.17%
Apache Spark - 20000000 - 100 - C.P.B (sec)	**79.16**	**79.73**	79.19	79.25
Normalized	100%	99.29%	99.96%	99.89%
Apache Spark - 20000000 - 100 - S.5.B.T (sec)	36.15	35.84	**34.03**	**44.06**
Normalized	94.14%	94.95%	100%	77.24%
Apache Spark - 20000000 - 500 - B.I.J.T.T (sec)	18.50	**18.12**	**20.08**	18.79
Normalized	97.95%	100%	90.24%	96.43%
Apache Spark - 20000000 - 500 - I.J.T.T (sec)	19.961697191	**19.46**	**20.95**	20.17
Normalized	97.49%	100%	92.89%	96.48%
Apache Spark - 20000000 - 500 - R.T.T (sec)	**19.18**	**23.49**	22.41	19.20
Normalized	100%	81.65%	85.59%	99.9%
Apache Spark - 20000000 - 500 - Group By Test Time (sec)	**21.82**	**19.57**	20.64	21.59
Normalized	89.69%	100%	94.82%	90.64%
Apache Spark - 20000000 - 500 - C.P.B.U.D (sec)	**6.29**	6.17	6.07	**5.91**
Normalized	93.96%	95.79%	97.36%	100%

spark svt

Apache Spark - 20000000 - 500 - C.P.B (sec)	80.53	79.91	80.02	79.76
Normalized	99.04%	99.81%	99.68%	100%
Apache Spark - 20000000 - 500 - S.5.B.T (sec)	37.11	37.95	40.03	36.68
Normalized	98.84%	96.65%	91.63%	100%
Apache Spark - 10000000 - 2000 - B.I.J.T.T (sec)	14.71	12.97	11.82	11.75
Normalized	79.88%	90.59%	99.41%	100%
Apache Spark - 10000000 - 2000 - I.J.T.T (sec)	17.08	14.95	15.38	14.55
Normalized	85.19%	97.32%	94.6%	100%
Apache Spark - 10000000 - 2000 - R.T.T (sec)	13.87	14.26	13.46	13.26
Normalized	95.6%	92.99%	98.51%	100%
Apache Spark - 10000000 - 2000 - Group By Test Time (sec)	19.09	17.98	18.47	16.30
Normalized	85.39%	90.66%	88.25%	100%
Apache Spark - 10000000 - 2000 - C.P.B.U.D (sec)	6.19	5.93	6.16	6.18
Normalized	95.8%	100%	96.27%	95.95%
Apache Spark - 10000000 - 2000 - C.P.B (sec)	80.74	80.67	79.91	80.06
Normalized	98.97%	99.06%	100%	99.81%
Apache Spark - 10000000 - 2000 - S.5.B.T (sec)	23.00	25.03	22.31	22.89
Normalized	97%	89.13%	100%	97.47%
Apache Spark - 10000000 - 1000 - B.I.J.T.T (sec)	12.97	11.13	11.95	11.71
Normalized	85.81%	100%	93.14%	95.05%
Apache Spark - 10000000 - 1000 - I.J.T.T (sec)	13.87	13.76	14.40	13.19
Normalized	95.1%	95.86%	91.6%	100%
Apache Spark - 10000000 - 1000 - R.T.T (sec)	15.37	13.65	14.85	14.34
Normalized	88.81%	100%	91.92%	95.19%
Apache Spark - 10000000 - 1000 - Group By Test Time (sec)	17.09	16.50	15.06	14.20
Normalized	83.09%	86.06%	94.29%	100%
Apache Spark - 10000000 - 1000 - C.P.B.U.D (sec)	6.18	6.28	6.50	6.37
Normalized	100%	98.41%	95.08%	97.02%
Apache Spark - 10000000 - 1000 - C.P.B (sec)	80.86	80.06	79.27	79.84
Normalized	98.03%	99.01%	100%	99.29%
Apache Spark - 10000000 - 1000 - S.5.B.T (sec)	22.56	22.14	21.85	21.02
Normalized	93.17%	94.94%	96.2%	100%
Apache Spark - 10000000 - 100 - B.I.J.T.T (sec)	11.67	10.56	11.12	10.93
Normalized	90.49%	100%	94.96%	96.61%

spark svt

Apache Spark - 10000000 - 100 - I.J.T.T (sec)	**13.51**	**11.79**	13.11	13.35
Normalized	87.27%	100%	89.93%	88.31%
Apache Spark - 10000000 - 100 - R.T.T (sec)	**12.31**	**16.72**	15.71	16.34
Normalized	100%	73.62%	78.36%	75.34%
Apache Spark - 10000000 - 100 - Group By Test Time (sec)	**14.55**	15.10	15.18	**17.51**
Normalized	100%	96.36%	95.85%	83.1%
Apache Spark - 10000000 - 100 - C.P.B.U.D (sec)	6.40	**6.27**	6.31	**6.43**
Normalized	97.97%	100%	99.37%	97.51%
Apache Spark - 10000000 - 100 - C.P.B (sec)	78.80	**79.50**	**78.44**	78.89
Normalized	99.54%	98.67%	100%	99.43%
Apache Spark - 10000000 - 100 - S.5.B.T (sec)	22.38	**21.43**	24.07	**24.39**
Normalized	95.76%	100%	89.03%	87.86%
Apache Spark - 10000000 - 500 - B.I.J.T.T (sec)	12.28	11.23	**12.29**	**10.81**
Normalized	88.03%	96.26%	87.96%	100%
Apache Spark - 10000000 - 500 - I.J.T.T (sec)	12.56	**12.17**	**14.15**	13.23
Normalized	96.89%	100%	86.01%	91.99%
Apache Spark - 10000000 - 500 - R.T.T (sec)	15.13	13.72	**12.64**	**15.26**
Normalized	83.54%	92.13%	100%	82.83%
Apache Spark - 10000000 - 500 - Group By Test Time (sec)	15.30	**19.54**	**15.00**	15.50
Normalized	98.04%	76.77%	100%	96.77%
Apache Spark - 10000000 - 500 - C.P.B.U.D (sec)	6.24	**6.14**	6.22	**6.56**
Normalized	98.4%	100%	98.71%	93.6%
Apache Spark - 10000000 - 500 - C.P.B (sec)	**79.68**	79.44	79.27	**79.22**
Normalized	99.42%	99.72%	99.94%	100%
Apache Spark - 10000000 - 500 - S.5.B.T (sec)	**21.91**	**22.51**	22.00	21.95
Normalized	100%	97.33%	99.59%	99.82%
SVT-AV1 - Preset 4 - Bosphorus 4K (FPS)	**0.945**	**1.008**	0.974	0.984
Normalized	93.75%	100%	96.63%	97.62%
Apache Spark - 1000000 - 2000 - B.I.J.T.T (sec)	5.62	**7.74**	**4.57**	5.71
Normalized	81.32%	59.04%	100%	80.04%
Apache Spark - 1000000 - 2000 - I.J.T.T (sec)	**5.74**	5.87	**6.52**	5.75
Normalized	100%	97.79%	88.04%	99.83%
Apache Spark - 1000000 - 2000 - R.T.T (sec)	6.19	**7.19**	**5.76**	6.97
Normalized	93.05%	80.11%	100%	82.64%

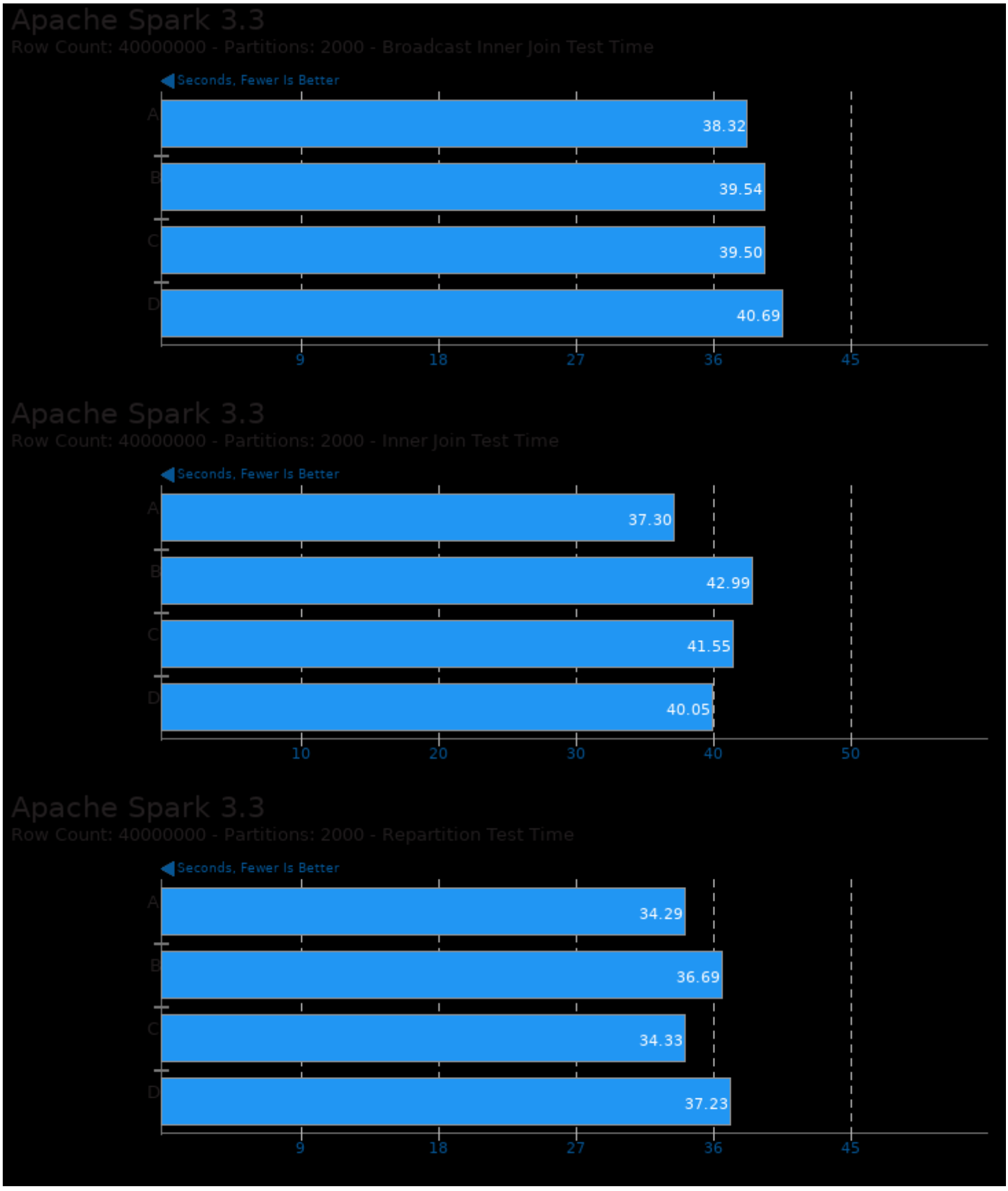
spark svt

Apache Spark - 1000000 - 2000 - Group By Test Time (sec)	10.05	**10.77**	10.13	**10.01**
Normalized	99.6%	92.94%	98.82%	100%
Apache Spark - 1000000 - 2000 - C.P.B.U.D (sec)	**6.57**	**6.20**	6.21	6.32
Normalized	94.37%	100%	99.84%	98.1%
Apache Spark - 1000000 - 2000 - C.P.B (sec)	78.84	**79.73**	78.70	**78.50**
Normalized	99.57%	98.46%	99.75%	100%
Apache Spark - 1000000 - 2000 - S.5.B.T (sec)	8.31	**8.61**	**7.71**	8.56
Normalized	92.78%	89.55%	100%	90.07%
Apache Spark - 1000000 - 1000 - B.I.J.T.T (sec)	**3.02**	**4.06**	3.25	3.28
Normalized	100%	74.38%	92.92%	92.07%
Apache Spark - 1000000 - 1000 - I.J.T.T (sec)	4.53	5.01	**5.47**	**4.33**
Normalized	95.58%	86.43%	79.16%	100%
Apache Spark - 1000000 - 1000 - R.T.T (sec)	5.43	5.68	**5.30**	**9.22**
Normalized	97.61%	93.31%	100%	57.48%
Apache Spark - 1000000 - 1000 - Group By Test Time (sec)	**9.99**	**10.81**	10.42	10.15
Normalized	100%	92.41%	95.87%	98.42%
Apache Spark - 1000000 - 1000 - C.P.B.U.D (sec)	**6.08**	6.40	**6.54**	6.11
Normalized	100%	95%	92.97%	99.51%
Apache Spark - 1000000 - 1000 - C.P.B (sec)	**78.88**	78.84	**78.56**	78.62
Normalized	99.59%	99.64%	100%	99.92%
Apache Spark - 1000000 - 1000 - S.5.B.T (sec)	7.76	**7.43**	**8.75**	7.81
Normalized	95.75%	100%	84.91%	95.13%
Apache Spark - 1000000 - 500 - B.I.J.T.T (sec)	3.16	3.56	**4.47**	**3.11**
Normalized	98.42%	87.36%	69.57%	100%
Apache Spark - 1000000 - 500 - I.J.T.T (sec)	**4.40**	4.58	4.43	**5.62**
Normalized	100%	96.07%	99.32%	78.29%
Apache Spark - 1000000 - 500 - R.T.T (sec)	**5.09**	5.98	**7.29**	5.83
Normalized	100%	85.12%	69.82%	87.31%
Apache Spark - 1000000 - 500 - Group By Test Time (sec)	9.85	**9.30**	9.53	**9.99**
Normalized	94.42%	100%	97.59%	93.09%
Apache Spark - 1000000 - 500 - C.P.B.U.D (sec)	**6.02**	**6.42**	6.26	6.30
Normalized	100%	93.77%	96.17%	95.56%
Apache Spark - 1000000 - 500 - C.P.B (sec)	**78.34**	**79.11**	78.54	78.64
Normalized	100%	99.03%	99.75%	99.62%

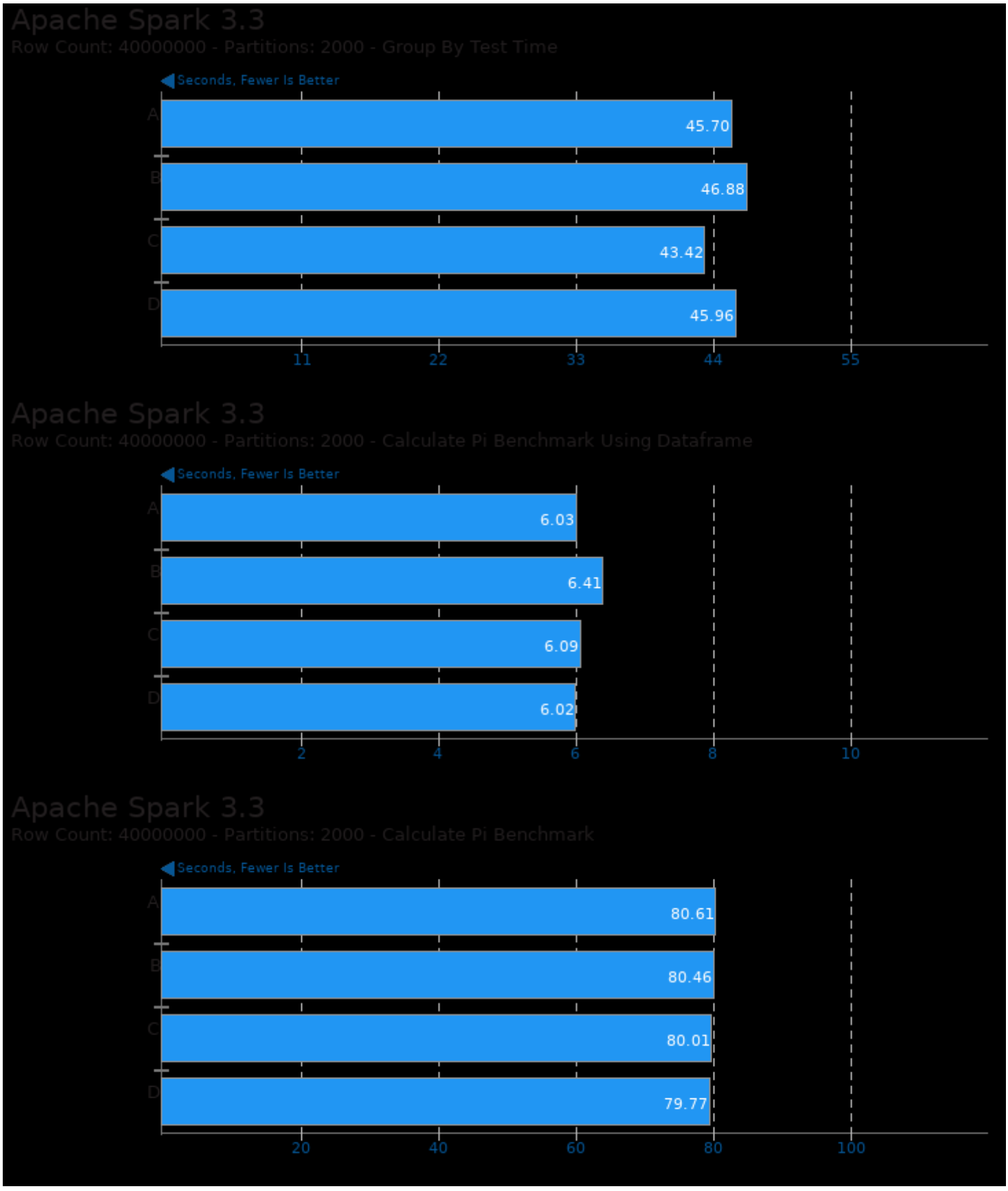
spark svt

Apache Spark - 1000000 - 500 - S.5.B.T (sec)	**8.57**	**7.57**	7.71	7.82
Normalized	88.33%	100%	98.18%	96.8%
Apache Spark - 1000000 - 100 - B.I.J.T.T (sec)	**3.10**	2.56	**2.05**	2.91
Normalized	66.13%	80.08%	100%	70.45%
Apache Spark - 1000000 - 100 - I.J.T.T (sec)	3.65	**5.01**	3.97	**3.52**
Normalized	96.44%	70.26%	88.66%	100%
Apache Spark - 1000000 - 100 - R.T.T (sec)	4.64	**3.87**	4.77	**7.94**
Normalized	83.41%	100%	81.13%	48.74%
Apache Spark - 1000000 - 100 - Group By Test Time (sec)	9.24	**8.93**	9.25	**9.56**
Normalized	96.65%	100%	96.54%	93.41%
Apache Spark - 1000000 - 100 - C.P.B.U.D (sec)	**6.05**	**6.36**	6.26	6.26
Normalized	100%	95.13%	96.65%	96.65%
Apache Spark - 1000000 - 100 - C.P.B (sec)	**78.68**	**80.09**	79.56	78.79
Normalized	100%	98.24%	98.89%	99.86%
Apache Spark - 1000000 - 100 - S.5.B.T (sec)	7.62	**7.16**	**7.69**	7.55
Normalized	93.96%	100%	93.11%	94.83%
SVT-AV1 - Preset 4 - Bosphorus 1080p (FPS)	**2.432**	**2.539**	2.492	2.529
Normalized	95.79%	100%	98.15%	99.61%
SVT-AV1 - Preset 8 - Bosphorus 4K (FPS)	**28.876**	**24.235**	25.169	25.415
Normalized	100%	83.93%	87.16%	88.01%
SVT-AV1 - Preset 10 - Bosphorus 4K (FPS)	**49.17**	**40.566**	42.506	46.591
Normalized	100%	82.5%	86.45%	94.75%
SVT-AV1 - Preset 12 - Bosphorus 4K (FPS)	**62.042**	60.306	**54.822**	55.324
Normalized	100%	97.2%	88.36%	89.17%
SVT-AV1 - Preset 8 - Bosphorus 1080p (FPS)	61.782	**57.433**	**62.241**	60.95
Normalized	99.26%	92.28%	100%	97.93%
SVT-AV1 - Preset 10 - Bosphorus 1080p (FPS)	129.462	**119.495**	131.751	**138.694**
Normalized	93.34%	86.16%	94.99%	100%
SVT-AV1 - Preset 12 - Bosphorus 1080p (FPS)	193.672	**170.278**	**196.229**	195.506
Normalized	98.7%	86.78%	100%	99.63%

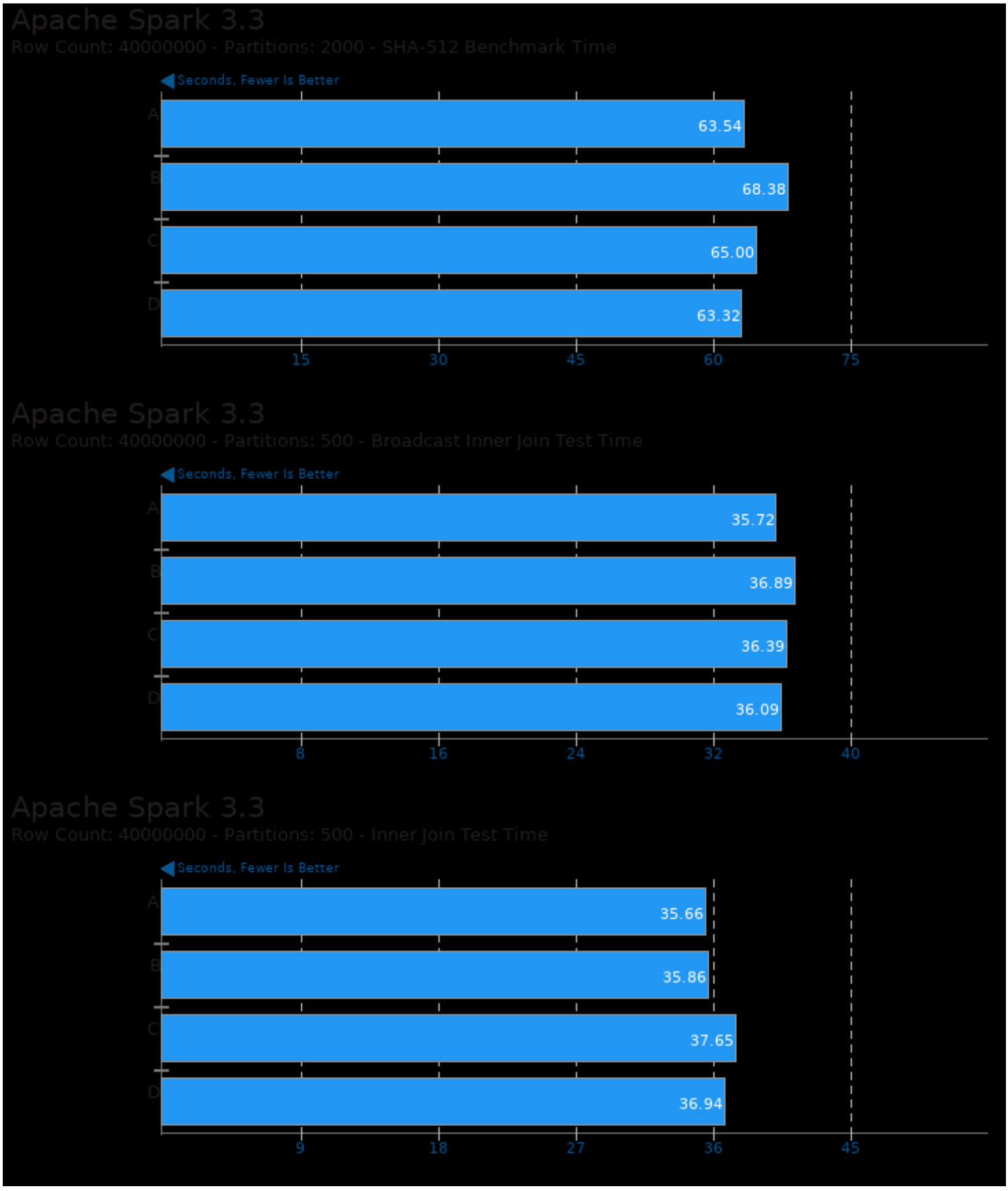
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
38.32
B
39.54
C
39.50
D
40.69
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Inner Join Test Time
Seconds, Fewer Is Better
A
37.30
B
42.99
C
41.55
D
40.05
10
20
30
40
50
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
34.29
B
36.69
C
34.33
D
37.23
9
18
27
36
45

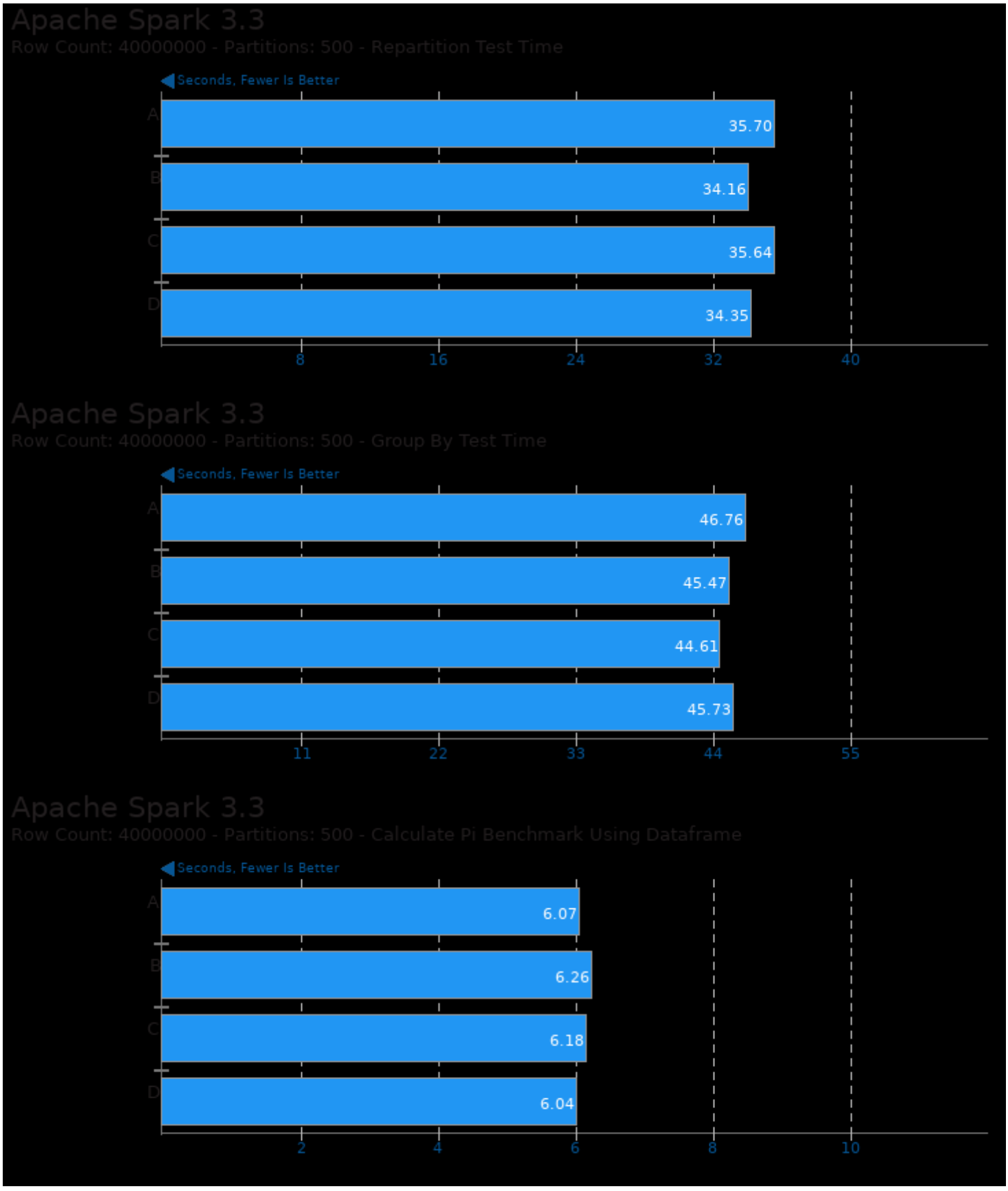
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
45.70
B
46.88
C
43.42
D
45.96
11
22
33
44
55
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.03
B
6.41
C
6.09
D
6.02
2
4
6
8
10
Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.61
B
80.46
C
80.01
D
79.77
20
40
60
80
100

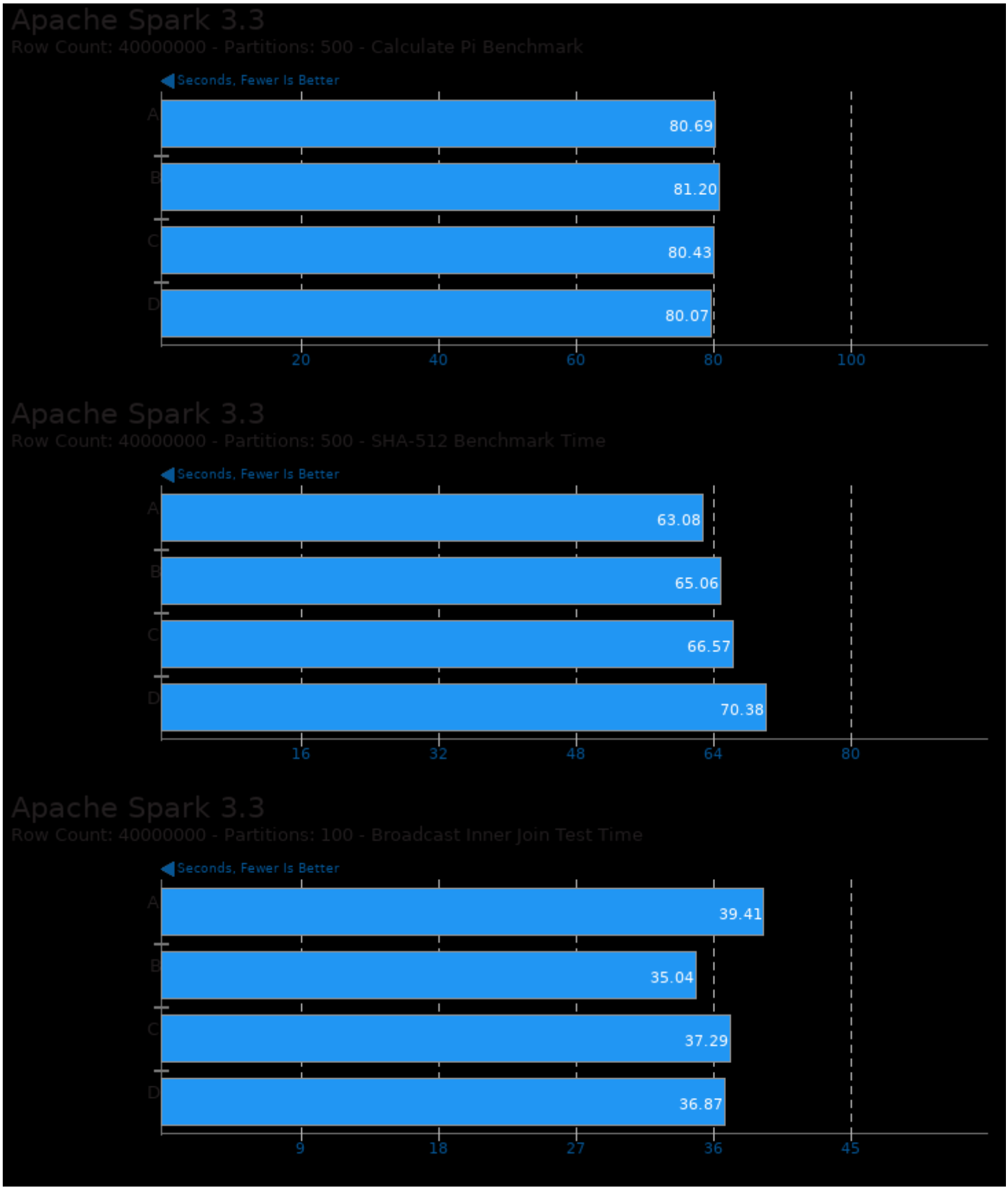
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
63.54
B
68.38
C
65.00
D
63.32
15
30
45
60
75
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
35.72
B
36.89
C
36.39
D
36.09
8
16
24
32
40
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
35.66
B
35.86
C
37.65
D
36.94
9
18
27
36
45

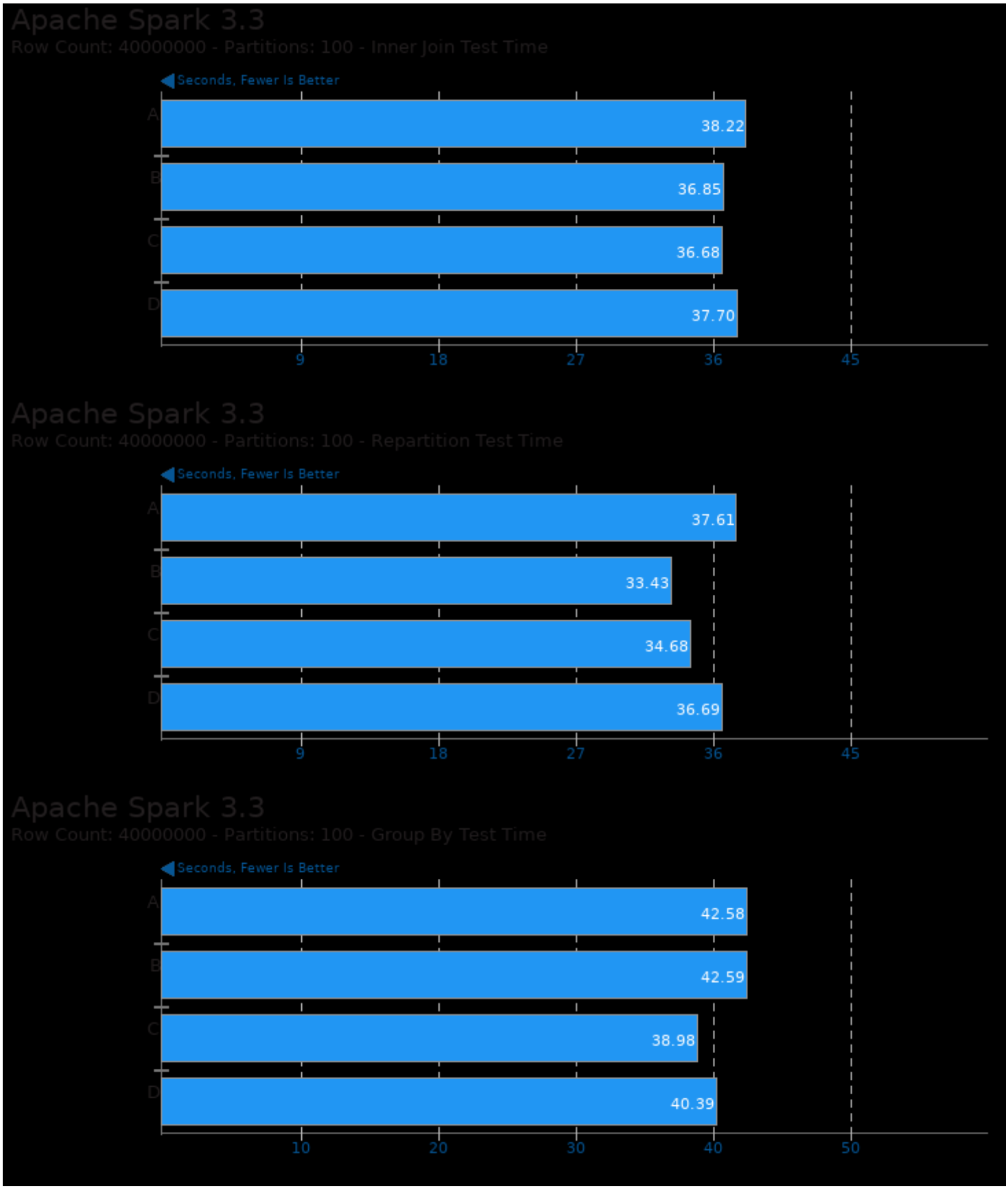
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
35.70
B
34.16
C
35.64
D
34.35
8
16
24
32
40
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
46.76
B
45.47
C
44.61
D
45.73
11
22
33
44
55
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.07
B
6.26
C
6.18
D
6.04
2
4
6
8
10

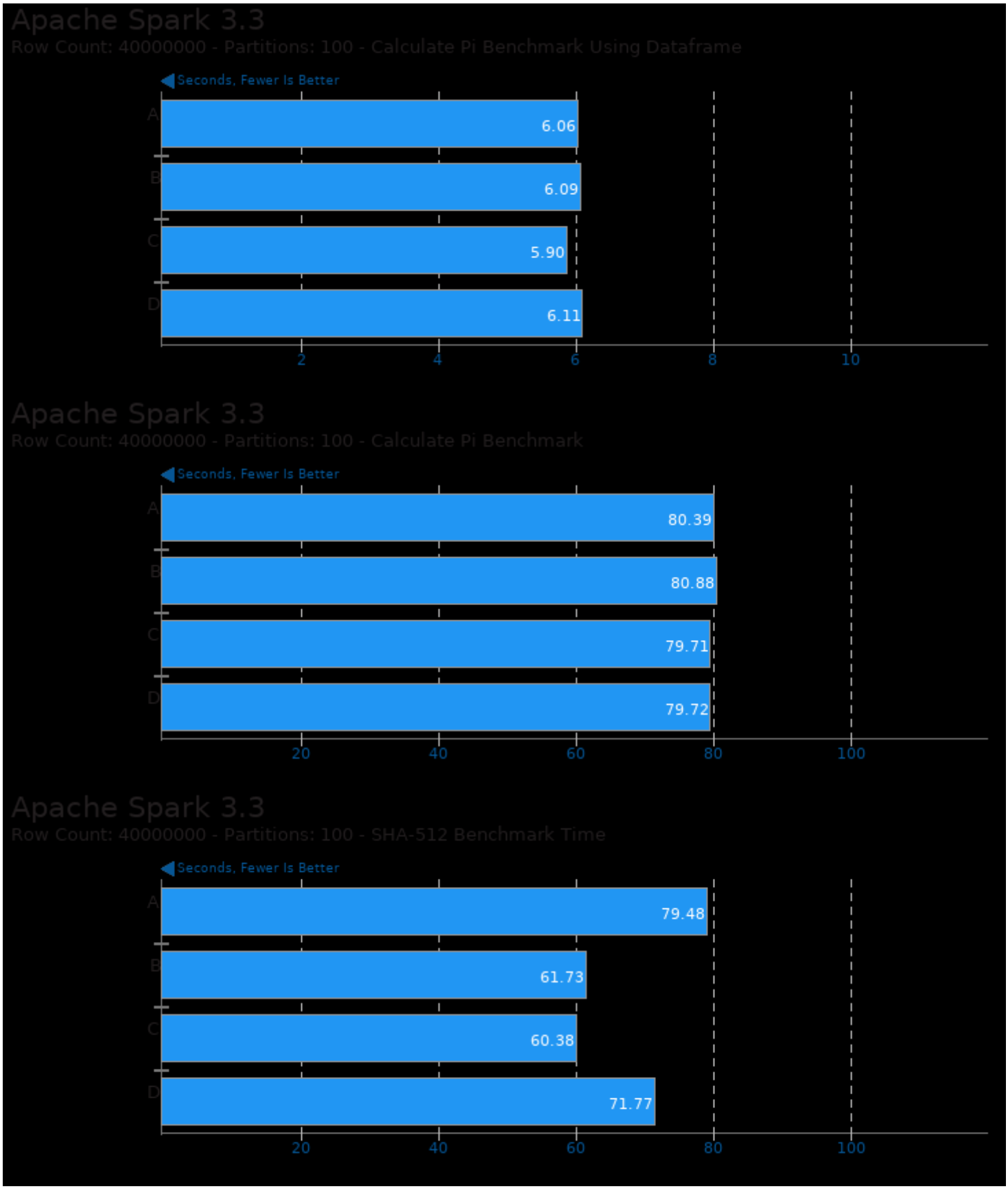
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.69
B
81.20
C
80.43
D
80.07
20
40
60
80
100
Apache Spark 3.3
Row Count: 40000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
63.08
B
65.06
C
66.57
D
70.38
16
32
48
64
80
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
39.41
B
35.04
C
37.29
D
36.87
9
18
27
36
45

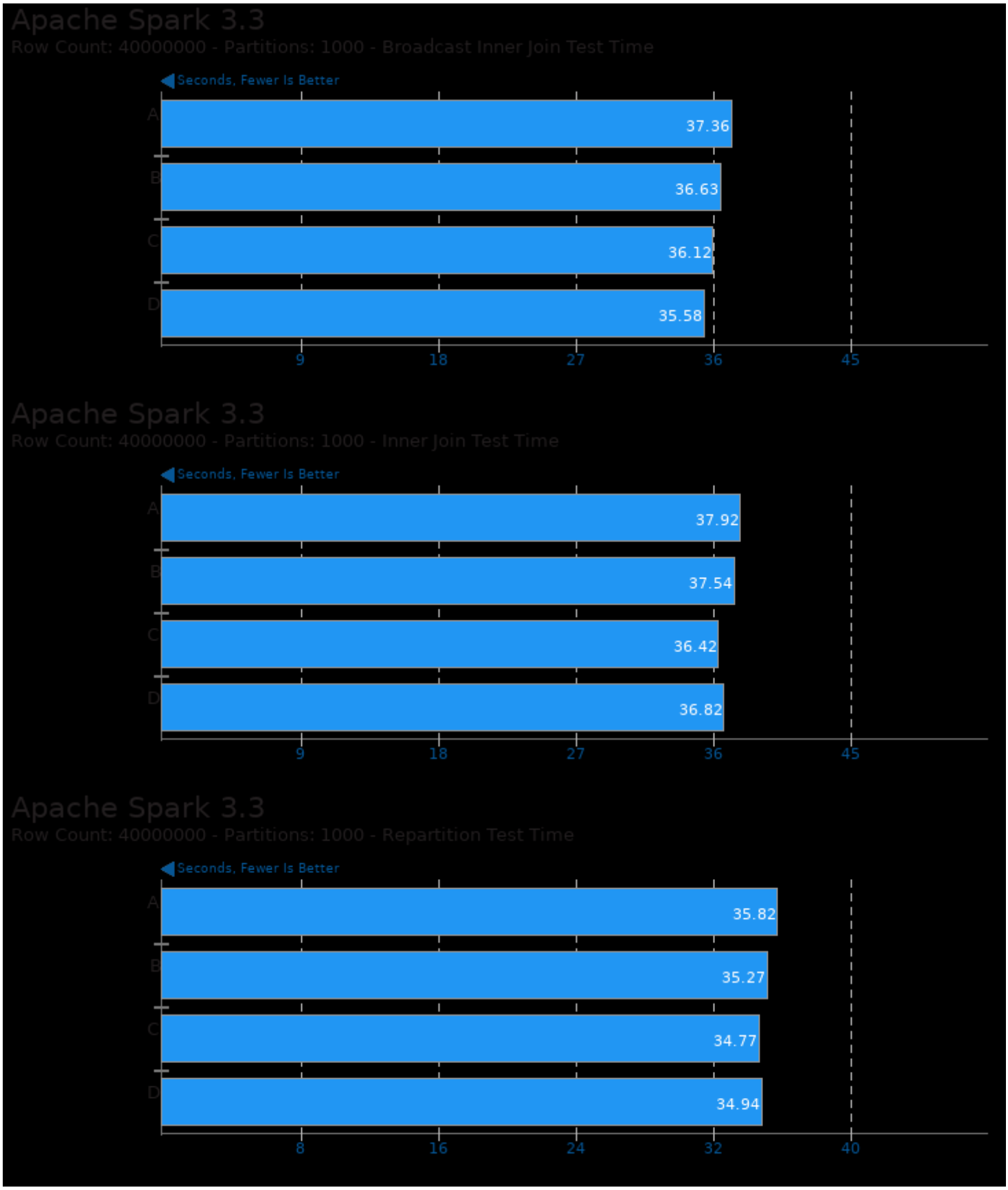
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
38.22
B
36.85
C
36.68
D
37.70
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
37.61
B
33.43
C
34.68
D
36.69
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
42.58
B
42.59
C
38.98
D
40.39
10
20
30
40
50

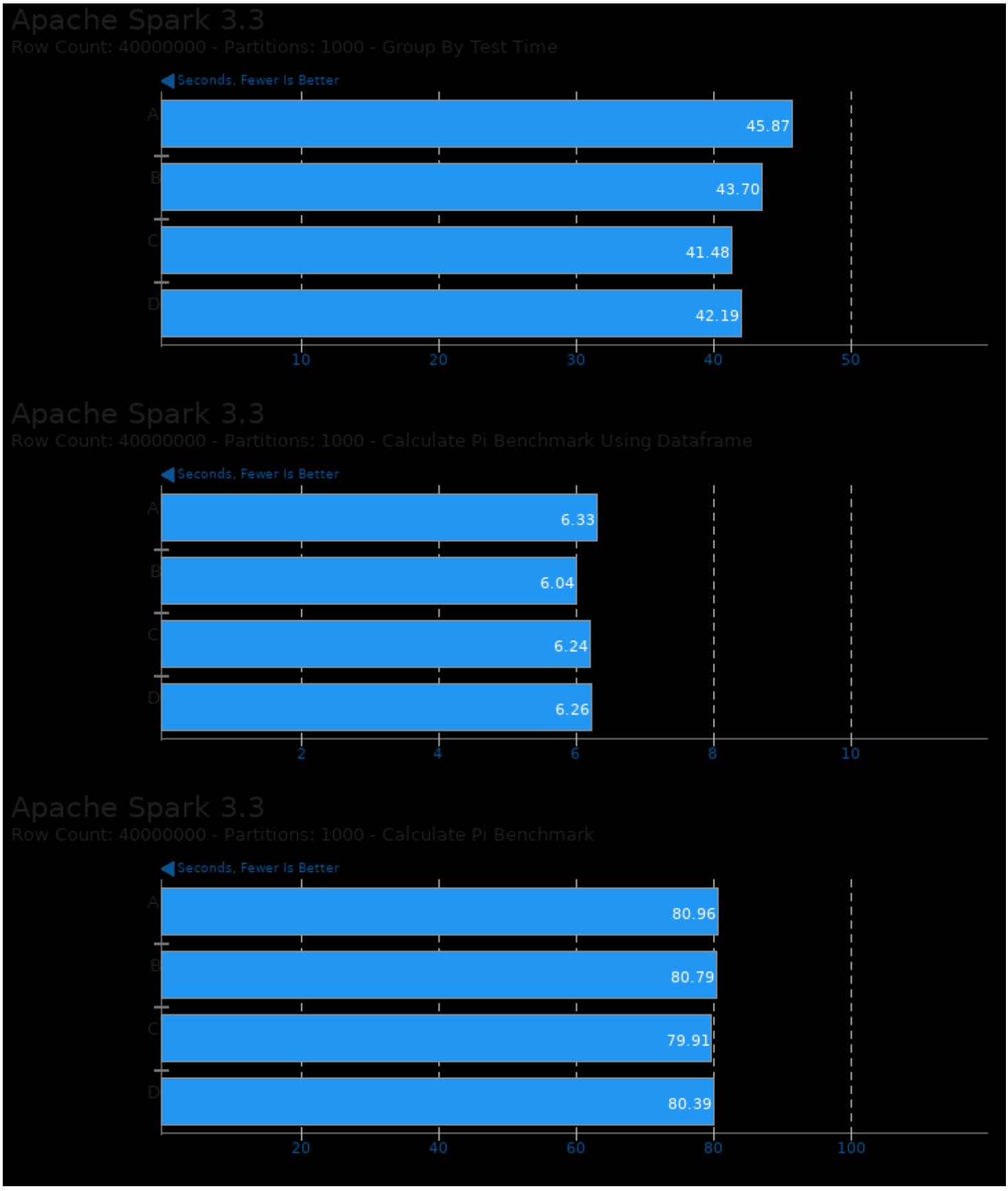
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.06
B
6.09
C
5.90
D
6.11
2
4
6
8
10
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.39
B
80.88
C
79.71
D
79.72
20
40
60
80
100
Apache Spark 3.3
Row Count: 40000000 - Partitions: 100 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
79.48
B
61.73
C
60.38
D
71.77
20
40
60
80
100

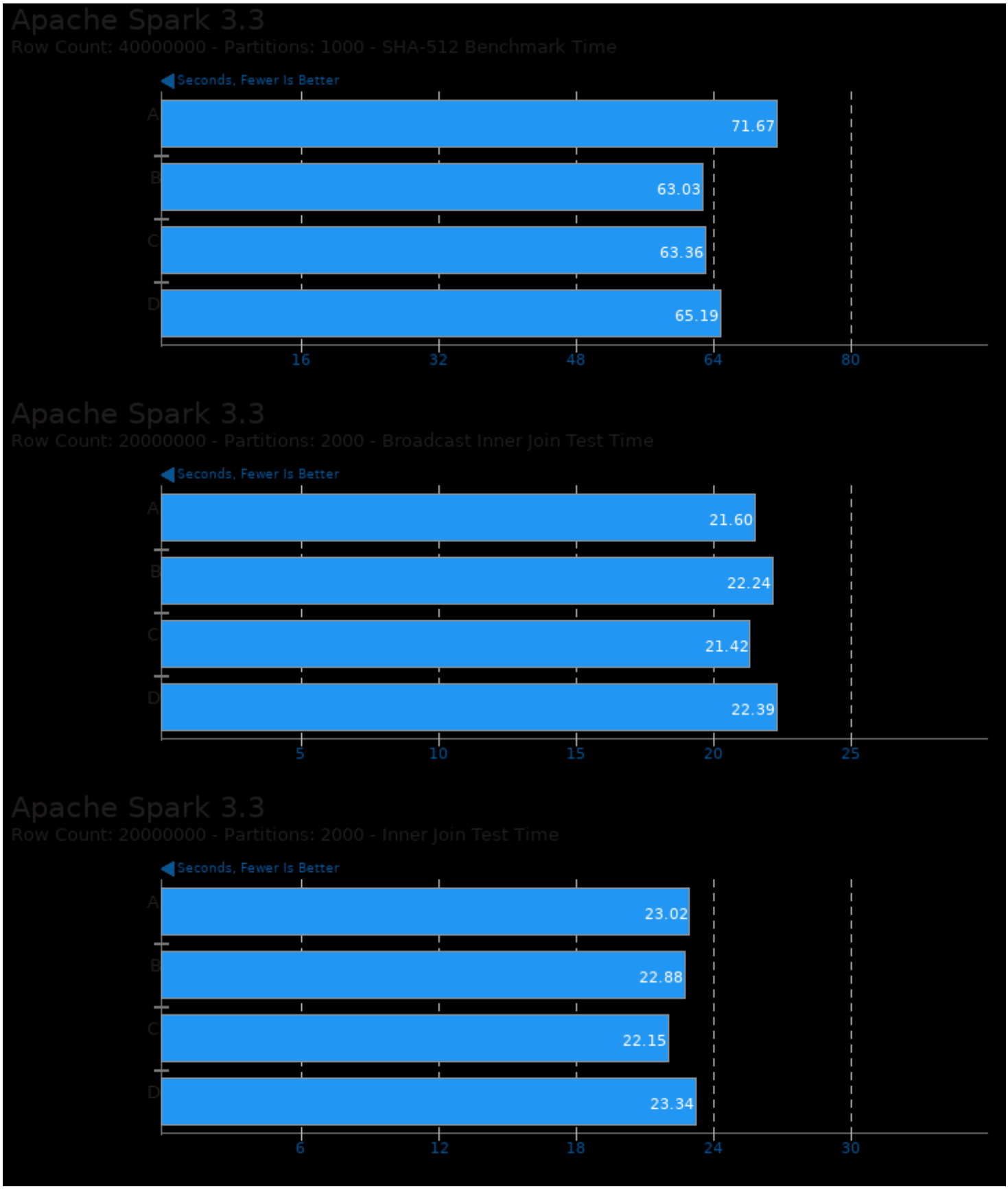
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
37.36
B
36.63
C
36.12
D
35.58
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
37.92
B
37.54
C
36.42
D
36.82
9
18
27
36
45
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
35.82
B
35.27
C
34.77
D
34.94
8
16
24
32
40

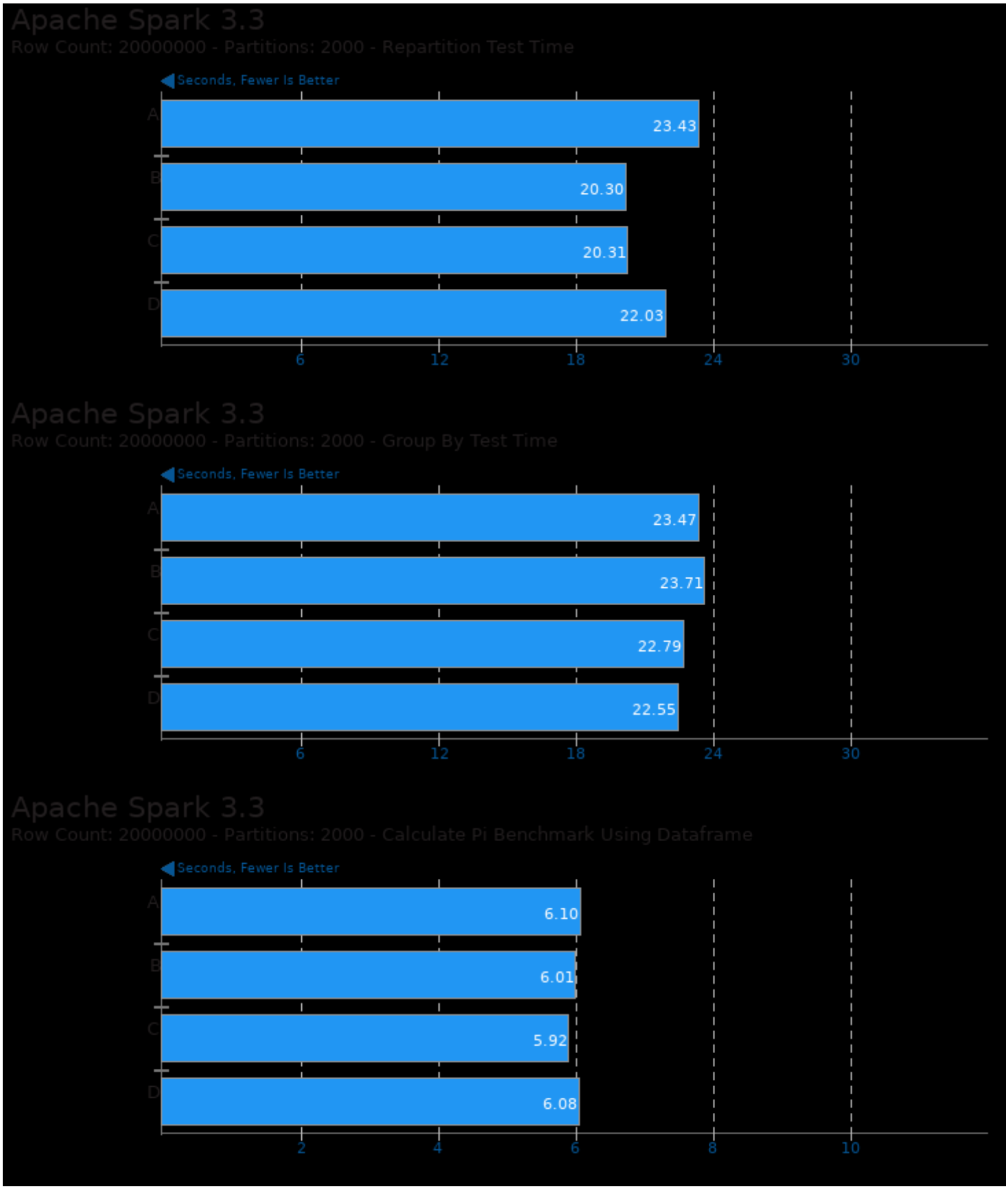
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
45.87
B
43.70
C
41.48
D
42.19
10
20
30
40
50
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.33
B
6.04
C
6.24
D
6.26
2
4
6
8
10
Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.96
B
80.79
C
79.91
D
80.39
20
40
60
80
100

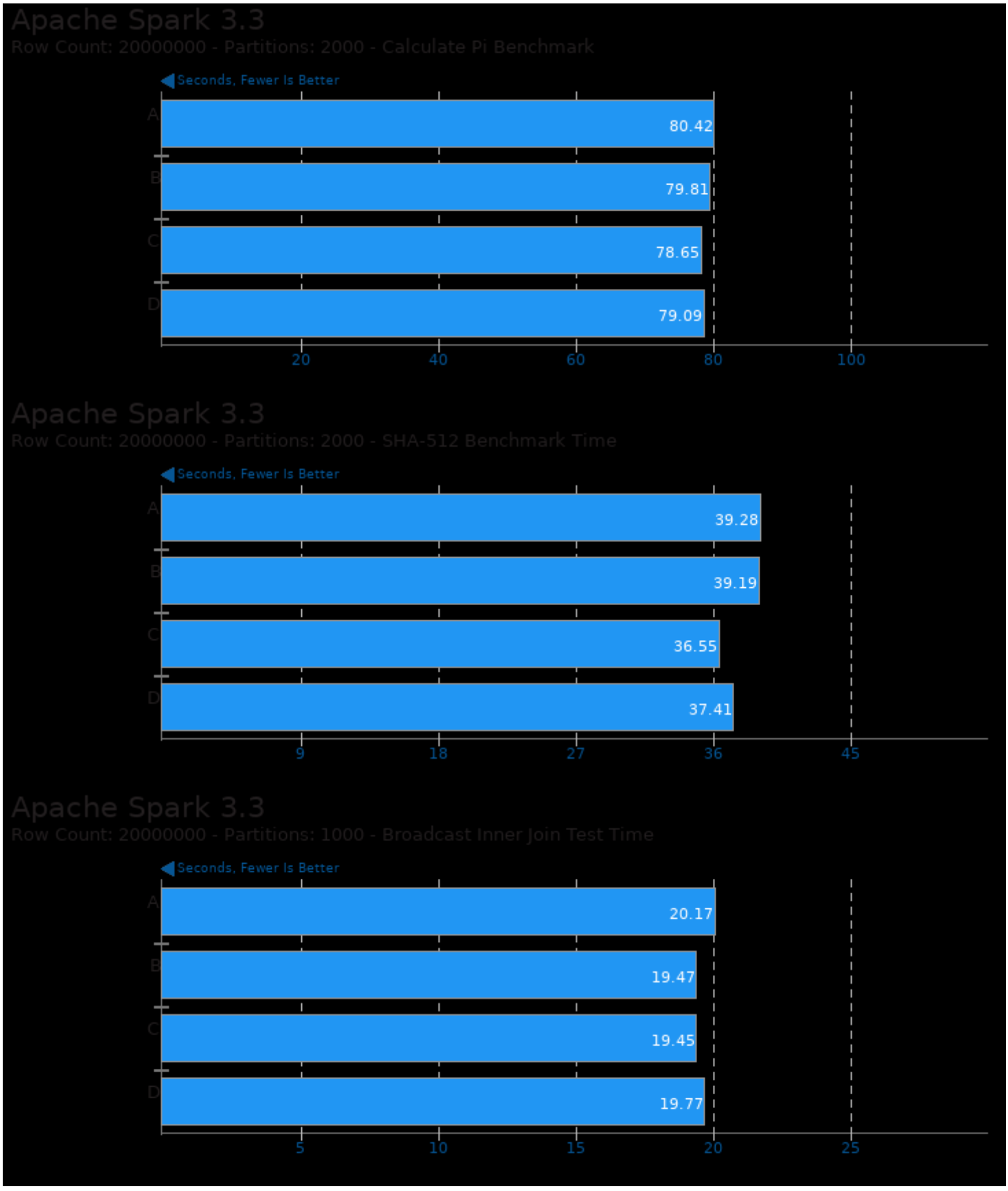
spark svt


Apache Spark 3.3
Row Count: 40000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
71.67
B
63.03
C
63.36
D
65.19
16
32
48
64
80
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
21.60
B
22.24
C
21.42
D
22.39
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Inner Join Test Time
Seconds, Fewer Is Better
A
23.02
B
22.88
C
22.15
D
23.34
6
12
18
24
30

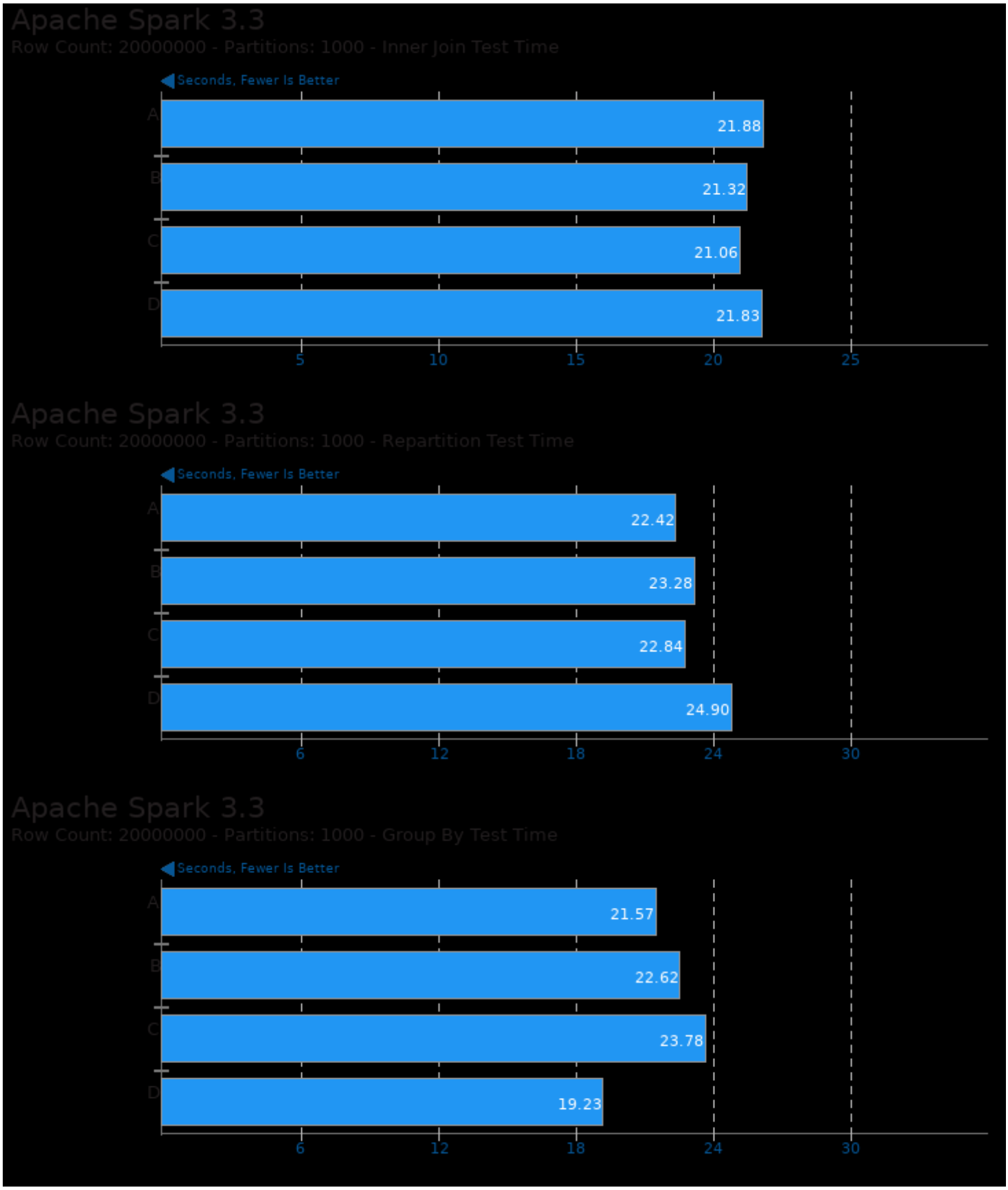
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
23.43
B
20.30
C
20.31
D
22.03
6
12
18
24
30
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
23.47
B
23.71
C
22.79
D
22.55
6
12
18
24
30
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.10
B
6.01
C
5.92
D
6.08
2
4
6
8
10

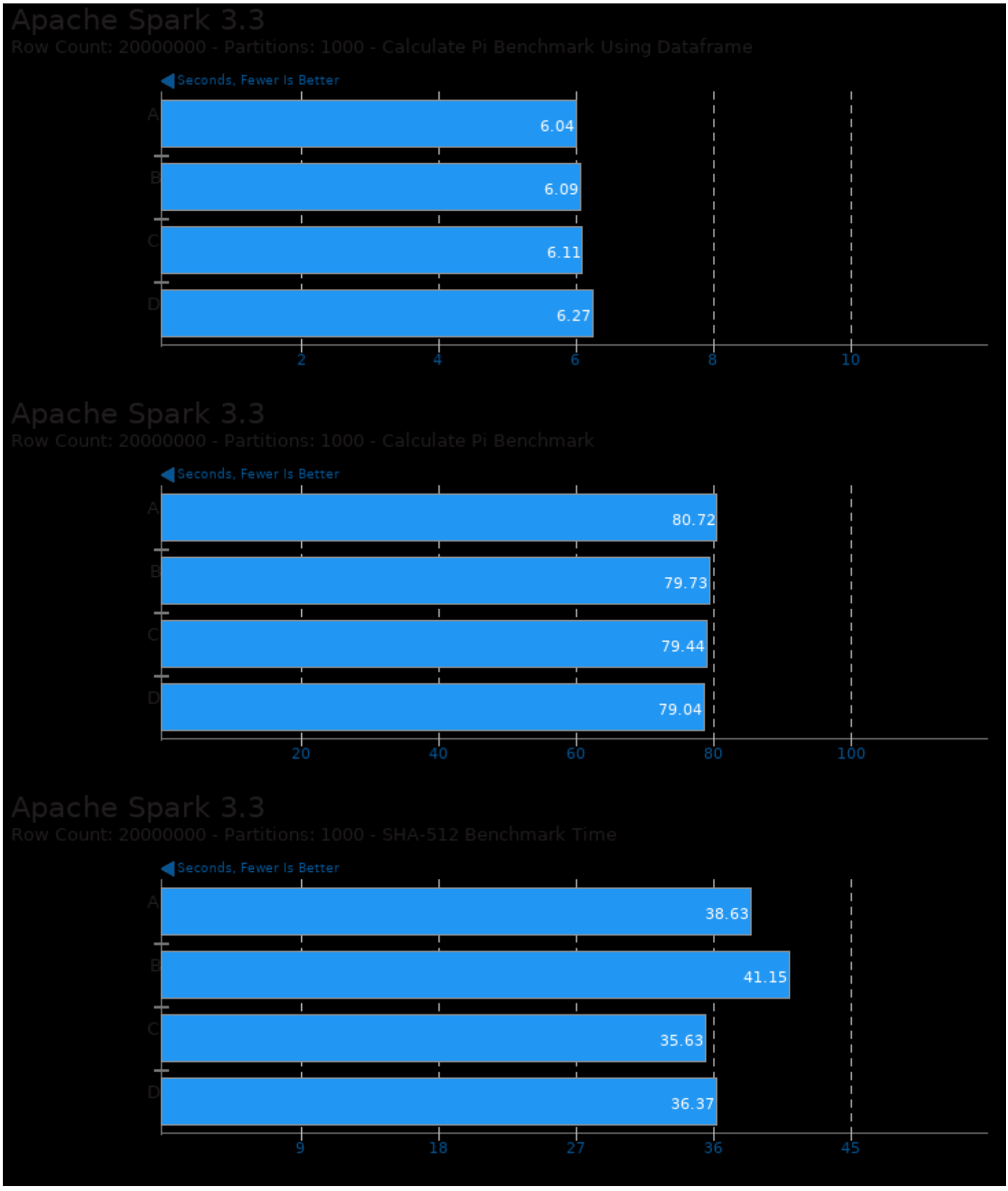
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.42
B
79.81
C
78.65
D
79.09
20
40
60
80
100
Apache Spark 3.3
Row Count: 20000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
39.28
B
39.19
C
36.55
D
37.41
9
18
27
36
45
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
20.17
B
19.47
C
19.45
D
19.77
5
10
15
20
25

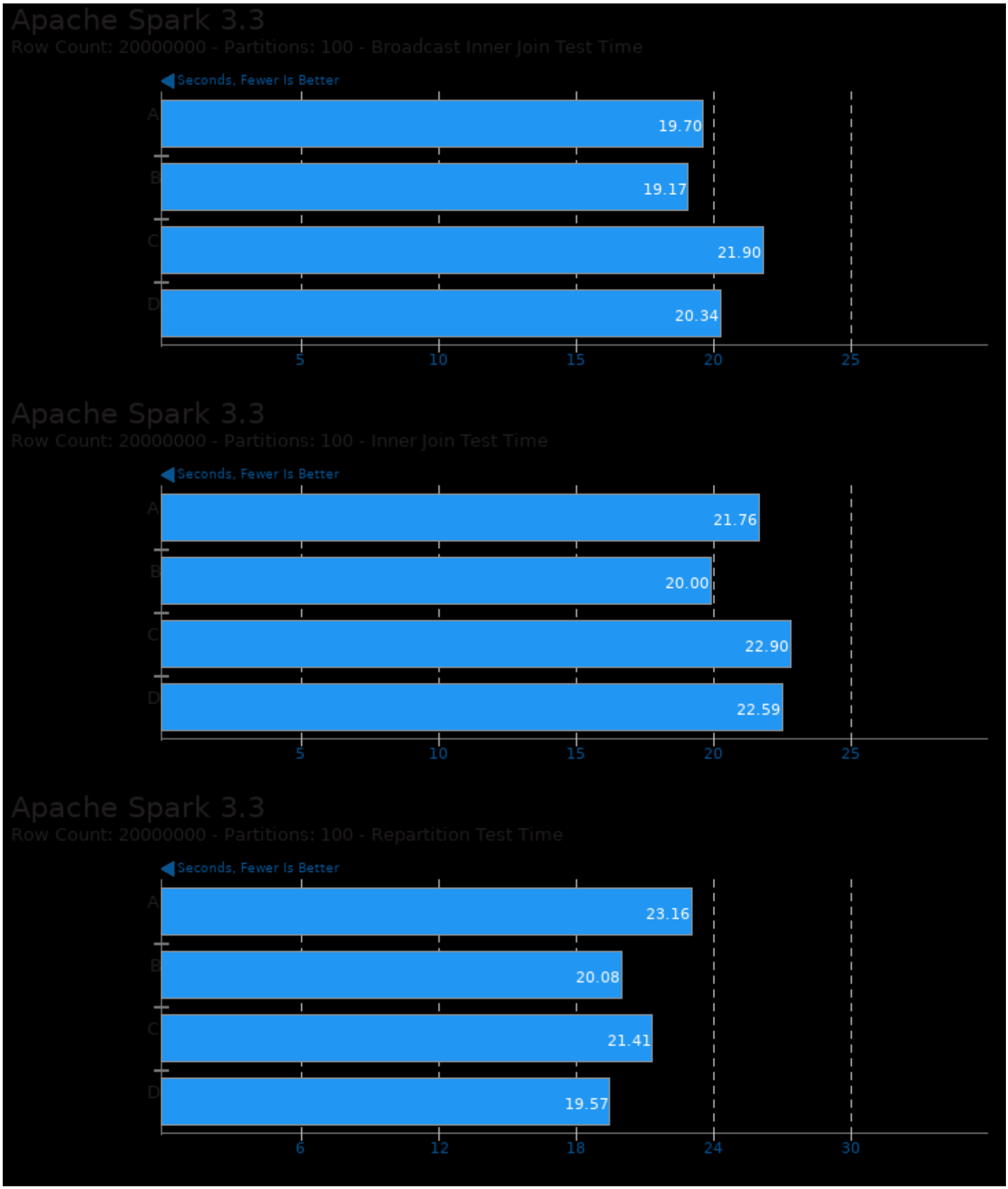
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
21.88
B
21.32
C
21.06
D
21.83
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
22.42
B
23.28
C
22.84
D
24.90
6
12
18
24
30
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
21.57
B
22.62
C
23.78
D
19.23
6
12
18
24
30

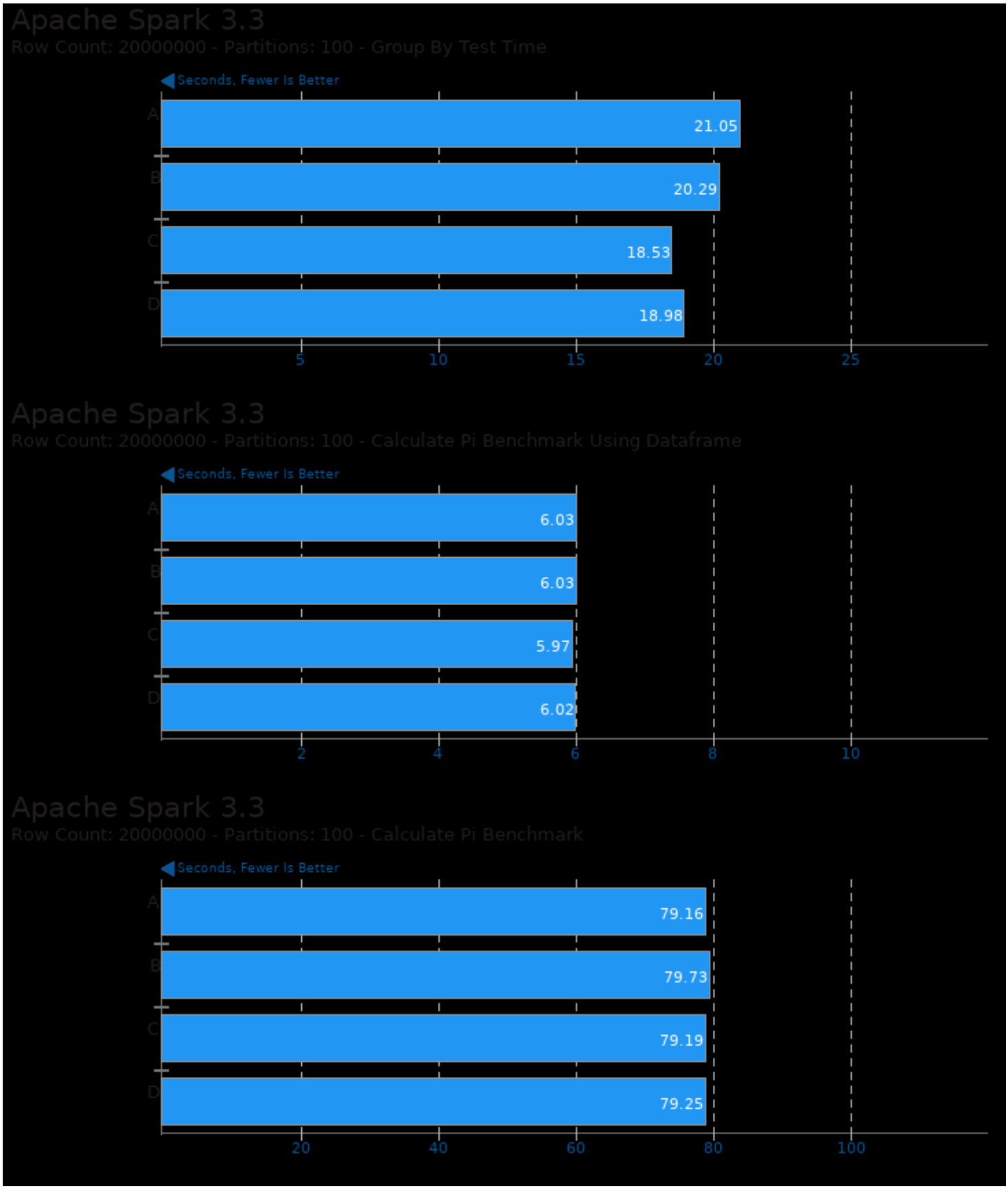
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.04
B
6.09
C
6.11
D
6.27
2
4
6
8
10
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.72
B
79.73
C
79.44
D
79.04
20
40
60
80
100
Apache Spark 3.3
Row Count: 20000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
38.63
B
41.15
C
35.63
D
36.37
9
18
27
36
45

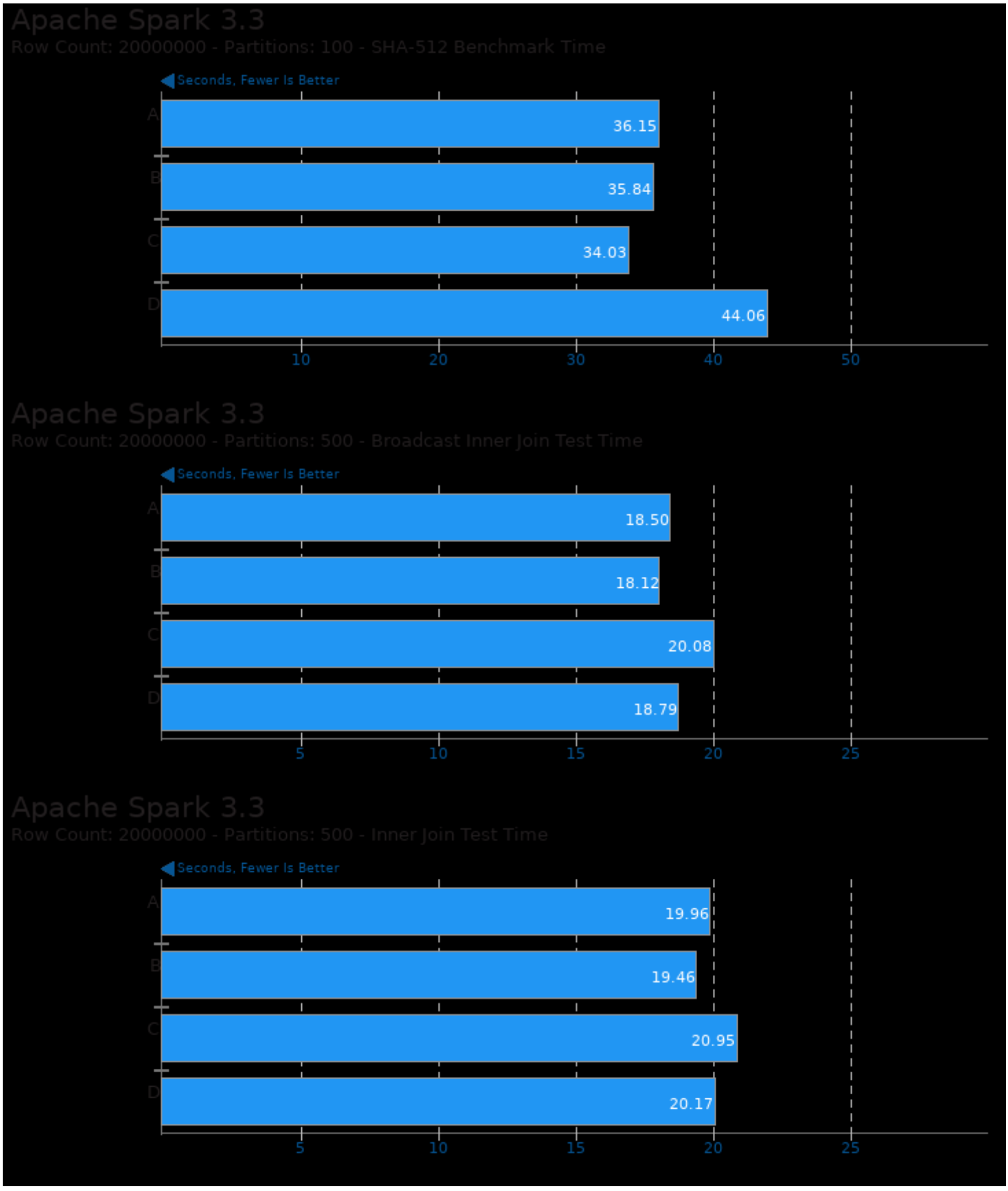
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
19.70
B
19.17
C
21.90
D
20.34
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
21.76
B
20.00
C
22.90
D
22.59
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
23.16
B
20.08
C
21.41
D
19.57
6
12
18
24
30

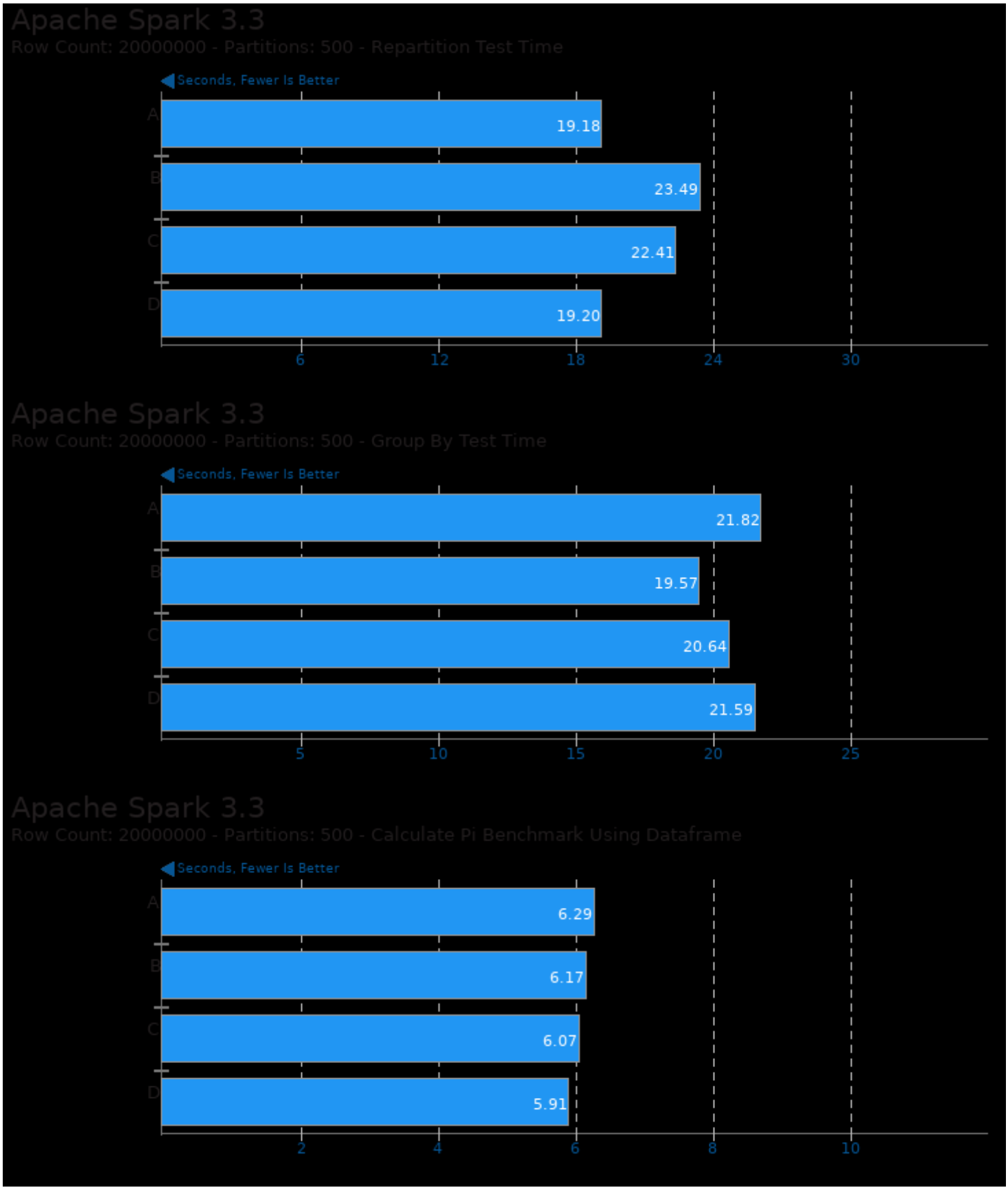
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
21.05
B
20.29
C
18.53
D
18.98
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.03
B
6.03
C
5.97
D
6.02
2
4
6
8
10
Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
79.16
B
79.73
C
79.19
D
79.25
20
40
60
80
100

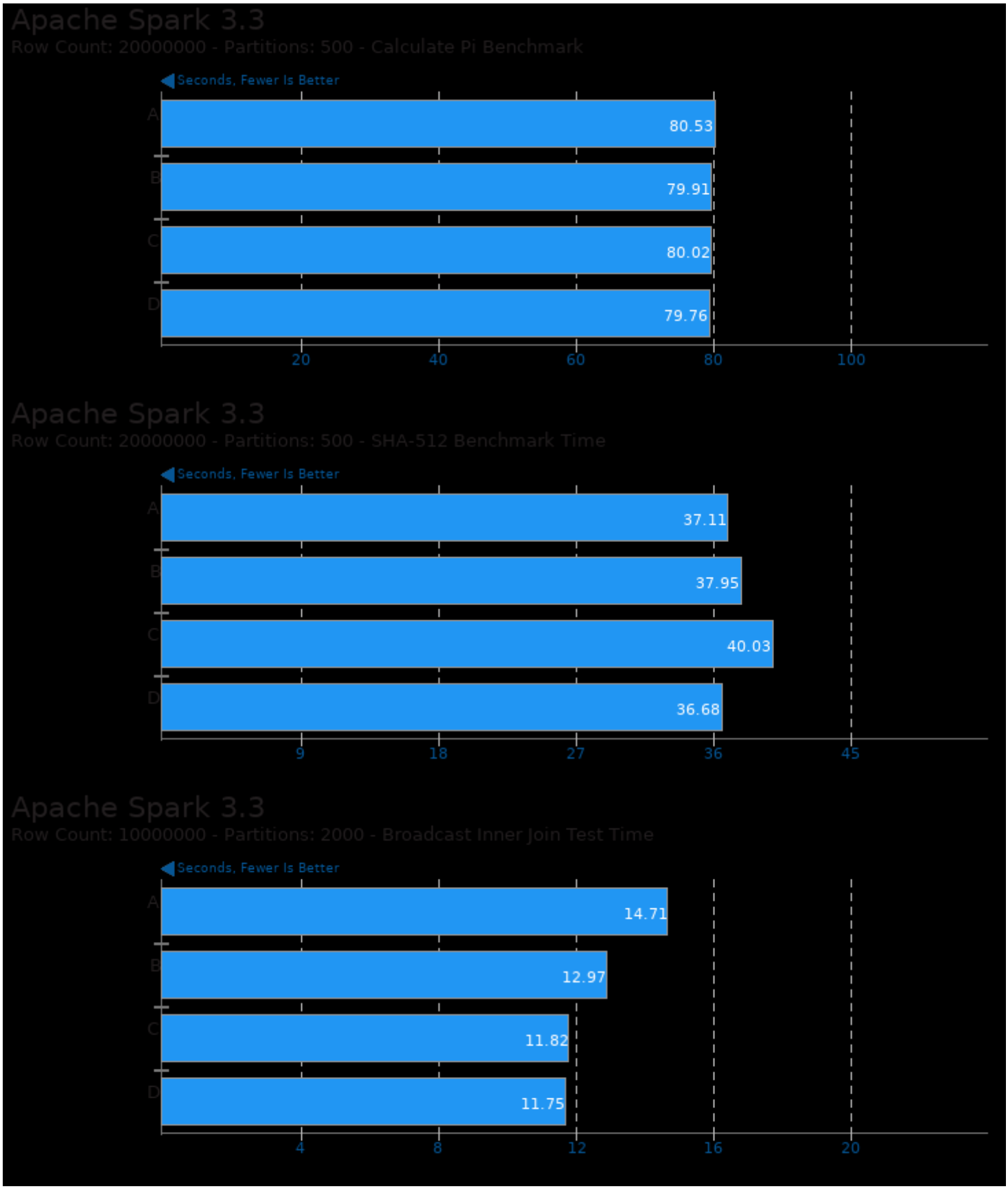
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 100 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
36.15
B
35.84
C
34.03
D
44.06
10
20
30
40
50
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
18.50
B
18.12
C
20.08
D
18.79
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
19.96
B
19.46
C
20.95
D
20.17
5
10
15
20
25

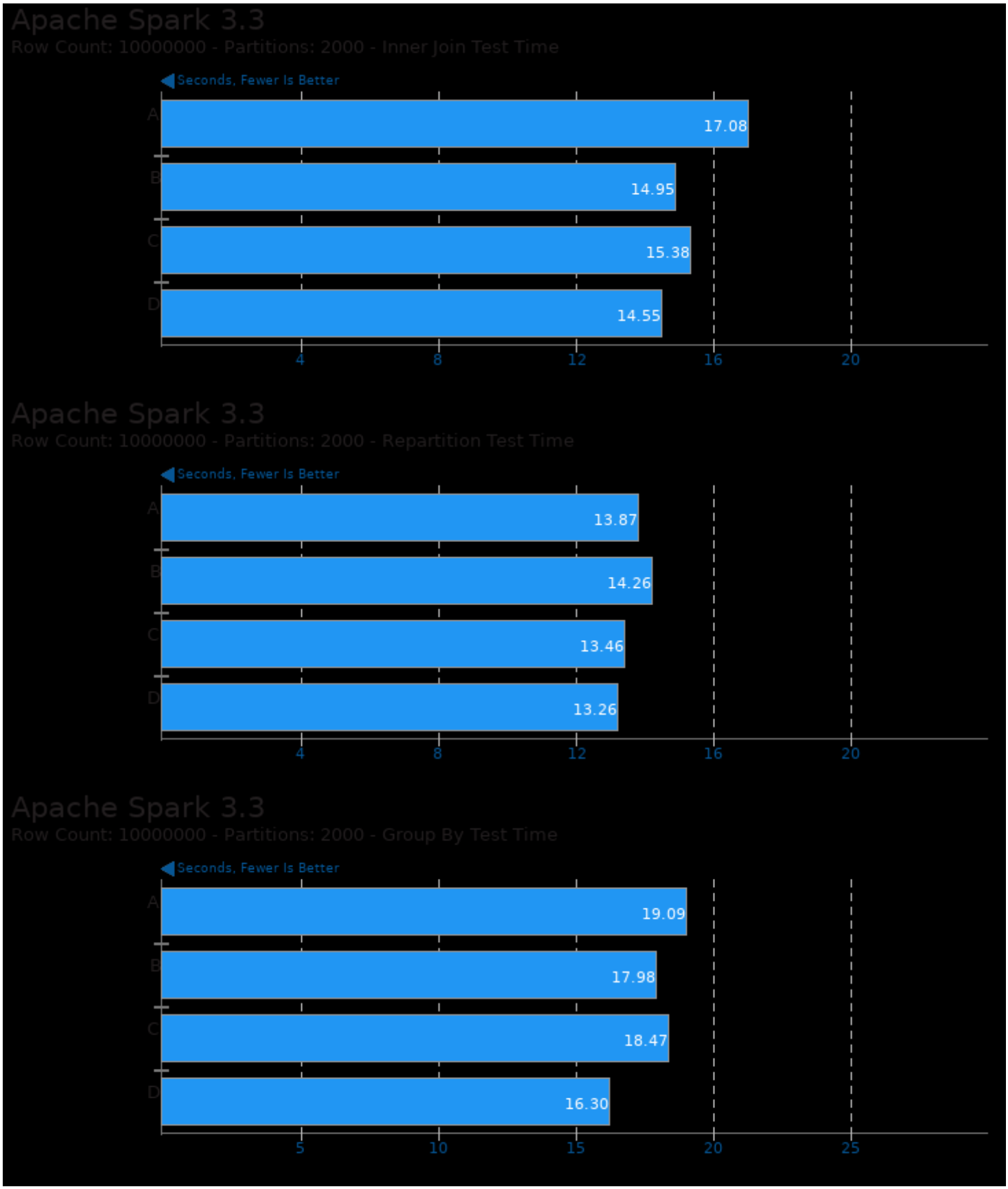
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
19.18
B
23.49
C
22.41
D
19.20
6
12
18
24
30
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
21.82
B
19.57
C
20.64
D
21.59
5
10
15
20
25
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.29
B
6.17
C
6.07
D
5.91
2
4
6
8
10

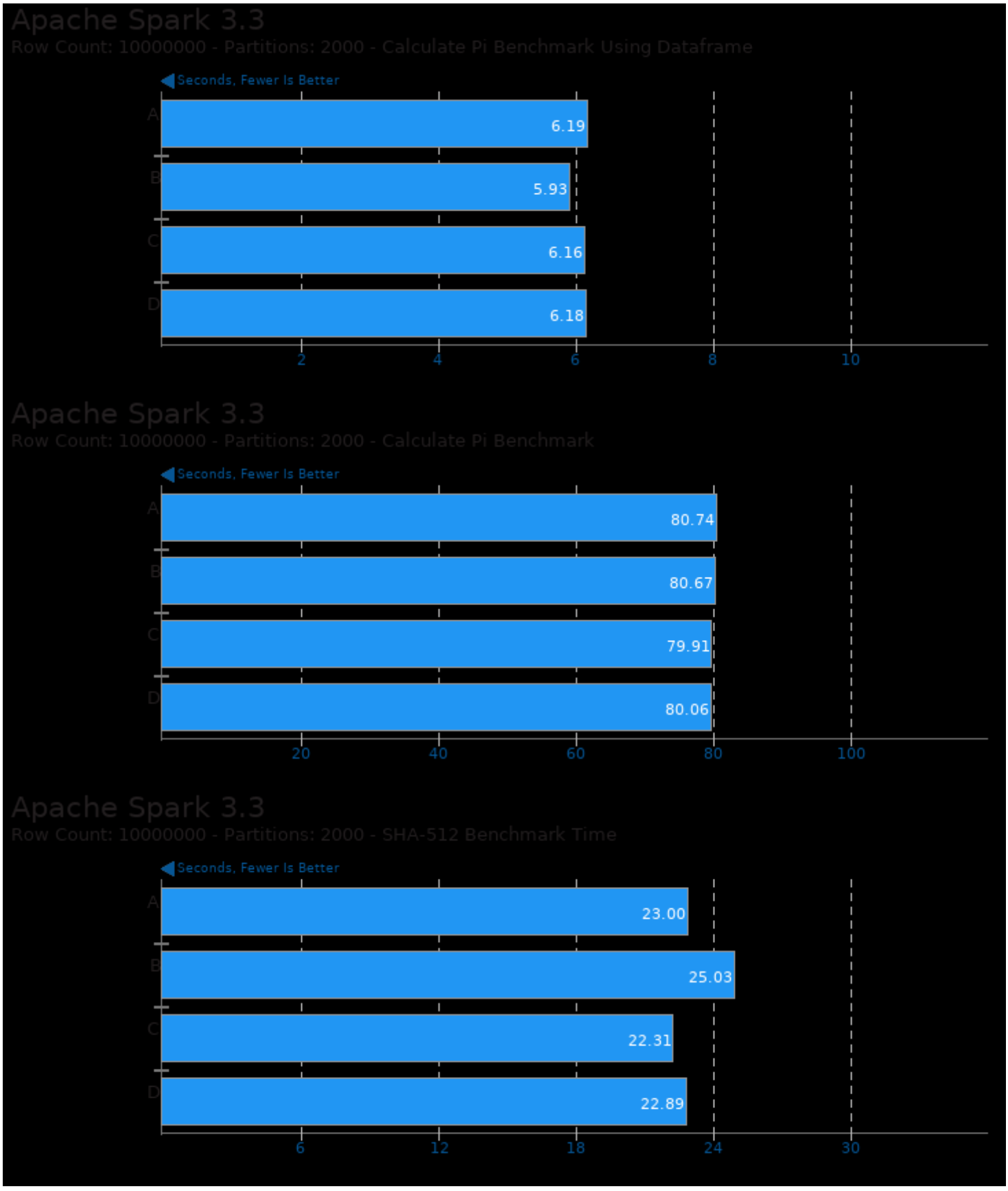
spark svt


Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.53
B
79.91
C
80.02
D
79.76
20
40
60
80
100
Apache Spark 3.3
Row Count: 20000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
37.11
B
37.95
C
40.03
D
36.68
9
18
27
36
45
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
14.71
B
12.97
C
11.82
D
11.75
4
8
12
16
20

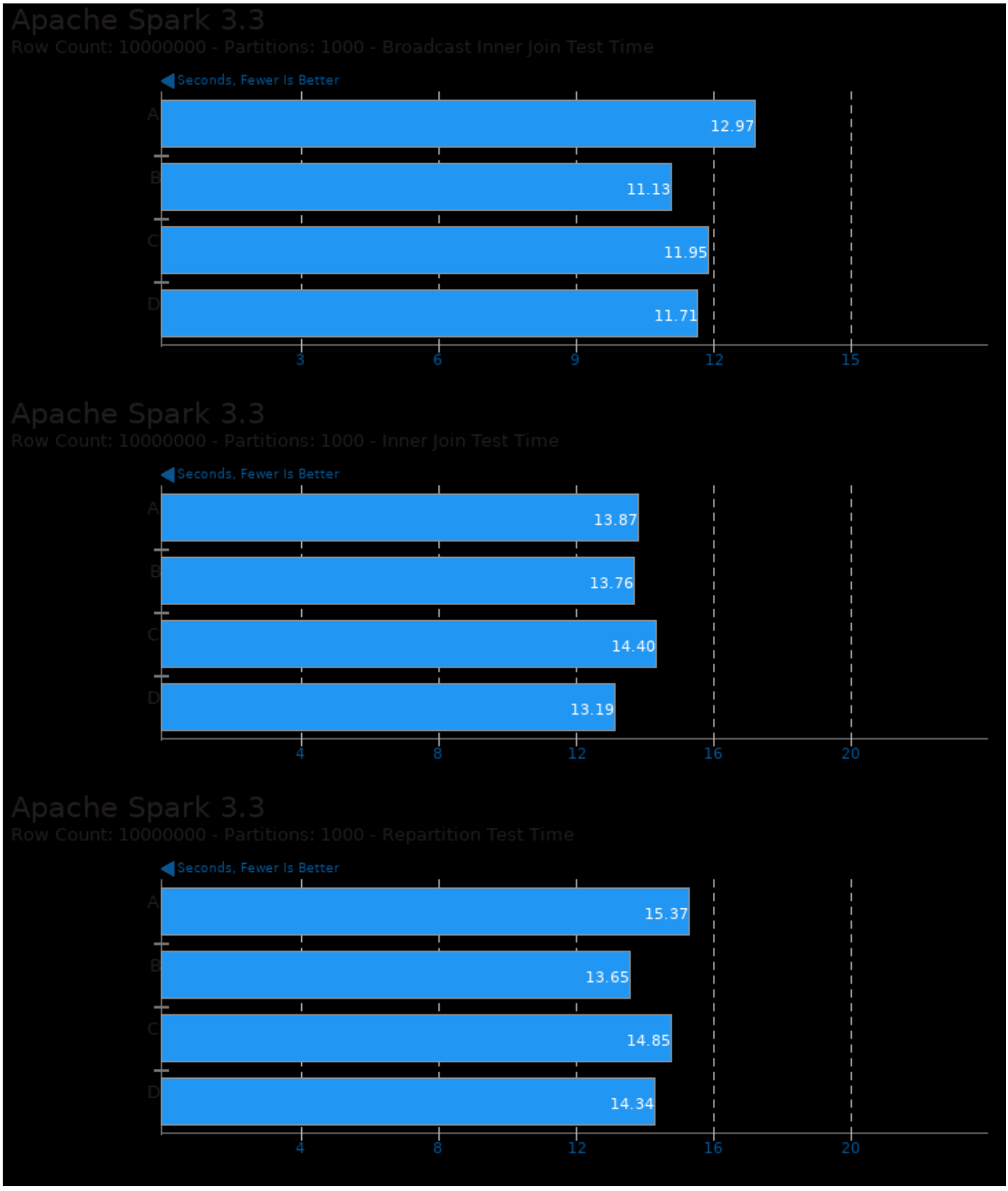
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Inner Join Test Time
Seconds, Fewer Is Better
A
17.08
B
14.95
C
15.38
D
14.55
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
13.87
B
14.26
C
13.46
D
13.26
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
19.09
B
17.98
C
18.47
D
16.30
5
10
15
20
25

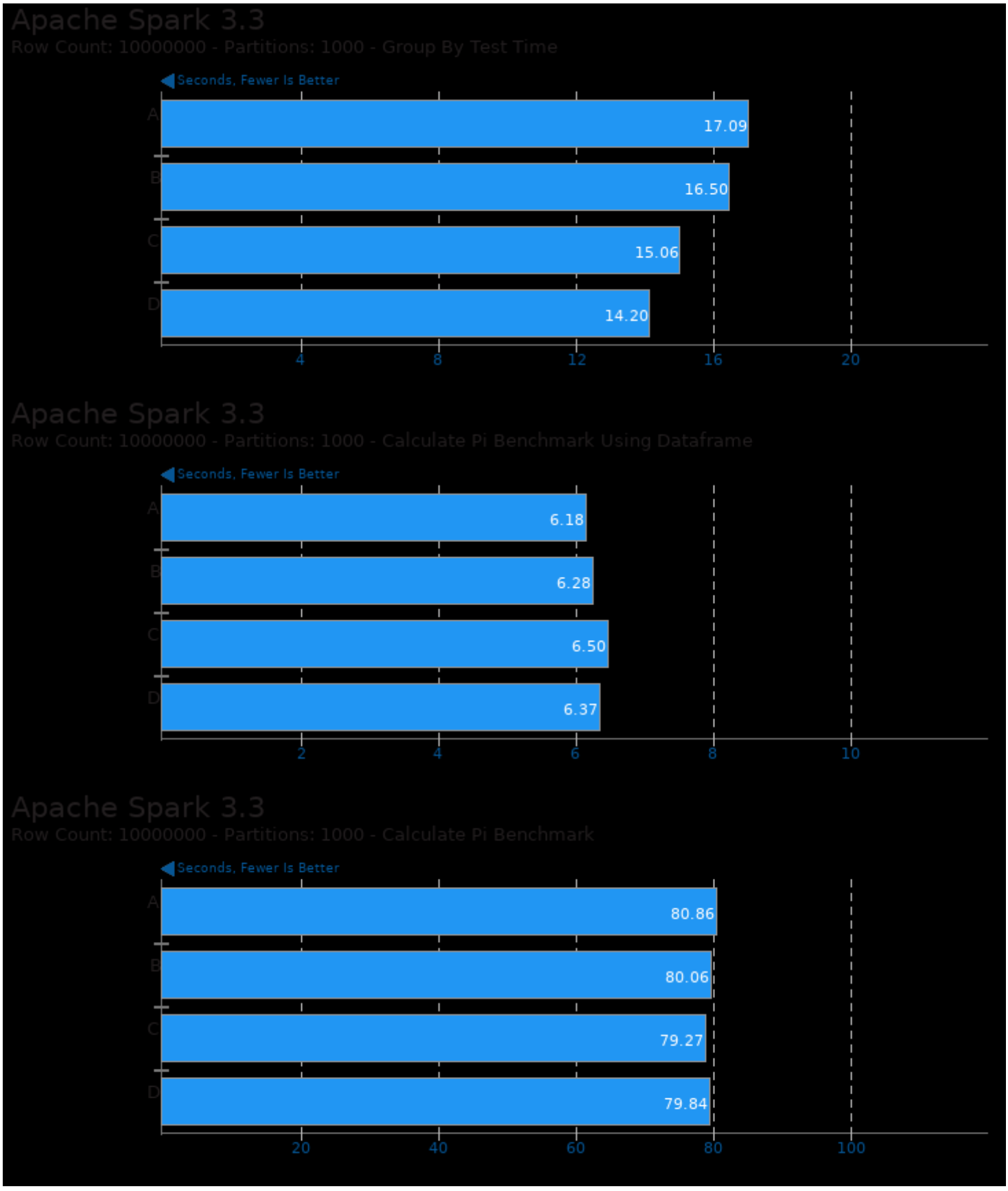
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.19
B
5.93
C
6.16
D
6.18
2
4
6
8
10
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.74
B
80.67
C
79.91
D
80.06
20
40
60
80
100
Apache Spark 3.3
Row Count: 10000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
23.00
B
25.03
C
22.31
D
22.89
6
12
18
24
30

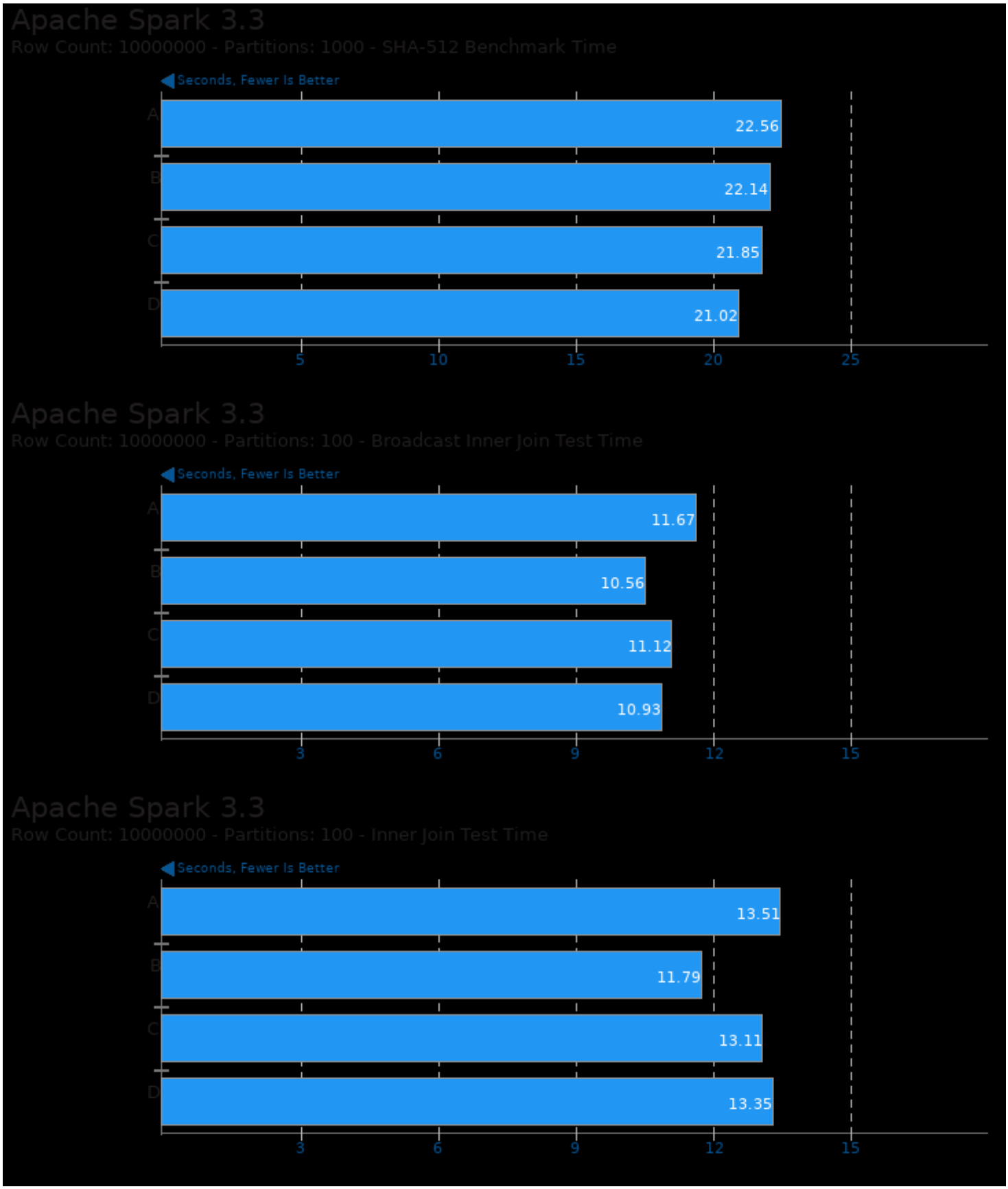
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
B
C
D
12.97
11.13
11.95
11.71
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
B
C
D
13.87
13.76
14.40
13.19
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
B
C
D
15.37
13.65
14.85
14.34
4
8
12
16
20

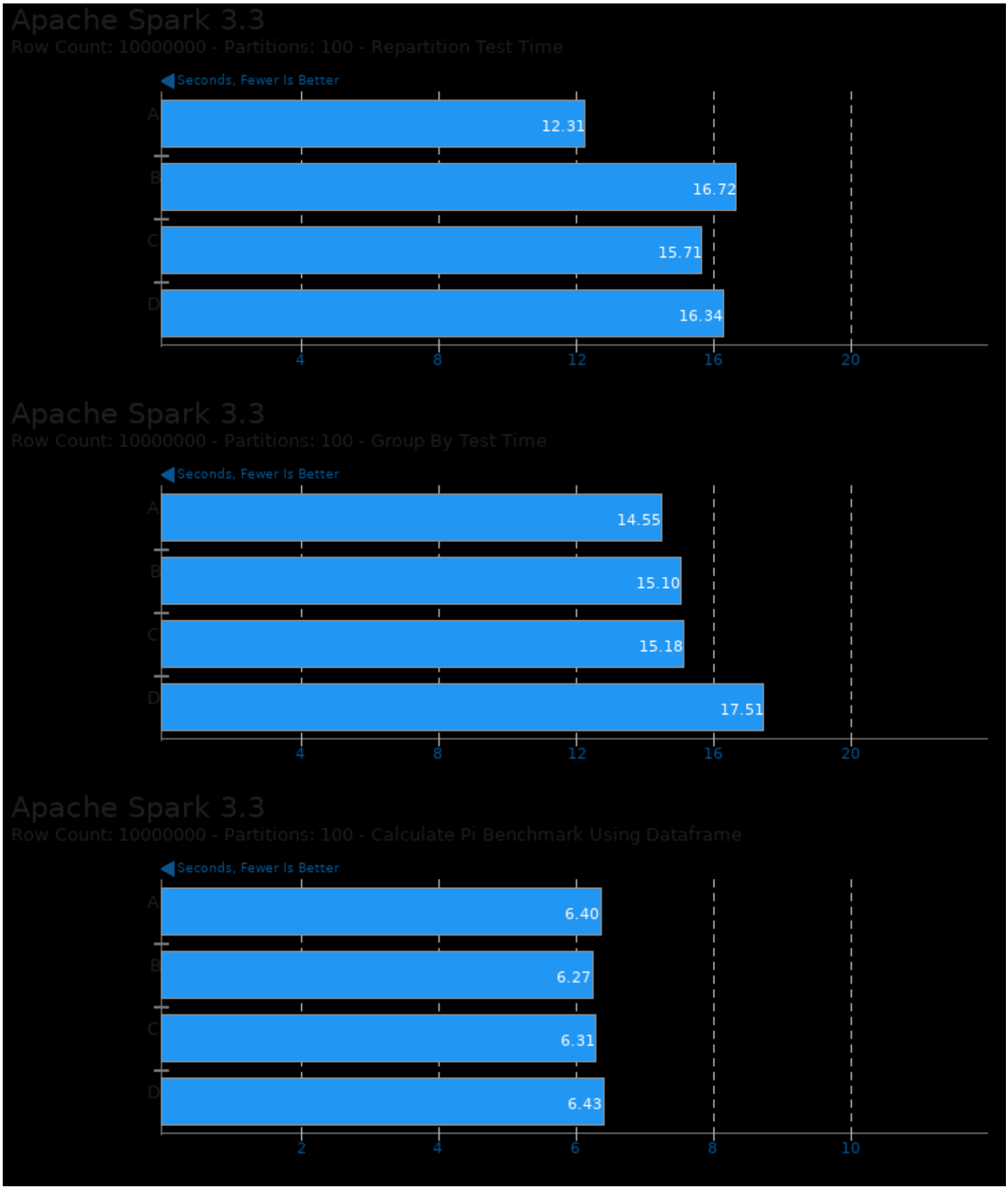
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
17.09
B
16.50
C
15.06
D
14.20
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.18
B
6.28
C
6.50
D
6.37
2
4
6
8
10
Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
80.86
B
80.06
C
79.27
D
79.84
20
40
60
80
100

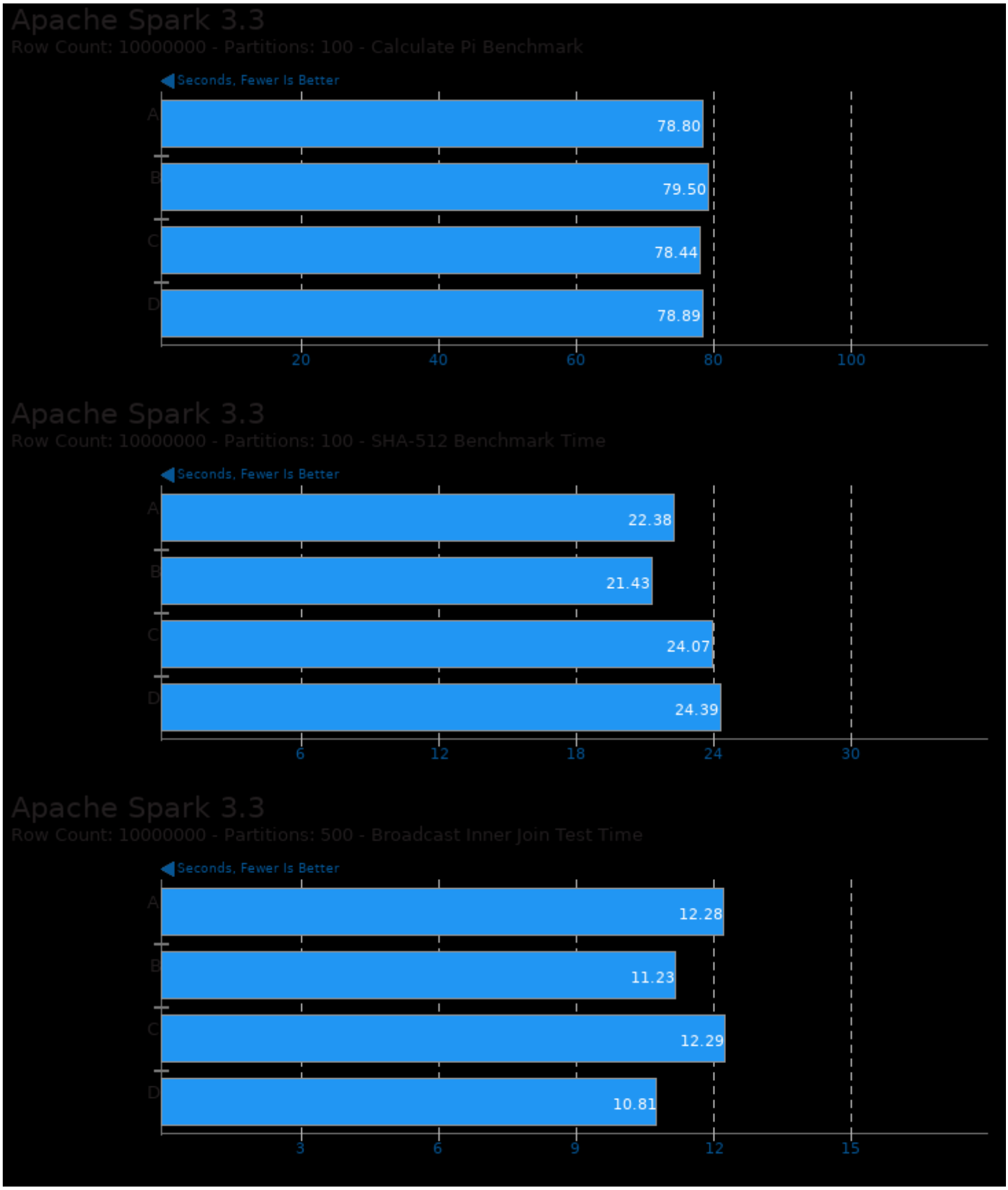
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
22.56
B
22.14
C
21.85
D
21.02
5
10
15
20
25
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
11.67
B
10.56
C
11.12
D
10.93
3
6
9
12
15
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
13.51
B
11.79
C
13.11
D
13.35
3
6
9
12
15

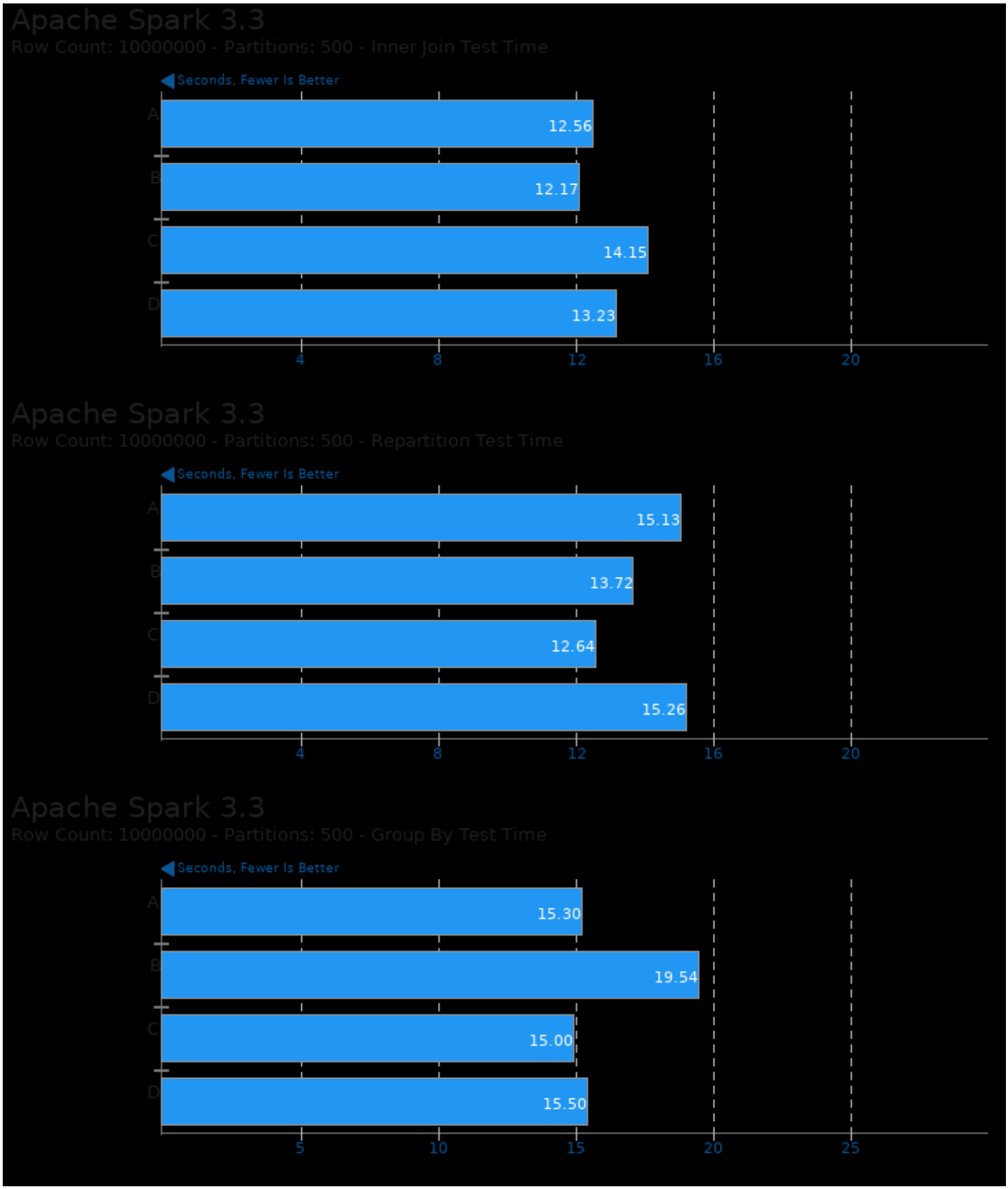
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
12.31
B
16.72
C
15.71
D
16.34
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
14.55
B
15.10
C
15.18
D
17.51
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.40
B
6.27
C
6.31
D
6.43
2
4
6
8
10

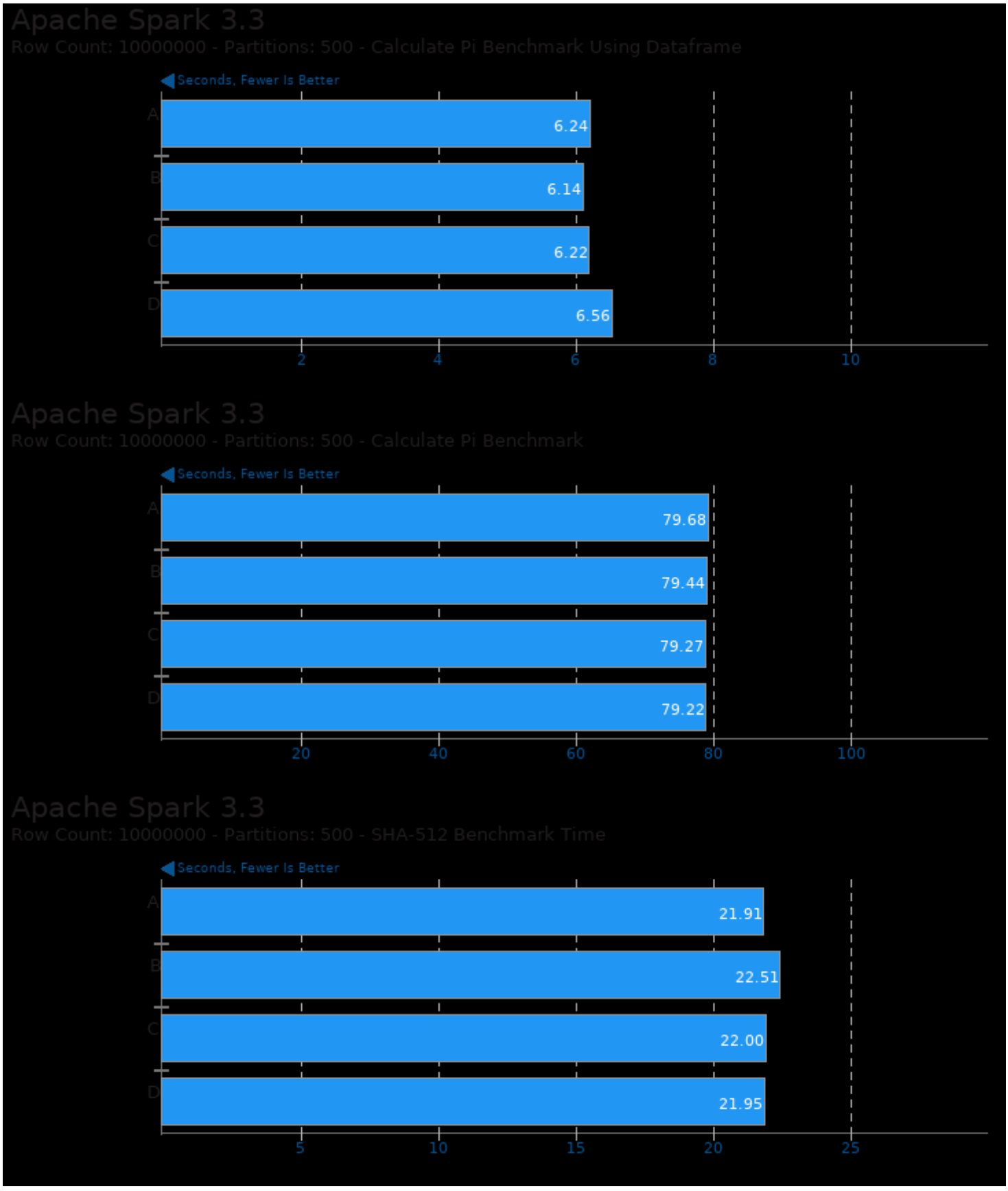
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
78.80
B
79.50
C
78.44
D
78.89
20
40
60
80
100
Apache Spark 3.3
Row Count: 10000000 - Partitions: 100 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
22.38
B
21.43
C
24.07
D
24.39
6
12
18
24
30
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
12.28
B
11.23
C
12.29
D
10.81
3
6
9
12
15

spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
12.56
B
12.17
C
14.15
D
13.23
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
15.13
B
13.72
C
12.64
D
15.26
4
8
12
16
20
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
15.30
B
19.54
C
15.00
D
15.50
5
10
15
20
25

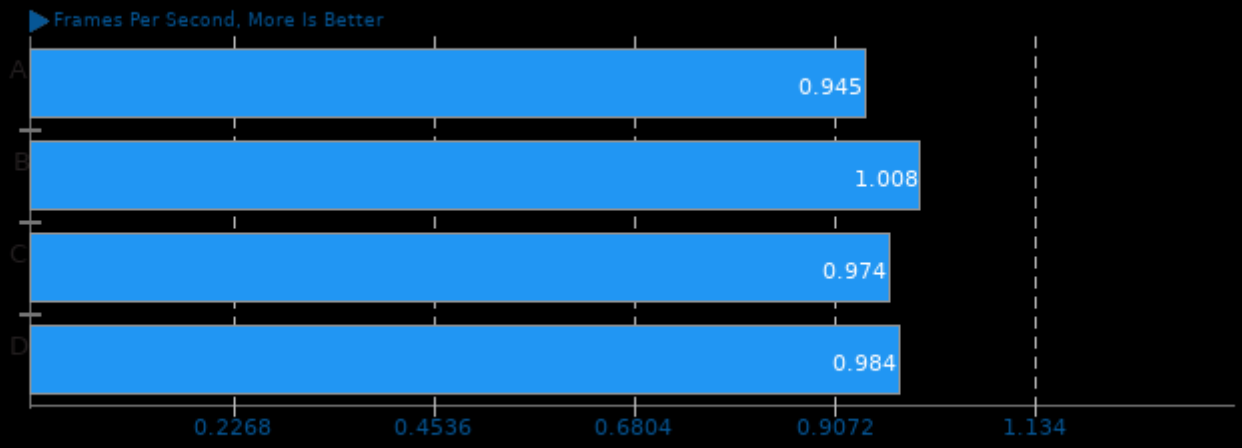
spark svt


Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.24
B
6.14
C
6.22
D
6.56
2
4
6
8
10
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
79.68
B
79.44
C
79.27
D
79.22
20
40
60
80
100
Apache Spark 3.3
Row Count: 10000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
21.91
B
22.51
C
22.00
D
21.95
5
10
15
20
25

spark svt

SVT-AV1 1.2

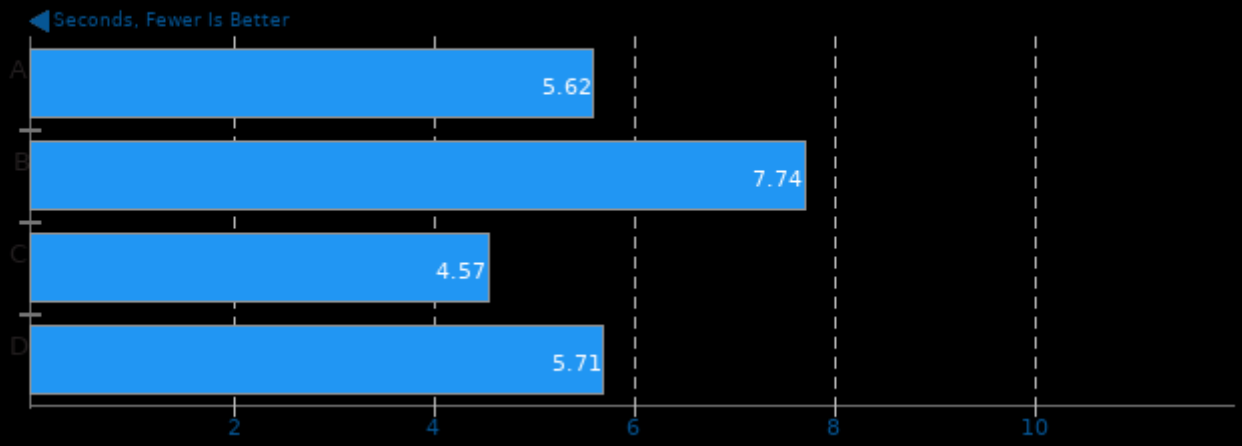
Encoder Mode: Preset 4 - Input: Bosphorus 4K



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

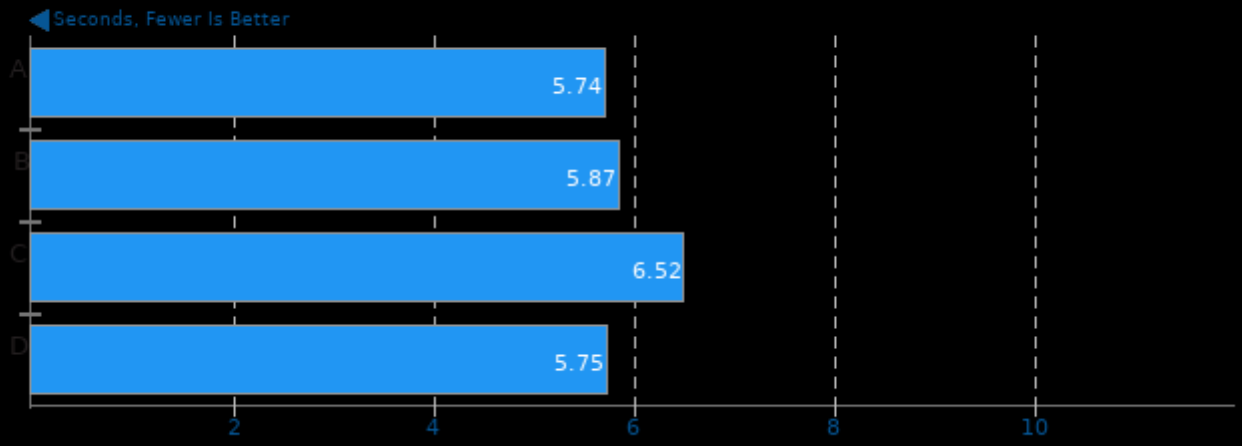
Apache Spark 3.3

Row Count: 1000000 - Partitions: 2000 - Broadcast Inner Join Test Time

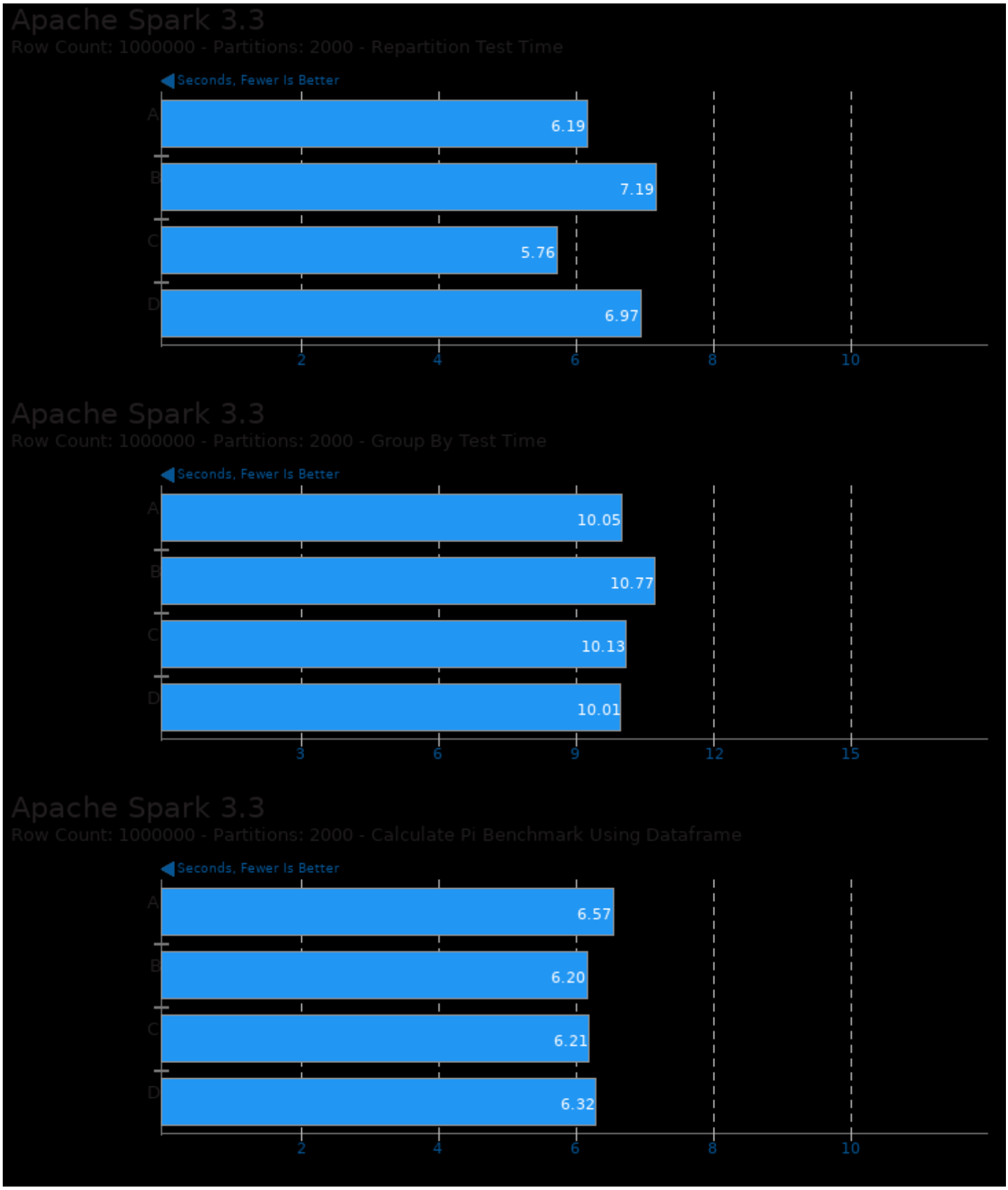


Apache Spark 3.3

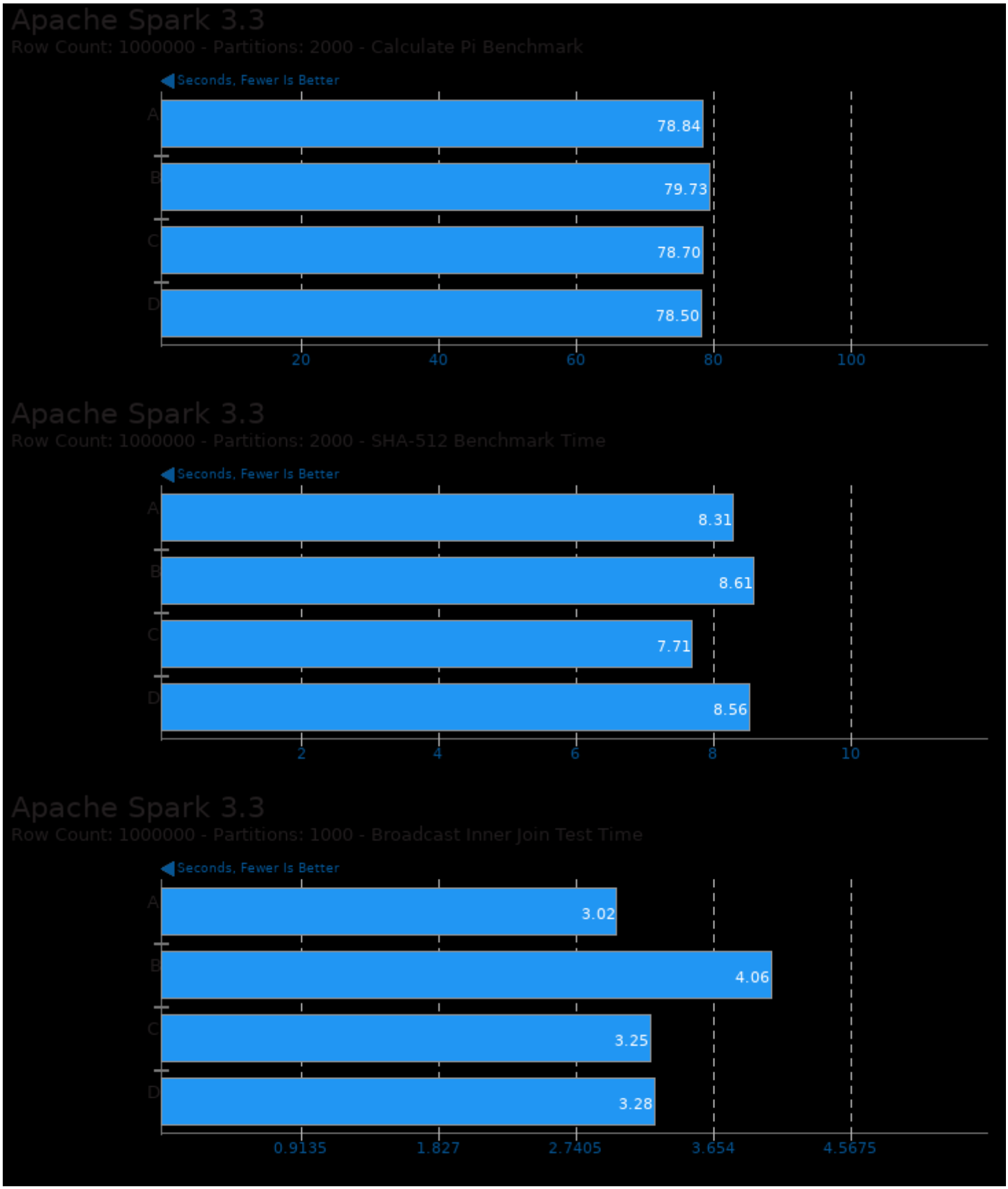
Row Count: 1000000 - Partitions: 2000 - Inner Join Test Time



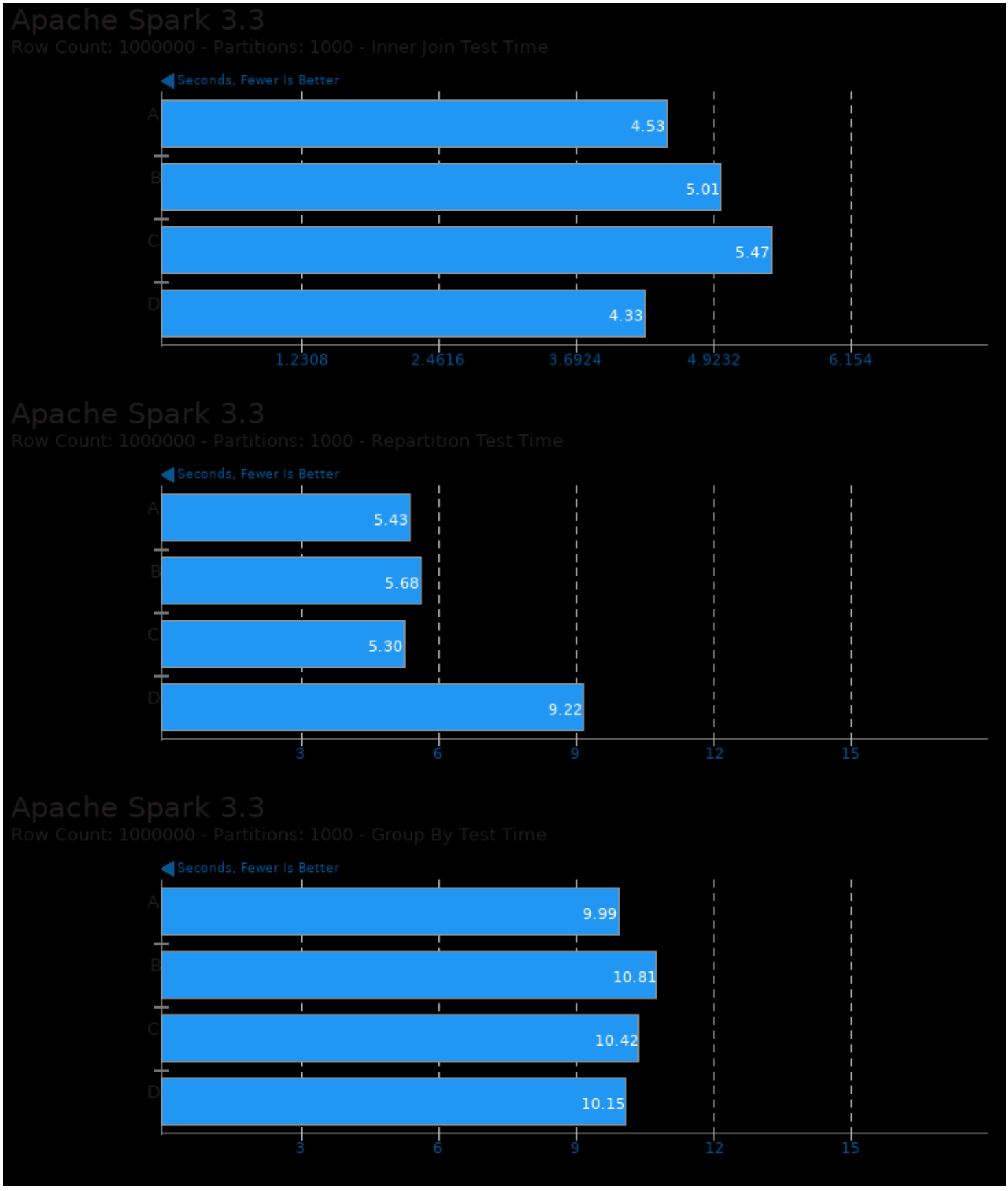
spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Repartition Test Time
Seconds, Fewer Is Better
A
6.19
B
7.19
C
5.76
D
6.97
2
4
6
8
10
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Group By Test Time
Seconds, Fewer Is Better
A
10.05
B
10.77
C
10.13
D
10.01
3
6
9
12
15
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.57
B
6.20
C
6.21
D
6.32
2
4
6
8
10

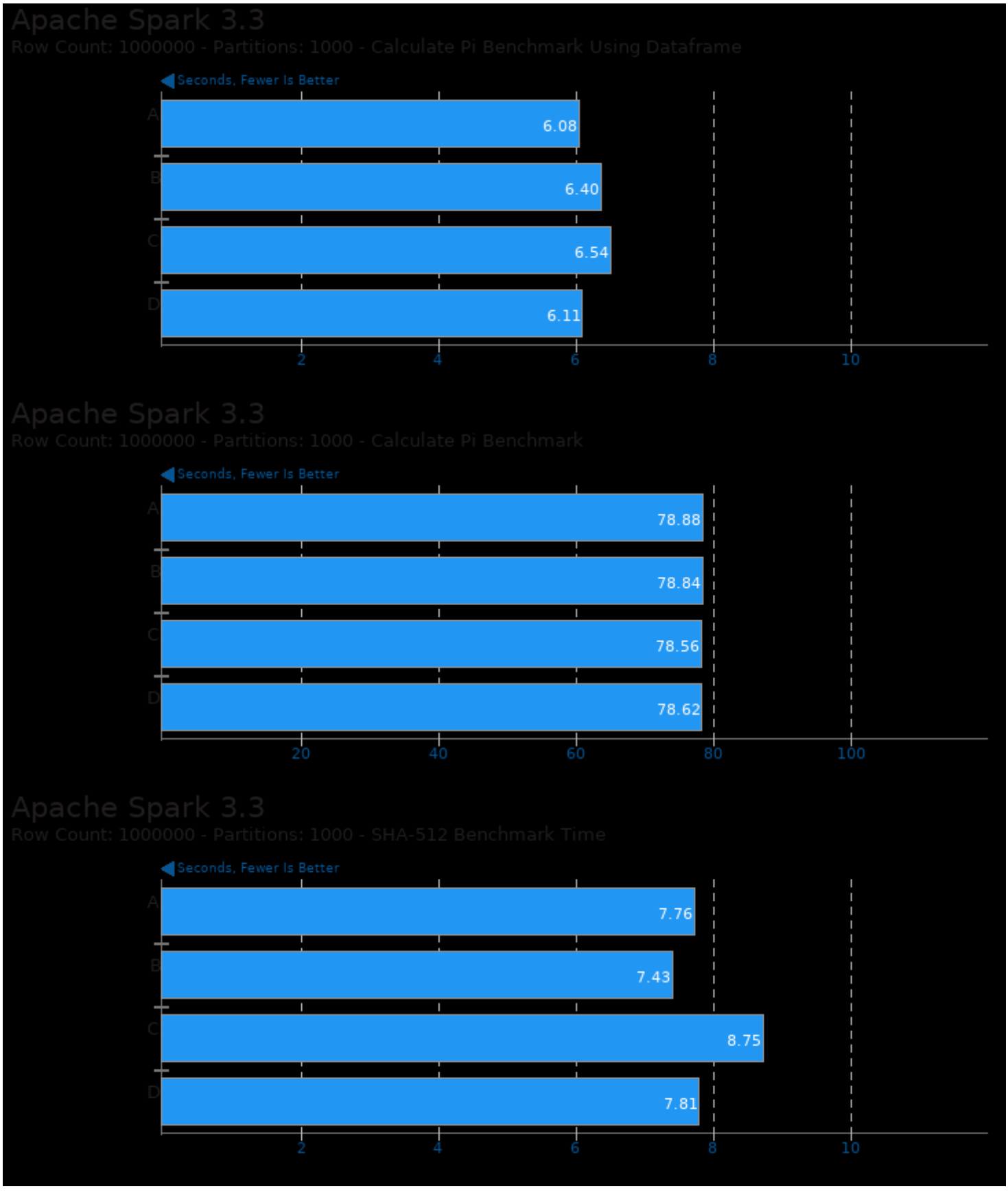
spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
78.84
B
79.73
C
78.70
D
78.50
20
40
60
80
100
Apache Spark 3.3
Row Count: 1000000 - Partitions: 2000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
8.31
B
8.61
C
7.71
D
8.56
2
4
6
8
10
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
3.02
B
4.06
C
3.25
D
3.28
0.9135
1.827
2.7405
3.654
4.5675

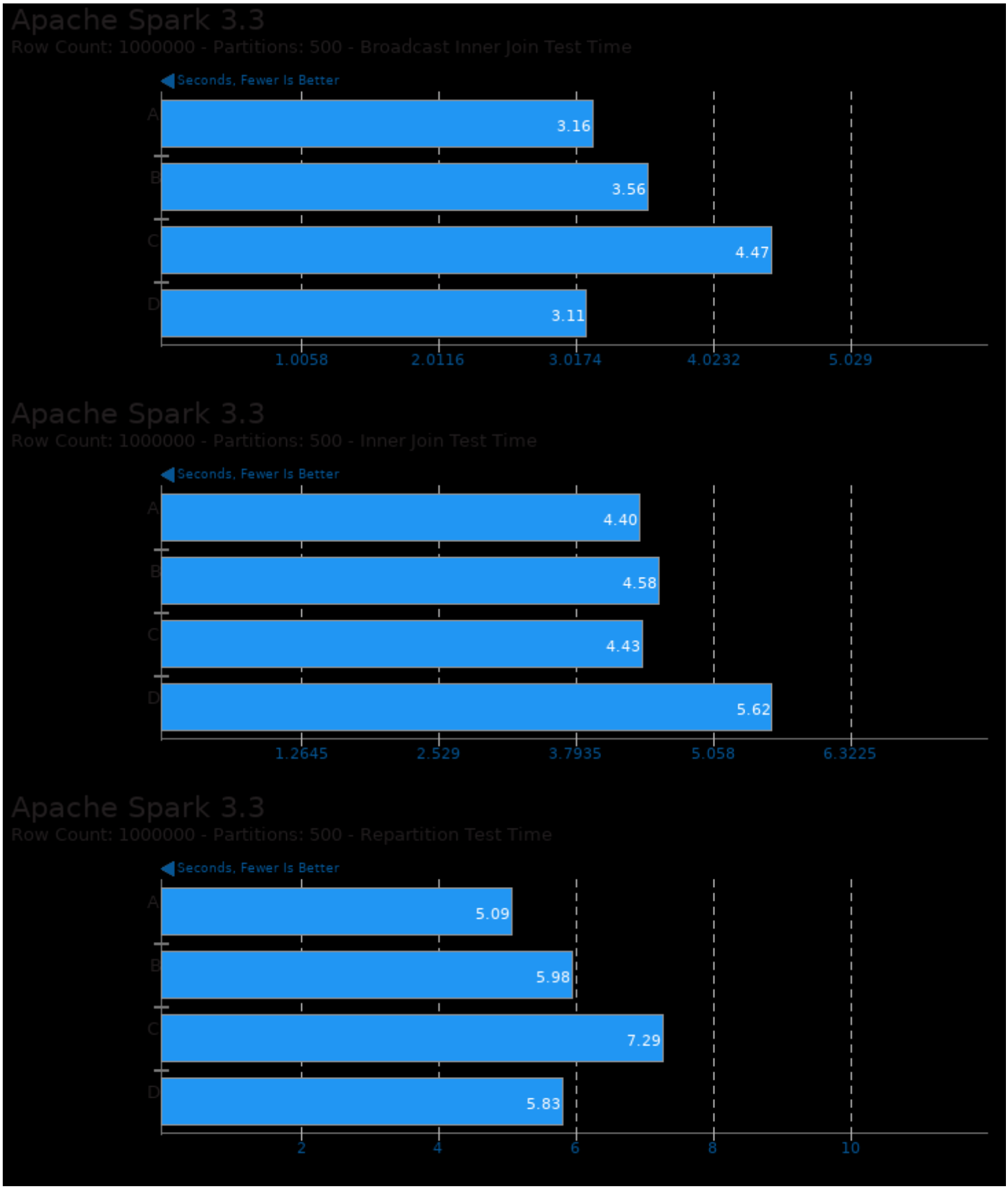
spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Inner Join Test Time
Seconds, Fewer Is Better
A
4.53
B
5.01
C
5.47
D
4.33
1.2308
2.4616
3.6924
4.9232
6.154
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Repartition Test Time
Seconds, Fewer Is Better
A
5.43
B
5.68
C
5.30
D
9.22
3
6
9
12
15
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Group By Test Time
Seconds, Fewer Is Better
A
9.99
B
10.81
C
10.42
D
10.15
3
6
9
12
15

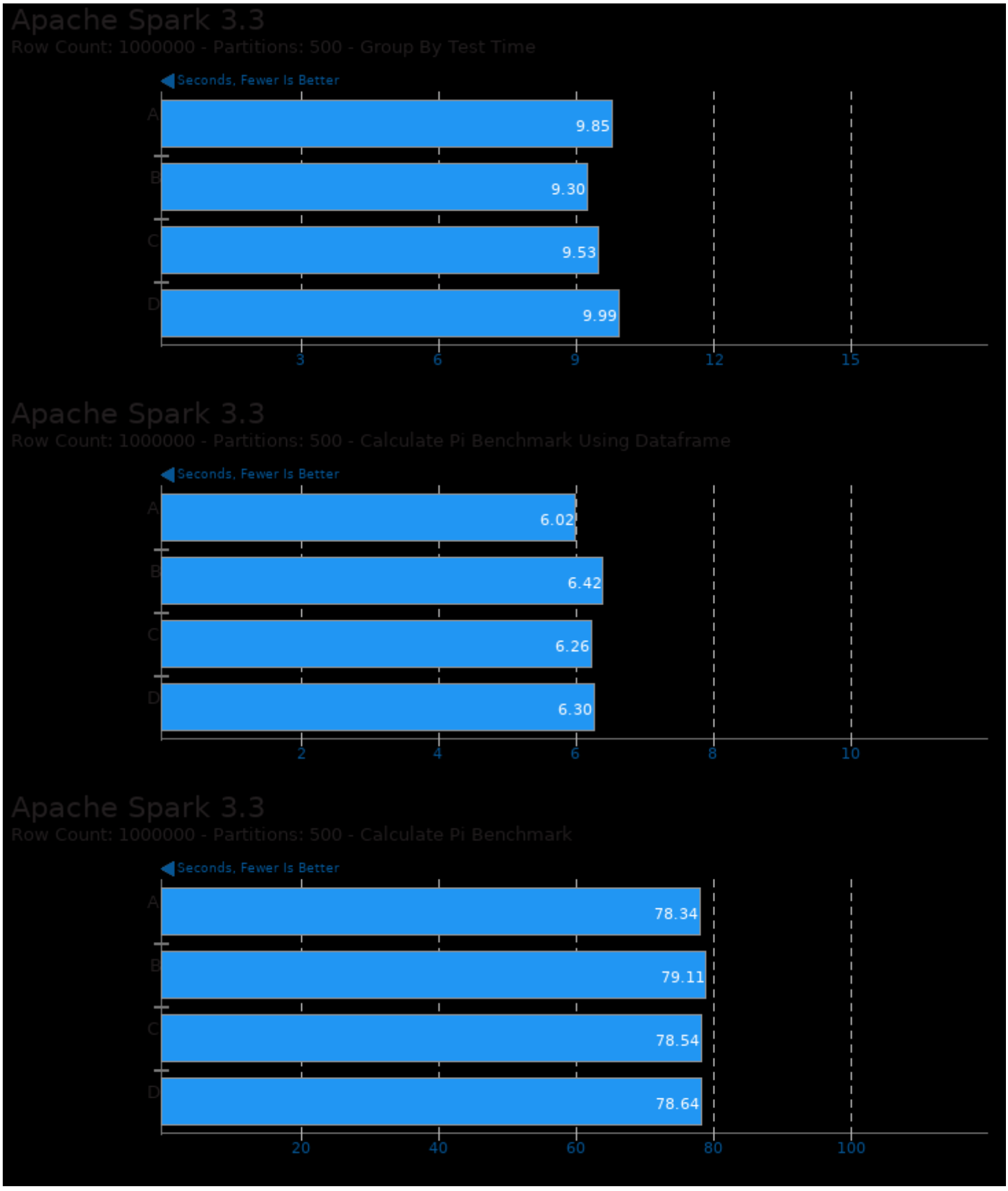
spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.08
B
6.40
C
6.54
D
6.11
2
4
6
8
10
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
78.88
B
78.84
C
78.56
D
78.62
20
40
60
80
100
Apache Spark 3.3
Row Count: 1000000 - Partitions: 1000 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
7.76
B
7.43
C
8.75
D
7.81
2
4
6
8
10

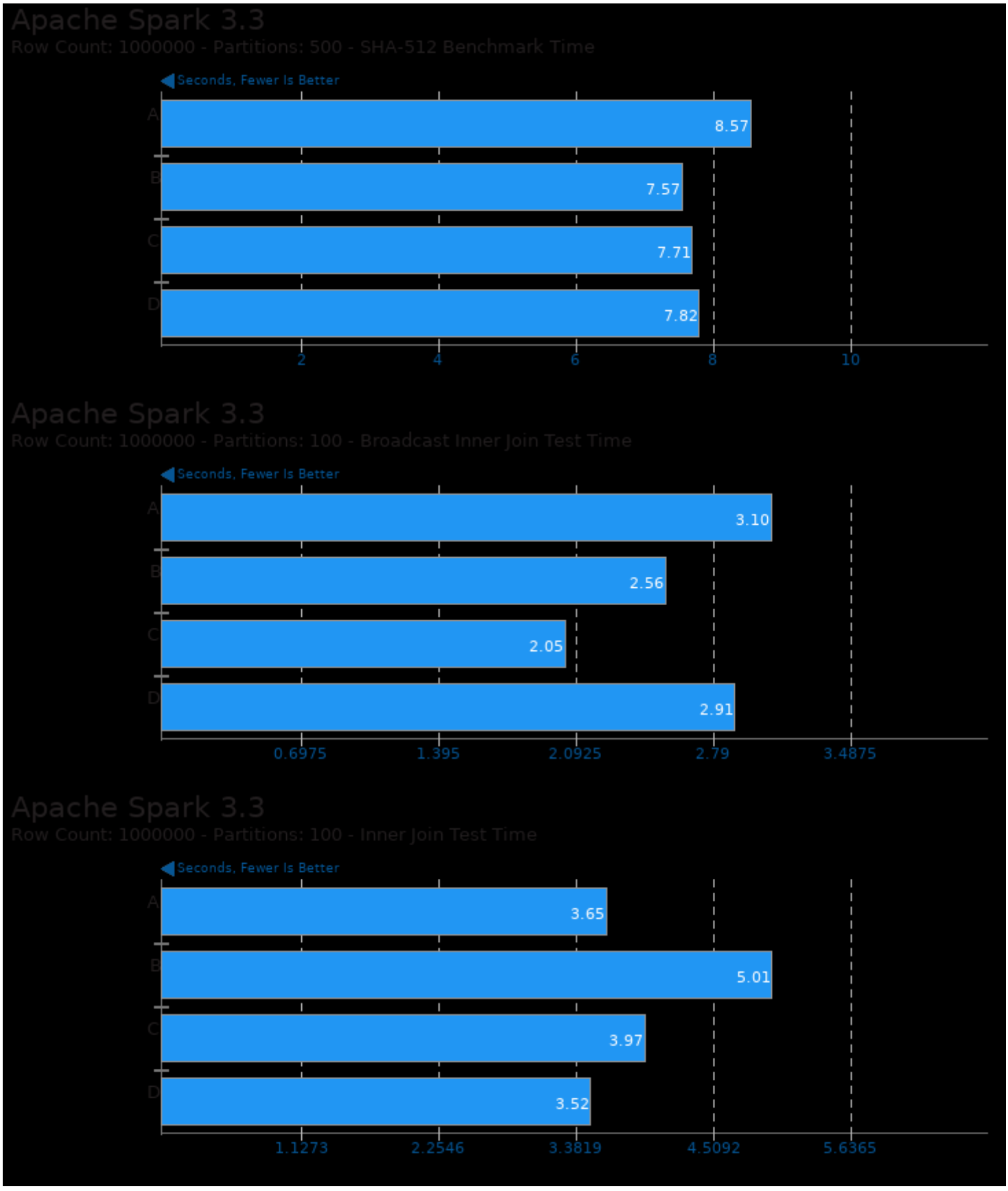
spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
3.16
B
3.56
C
4.47
D
3.11
1.0058
2.0116
3.0174
4.0232
5.029
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Inner Join Test Time
Seconds, Fewer Is Better
A
4.40
B
4.58
C
4.43
D
5.62
1.2645
2.529
3.7935
5.058
6.3225
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Repartition Test Time
Seconds, Fewer Is Better
A
5.09
B
5.98
C
7.29
D
5.83
2
4
6
8
10

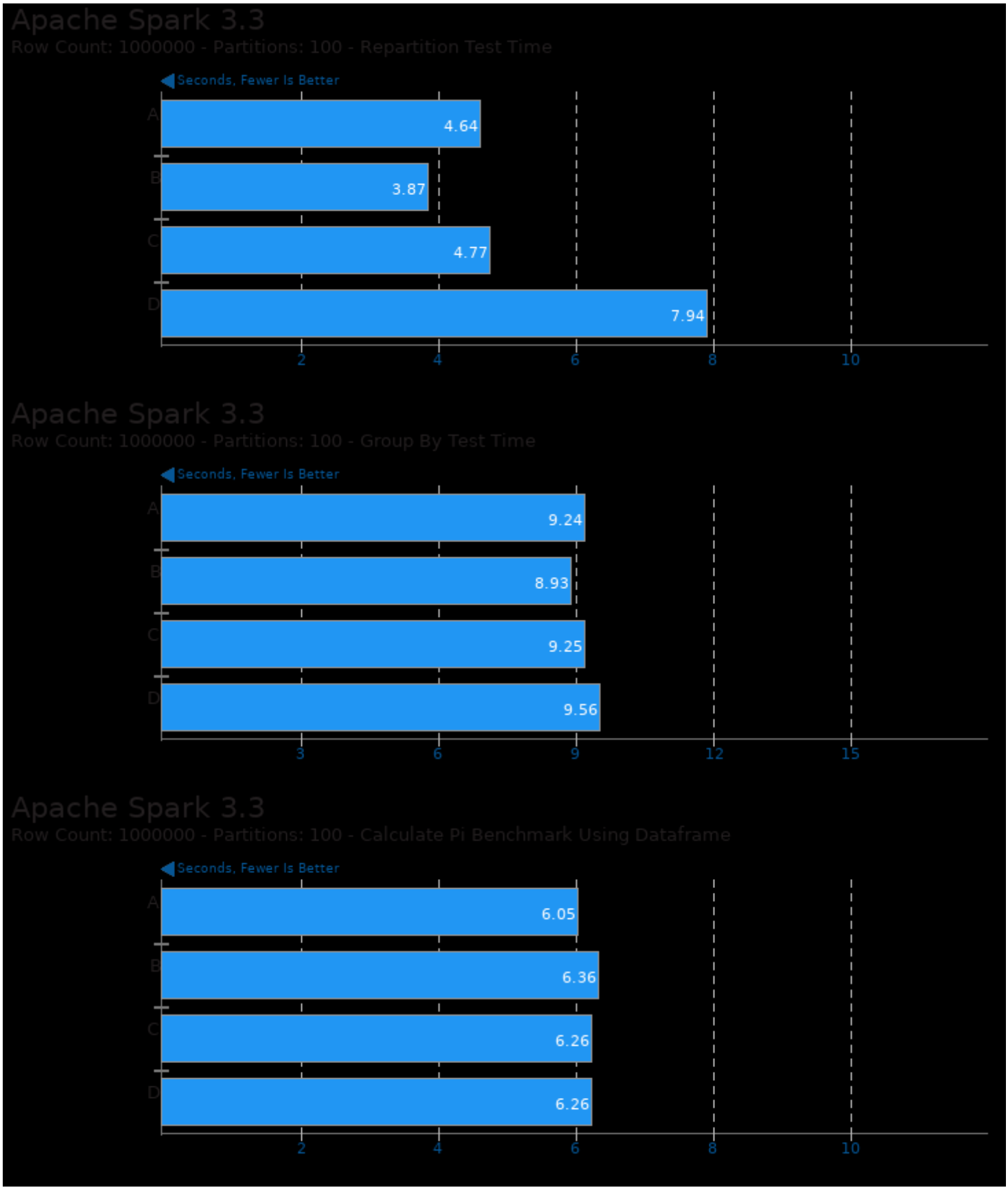
spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Group By Test Time
Seconds, Fewer Is Better
A
9.85
B
9.30
C
9.53
D
9.99
3
6
9
12
15
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.02
B
6.42
C
6.26
D
6.30
2
4
6
8
10
Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - Calculate Pi Benchmark
Seconds, Fewer Is Better
A
78.34
B
79.11
C
78.54
D
78.64
20
40
60
80
100

spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 500 - SHA-512 Benchmark Time
Seconds, Fewer Is Better
A
8.57
B
7.57
C
7.71
D
7.82
2
4
6
8
10
Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Broadcast Inner Join Test Time
Seconds, Fewer Is Better
A
3.10
B
2.56
C
2.05
D
2.91
0.6975
1.395
2.0925
2.79
3.4875
Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Inner Join Test Time
Seconds, Fewer Is Better
A
3.65
B
5.01
C
3.97
D
3.52
1.1273
2.2546
3.3819
4.5092
5.6365

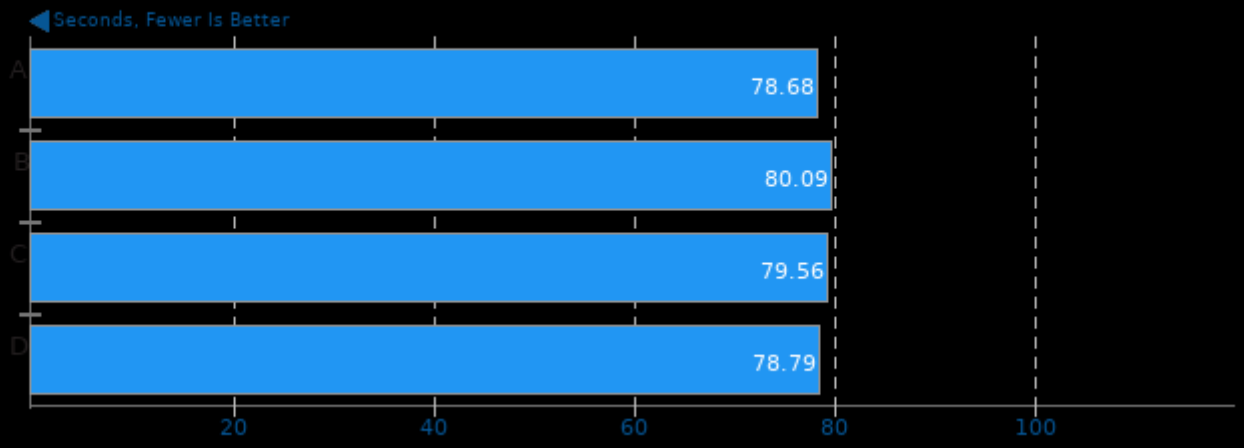
spark svt


Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Repartition Test Time
Seconds, Fewer Is Better
A
4.64
B
3.87
C
4.77
D
7.94
2
4
6
8
10
Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Group By Test Time
Seconds, Fewer Is Better
A
9.24
B
8.93
C
9.25
D
9.56
3
6
9
12
15
Apache Spark 3.3
Row Count: 1000000 - Partitions: 100 - Calculate Pi Benchmark Using Dataframe
Seconds, Fewer Is Better
A
6.05
B
6.36
C
6.26
D
6.26
2
4
6
8
10

spark svt

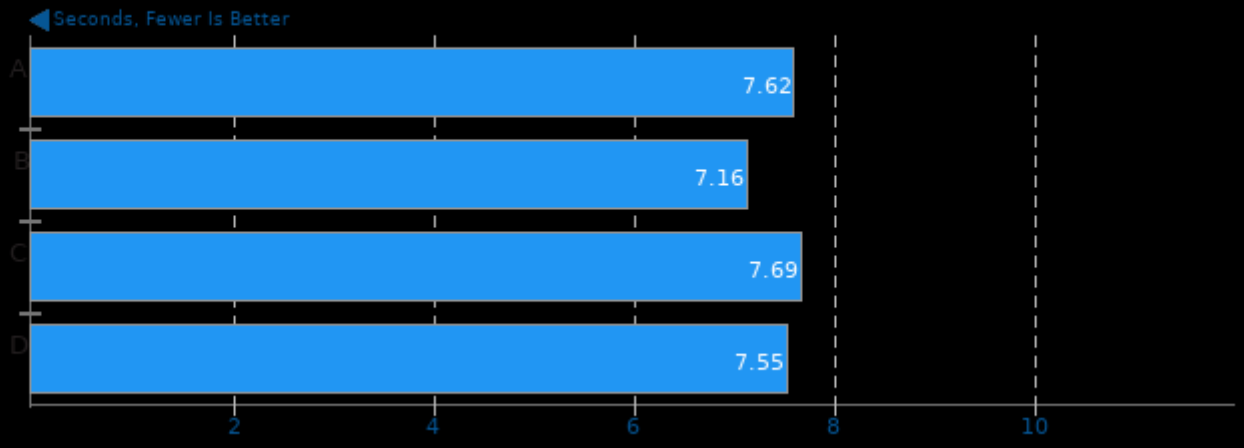
Apache Spark 3.3

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



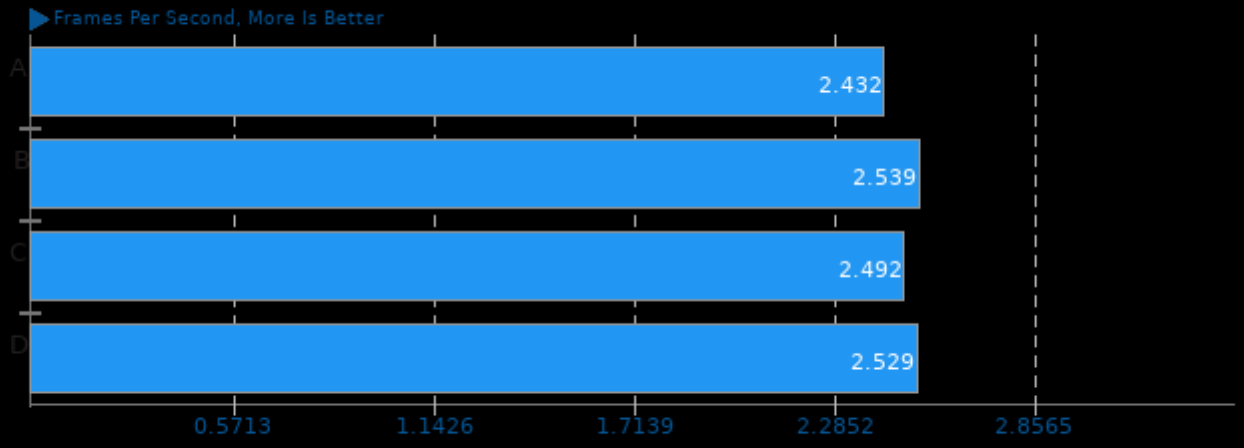
Apache Spark 3.3

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



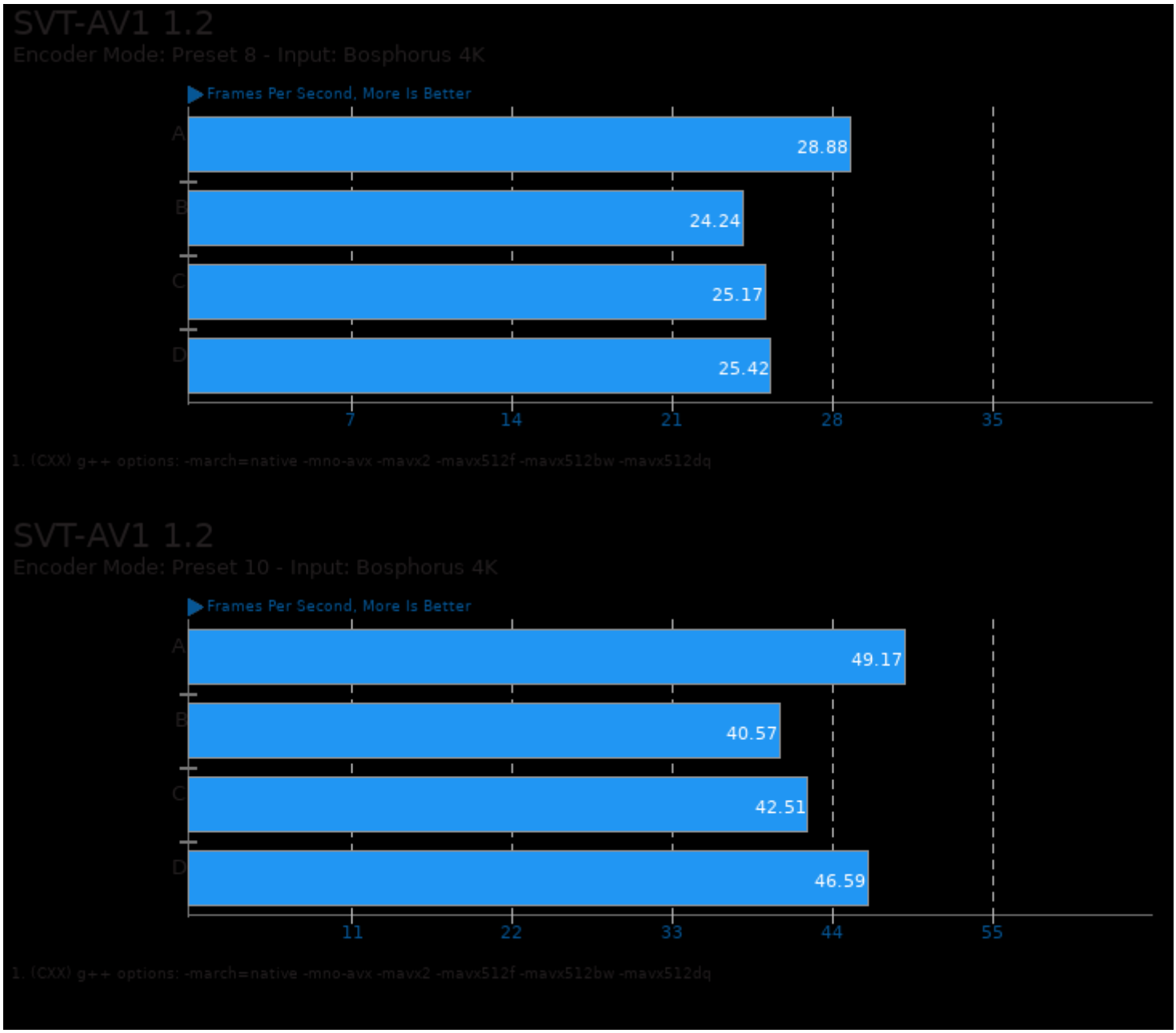
SVT-AV1 1.2

Encoder Mode: Preset 4 - Input: Bosphorus 1080p

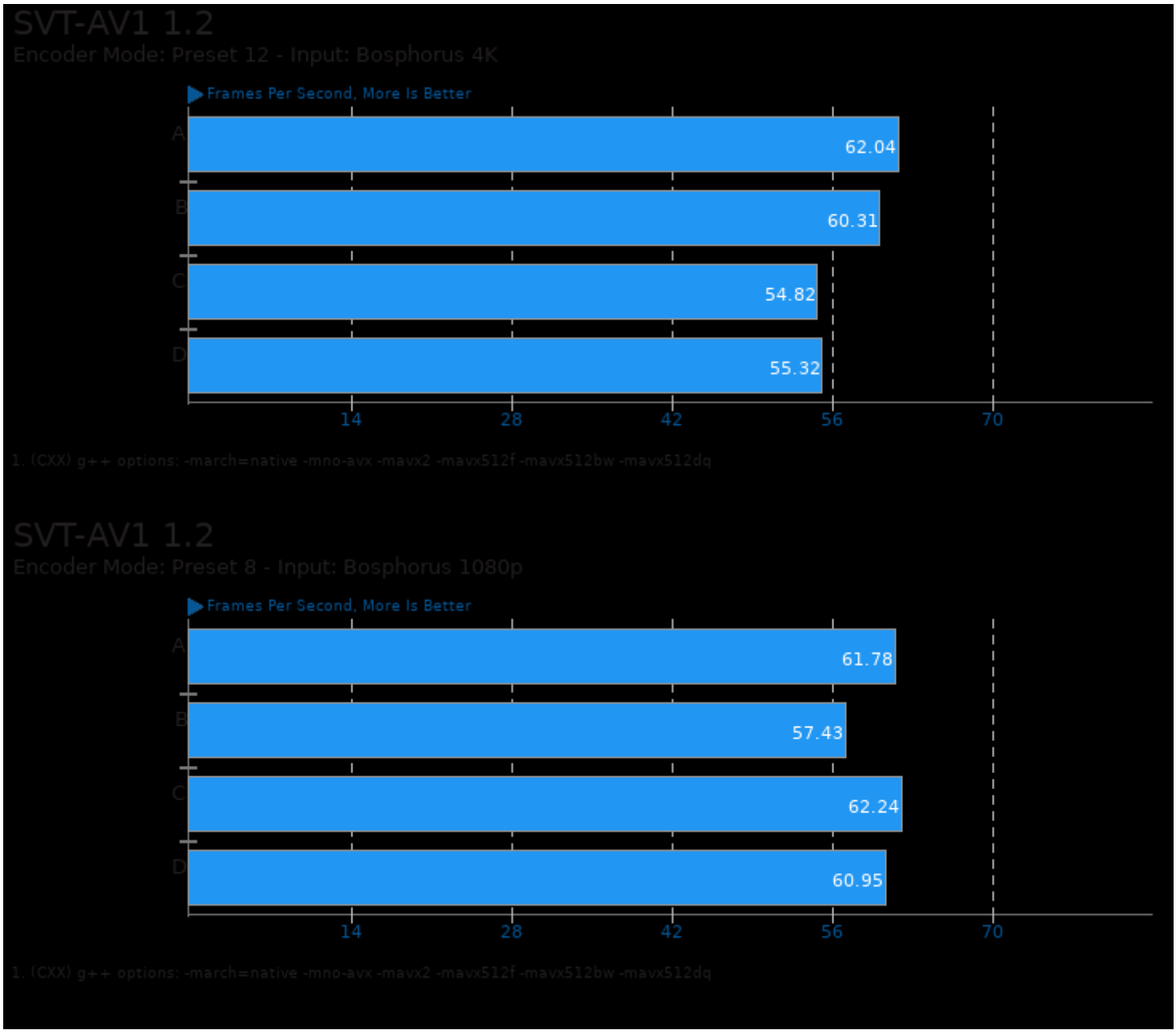


1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq

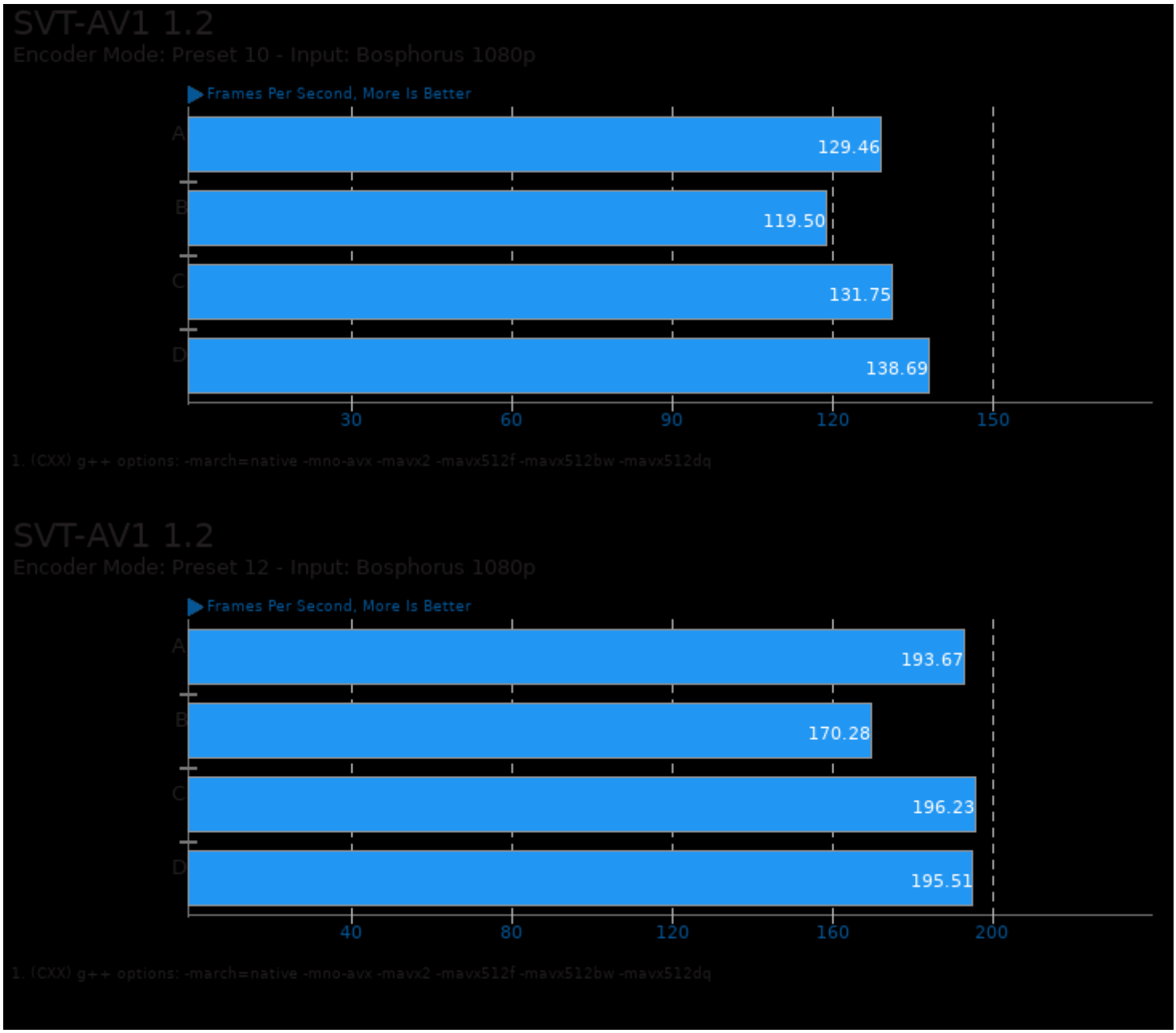
spark svt


SVT-AV1 1.2
Encoder Mode: Preset 8 - Input: Bosphorus 4K
Frames Per Second, More Is Better
A
28.88
B
24.24
C
25.17
D
25.42
7
14
21
28
35
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
49.17
B
40.57
C
42.51
D
46.59
11
22
33
44
55
1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq

spark svt


SVT-AV1 1.2
Encoder Mode: Preset 12 - Input: Bosphorus 4K
Frames Per Second, More Is Better
A
62.04
B
60.31
C
54.82
D
55.32
14
28
42
56
70
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
61.78
B
57.43
C
62.24
D
60.95
14
28
42
56
70
1. (CXX) g++ options: -march=native -mno-avx -mavx2 -mavx512f -mavx512bw -mavx512dq

spark svt



This file was automatically generated via the Phoronix Test Suite benchmarking software on Friday, 29 March 2024 02:48.