

www.phoronix-test-suite.com

icelake stuff

Intel Core i7-1065G7 testing with a Dell 06CDVY (1.0.9 BIOS) and Intel Iris Plus ICL GT2 16GB on Ubuntu 22.04 via the Phoronix Test Suite.

Automated Executive Summary

A had the most wins, coming in first place for 49% of the tests.

Based on the geometric mean of all complete results, the fastest (A) was 1.074x the speed of the slowest (C). B was 0.953x the speed of A and C was 0.977x the speed of B.

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

Redis (Test: LPOP - Parallel Connections: 50) at 1.692x
etcd (Test: RANGE - Connections: 500 - Clients: 100 - Average Latency) at 1.658x
etcd (Test: RANGE - Connections: 500 - Clients: 100) at 1.638x
etcd (Test: RANGE - Connections: 50 - Clients: 100 - Average Latency) at 1.629x
etcd (Test: RANGE - Connections: 50 - Clients: 100) at 1.624x
Redis (Test: LPOP - Parallel Connections: 500) at 1.606x
etcd (Test: PUT - Connections: 500 - Clients: 100) at 1.579x
etcd (Test: PUT - Connections: 500 - Clients: 100 - Average Latency) at 1.579x
etcd (Test: RANGE - Connections: 100 - Clients: 100 - Average Latency) at 1.553x

etcd (Test: RANGE - Connections: 100 - Clients: 100) at 1.54x.

Test Systems:

A

B

C

Processor: Intel Core i7-1065G7 @ 3.90GHz (4 Cores / 8 Threads), Motherboard: Dell 06CDVY (1.0.9 BIOS), Chipset: Intel Ice Lake-LP DRAM, Memory: 16GB, Disk: Toshiba KBG40ZPZ512G NVMe 512GB, Graphics: Intel Iris Plus ICL GT2 16GB (1100MHz), Audio: Realtek ALC289, Network: Intel Ice Lake-LP PCH CNVi WiFi

OS: Ubuntu 22.04, Kernel: 5.18.8-051808-generic (x86_64), Desktop: GNOME Shell 42.2, Display Server: X Server + Wayland, OpenGL: 4.6 Mesa 22.0.1, Vulkan: 1.3.204, Compiler: GCC 11.2.0, File-System: ext4, Screen Resolution: 1920x1200

Kernel Notes: Transparent Huge Pages: madvise
Compiler Notes: --build=x86_64-linux-gnu --disable-vtable-verify --disable-werror --enable-bootstrap --enable-cet --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-serialization=2 --enable-multiarch --enable-multilib --enable-nls --enable-objc-gc=auto --enable-offload-targets=nvptx-none=/build/gcc-11-gBFGDP/gcc-11-11.2.0/debian/tmp-nvptx/usr,amdgcn-amdhsa=/build/gcc-11-gBFGDP/gcc-11-11.2.0/debian/tmp-gcn/usr --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
Disk Notes: NONE / errors=remount-ro,relatime,rw / Block Size: 4096
Processor Notes: Scaling Governor: intel_pstate powersave (EPP: balance_performance) - CPU Microcode: 0xb0 - Thermald 2.4.9
Security Notes: itlb_multihit: KVM: Mitigation of VMX disabled + l1tf: Not affected + mds: Not affected + meltdown: Not affected + mmio_stale_data: Mitigation of Clear buffers; SMT vulnerable + spec_store_bypass: Mitigation of SSB disabled via prctl + spectre_v1: Mitigation of usercopy/swapgs barriers and __user pointer sanitization + spectre_v2: Mitigation of Enhanced IBRS IBPB: conditional RSB filling + srbds: Mitigation of Microcode + tsx_async_abort: Not affected

	A	B	C
Unpacking The Linux Kernel - linux-5.19.tar.xz (sec)	**8.356**	**8.268**	8.287
Normalized	98.95%	100%	99.77%
C-Blosc - blosclz shuffle (MB/s)	**10042**	**9658**	9924
Normalized	100%	96.17%	98.82%
C-Blosc - blosclz bitshuffle (MB/s)	6567	**6487**	**6621**
Normalized	99.19%	97.98%	100%
GraphicsMagick - Swirl (Iterations/min)	**163**	**145**	162
Normalized	100%	88.96%	99.39%
GraphicsMagick - Rotate (Iterations/min)	**857**	**838**	840
Normalized	100%	97.78%	98.02%
GraphicsMagick - Sharpen (Iterations/min)	**52**	**40**	**52**
Normalized	100%	76.92%	100%

icelake stuff

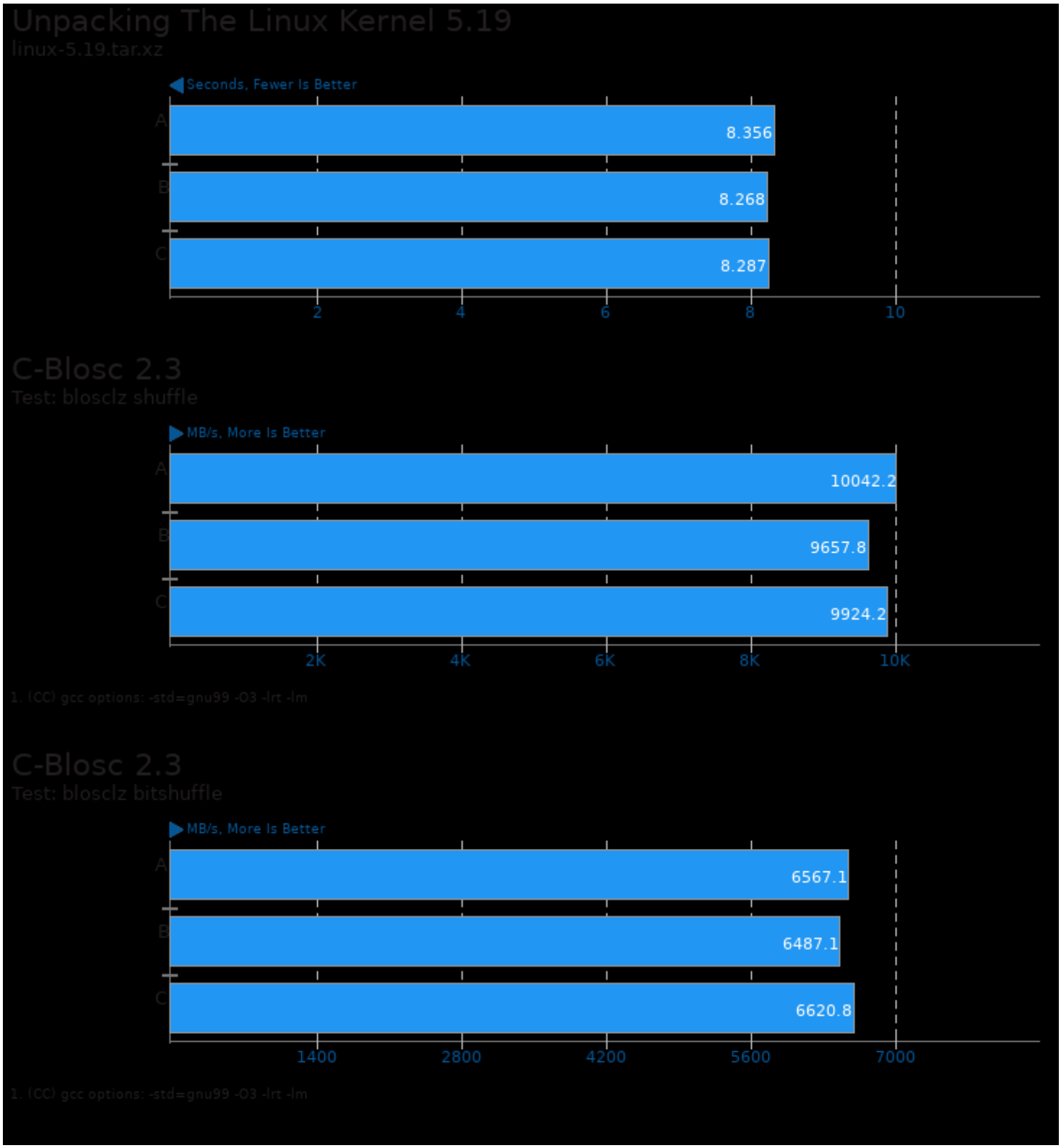
GraphicsMagick - Enhanced (Iterations/min)	62	**56**	**72**
Normalized	86.11%	77.78%	100%
GraphicsMagick - Resizing (Iterations/min)	283	**263**	**356**
Normalized	79.49%	73.88%	100%
GraphicsMagick - Noise-Gaussian (Iterations/min)	81	**75**	**85**
Normalized	95.29%	88.24%	100%
GraphicsMagick - HWB Color Space (Iterations/min)	497	**489**	**554**
Normalized	89.71%	88.27%	100%
etcd - PUT - 50 - 100 (Reqs/sec)	**30779**	**27993**	30301
Normalized	100%	90.95%	98.45%
etcd - PUT - 50 - 100 - Average Latency (ms)	**3.2**	**3.6**	3.3
Normalized	100%	88.89%	96.97%
etcd - PUT - 100 - 100 (Reqs/sec)	**26350**	25567	**21285**
Normalized	100%	97.03%	80.78%
etcd - PUT - 100 - 100 - Average Latency (ms)	**3.8**	3.9	**4.7**
Normalized	100%	97.44%	80.85%
etcd - PUT - 50 - 1000 (Reqs/sec)	**46893**	44172	**44159**
Normalized	100%	94.2%	94.17%
etcd - PUT - 50 - 1000 - Average Latency (ms)	**20.8**	22.1	**22.3**
Normalized	100%	94.12%	93.27%
etcd - PUT - 500 - 100 (Reqs/sec)	**26352**	25223	**16690**
Normalized	100%	95.71%	63.33%
etcd - PUT - 500 - 100 - Average Latency (ms)	**3.8**	3.9	**6**
Normalized	100%	97.44%	63.33%
etcd - PUT - 100 - 1000 (Reqs/sec)	42255	**43136**	**40575**
Normalized	97.96%	100%	94.06%
etcd - PUT - 100 - 1000 - Average Latency	23	**22.5**	**24.2**
Normalized	97.83%	100%	92.98%
etcd - PUT - 500 - 1000 (Reqs/sec)	**29439**	**33765**	32717
Normalized	87.19%	100%	96.9%
etcd - PUT - 500 - 1000 - Average Latency	**32.6**	**28.1**	29.3
Normalized	86.2%	100%	95.9%
etcd - RANGE - 50 - 100 (Reqs/sec)	27094	**28578**	**17600**
Normalized	94.81%	100%	61.58%
etcd - RANGE - 50 - 100 - Average Latency (ms)	3.7	**3.5**	**5.7**
Normalized	94.59%	100%	61.4%
etcd - RANGE - 100 - 100 (Reqs/sec)	25816	**26010**	**16887**
Normalized	99.26%	100%	64.92%
etcd - RANGE - 100 - 100 - Average Latency (ms)	3.9	**3.8**	**5.9**
Normalized	97.44%	100%	64.41%
etcd - RANGE - 50 - 1000 (Reqs/sec)	**44273**	**43498**	43697
Normalized	100%	98.25%	98.7%
etcd - RANGE - 50 - 1000 - Average Latency (ms)	**22.1**	**22.5**	**22.5**
Normalized	100%	98.22%	98.22%
etcd - RANGE - 500 - 100 (Reqs/sec)	**26033**	25161	**15891**
Normalized	100%	96.65%	61.04%

icelake stuff

etcd - RANGE - 500 - 100 - Average Latency (ms)	**3.8**	4	**6.3**
Normalized	100%	95%	60.32%
etcd - RANGE - 100 - 1000 (Reqs/sec)	**42392**	**35588**	40357
Normalized	100%	83.95%	95.2%
etcd - RANGE - 100 - 1000 - Average Latency (ms)	**22.9**	**27.5**	24.3
Normalized	100%	83.27%	94.24%
etcd - RANGE - 500 - 1000 (Reqs/sec)	**33675**	**30708**	31633
Normalized	100%	91.19%	93.94%
etcd - RANGE - 500 - 1000 - Average Latency (ms)	**28.2**	**31.2**	30.4
Normalized	100%	90.38%	92.76%
Dragonflydb - 50 - 1:1 (Ops/sec)	**1223644**	**1161835**	1205679
Normalized	100%	94.95%	98.53%
Dragonflydb - 50 - 1:5 (Ops/sec)	**1310577**	1323728	**1353256**
Normalized	96.85%	97.82%	100%
Dragonflydb - 50 - 5:1 (Ops/sec)	**1209028**	1168744	**1147623**
Normalized	100%	96.67%	94.92%
Dragonflydb - 200 - 1:1 (Ops/sec)	**1123969**	**1154501**	1127315
Normalized	97.36%	100%	97.65%
Dragonflydb - 200 - 1:5 (Ops/sec)	**1220448**	1225085	**1236911**
Normalized	98.67%	99.04%	100%
Dragonflydb - 200 - 5:1 (Ops/sec)	1083080	**1036751**	**1087585**
Normalized	99.59%	95.33%	100%
Redis - GET - 50 (Reqs/sec)	**2613196**	**2006295**	2054067
Normalized	100%	76.78%	78.6%
Redis - SET - 50 (Reqs/sec)	1573147	**1474518**	**1722387**
Normalized	91.34%	85.61%	100%
Redis - GET - 500 (Reqs/sec)	**2243903**	2227708	**2204528**
Normalized	100%	99.28%	98.25%
Redis - LPOP - 50 (Reqs/sec)	**2448384**	1476085	**1447330**
Normalized	100%	60.29%	59.11%
Redis - SADD - 50 (Reqs/sec)	**1958209**	2024142	**2167544**
Normalized	90.34%	93.38%	100%
Redis - SET - 500 (Reqs/sec)	**1733855**	1726908	**1691620**
Normalized	100%	99.6%	97.56%
Redis - GET - 1000 (Reqs/sec)	**2210557**	2171456	**1856685**
Normalized	100%	98.23%	83.99%
Redis - LPOP - 500 (Reqs/sec)	**2370160**	**1475823**	1487060
Normalized	100%	62.27%	62.74%
Redis - LPUSH - 50 (Reqs/sec)	**1904869**	**1453472**	1886043
Normalized	100%	76.3%	99.01%
Redis - SADD - 500 (Reqs/sec)	1961820	**1776357**	**2037942**
Normalized	96.26%	87.16%	100%
Redis - SET - 1000 (Reqs/sec)	1661297	**1514481**	**1929917**
Normalized	86.08%	78.47%	100%
Redis - LPOP - 1000 (Reqs/sec)	**1448923**	**1555619**	1468438
Normalized	93.14%	100%	94.4%
Redis - LPUSH - 500 (Reqs/sec)	1502250	**1364318**	**1574137**
Normalized	95.43%	86.67%	100%
Redis - SADD - 1000 (Reqs/sec)	**2050489**	1914215	**1892381**
Normalized	100%	93.35%	92.29%

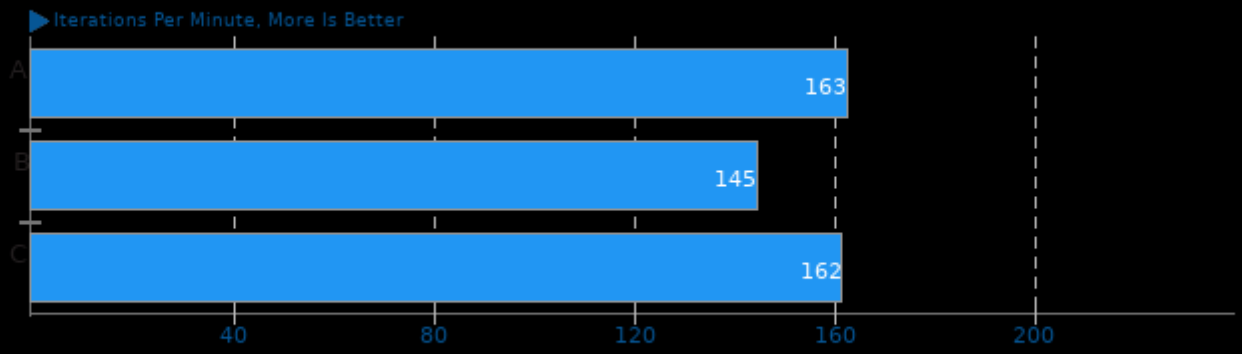
icelake stuff

Redis - LPUSH - 1000 (Reqs/sec)	1484244	**1457510**	**1521577**
Normalized	97.55%	95.79%	100%
memtier_benchmark - Redis - 50 - 1:1 (Ops/sec)	**1217910**	1168087	**1164388**
Normalized	100%	95.91%	95.61%
memtier_benchmark - Redis - 50 - 1:5 (Ops/sec)	**1255526**	**1195319**	1233231
Normalized	100%	95.2%	98.22%
memtier_benchmark - Redis - 50 - 5:1 (Ops/sec)	1082060	**1082707**	**1062734**
Normalized	99.94%	100%	98.16%
memtier_benchmark - Redis - 100 - 1:1 (Ops/sec)	**1136347**	**1096855**	1102035
Normalized	100%	96.52%	96.98%
memtier_benchmark - Redis - 100 - 1:5 (Ops/sec)	**1167809**	1190357	**1201792**
Normalized	97.17%	99.05%	100%
memtier_benchmark - Redis - 100 - 5:1 (Ops/sec)	**1035475**	1064869	**1070231**
Normalized	96.75%	99.5%	100%
memtier_benchmark - Redis - 50 - 1:10 (Ops/sec)	**1188774**	**1248746**	1223149
Normalized	95.2%	100%	97.95%
memtier_benchmark - Redis - 500 - 1:1 (Ops/sec)	**944916**	**1008397**	973167
Normalized	93.7%	100%	96.51%
memtier_benchmark - Redis - 500 - 1:5 (Ops/sec)	**1022392**	1063748	**1069278**
Normalized	95.62%	99.48%	100%
memtier_benchmark - Redis - 500 - 5:1 (Ops/sec)	**913341**	944099	**993069**
Normalized	91.97%	95.07%	100%
memtier_benchmark - Redis - 100 - 1:10 (Ops/sec)	**1214374**	**1232518**	1225347
Normalized	98.53%	100%	99.42%
memtier_benchmark - Redis - 500 - 1:10 (Ops/sec)	**1019436**	1030580	**1069539**
Normalized	95.32%	96.36%	100%


Unpacking The Linux Kernel 5.19
linux-5.19.tar.xz
Seconds, Fewer Is Better
A
B
C
8.356
8.268
8.287
2
4
6
8
10
C-Blosc 2.3
Test: blosclz shuffle
MB/s, More Is Better
A
B
C
10042.2
9657.8
9924.2
2K
4K
6K
8K
10K
1. (CC) gcc options: -std=gnu99 -O3 -lrt -lm
C-Blosc 2.3
Test: blosclz bitshuffle
MB/s, More Is Better
A
B
C
6567.1
6487.1
6620.8
1400
2800
4200
5600
7000
1. (CC) gcc options: -std=gnu99 -O3 -lrt -lm

GraphicsMagick 1.3.38

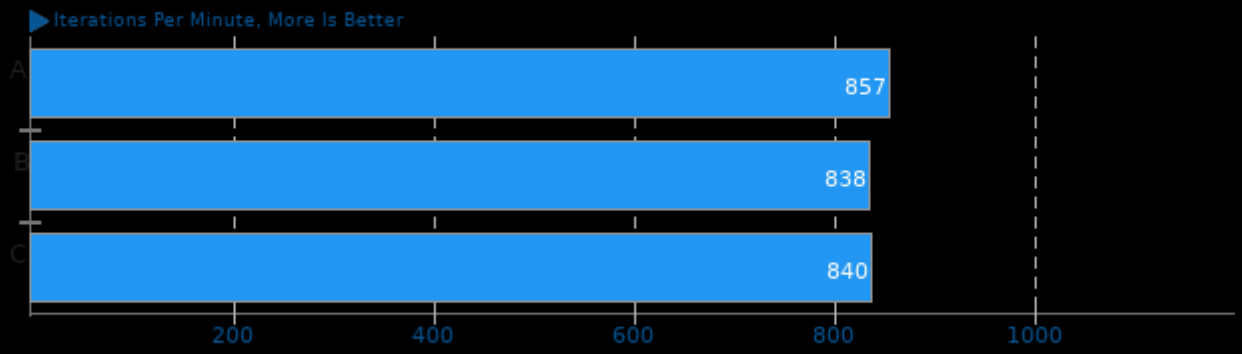
Operation: Swirl



1. (CC) gcc options: -fopenmp -O2 -ljbig -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lxml2 -lz -lm -lpthread

GraphicsMagick 1.3.38

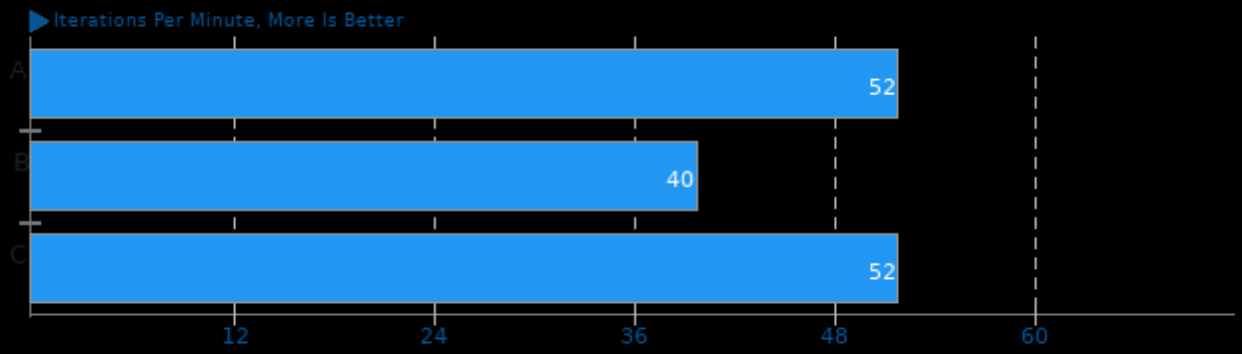
Operation: Rotate



1. (CC) gcc options: -fopenmp -O2 -ljbig -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lxml2 -lz -lm -lpthread

GraphicsMagick 1.3.38

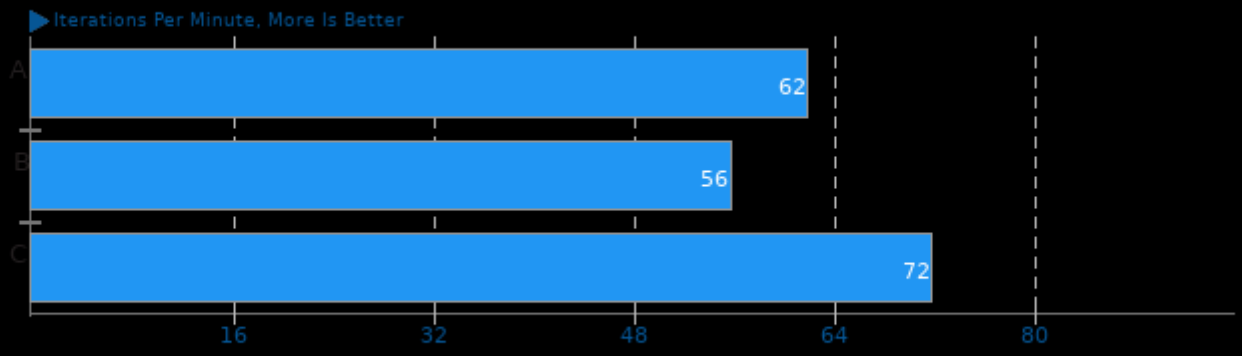
Operation: Sharpen



1. (CC) gcc options: -fopenmp -O2 -ljbig -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lxml2 -lz -lm -lpthread

GraphicsMagick 1.3.38

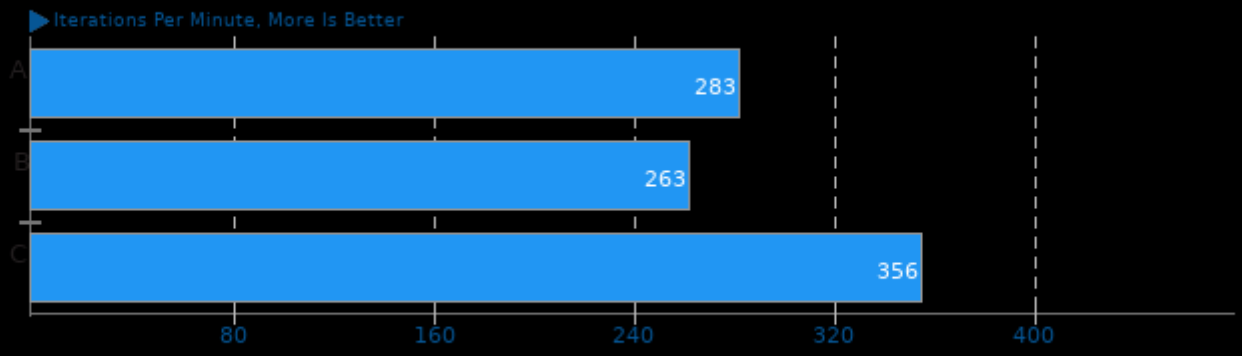
Operation: Enhanced



1. (CC) gcc options: -fopenmp -O2 -ljbig -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lxml2 -lz -lm -lpthread

GraphicsMagick 1.3.38

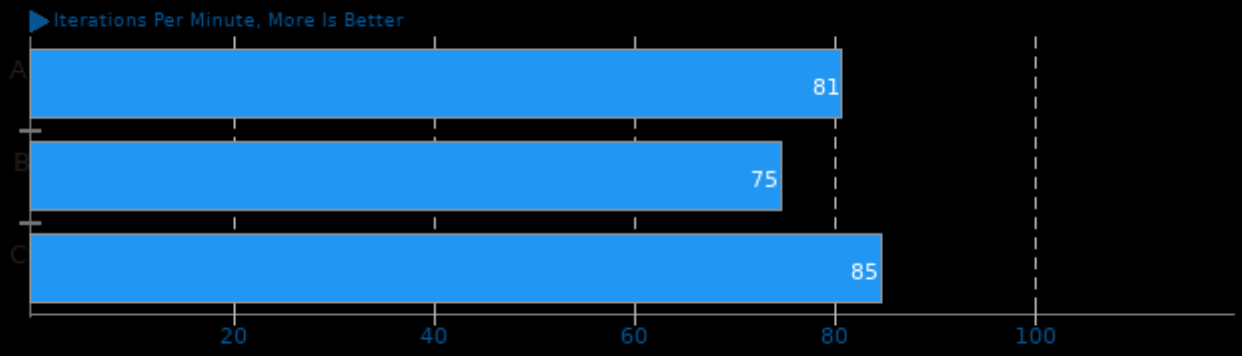
Operation: Resizing



1. (CC) gcc options: -fopenmp -O2 -ljbig -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lxml2 -lz -lm -lpthread

GraphicsMagick 1.3.38

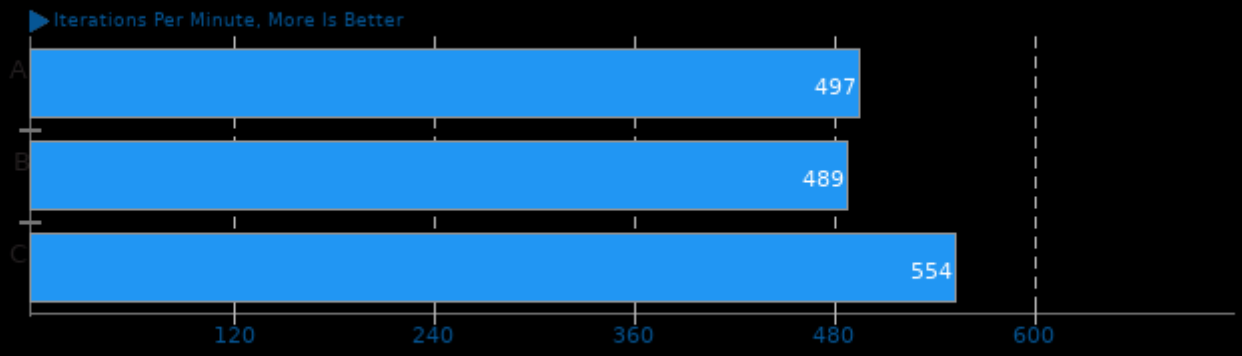
Operation: Noise-Gaussian



1. (CC) gcc options: -fopenmp -O2 -ljbig -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lxml2 -lz -lm -lpthread

GraphicsMagick 1.3.38

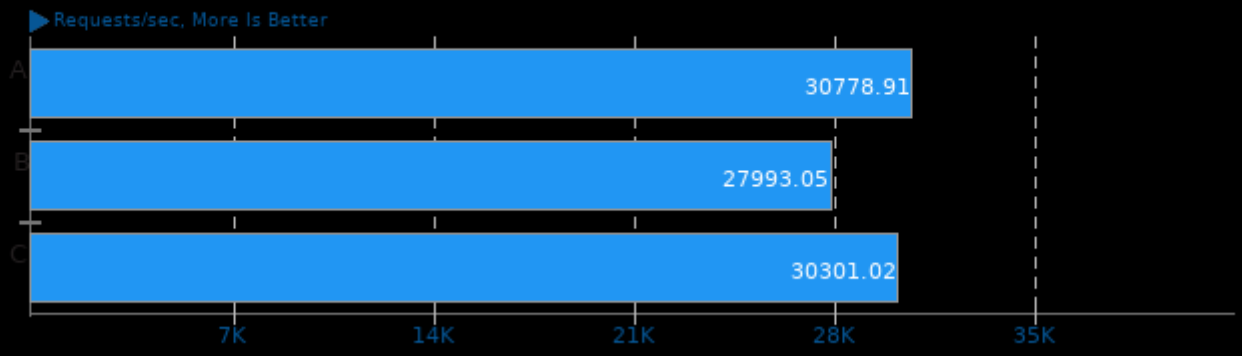
Operation: HWB Color Space



1. (CC) gcc options: -fopenmp -O2 -ljbig -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lxml2 -lz -lm -lpthread

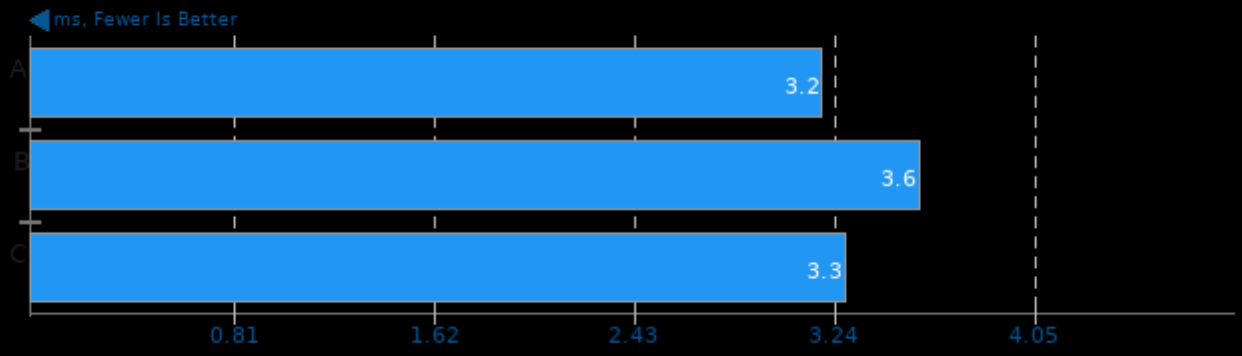
etcd 3.5.4

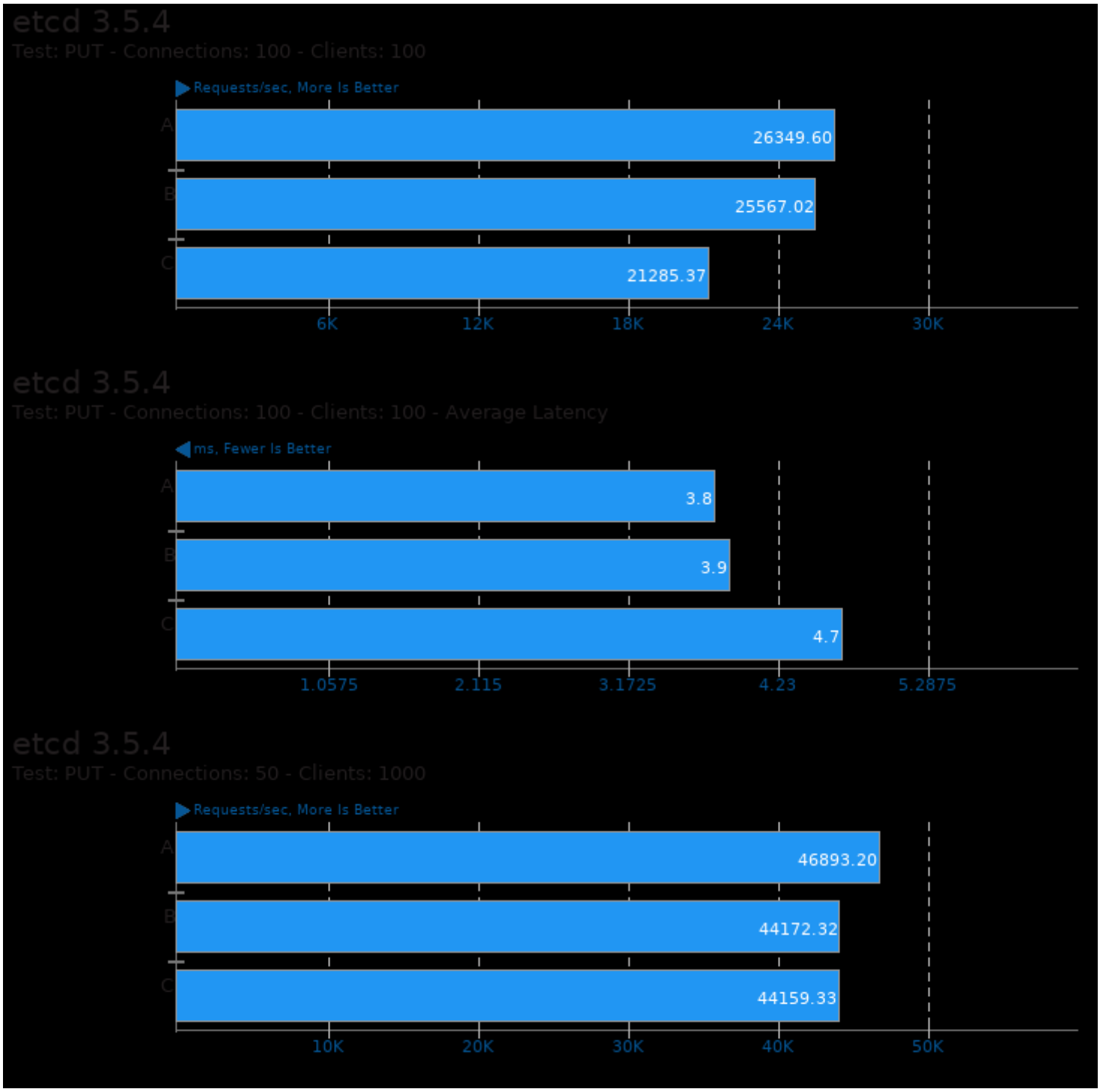
Test: PUT - Connections: 50 - Clients: 100

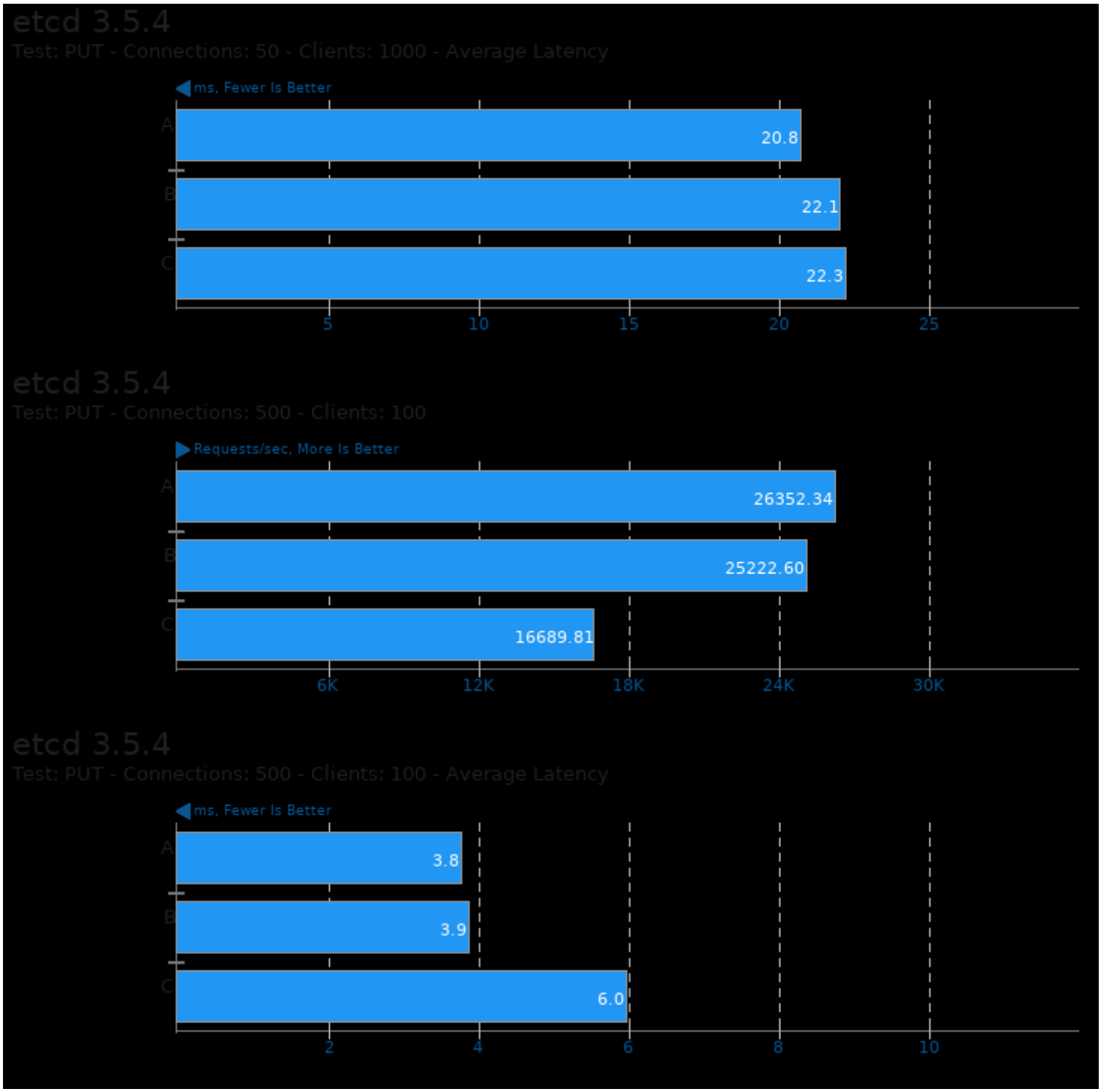


etcd 3.5.4

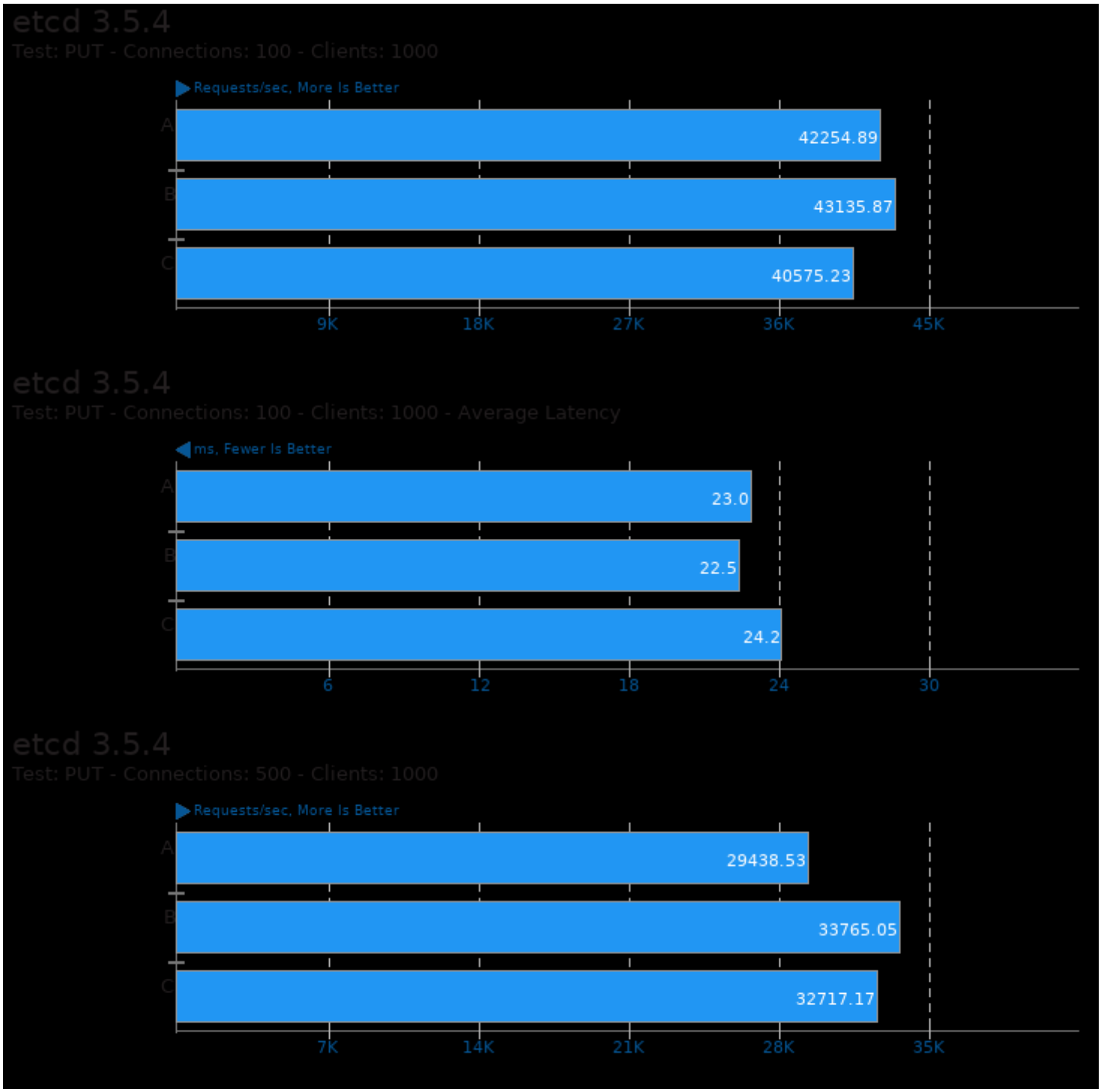
Test: PUT - Connections: 50 - Clients: 100 - Average Latency

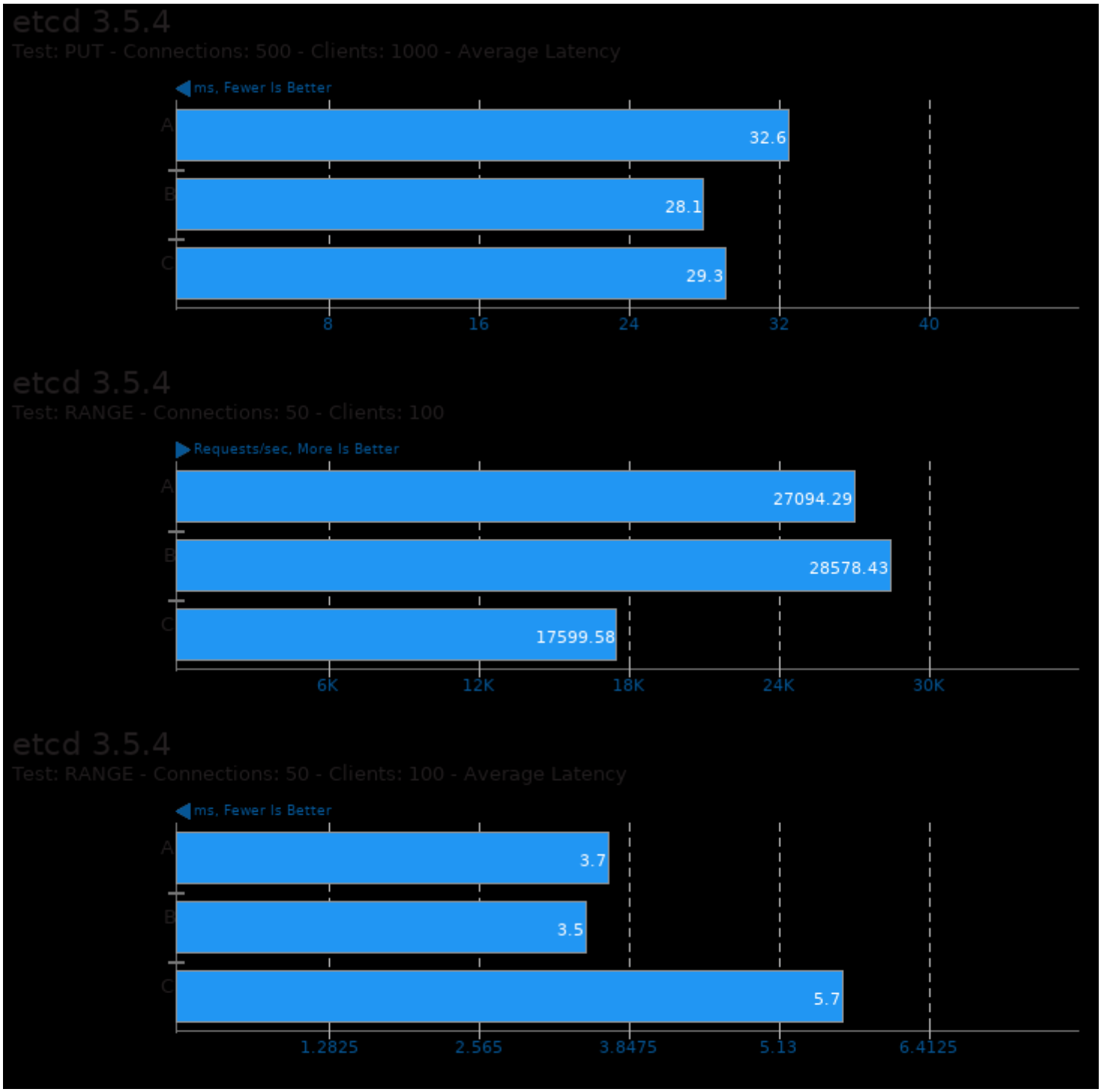


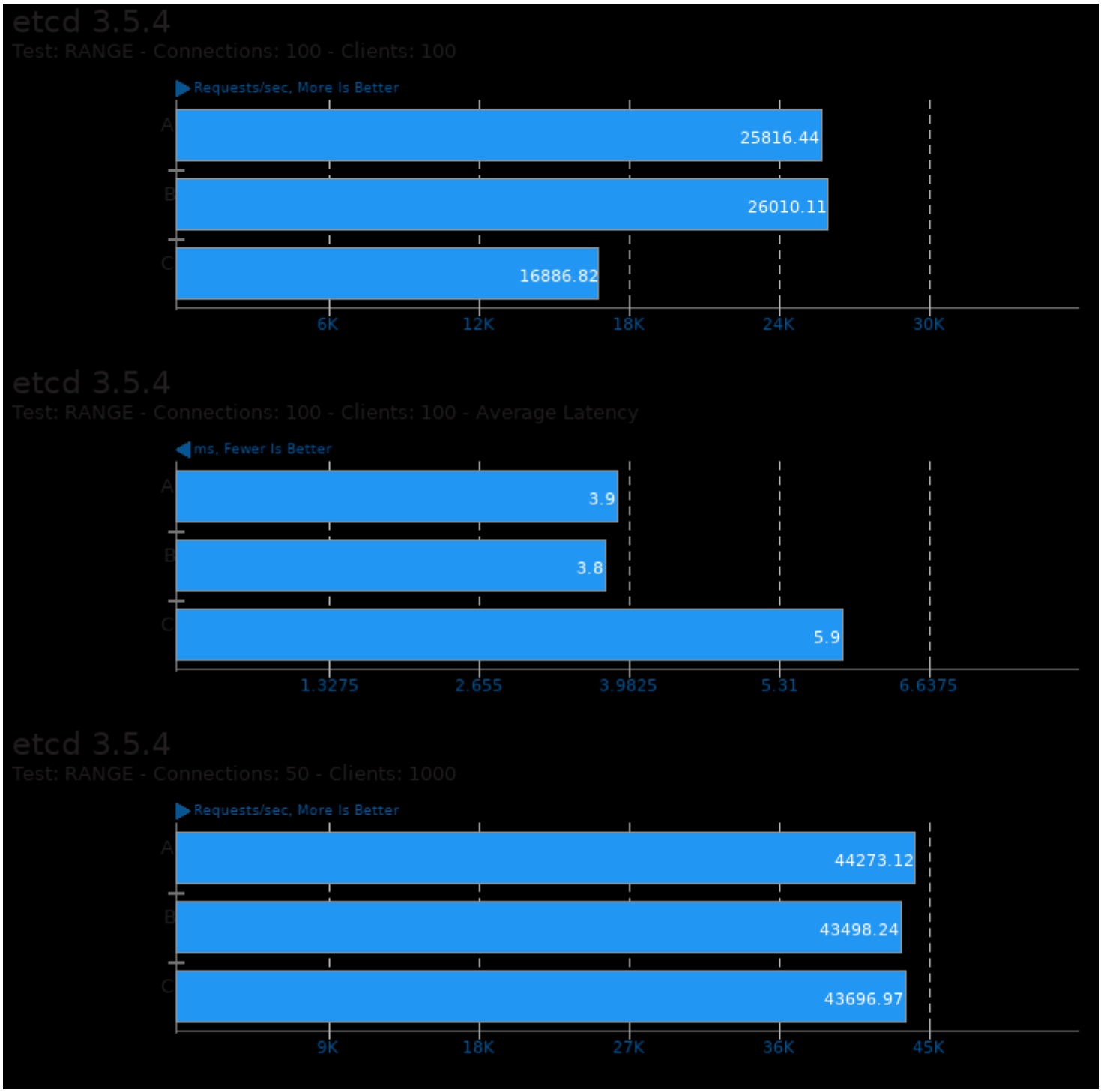

etcd 3.5.4
Test: PUT - Connections: 100 - Clients: 100
Requests/sec, More Is Better
A
B
C
26349.60
25567.02
21285.37
6K
12K
18K
24K
30K
etcd 3.5.4
Test: PUT - Connections: 100 - Clients: 100 - Average Latency
ms, Fewer Is Better
A
B
C
3.8
3.9
4.7
1.0575
2.115
3.1725
4.23
5.2875
etcd 3.5.4
Test: PUT - Connections: 50 - Clients: 1000
Requests/sec, More Is Better
A
B
C
46893.20
44172.32
44159.33
10K
20K
30K
40K
50K

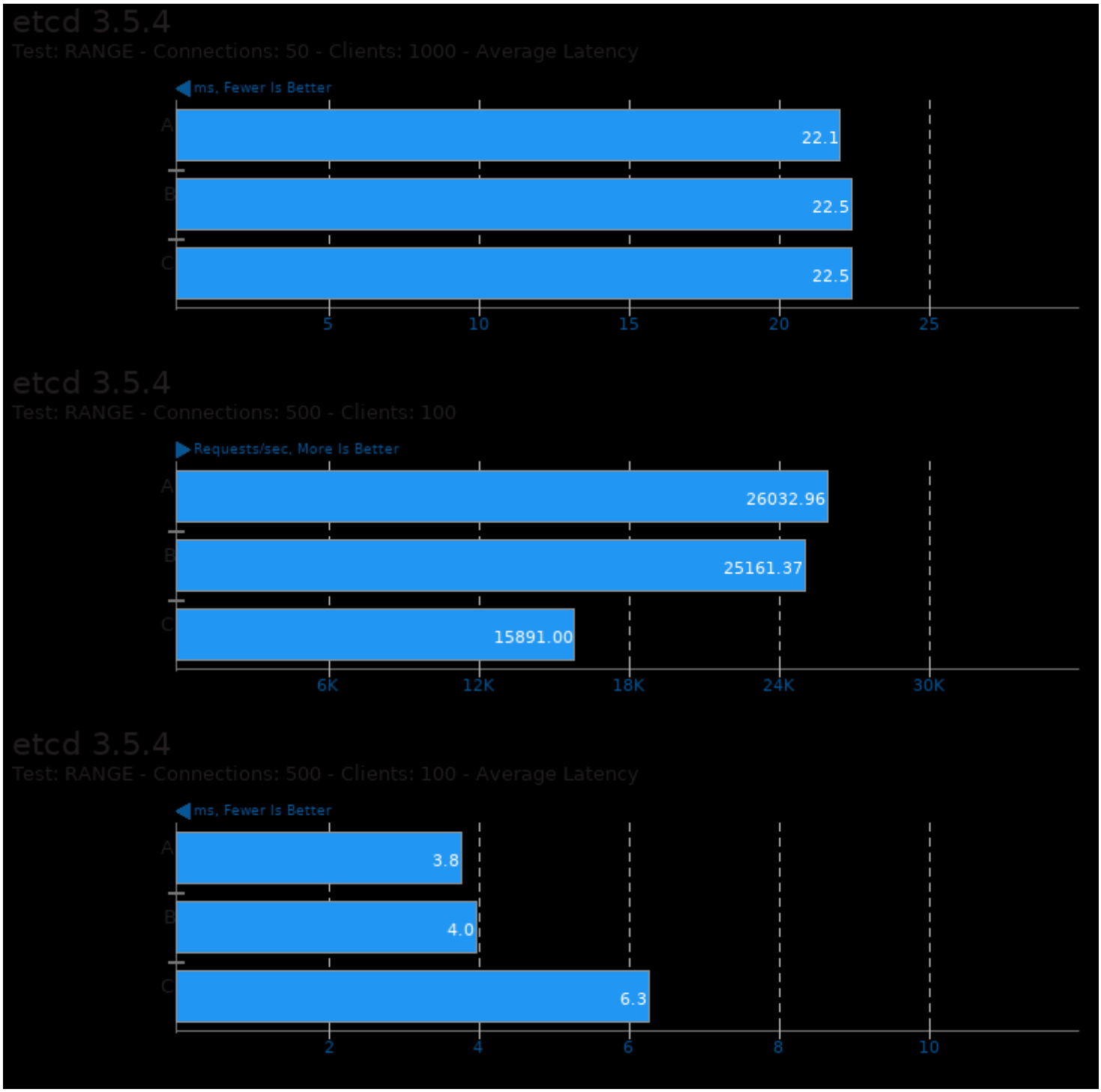

etcd 3.5.4
Test: PUT - Connections: 50 - Clients: 1000 - Average Latency
ms, Fewer Is Better
A
B
C
20.8
22.1
22.3
5
10
15
20
25
etcd 3.5.4
Test: PUT - Connections: 500 - Clients: 100
Requests/sec, More Is Better
A
B
C
26352.34
25222.60
16689.81
6K
12K
18K
24K
30K
etcd 3.5.4
Test: PUT - Connections: 500 - Clients: 100 - Average Latency
ms, Fewer Is Better
A
B
C
3.8
3.9
6.0
2
4
6
8
10

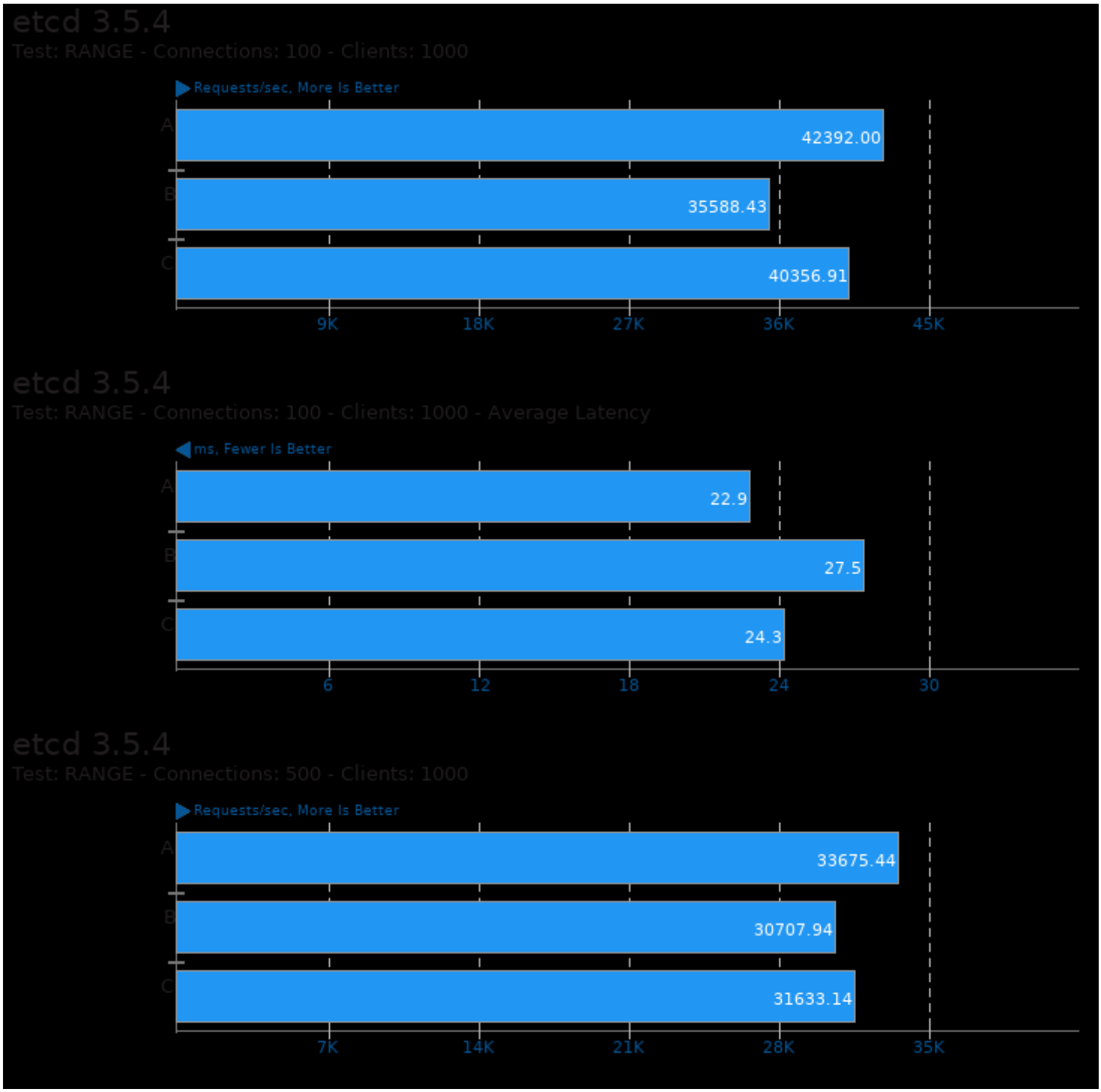
icelake stuff

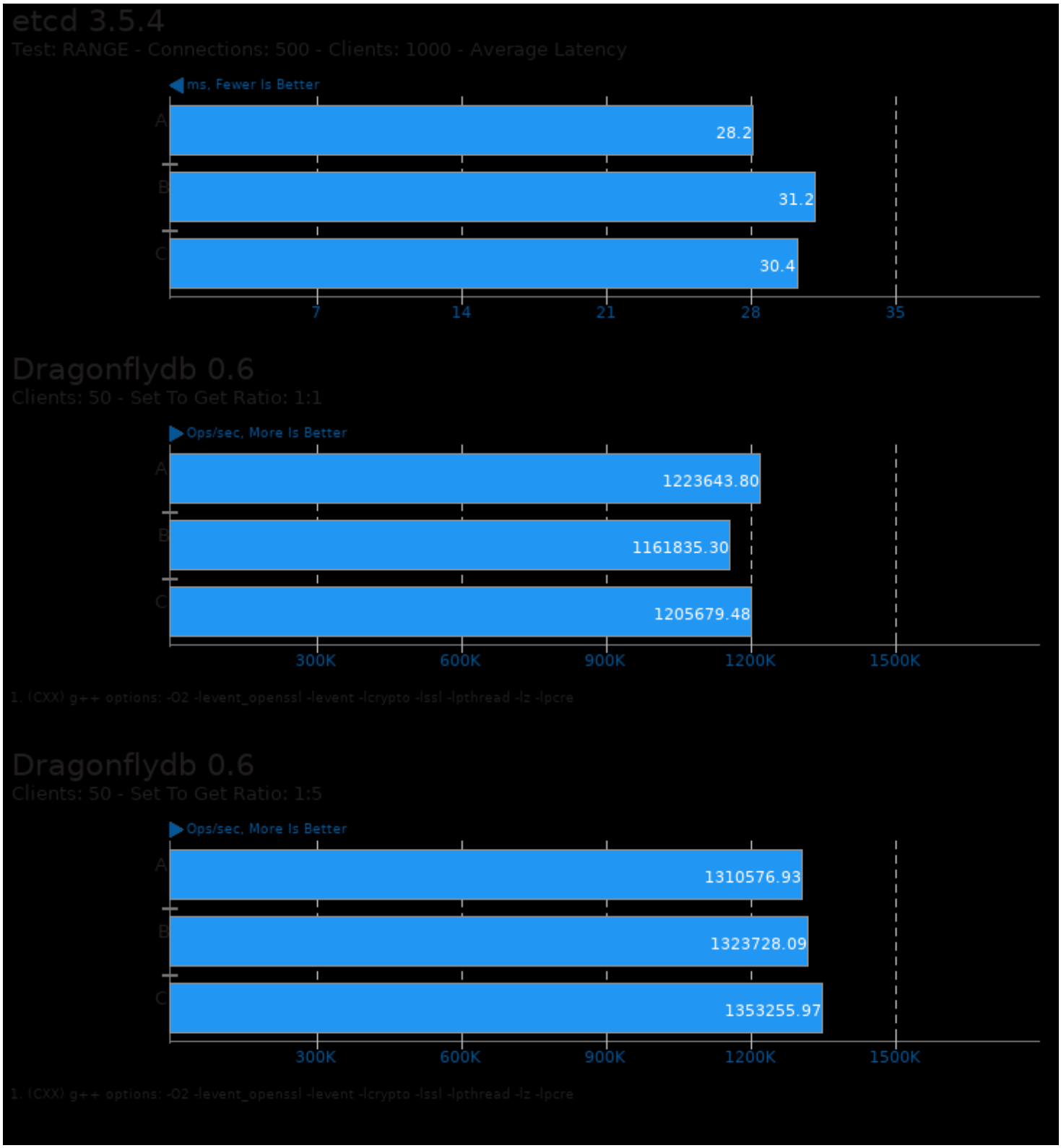


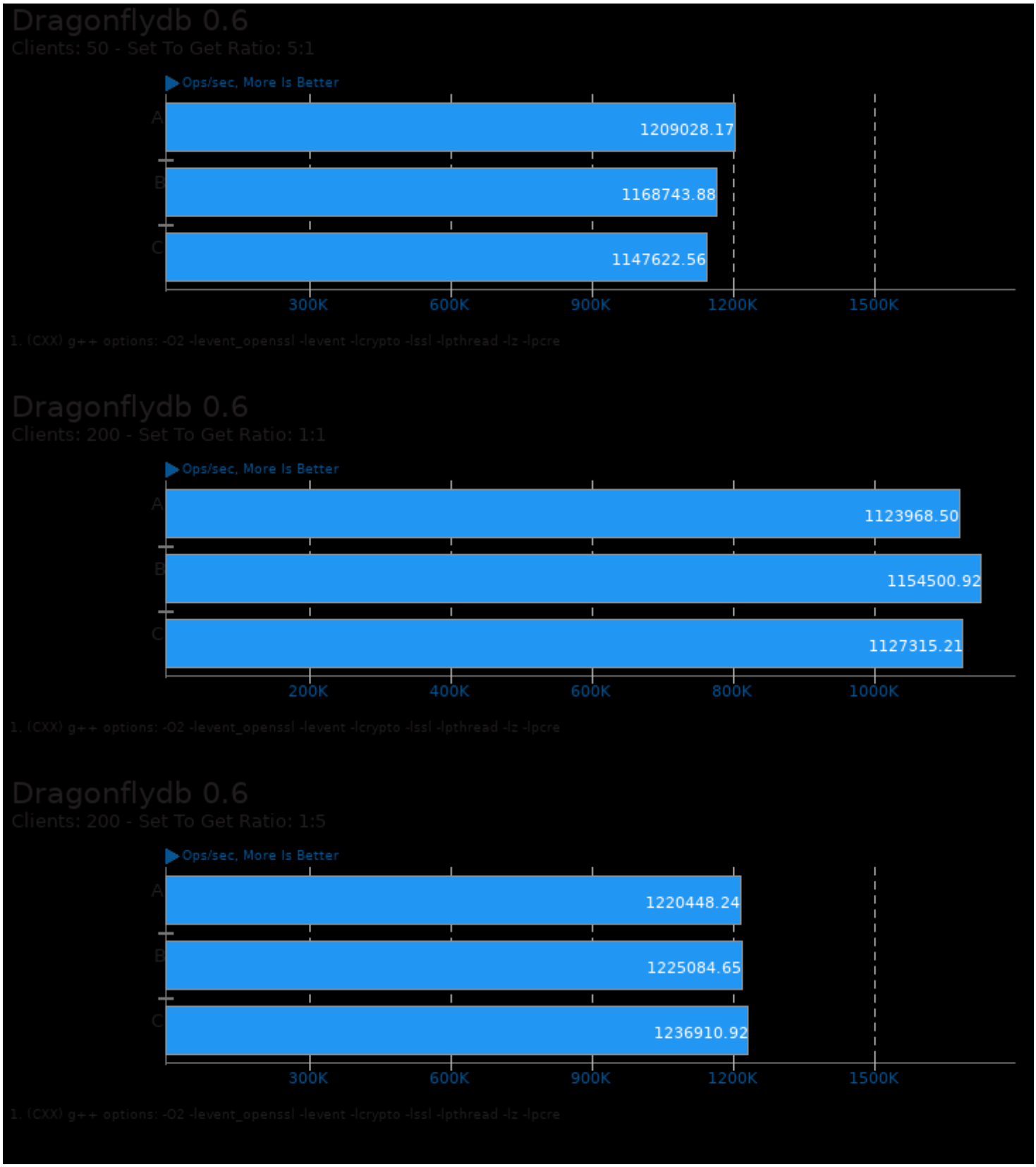

etcd 3.5.4
Test: PUT - Connections: 500 - Clients: 1000 - Average Latency
ms, Fewer Is Better
A
B
C
32.6
28.1
29.3
8
16
24
32
40
etcd 3.5.4
Test: RANGE - Connections: 50 - Clients: 100
Requests/sec, More Is Better
A
B
C
27094.29
28578.43
17599.58
6K
12K
18K
24K
30K
etcd 3.5.4
Test: RANGE - Connections: 50 - Clients: 100 - Average Latency
ms, Fewer Is Better
A
B
C
3.7
3.5
5.7
1.2825
2.565
3.8475
5.13
6.4125


etcd 3.5.4
Test: RANGE - Connections: 100 - Clients: 100
Requests/sec, More Is Better
A
25816.44
B
26010.11
C
16886.82
6K
12K
18K
24K
30K
etcd 3.5.4
Test: RANGE - Connections: 100 - Clients: 100 - Average Latency
ms, Fewer Is Better
A
3.9
B
3.8
C
5.9
1.3275
2.655
3.9825
5.31
6.6375
etcd 3.5.4
Test: RANGE - Connections: 50 - Clients: 1000
Requests/sec, More Is Better
A
44273.12
B
43498.24
C
43696.97
9K
18K
27K
36K
45K


etcd 3.5.4
Test: RANGE - Connections: 50 - Clients: 1000 - Average Latency
ms, Fewer Is Better
A
B
C
22.1
22.5
22.5
5
10
15
20
25
etcd 3.5.4
Test: RANGE - Connections: 500 - Clients: 100
Requests/sec, More Is Better
A
B
C
26032.96
25161.37
15891.00
6K
12K
18K
24K
30K
etcd 3.5.4
Test: RANGE - Connections: 500 - Clients: 100 - Average Latency
ms, Fewer Is Better
A
B
C
3.8
4.0
6.3
2
4
6
8
10

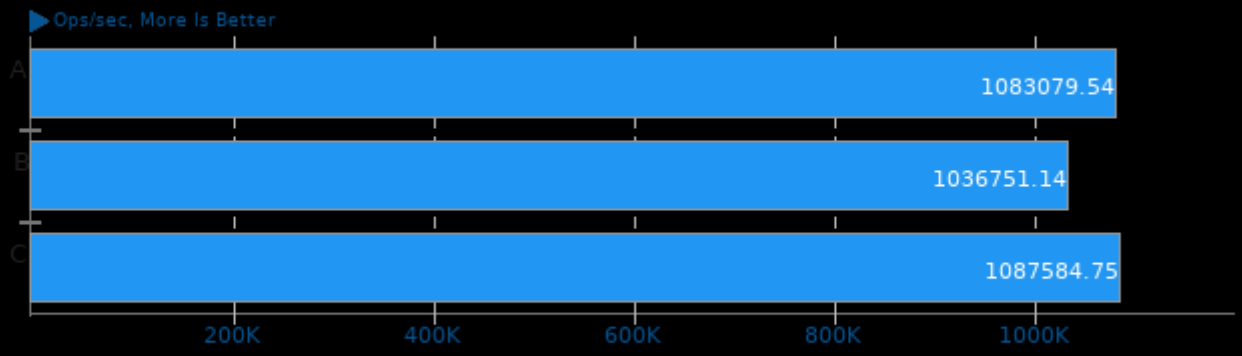

etcd 3.5.4
Test: RANGE - Connections: 100 - Clients: 1000
Requests/sec, More Is Better
A
42392.00
B
35588.43
C
40356.91
9K
18K
27K
36K
45K
etcd 3.5.4
Test: RANGE - Connections: 100 - Clients: 1000 - Average Latency
ms, Fewer Is Better
A
22.9
B
27.5
C
24.3
6
12
18
24
30
etcd 3.5.4
Test: RANGE - Connections: 500 - Clients: 1000
Requests/sec, More Is Better
A
33675.44
B
30707.94
C
31633.14
7K
14K
21K
28K
35K


etcd 3.5.4
Test: RANGE - Connections: 500 - Clients: 1000 - Average Latency
ms, Fewer Is Better
A
28.2
B
31.2
C
30.4
7
14
21
28
35
Dragonflydb 0.6
Clients: 50 - Set To Get Ratio: 1:1
Ops/sec, More Is Better
A
1223643.80
B
1161835.30
C
1205679.48
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
Dragonflydb 0.6
Clients: 50 - Set To Get Ratio: 1:5
Ops/sec, More Is Better
A
1310576.93
B
1323728.09
C
1353255.97
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre


Dragonflydb 0.6
Clients: 50 - Set To Get Ratio: 5:1
Ops/sec, More Is Better
A
1209028.17
B
1168743.88
C
1147622.56
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
Dragonflydb 0.6
Clients: 200 - Set To Get Ratio: 1:1
Ops/sec, More Is Better
A
1123968.50
B
1154500.92
C
1127315.21
200K
400K
600K
800K
1000K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
Dragonflydb 0.6
Clients: 200 - Set To Get Ratio: 1:5
Ops/sec, More Is Better
A
1220448.24
B
1225084.65
C
1236910.92
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre

Dragonflydb 0.6

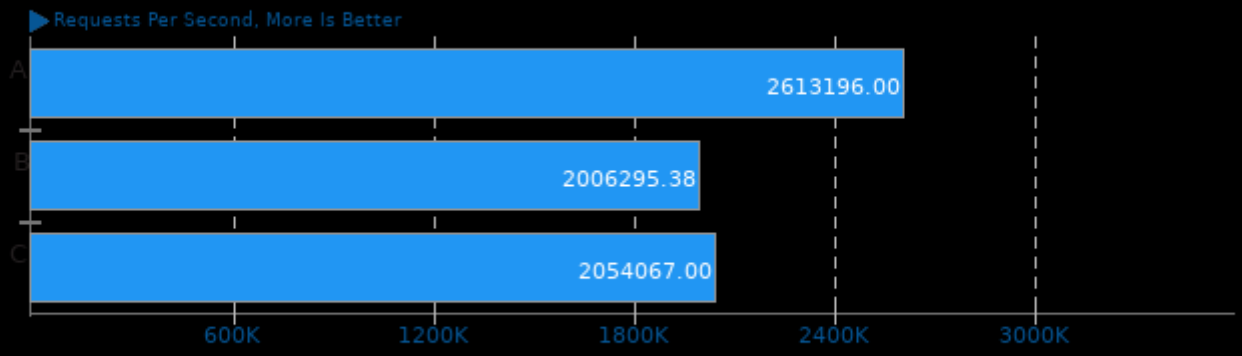
Clients: 200 - Set To Get Ratio: 5:1



1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre

Redis 7.0.4

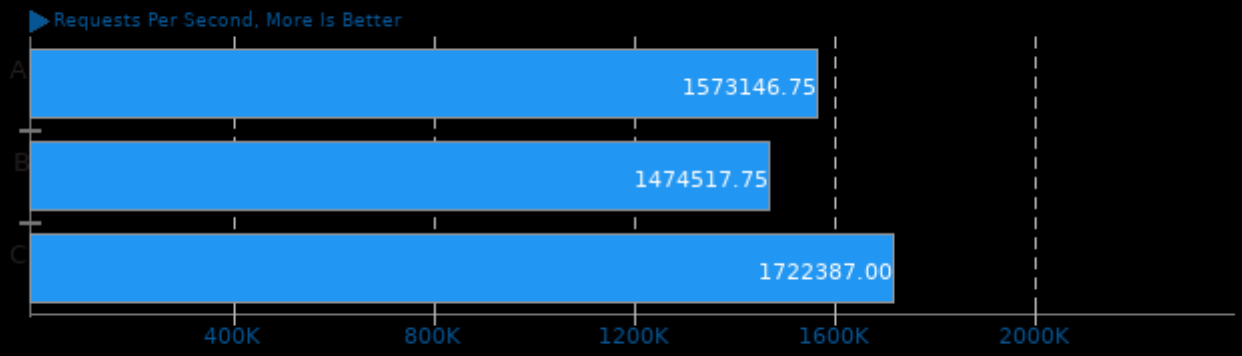
Test: GET - Parallel Connections: 50



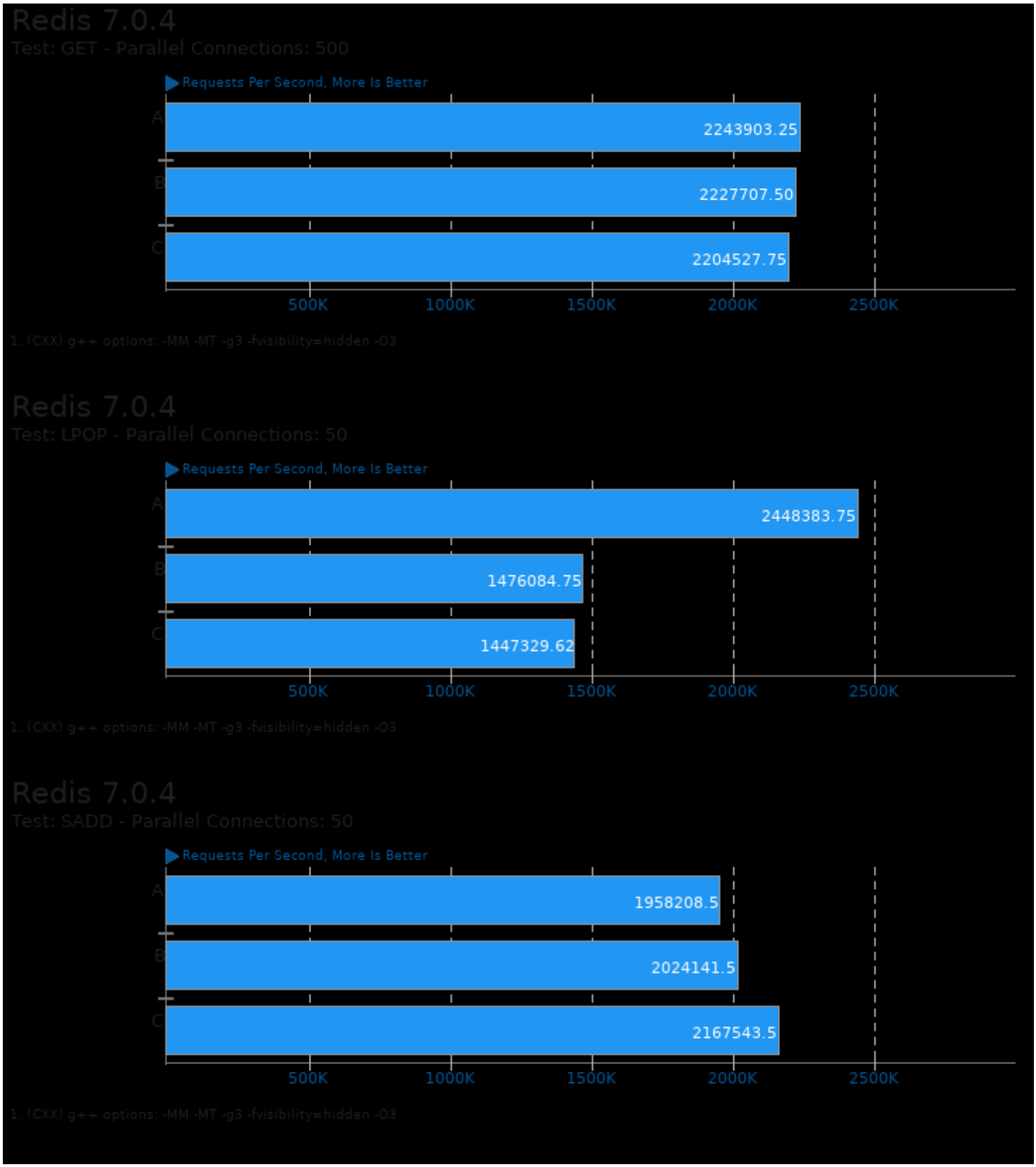
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3

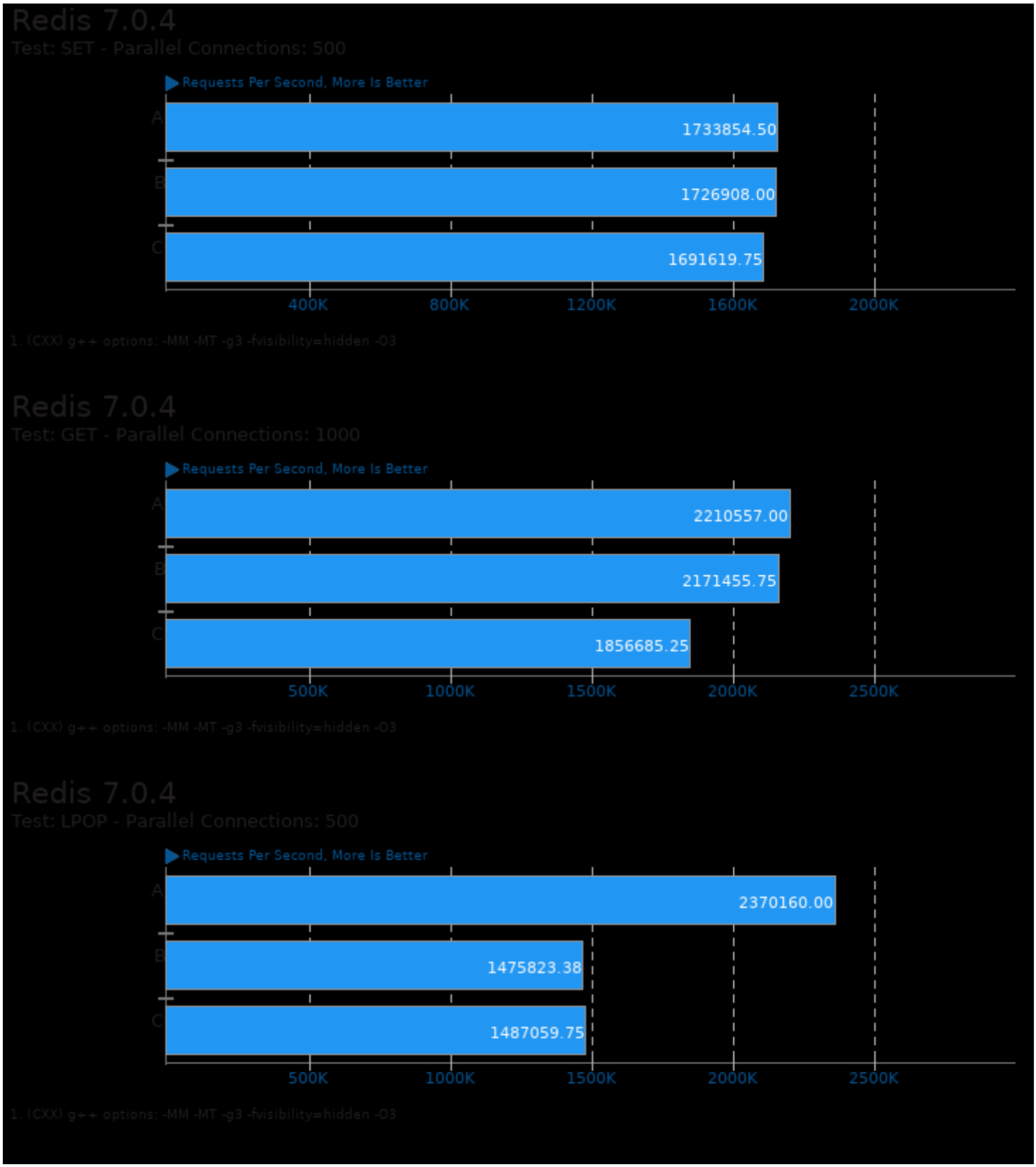
Redis 7.0.4

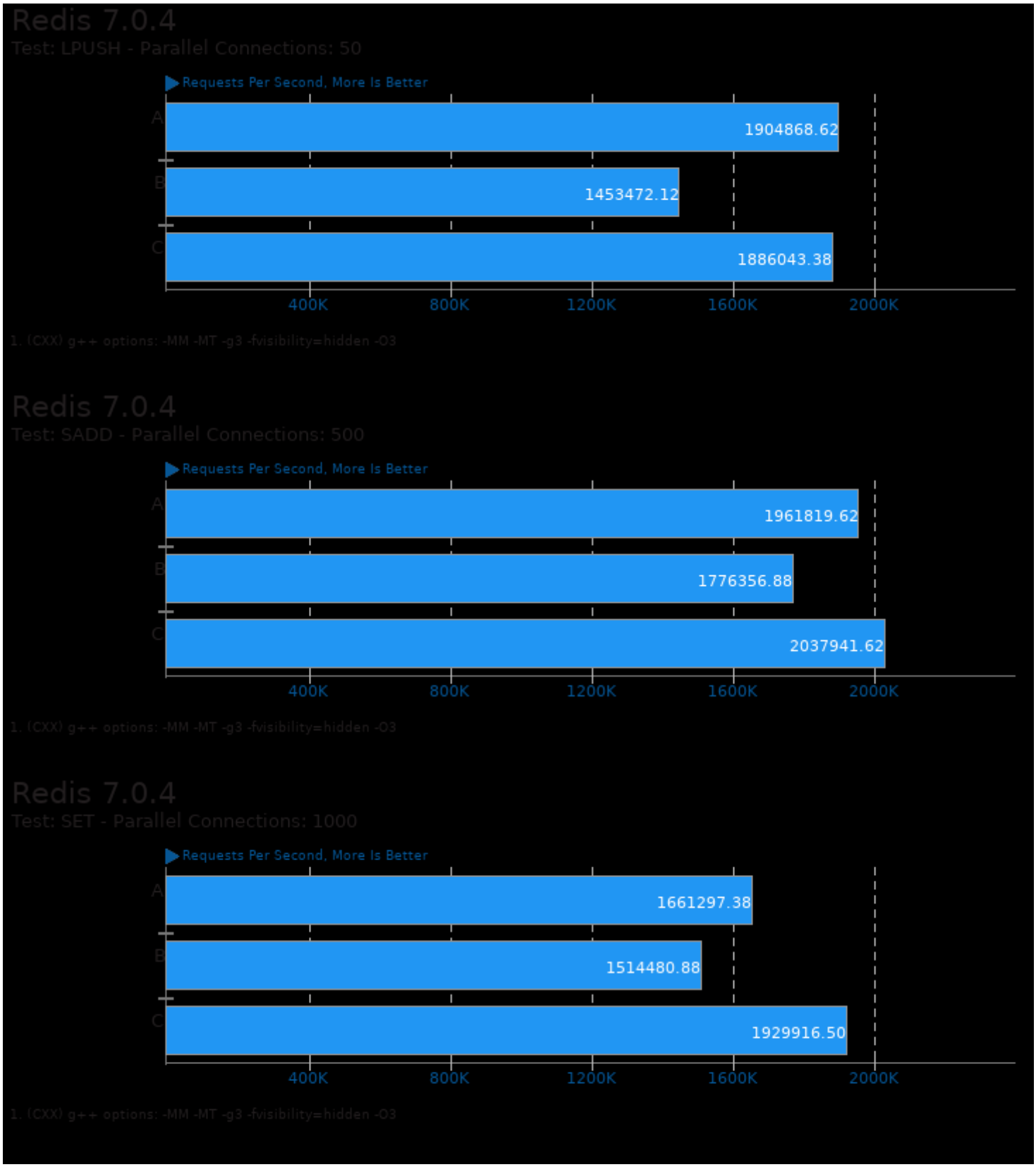
Test: SET - Parallel Connections: 50

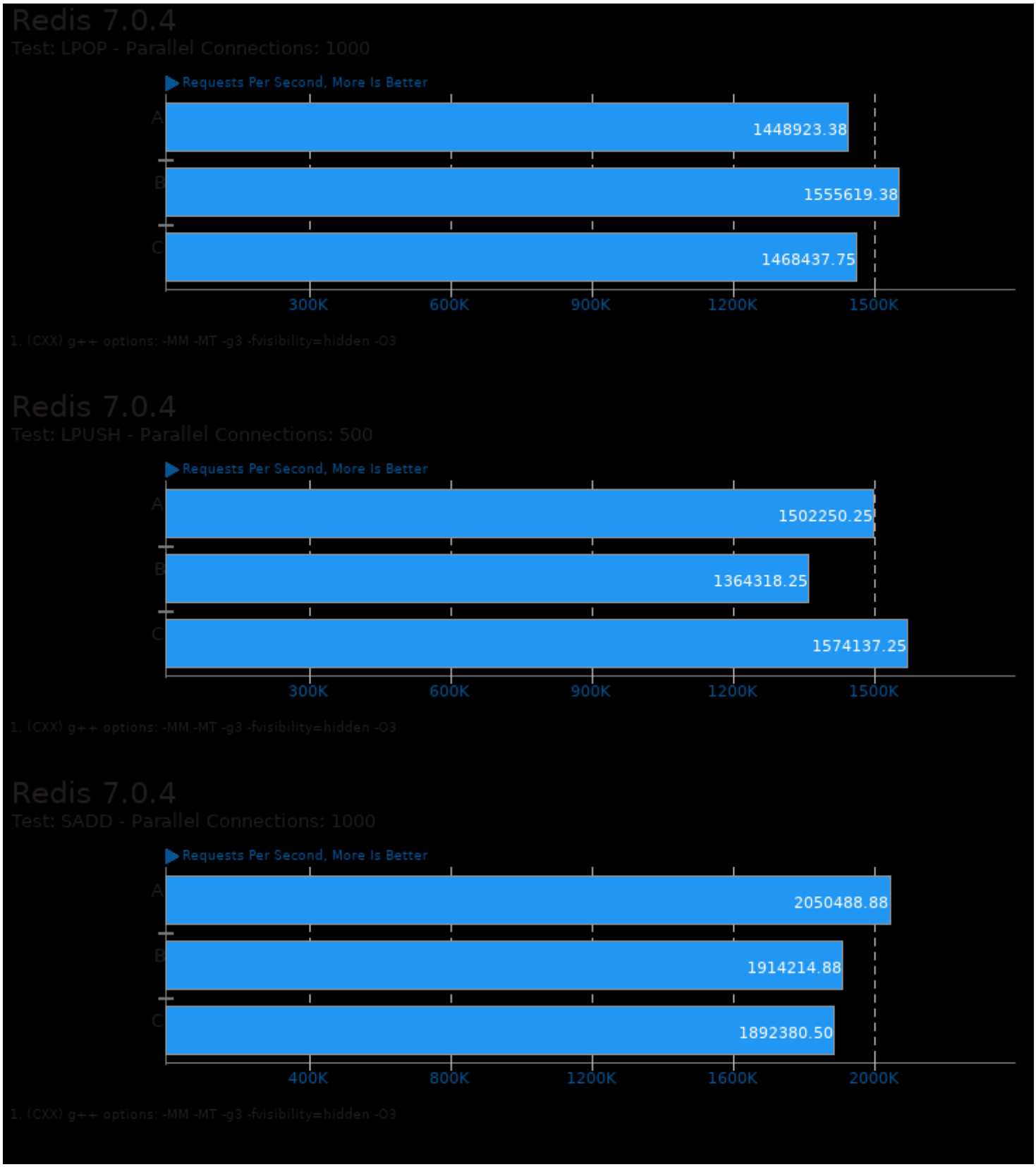


1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3


Redis 7.0.4
Test: GET - Parallel Connections: 500
Requests Per Second, More Is Better
A
2243903.25
B
2227707.50
C
2204527.75
500K
1000K
1500K
2000K
2500K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: LPOP - Parallel Connections: 50
Requests Per Second, More Is Better
A
2448383.75
B
1476084.75
C
1447329.62
500K
1000K
1500K
2000K
2500K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: SADD - Parallel Connections: 50
Requests Per Second, More Is Better
A
1958208.5
B
2024141.5
C
2167543.5
500K
1000K
1500K
2000K
2500K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3

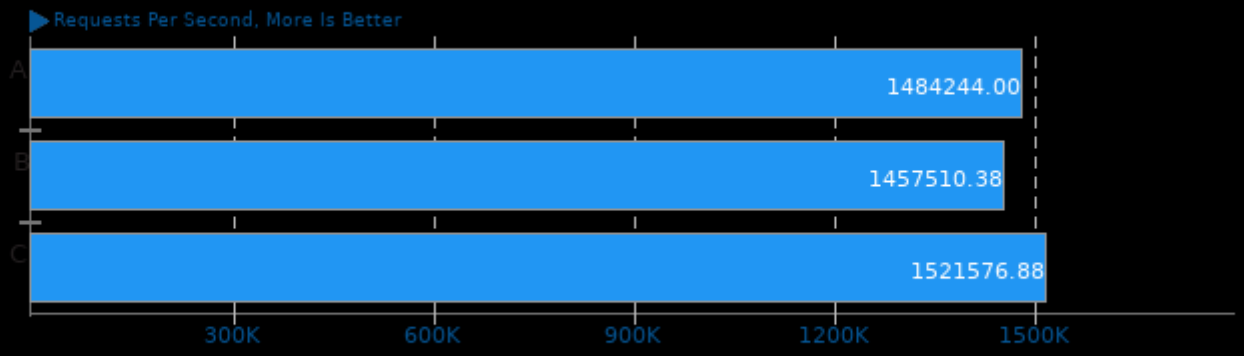

Redis 7.0.4
Test: SET - Parallel Connections: 500
Requests Per Second, More Is Better
A
1733854.50
B
1726908.00
C
1691619.75
400K
800K
1200K
1600K
2000K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: GET - Parallel Connections: 1000
Requests Per Second, More Is Better
A
2210557.00
B
2171455.75
C
1856685.25
500K
1000K
1500K
2000K
2500K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: LPOP - Parallel Connections: 500
Requests Per Second, More Is Better
A
2370160.00
B
1475823.38
C
1487059.75
500K
1000K
1500K
2000K
2500K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3


Redis 7.0.4
Test: LPUSH - Parallel Connections: 50
Requests Per Second, More Is Better
A
B
C
1904868.62
1453472.12
1886043.38
400K
800K
1200K
1600K
2000K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: SADD - Parallel Connections: 500
Requests Per Second, More Is Better
A
B
C
1961819.62
1776356.88
2037941.62
400K
800K
1200K
1600K
2000K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: SET - Parallel Connections: 1000
Requests Per Second, More Is Better
A
B
C
1661297.38
1514480.88
1929916.50
400K
800K
1200K
1600K
2000K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3


Redis 7.0.4
Test: LPOP - Parallel Connections: 1000
Requests Per Second, More Is Better
A
B
C
1448923.38
1555619.38
1468437.75
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: LPUSH - Parallel Connections: 500
Requests Per Second, More Is Better
A
B
C
1502250.25
1364318.25
1574137.25
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3
Redis 7.0.4
Test: SADD - Parallel Connections: 1000
Requests Per Second, More Is Better
A
B
C
2050488.88
1914214.88
1892380.50
400K
800K
1200K
1600K
2000K
1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3

Redis 7.0.4

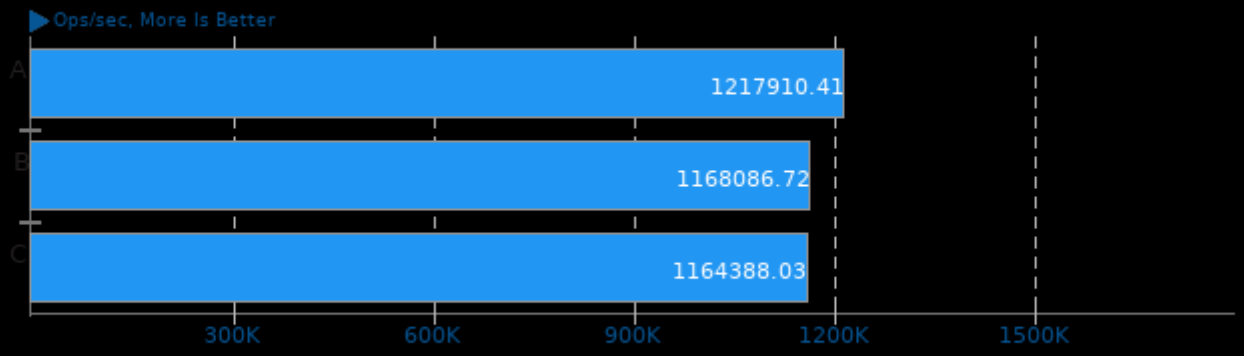
Test: LPUSH - Parallel Connections: 1000



1. (CXX) g++ options: -MM -MT -g3 -fvisibility=hidden -O3

memtier_benchmark 1.4

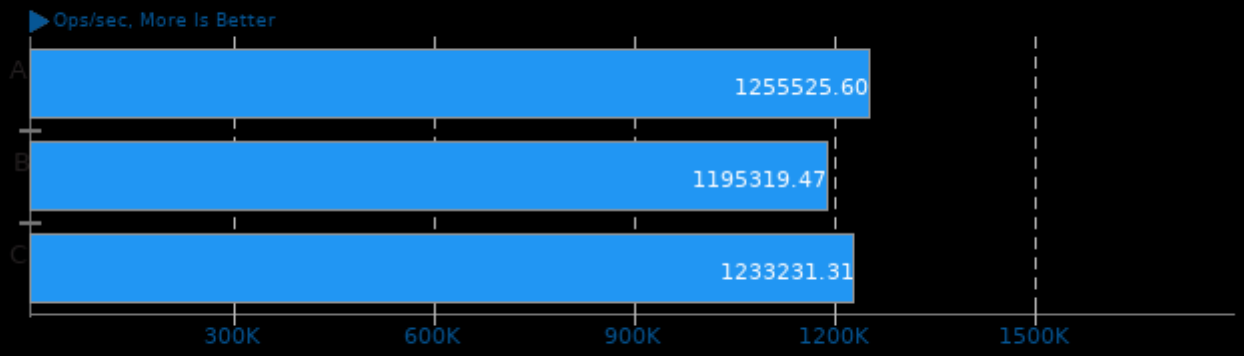
Protocol: Redis - Clients: 50 - Set To Get Ratio: 1:1



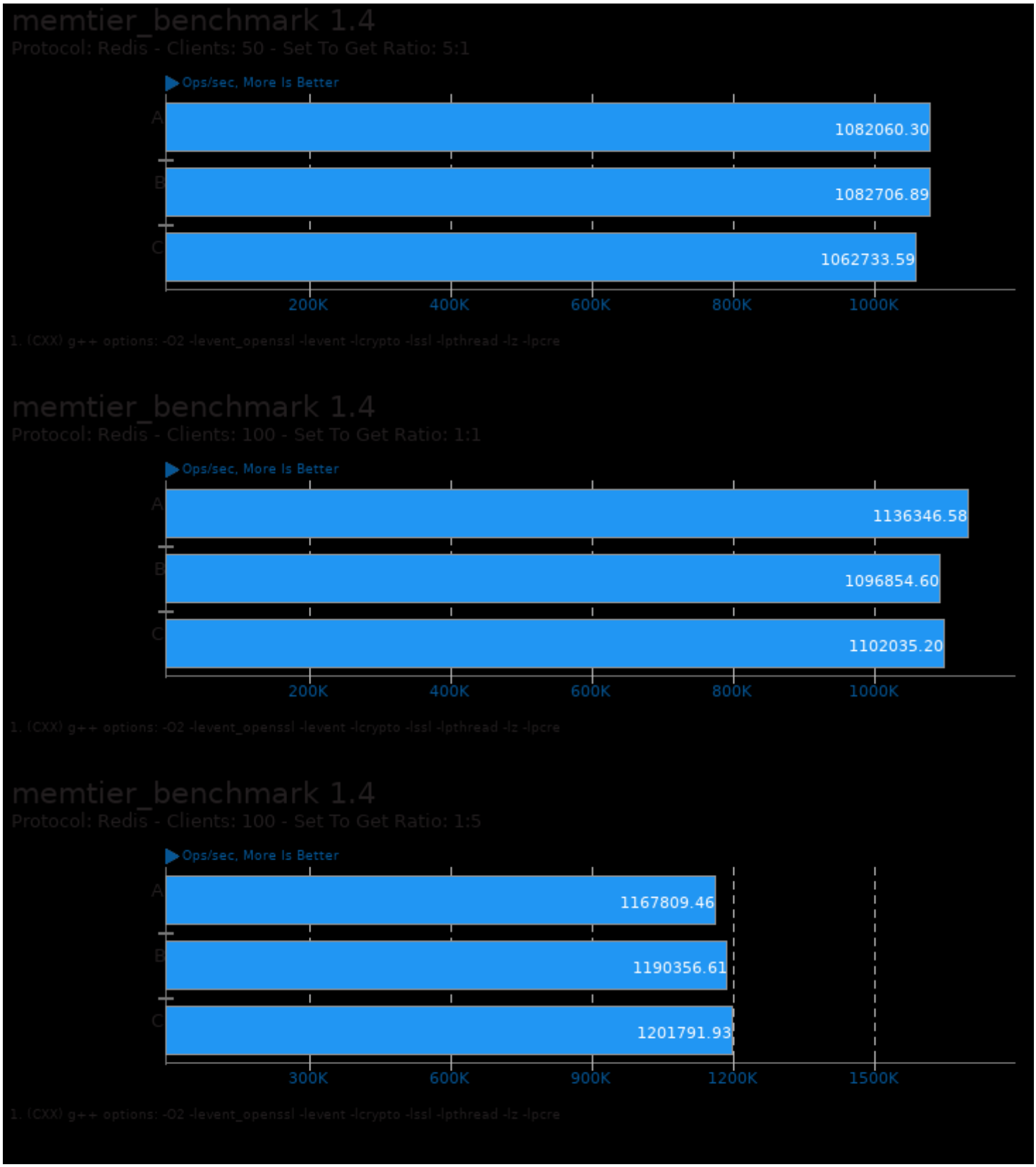
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre

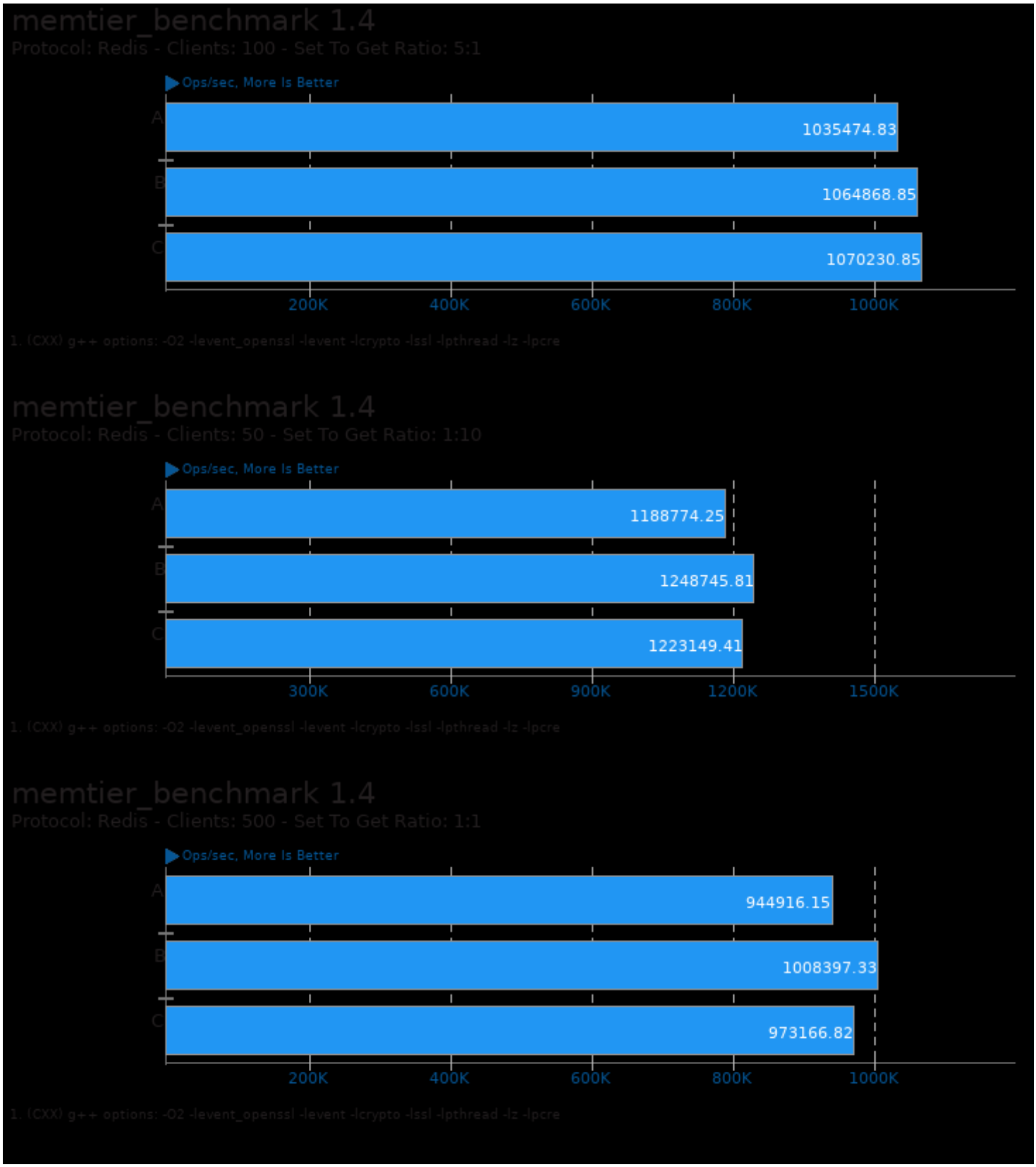
memtier_benchmark 1.4

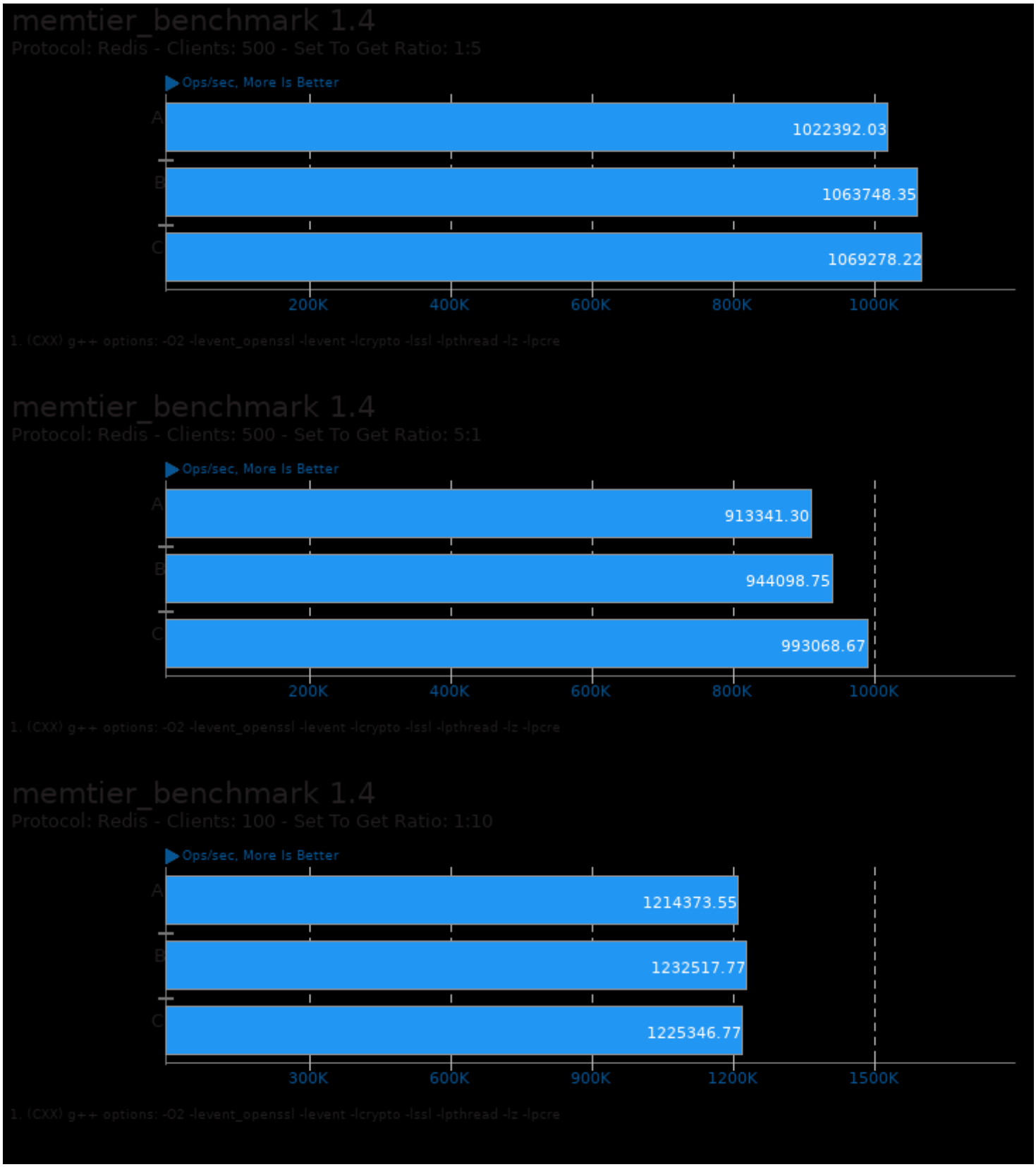
Protocol: Redis - Clients: 50 - Set To Get Ratio: 1:5



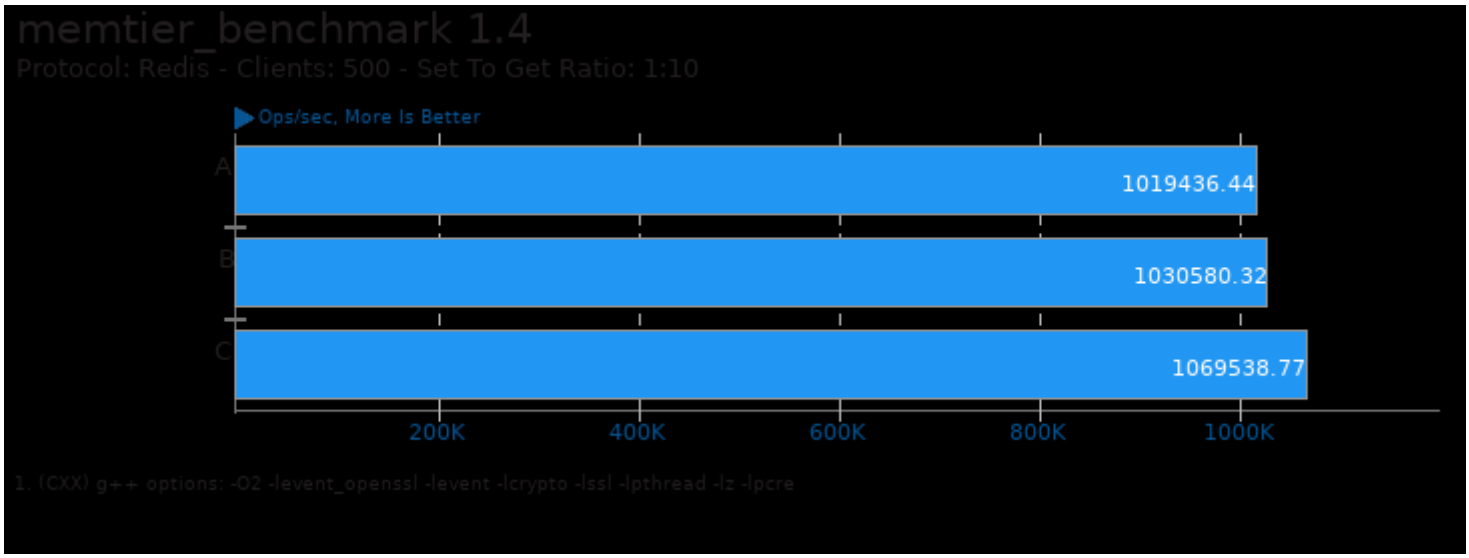
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre


memtier_benchmark 1.4
Protocol: Redis - Clients: 50 - Set To Get Ratio: 5:1
Ops/sec, More Is Better
A
1082060.30
B
1082706.89
C
1062733.59
200K
400K
600K
800K
1000K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
memtier_benchmark 1.4
Protocol: Redis - Clients: 100 - Set To Get Ratio: 1:1
Ops/sec, More Is Better
A
1136346.58
B
1096854.60
C
1102035.20
200K
400K
600K
800K
1000K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
memtier_benchmark 1.4
Protocol: Redis - Clients: 100 - Set To Get Ratio: 1:5
Ops/sec, More Is Better
A
1167809.46
B
1190356.61
C
1201791.93
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre


memtier_benchmark 1.4
Protocol: Redis - Clients: 100 - Set To Get Ratio: 5:1
Ops/sec, More Is Better
A
1035474.83
B
1064868.85
C
1070230.85
200K
400K
600K
800K
1000K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
memtier_benchmark 1.4
Protocol: Redis - Clients: 50 - Set To Get Ratio: 1:10
Ops/sec, More Is Better
A
1188774.25
B
1248745.81
C
1223149.41
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
memtier_benchmark 1.4
Protocol: Redis - Clients: 500 - Set To Get Ratio: 1:1
Ops/sec, More Is Better
A
944916.15
B
1008397.33
C
973166.82
200K
400K
600K
800K
1000K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre

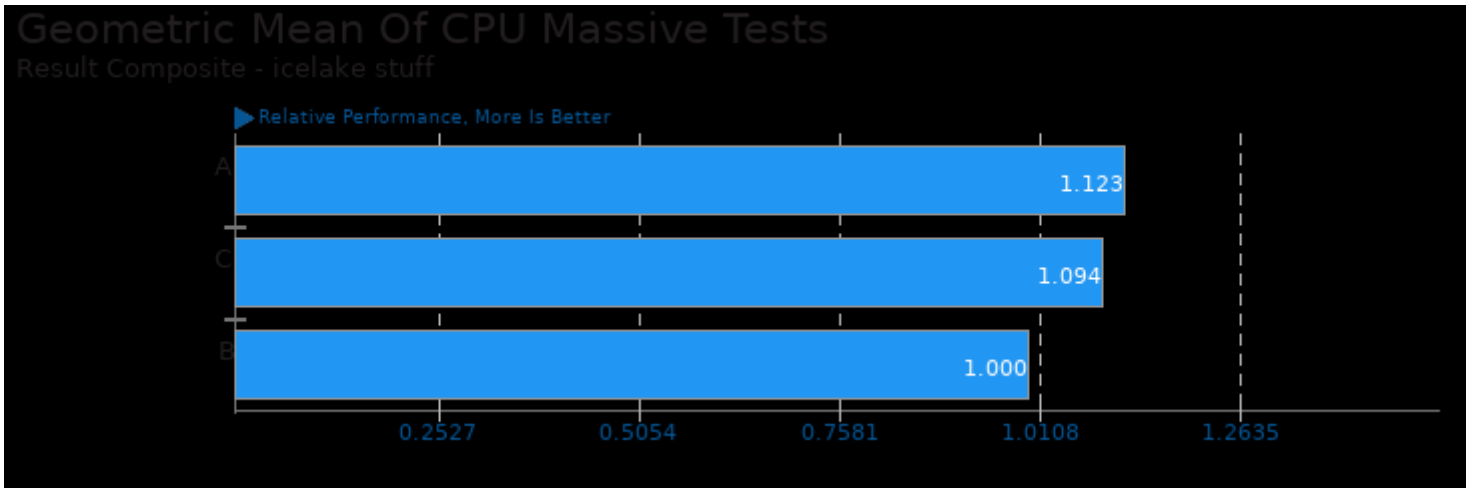

memtier_benchmark 1.4
Protocol: Redis - Clients: 500 - Set To Get Ratio: 1:5
Ops/sec, More Is Better
A
B
C
1022392.03
1063748.35
1069278.22
200K
400K
600K
800K
1000K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
memtier_benchmark 1.4
Protocol: Redis - Clients: 500 - Set To Get Ratio: 5:1
Ops/sec, More Is Better
A
B
C
913341.30
944098.75
993068.67
200K
400K
600K
800K
1000K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre
memtier_benchmark 1.4
Protocol: Redis - Clients: 100 - Set To Get Ratio: 1:10
Ops/sec, More Is Better
A
B
C
1214373.55
1232517.77
1225346.77
300K
600K
900K
1200K
1500K
1. (CXX) g++ options: -O2 -levent_openssl -levent -lcrypto -lssl -lpthread -lz -lpcre

icelake stuff



icelake stuff

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



Geometric mean based upon tests: pts/graphics-magick and pts/redis



Geometric mean based upon tests: pts/etcd, pts/redis, pts/dragonflydb and pts/memtier-benchmark



Geometric mean based upon tests: pts/memtier-benchmark, pts/redis, pts/etcd and pts/dragonflydb

This file was automatically generated via the Phoronix Test Suite benchmarking software on Thursday, 28 March 2024 05:10.