



[www.phoronix-test-suite.com](http://www.phoronix-test-suite.com)

## **Huawei TaiShan 2280 | Ubuntu 19.10, Ubuntu 20.04, openEuler 20.03**

Huawei TaiShan 2280 | Ubuntu 19.10, Ubuntu 20.04, openEuler 20.03 | Details: <https://servernews.ru/1010630/>

### **Automated Executive Summary**

*openEuler 20.03 had the most wins, coming in first place for 61% of the tests.*

*Based on the geometric mean of all complete results, the fastest (openEuler 20.03) was 1.05x the speed of the slowest (Ubuntu 19.10). Ubuntu 20.04 was 0.99x the speed of openEuler 20.03 and Ubuntu 19.10 was 0.96x the speed of Ubuntu 20.04.*

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

*Dbench (Client Count: 12) at 4.83x  
Dbench (Client Count: 6) at 4.03x  
Tinymembench (Standard Memset) at 3.18x  
CacheBench (Test: Write) at 1.97x  
Stream (Type: Scale) at 1.75x  
Mlpack Benchmark (Benchmark: scikit\_svm) at 1.65x  
PostMark (Disk Transaction Performance) at 1.61x  
OSBench (Test: Create Files) at 1.51x  
OSBench (Test: Memory Allocations) at 1.51x*

MBW (Test: Memory Copy - Array Size: 512 MiB) at 1.48x.

## Test Systems:

### Ubuntu 19.10

Processor: 2 x HiSilicon Kunpeng 920-7260 (64 Cores), Motherboard: Huawei BC82AMDD (1.05 BIOS), Chipset: Huawei HiSilicon, Memory: 16 x 32 GB DDR4-2933MT/s Samsung M393A4K40CB2-CVF, Disk: 2 x 1199GB HW-SAS3508 + 4796GB HW-SAS3508, Graphics: Huawei Hi1710 [iBMC Intelligent Management chip w/VGA support], Network: 8 x Huawei Hi1822 + 8 x Huawei HNS GE/10GE/25GE

OS: Ubuntu 19.10, Kernel: 5.3.0-46-generic (aarch64), Compiler: GCC 9.2.1 20191008, File-System: xfs, Screen Resolution: 640x480

Compiler Notes: --build=aarch64-linux-gnu --disable-libquadmath --disable-libquadmath-support --disable-werror --enable-bootstrap --enable-checking=release --enable-clocale-gnu --enable-default-pie --enable-fix-cortex-a53-843419 --enable-gnu-unique-object --enable-languages=c,ada,c++,go,d,fortran,objc,obj-c++,gm2 --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-multiarch --enable-nls --enable-plugin --enable-shared --enable-threads=posix --host=aarch64-linux-gnu --program-prefix=aarch64-linux-gnu- --target=aarch64-linux-gnu --with-default-libstdcxx-abi=new --with-gcc-major-version-only --with-target-system-zlib=auto -v  
Security Notes: itlb\_multihit: Not affected + l1tf: Not affected + mds: Not affected + meltdown: Not affected + spec\_store\_bypass: Vulnerable + spectre\_v1: Mitigation of \_\_user pointer sanitization + spectre\_v2: Not affected + tsx\_async\_abort: Not affected

### Ubuntu 20.04

Processor: 2 x HiSilicon Kunpeng 920-7260 (64 Cores), Motherboard: Huawei BC82AMDD (1.05 BIOS), Chipset: Huawei HiSilicon, Memory: 16 x 32 GB DDR4-2933MT/s Samsung M393A4K40CB2-CVF, Disk: 2 x 1199GB HW-SAS3508 + 4796GB HW-SAS3508, Graphics: Huawei Hi1710 [iBMC Intelligent Management chip w/VGA support], Network: 8 x Huawei Hi1822 + 8 x Huawei HNS GE/10GE/25GE

OS: Ubuntu 20.04, Kernel: 5.4.0-28-generic (aarch64), Compiler: GCC 9.3.0, File-System: xfs, Screen Resolution: 640x480

Compiler Notes: --build=aarch64-linux-gnu --disable-libquadmath --disable-libquadmath-support --disable-werror --enable-checking=release --enable-clocale-gnu --enable-default-pie --enable-fix-cortex-a53-843419 --enable-gnu-unique-object --enable-languages=c,ada,c++,go,d,fortran,objc,obj-c++,gm2 --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-multiarch --enable-nls --enable-plugin --enable-shared --enable-threads=posix --host=aarch64-linux-gnu --program-prefix=aarch64-linux-gnu- --target=aarch64-linux-gnu --with-default-libstdcxx-abi=new --with-gcc-major-version-only --with-target-system-zlib=auto -v  
Disk Notes: MQ\_DEADLINE / attr2,inode64,logbsize=32k,logbufs=8,noquota,relatime,rw,sunit=512,swidth=1024  
Java Notes: OpenJDK Runtime Environment (build 11.0.7+10-post-Ubuntu-3ubuntu1)  
Python Notes: Python 2.7.18rc1 + Python 3.8.2  
Security Notes: itlb\_multihit: Not affected + l1tf: Not affected + mds: Not affected + meltdown: Not affected + spec\_store\_bypass: Vulnerable + spectre\_v1: Mitigation of \_\_user pointer sanitization + spectre\_v2: Not affected + tsx\_async\_abort: Not affected

### openEuler 20.03

Processor: 2 x HiSilicon Kunpeng 920-7260 (64 Cores), Motherboard: Huawei BC82AMDD (1.05 BIOS), Chipset: Huawei HiSilicon, Memory: 16 x 32 GB DDR4-2933MT/s Samsung M393A4K40CB2-CVF, Disk: 2 x 1199GB HW-SAS3508 + 4796GB HW-SAS3508, Graphics: Huawei Hi1710 [iBMC Intelligent Management chip w/VGA support], Network: 8 x Huawei Hi1822 + 8 x Huawei HNS GE/10GE/25GE

OS: openEuler 20.03, Kernel: 4.19.90-2003.4.0.0036.oe1.aarch64 (aarch64) 20200323, Compiler: GCC 7.3.0, File-System: xfs, Screen Resolution: 640x480

Compiler Notes: --build=aarch64-linux-gnu --disable-libgcj --disable-libunwind-exceptions --enable-\_\_cxa\_atexit --enable-checking=release --enable-gnu-indirect-function

```
--enable-gnu-unique-object --enable-initfini-array --enable-languages=c,c++,objc,obj-c++,fortran,ito --enable-plugin --enable-shared --enable-threads=posix
--mandir=/usr/share/man --with-boot-ldflags=' --with-linker-hash-style=gnu --with-multilib-list=lp64 --with-stage1-ldflags=' --without-cloog --without-isl -Wl,-z,relro,-z,now'
-Disk Notes: BFQ / attr2,inode64,noquota,relatime,rw,seclabel,sunit=512,swidth=1024
-Java Notes: OpenJDK Runtime Environment (build 1.8.0_242-b08)
-Python Notes: Python 2.7.16 + Python 3.7.4
```

```
Security Notes: SELinux + itlb_multihit: Not affected + l1tf: Not affected + mds: Not affected + meltdown: Not affected + spec_store_bypass: Vulnerable + spectre_v1:
Mitigation of __user pointer sanitization + spectre_v2: Not affected + tsx_async_abort: Not affected
```

	Ubuntu 19.10	Ubuntu 20.04	openEuler 20.03
<b>High Performance Conjugate Gradient (GFLOP/s)</b>	<b>29.6005</b>	29.5027	<b>28.8991</b>
Normalized	100%	99.67%	97.63%
Standard Deviation	1.1%	1.4%	6.4%
<b>GNU MPC - M.P.B (Global Score)</b>	<b>4603</b>	4607	<b>4833</b>
Normalized	95.24%	95.32%	100%
Standard Deviation	0.3%	0.1%	0.1%
<b>Rodinia - OpenMP LavaMD (sec)</b>	<b>8.157</b>	7.778	<b>7.391</b>
Normalized	90.61%	95.02%	100%
Standard Deviation	1.3%	2.4%	0.6%
<b>Rodinia - OpenMP CFD Solver (sec)</b>	<b>16.416</b>	14.222	<b>13.071</b>
Normalized	79.62%	91.91%	100%
Standard Deviation	9.8%	26.8%	5%
<b>Rodinia - O.S (sec)</b>	16.267	<b>15.933</b>	<b>18.405</b>
Normalized	97.95%	100%	86.57%
Standard Deviation	5.3%	6.5%	1%
<b>Coremark - CoreMark Size 666 - I.P.S (Iterations/Sec)</b>	<b>1916064</b>	2190153	<b>2248994</b>
Normalized	85.2%	97.38%	100%
Standard Deviation	4.7%	1.9%	5.1%
<b>Stockfish - Total Time (Nodes/s)</b>	151586151	<b>149367032</b>	<b>169954198</b>
Normalized	89.19%	87.89%	100%
Standard Deviation	1.2%	1.4%	1.2%
<b>RAMspeed SMP - Add - Integer (MB/s)</b>	23079	<b>21889</b>	<b>23116</b>
Normalized	99.84%	94.69%	100%
Standard Deviation	0.6%	1.1%	1.5%
<b>RAMspeed SMP - Copy - Integer (MB/s)</b>	<b>22782</b>	21093	<b>20706</b>
Normalized	100%	92.59%	90.89%
Standard Deviation	0.5%	0.3%	1.3%
<b>RAMspeed SMP - Scale - Integer (MB/s)</b>	13876	<b>13543</b>	<b>15782</b>
Normalized	87.92%	85.81%	100%
Standard Deviation	0.4%	0.6%	0.5%
<b>RAMspeed SMP - Triad - Integer (MB/s)</b>	18378	<b>18531</b>	<b>15817</b>
Normalized	99.17%	100%	85.36%
Standard Deviation	4.4%	2.5%	0.8%
<b>RAMspeed SMP - Average - Integer (MB/s)</b>	<b>19600</b>	<b>18676</b>	19002
Normalized	100%	95.29%	96.95%
Standard Deviation	1.4%	1.1%	0.1%
<b>RAMspeed SMP - Add - Floating Point (MB/s)</b>	<b>23303</b>	<b>22046</b>	22872
Normalized	100%	94.61%	98.15%
Standard Deviation	1.5%	0.4%	0.3%
<b>RAMspeed SMP - Copy - Floating Point</b>	<b>22497</b>	21038	<b>20541</b>
Normalized	100%	93.52%	91.31%

<b>RAMspeed SMP - Scale - Floating Point</b>	Standard Deviation Normalized	2.8% 95.46%	0.1% 97.16%	0.9% 100%
<b>RAMspeed SMP - Triad - Floating Point</b>	Standard Deviation Normalized	1.4% 82.34%	0.1% 83.26%	0.1% 100%
<b>RAMspeed SMP - Average - Floating Point</b>	Standard Deviation (MB/s)	0.5% 97.6%	1.7% 94.22%	0% 100%
<b>Stream - Copy (MB/s)</b>	Normalized Standard Deviation	212013 0.6%	<b>213498</b> 0.4%	<b>129799</b> 0.7%
<b>Stream - Scale (MB/s)</b>	Normalized Standard Deviation	<b>231955</b> 12.2%	230358 13.3%	<b>132901</b> 21%
<b>Stream - Triad (MB/s)</b>	Normalized Standard Deviation	214743 100%	<b>224687</b> 99.31%	<b>155296</b> 57.3%
<b>Stream - Add (MB/s)</b>	Normalized Standard Deviation	220225 12.9%	<b>227186</b> 4.3%	<b>168356</b> 2.8%
<b>Tinymembench - Standard Memcpy (MB/s)</b>	Normalized Standard Deviation	<b>5624</b> 96.94%	5872 9%	<b>6278</b> 21.9%
<b>Tinymembench - Standard Memset (MB/s)</b>	Normalized Standard Deviation	<b>11570</b> 89.59%	12530 93.54%	<b>36749</b> 0.4%
<b>MBW - Memory Copy - 128 MiB (MiB/s)</b>	Normalized Standard Deviation	6169 31.48%	<b>5964</b> 34.1%	<b>7024</b> 100%
<b>MBW - Memory Copy - 512 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>4674</b> 0.1%	6395 0.2%	<b>6938</b> 1.8%
<b>MBW - Memory Copy - 1024 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>4950</b> 0.1%	6250 0.2%	<b>6960</b> 0.2%
<b>MBW - Memory Copy - 4096 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>4978</b> 0.5%	6299 1%	<b>6987</b> 0.6%
<b>MBW - Memory Copy - 8192 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>4941</b> 0.2%	6335 0.1%	<b>6932</b> 0.3%
<b>MBW - M.C.F.B.S - 128 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>6152</b> 71.12%	5671 90.15%	<b>6556</b> 100%
<b>MBW - M.C.F.B.S - 512 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>4683</b> 0.7%	5751 0.9%	<b>6575</b> 0.3%
<b>MBW - M.C.F.B.S - 1024 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>4881</b> 0.4%	5705 0.2%	<b>6578</b> 0.7%
<b>MBW - M.C.F.B.S - 4096 MiB (MiB/s)</b>	Normalized Standard Deviation	<b>4969</b> 74.2%	5649 86.73%	<b>6539</b> 0.1%
	Normalized	75.99%	86.4%	100%

<b>MBW - M.C.F.B.S - 8192 MiB (MiB/s)</b>	<b>4784</b>	0.1%	2.5%
Normalized	72.46%	85.61%	100%
Standard Deviation	5%	0.1%	1.3%
<b>CacheBench - Read (MB/s)</b>	<b>4957</b>	<b>4956</b>	4956
Normalized	100%	99.99%	99.99%
Standard Deviation	0%	0%	0%
<b>CacheBench - Write (MB/s)</b>	<b>9772</b>	<b>9770</b>	<b>19244</b>
Normalized	50.78%	50.77%	100%
Standard Deviation	0%	0%	0%
<b>CacheBench - R.M.W (MB/s)</b>	<b>29801</b>	29806	<b>29826</b>
Normalized	99.92%	99.93%	100%
Standard Deviation	0%	0%	0%
<b>Sockperf - Throughput (Messages/sec)</b>	291853	<b>255363</b>	<b>308126</b>
Normalized	94.72%	82.88%	100%
Standard Deviation	2.9%	3%	2.6%
<b>Sockperf - Latency Ping Pong (usec)</b>	6.421	<b>5.215</b>	<b>6.614</b>
Normalized	81.22%	100%	78.85%
Standard Deviation	7.8%	0.4%	12.2%
<b>Loopback TCP Network Performance - T.T.T.V.L (sec)</b>	<b>16.750</b>	16.380	<b>14.425</b>
Normalized	86.12%	88.06%	100%
Standard Deviation	9.4%	1.1%	7.1%
<b>Timed Linux Kernel Compilation - Time To Compile (sec)</b>	79.165	<b>85.216</b>	<b>68.567</b>
Normalized	86.61%	80.46%	100%
Standard Deviation	1.6%	1.9%	0.8%
<b>Timed LLVM Compilation - Time To Compile (sec)</b>	300.455	<b>335.055</b>	<b>253.595</b>
Normalized	84.4%	75.69%	100%
Standard Deviation	0.3%	1.5%	0.6%
<b>C-Ray - Total Time - 4.1.R.P.P (sec)</b>	14.279	<b>14.244</b>	<b>14.308</b>
Normalized	99.75%	100%	99.55%
Standard Deviation	0.4%	0.9%	0.4%
<b>POV-Ray - Trace Time (sec)</b>	<b>24.821</b>	24.496	<b>22.475</b>
Normalized	90.55%	91.75%	100%
Standard Deviation	5%	5.2%	3.2%
<b>Smallpt - G.I.R.1.S (sec)</b>	2.903	<b>3.046</b>	<b>2.597</b>
Normalized	89.46%	85.26%	100%
Standard Deviation	4%	4.7%	0.3%
<b>Sysbench - Memory (Events/sec)</b>	7750721	<b>7783172</b>	<b>6049767</b>
Normalized	99.58%	100%	77.73%
Standard Deviation	0.5%	1.5%	1.7%
<b>Sysbench - CPU (Events/sec)</b>	415977	<b>402083</b>	<b>418474</b>
Normalized	99.4%	96.08%	100%
Standard Deviation	0.2%	1.8%	0.5%
<b>Sunflow Rendering System - G.I.I.S (sec)</b>	<b>2.678</b>	2.662	<b>0.906</b>
Normalized	33.83%	34.03%	100%
Standard Deviation	20.3%	10.3%	16.1%
<b>OSBench - Create Files (us/Event)</b>	<b>54.384858</b>	55.085366	<b>82.322768</b>
Normalized	100%	98.73%	66.06%
Standard Deviation	0.7%	1.3%	0.3%
<b>OSBench - Create Threads (us/Event)</b>	<b>28.223991</b>	26.253859	<b>20.967325</b>
Normalized	74.29%	79.86%	100%
Standard Deviation	2.3%	1.6%	0.6%

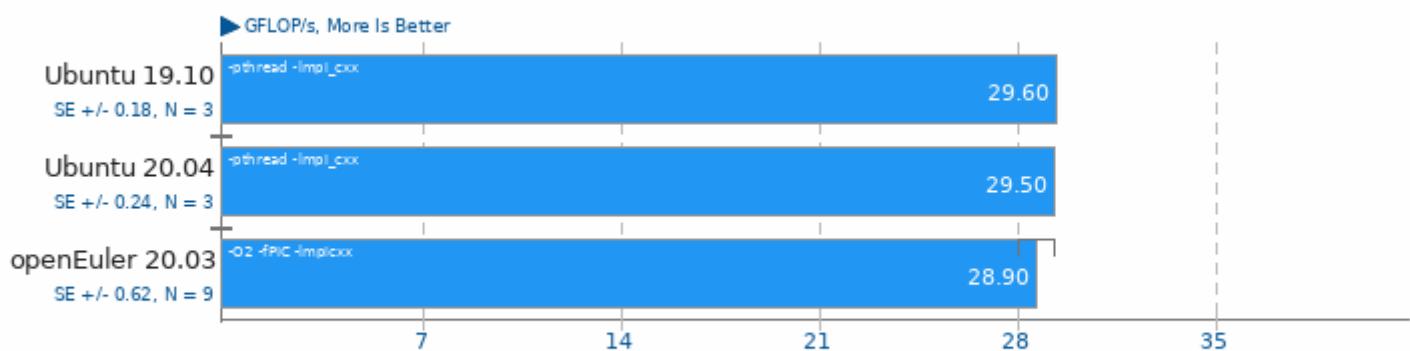
<b>OSBench - Launch Programs (us/Event)</b>	<b>67.972342</b>	64.999263	<b>63.133240</b>
Normalized	92.88%	97.13%	100%
Standard Deviation	0.2%	1.4%	1.9%
<b>OSBench - Create Processes (us/Event)</b>	<b>57.847500</b>	55.907568	<b>51.514506</b>
Normalized	89.05%	92.14%	100%
Standard Deviation	1.4%	2.4%	2.7%
<b>OSBench - Memory Allocations (Ns/Event)</b>	134.985745	<b>134.991408</b>	<b>89.430253</b>
Normalized	66.25%	66.25%	100%
Standard Deviation	2.6%	1.2%	0.7%
<b>SQLite - 1 (sec)</b>	3.513	<b>3.690</b>	<b>3.311</b>
Normalized	94.25%	89.73%	100%
Standard Deviation	0.7%	0.2%	0.6%
<b>SQLite - 8 (sec)</b>	13.720	<b>6.360</b>	<b>31.444</b>
Normalized	46.36%	100%	20.23%
Standard Deviation	7.4%	0.5%	15.8%
<b>SQLite - 32 (sec)</b>	<b>74.878</b>	<b>24.066</b>	56.711
Normalized	32.14%	100%	42.44%
Standard Deviation	2.9%	1.2%	9.2%
<b>SQLite - 64 (sec)</b>	99.172	<b>55.645</b>	<b>102.054</b>
Normalized	56.11%	100%	54.53%
Standard Deviation	11.4%	2.8%	6.7%
<b>SQLite - 128 (sec)</b>	<b>271.346</b>	<b>113.415</b>	234.644
Normalized	41.8%	100%	48.33%
Standard Deviation	2.8%	0.6%	7%
<b>Dbench - 1 (MB/s)</b>	636.274	<b>631.767</b>	<b>653.282</b>
Normalized	97.4%	96.71%	100%
Standard Deviation	0.3%	0.2%	0.2%
<b>Dbench - 6 (MB/s)</b>	<b>2917</b>	2837	<b>722.959</b>
Normalized	100%	97.27%	24.79%
Standard Deviation	0.7%	2.1%	1.2%
<b>Dbench - 12 (MB/s)</b>	<b>4749</b>	4621	<b>982.951</b>
Normalized	100%	97.3%	20.7%
Standard Deviation	1.6%	1.8%	0.6%
<b>Dbench - 48 (MB/s)</b>	<b>4849</b>	4925	<b>6738</b>
Normalized	71.96%	73.08%	100%
Standard Deviation	1.4%	1.6%	1.1%
<b>Dbench - 128 (MB/s)</b>	<b>4056</b>	4567	<b>5927</b>
Normalized	68.44%	77.06%	100%
Standard Deviation	0.4%	0.3%	0.4%
<b>Dbench - 256 (MB/s)</b>	<b>4201</b>	4486	<b>5813</b>
Normalized	72.28%	77.17%	100%
Standard Deviation	0.1%	0.3%	0.3%
<b>PostMark - D.T.P (TPS)</b>	3488	<b>3440</b>	<b>5525</b>
Normalized	63.13%	62.26%	100%
Standard Deviation	0.8%	0.8%	4.6%
<b>Java SciMark - Composite (Mflops)</b>	<b>1254</b>	<b>1259</b>	1257
Normalized	99.62%	100%	99.84%
Standard Deviation	0.1%	0.1%	2.8%
<b>Java SciMark - Monte Carlo (Mflops)</b>	<b>804.49</b>	803.41	<b>790.39</b>
Normalized	100%	99.87%	98.25%
Standard Deviation	0.8%	0.5%	0.3%
<b>Java SciMark - F.F.T (Mflops)</b>	<b>935.19</b>	933.88	<b>912.86</b>
Normalized	100%	99.86%	97.61%
Standard Deviation	0.1%	0.3%	0.2%
<b>Java SciMark - S.M.M (Mflops)</b>	964.92	<b>962.53</b>	<b>1047</b>

Normalized	92.16%	91.93%	100%
Standard Deviation	0.2%	0.2%	0%
<b>Java SciMark - D.L.M.F (Mflops)</b>	<b>2821</b>	<b>2849</b>	<b>2791</b>
Normalized	99.04%	100%	97.96%
Standard Deviation	0.1%	0.1%	6.3%
<b>Java SciMark - J.S.O.R (Mflops)</b>	<b>746.09</b>	<b>747.32</b>	<b>745.15</b>
Normalized	99.84%	100%	99.71%
Standard Deviation	0.1%	0%	0%
<b>Bork File Encrypter - F.E.T (sec)</b>	<b>13.910</b>	<b>14.198</b>	<b>13.155</b>
Normalized	94.57%	92.65%	100%
Standard Deviation	0.8%	2%	0.8%
<b>DaCapo Benchmark - H2 (msec)</b>	<b>5743</b>	<b>5601</b>	<b>5414</b>
Normalized	94.27%	96.66%	100%
Standard Deviation	4.1%	2.7%	3.9%
<b>DaCapo Benchmark - Jython (msec)</b>	<b>7535</b>	<b>7463</b>	<b>5771</b>
Normalized	76.59%	77.33%	100%
Standard Deviation	5.5%	5.4%	1.1%
<b>DaCapo Benchmark - Tradesoap (msec)</b>	<b>5647</b>	<b>5356</b>	<b>5201</b>
Normalized	92.1%	97.11%	100%
Standard Deviation	1.9%	2.9%	2.3%
<b>DaCapo Benchmark - Tradebeans (msec)</b>	<b>6339</b>	<b>6249</b>	<b>4466</b>
Normalized	70.45%	71.47%	100%
Standard Deviation	1.2%	2.9%	3.5%
<b>Botan - KASUMI (MiB/s)</b>	<b>51.220</b>	<b>51.231</b>	<b>52.562</b>
Normalized	97.45%	97.47%	100%
Standard Deviation	0%	0%	0.1%
<b>Botan - AES-256 (MiB/s)</b>	<b>2428</b>	<b>2432</b>	<b>2432</b>
Normalized	99.8%	99.97%	100%
Standard Deviation	0.1%	0%	0.1%
<b>Botan - Twofish (MiB/s)</b>	<b>180.092</b>	<b>180.267</b>	<b>173.886</b>
Normalized	99.9%	100%	96.46%
Standard Deviation	0.1%	0.1%	0%
<b>Botan - Blowfish (MiB/s)</b>	<b>219.029</b>	<b>218.950</b>	<b>225.671</b>
Normalized	97.06%	97.02%	100%
Standard Deviation	0.1%	0%	0%
<b>Botan - CAST-256 (MiB/s)</b>	<b>82.317</b>	<b>82.328</b>	<b>82.143</b>
Normalized	99.99%	100%	99.78%
Standard Deviation	0.1%	0%	0%
<b>John The Ripper - Blowfish (Real C/S)</b>	<b>47923</b>	<b>52918</b>	<b>72764</b>
Normalized	65.86%	72.73%	100%
Standard Deviation	25.8%	21.1%	0.6%
<b>John The Ripper - MD5 (Real C/S)</b>	<b>1506933</b>	<b>1634833</b>	<b>1762800</b>
Normalized	85.49%	92.74%	100%
Standard Deviation	15.8%	11.6%	2.7%
<b>dav1d - Chimera 1080p (FPS)</b>	<b>555.10</b>	<b>546.82</b>	<b>616.59</b>
Normalized	90.03%	88.68%	100%
Standard Deviation	1.5%	1.1%	0.5%
<b>dav1d - Summer Nature 4K (FPS)</b>	<b>213.56</b>	<b>215.86</b>	<b>252.63</b>
Normalized	84.53%	85.45%	100%
Standard Deviation	0.7%	1.3%	0.4%
<b>dav1d - S.N.1 (FPS)</b>	<b>639.50</b>	<b>639.72</b>	<b>729.01</b>
Normalized	87.72%	87.75%	100%
Standard Deviation	2.6%	1.2%	1.2%
<b>dav1d - C.1.1.b (FPS)</b>	<b>298.69</b>	<b>296.69</b>	<b>336.66</b>
Normalized	88.72%	88.13%	100%

	Standard Deviation	1.1%	0.8%	0.8%
<b>VP9 libvpx Encoding - Speed 0 (FPS)</b>	1.55	<b>1.54</b>	<b>1.66</b>	
Normalized	93.37%	92.77%	100%	
Standard Deviation	0.7%	0.4%	0.9%	
<b>VP9 libvpx Encoding - Speed 5 (FPS)</b>	<b>5.73</b>	<b>5.79</b>	<b>6.26</b>	
Normalized	91.53%	92.49%	100%	
Standard Deviation	1.1%	0.9%	0.7%	
<b>x264 - H.2.V.E (FPS)</b>	<b>132.03</b>	132.21	<b>150.92</b>	
Normalized	87.48%	87.6%	100%	
Standard Deviation	2.8%	4.9%	1.7%	
<b>7-Zip Compression - C.S.T (MIPS)</b>	195030	<b>193778</b>	<b>225882</b>	
Normalized	86.34%	85.79%	100%	
Standard Deviation	0.6%	1.7%	0.6%	
<b>Node.js Octane Benchmark (Score)</b>	<b>22276</b>	22500	<b>23337</b>	
Normalized	95.45%	96.41%	100%	
Standard Deviation	0.5%	1.8%	1.2%	
<b>Gzip Compression - L.S.T.A.T.t.g (sec)</b>	61.312	<b>61.340</b>	<b>54.323</b>	
Normalized	88.6%	88.56%	100%	
Standard Deviation	0.3%	0.3%	0.5%	
<b>XZ Compression - C.u.1.0.3.s.i.i.C.L.9 (sec)</b>	<b>32.040</b>	31.323	<b>25.625</b>	
Normalized	79.98%	81.81%	100%	
Standard Deviation	4.5%	4.4%	0.8%	
<b>Zstd Compression - C.u.1.0.3.s.i.i.C.L.1 (sec)</b>	14.631	<b>14.896</b>	<b>13.664</b>	
Normalized	93.39%	91.73%	100%	
Standard Deviation	2.9%	3.6%	1.1%	
<b>FLAC Audio Encoding - WAV To FLAC (sec)</b>	<b>41.785</b>	41.921	<b>45.437</b>	
Normalized	100%	99.68%	91.96%	
Standard Deviation	0.1%	0.1%	0.1%	
<b>LAME MP3 Encoding - WAV To MP3 (sec)</b>	<b>14.365</b>	14.410	<b>14.919</b>	
Normalized	100%	99.69%	96.29%	
Standard Deviation	0.1%	0.1%	0%	
<b>OpenSSL - R.4.b.P (Signs/sec)</b>	<b>14806</b>	<b>15037</b>	14993	
Normalized	98.47%	100%	99.71%	
Standard Deviation	2.8%	0.2%	1.7%	
<b>SQLite Speedtest - Timed Time - Size 1,000 (sec)</b>	124.875	<b>126.848</b>	<b>109.101</b>	
Normalized	87.37%	86.01%	100%	
Standard Deviation	0.9%	1.3%	0.1%	
<b>Memtier_benchmark - Redis (Ops/sec)</b>	<b>1647523</b>	<b>1316355</b>	1517267	
Normalized	100%	79.9%	92.09%	
Standard Deviation	2.2%	0.1%	8%	
<b>Redis - LPOP (Reqs/sec)</b>	<b>1604865</b>	1550925	<b>1508298</b>	
Normalized	100%	96.64%	93.98%	
Standard Deviation	4.7%	2.3%	0.2%	
<b>Redis - SADD (Reqs/sec)</b>	<b>1304919</b>	1228154	<b>1216119</b>	
Normalized	100%	94.12%	93.2%	
Standard Deviation	0.2%	2.8%	0.9%	
<b>Redis - LPUSH (Reqs/sec)</b>	827345	<b>812331</b>	<b>868863</b>	
Normalized	95.22%	93.49%	100%	
Standard Deviation	3.7%	1.9%	1%	
<b>Redis - GET (Reqs/sec)</b>	<b>1504542</b>	1457759	<b>1403510</b>	
Normalized	100%	96.89%	93.28%	
Standard Deviation	0.5%	0.6%	1.9%	
<b>Redis - SET (Reqs/sec)</b>	1083827	<b>1084803</b>	<b>1053800</b>	
Normalized	99.91%	100%	97.14%	

	Standard Deviation	2.3%	2.9%	0.9%
<b>Apache Benchmark - S.W.P.S (Reqs/sec)</b>	<b>14162</b>	13790	<b>13007</b>	
	Normalized	100%	97.37%	91.84%
	Standard Deviation	0.5%	0.3%	3.3%
<b>PHPBench - P.B.S (Score)</b>	<b>293810</b>	<b>357887</b>	<b>278937</b>	
	Normalized	82.1%	100%	77.94%
	Standard Deviation	0.2%	0.1%	0.3%
<b>Milpack Benchmark - scikit_ica (sec)</b>	<b>148.94</b>	193.79	<b>287.70</b>	
	Normalized	100%	76.86%	51.77%
	Standard Deviation	24.3%	32.6%	
<b>Milpack Benchmark - scikit_svm (sec)</b>	<b>37.09</b>	<b>61.06</b>	37.88	
	Normalized	100%	60.74%	97.91%
	Standard Deviation	0.4%	1.7%	1.2%
<b>Milpack Benchmark - scikit_linearridge_regression (sec)</b>	<b>5.18</b>	5.24	<b>7.99</b>	
	Normalized	100%	98.85%	64.83%
	Standard Deviation	10.2%	8.1%	32.3%

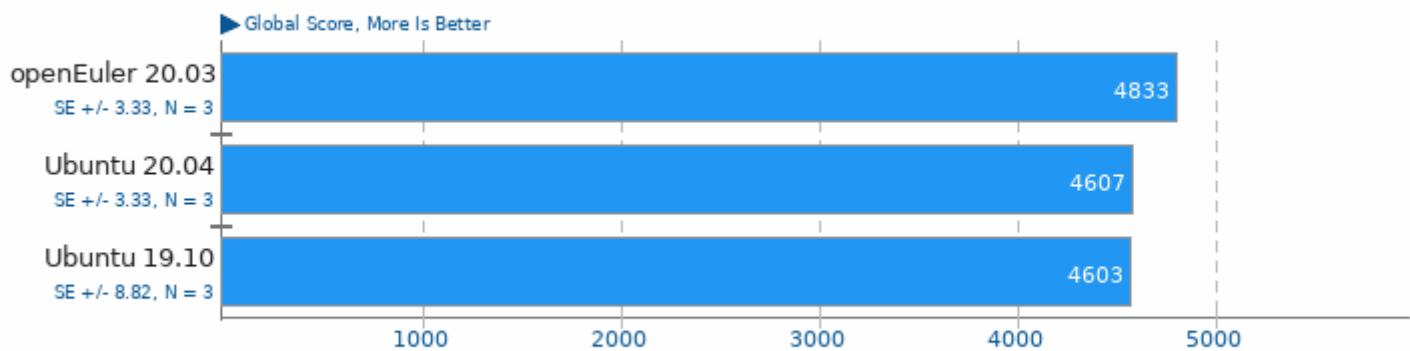
## High Performance Conjugate Gradient v3.1



1. (CXX) g++ options: -O3 -ffast-math -fno-vectorize -fmpic

## GNU MPC v1.1.0

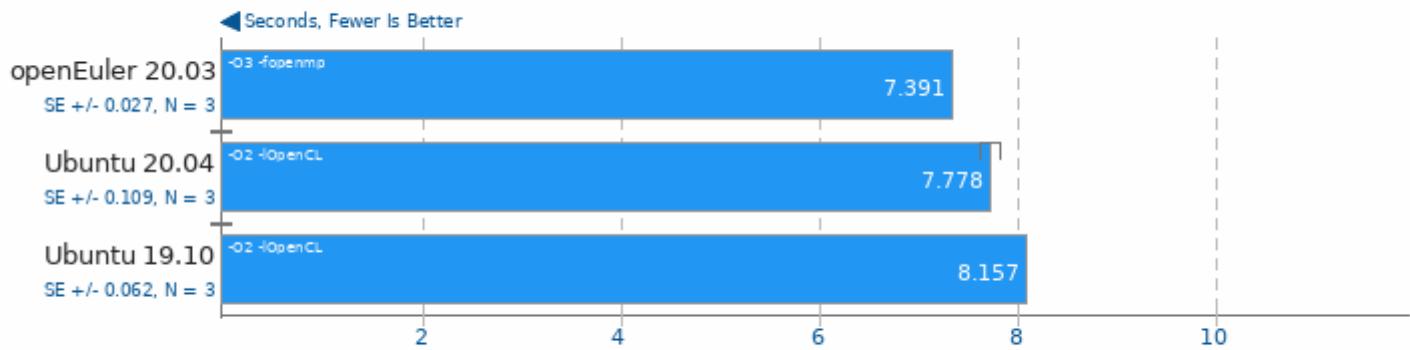
Multi-Precision Benchmark



1. (CC) gcc options: -lm -O2 -pedantic -march=armv8-a -MT -MD -MP -MF

## Rodinia v2.4

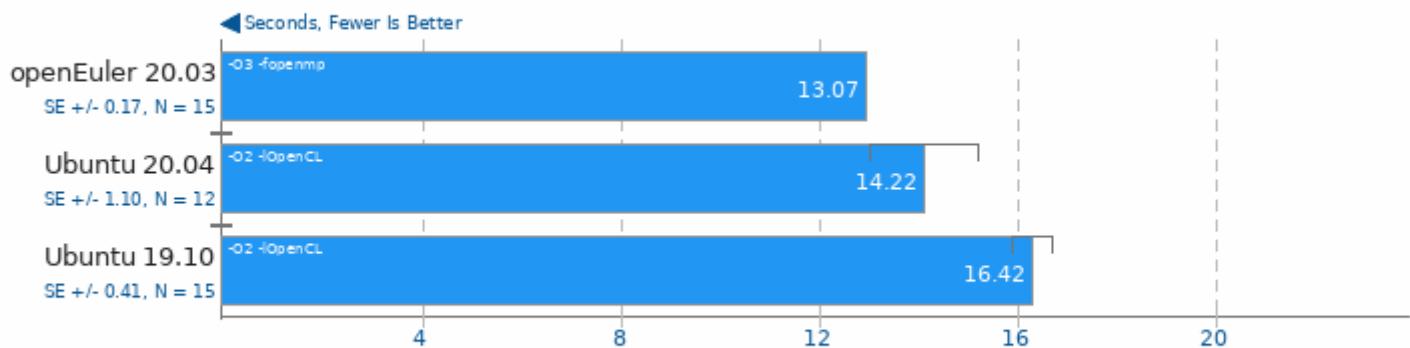
Test: OpenMP LavaMD



1. (CXX) g++ options:

## Rodinia v2.4

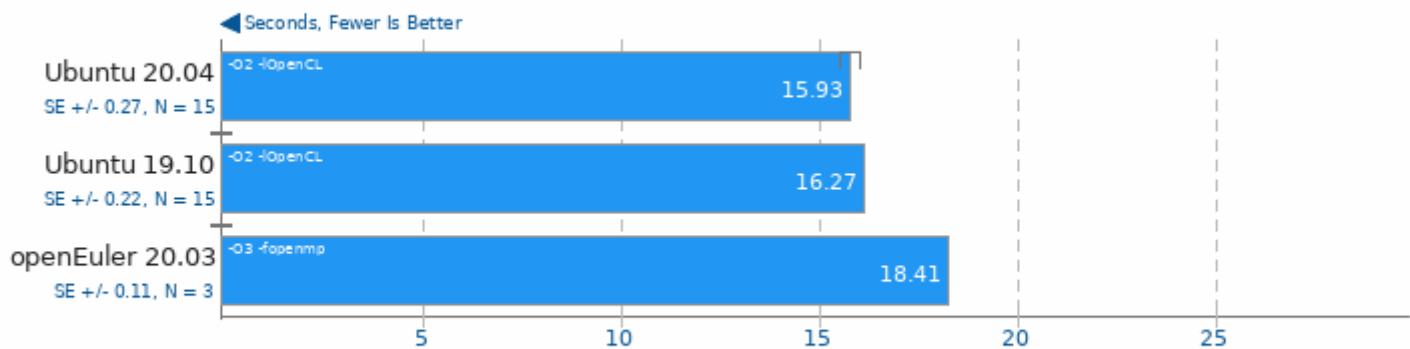
Test: OpenMP CFD Solver



1. (CXX) g++ options:

## Rodinia v2.4

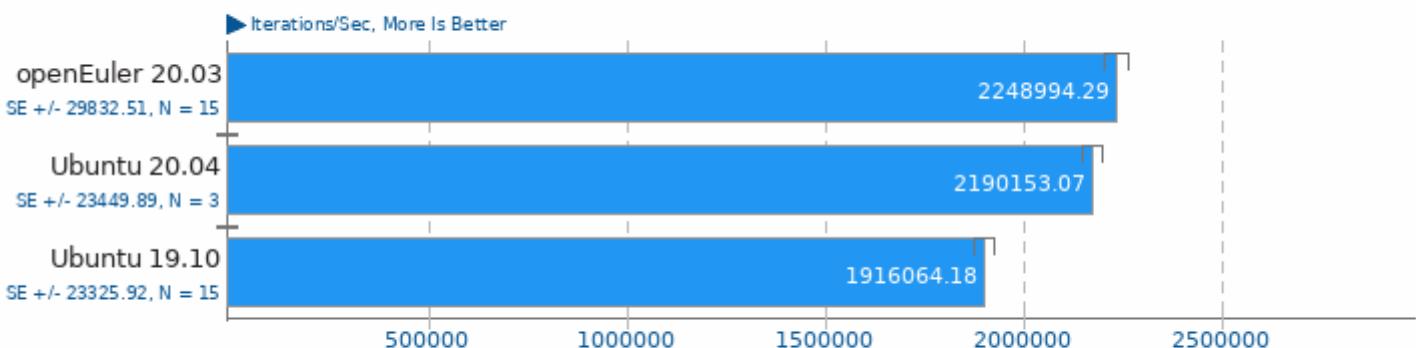
Test: OpenMP Streamcluster



1. (CXX) g++ options:

## Coremark v1.0

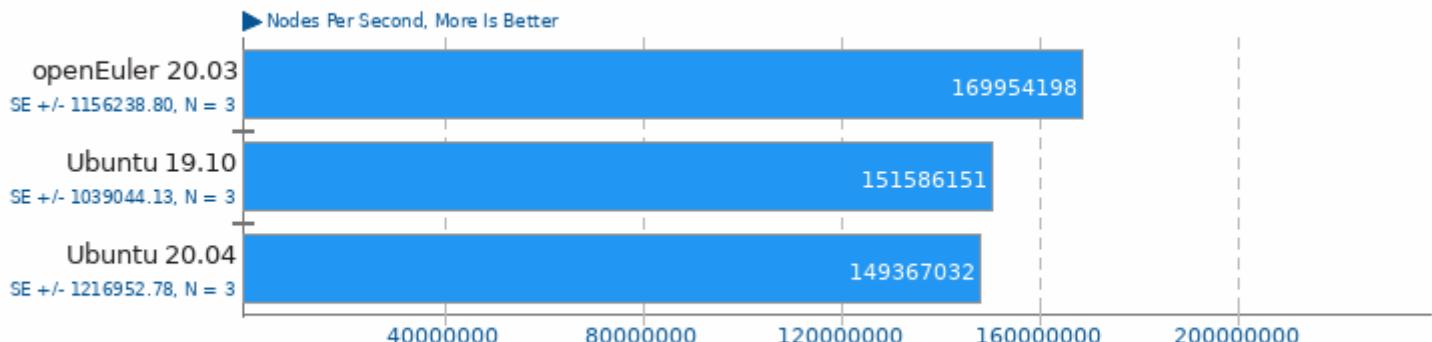
CoreMark Size 666 - Iterations Per Second



1. (CC) gcc options: -O2 -firt -lrt

## Stockfish 9

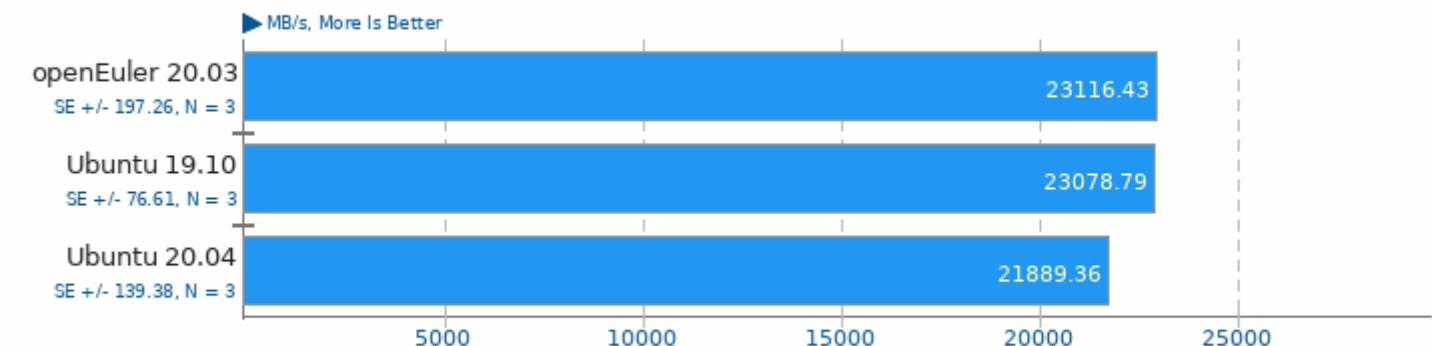
Total Time



1. (CXX) g++ options: -lpthread -fno-exceptions -std=c++11 -pedantic -O3 -fno

## RAMspeed SMP v3.5.0

Type: Add - Benchmark: Integer



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

## RAMspeed SMP v3.5.0

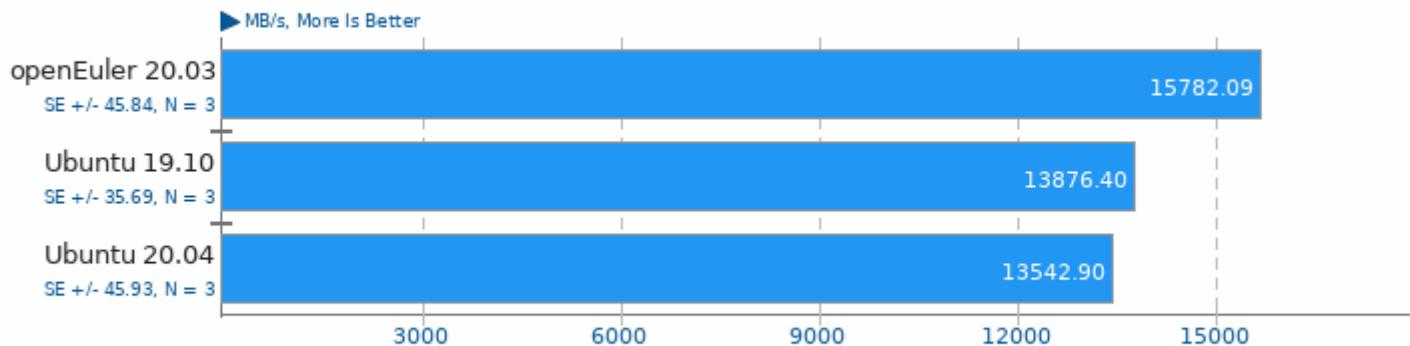
Type: Copy - Benchmark: Integer



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

## RAMspeed SMP v3.5.0

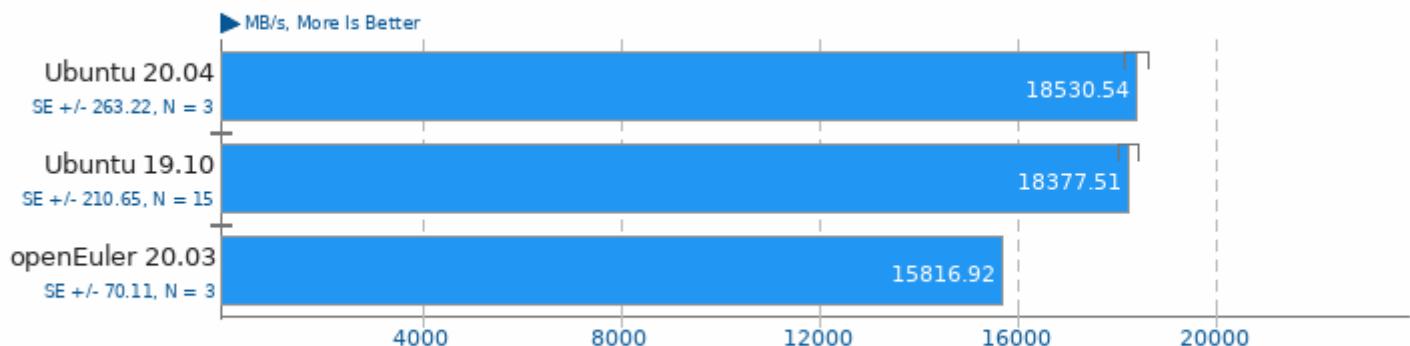
Type: Scale - Benchmark: Integer



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

## RAMspeed SMP v3.5.0

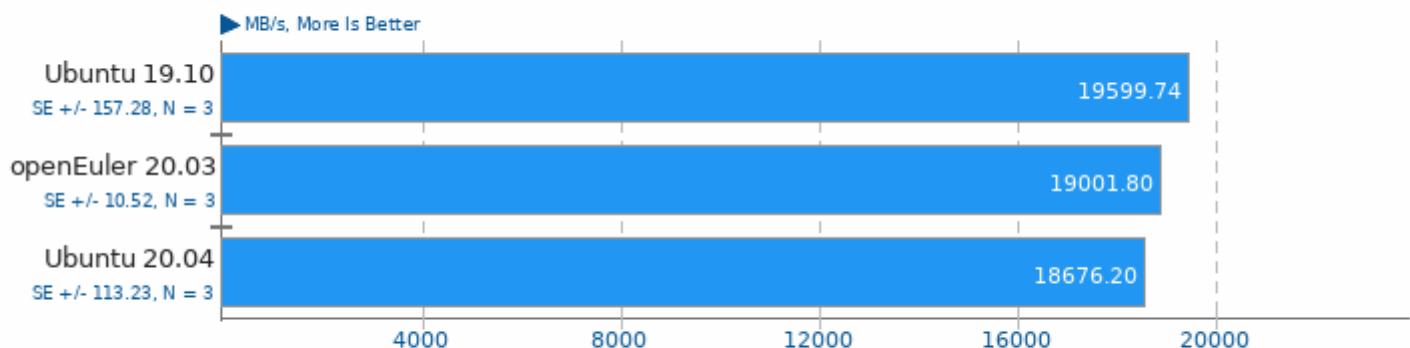
Type: Triad - Benchmark: Integer



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

## RAMspeed SMP v3.5.0

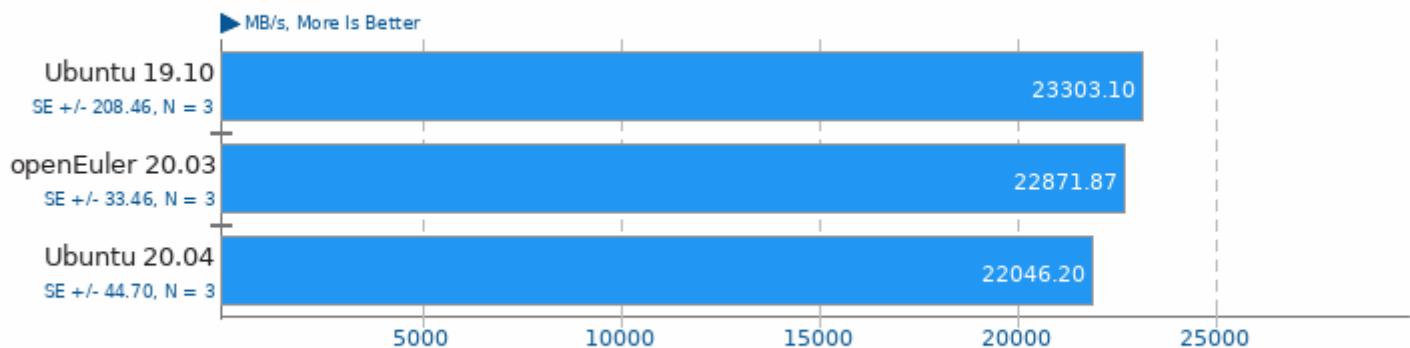
Type: Average - Benchmark: Integer



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

## RAMspeed SMP v3.5.0

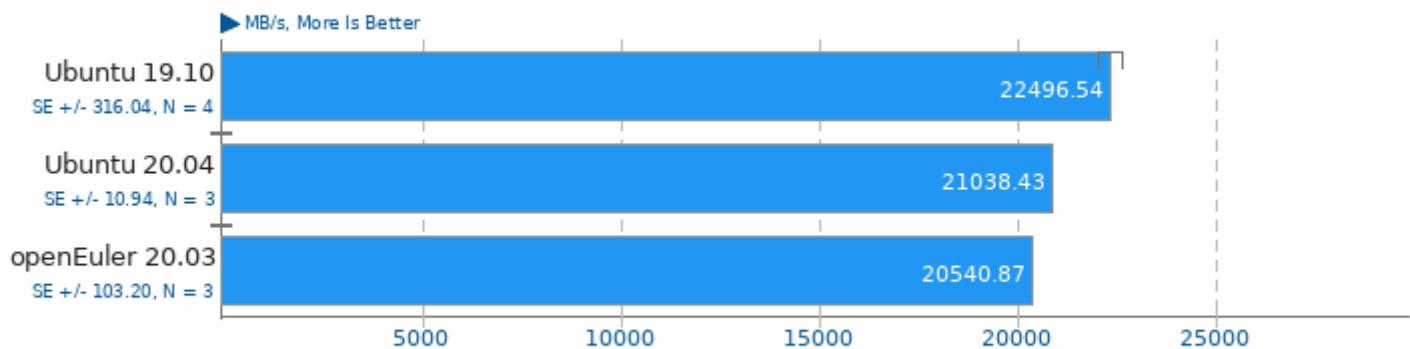
Type: Add - Benchmark: Floating Point



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

## RAMspeed SMP v3.5.0

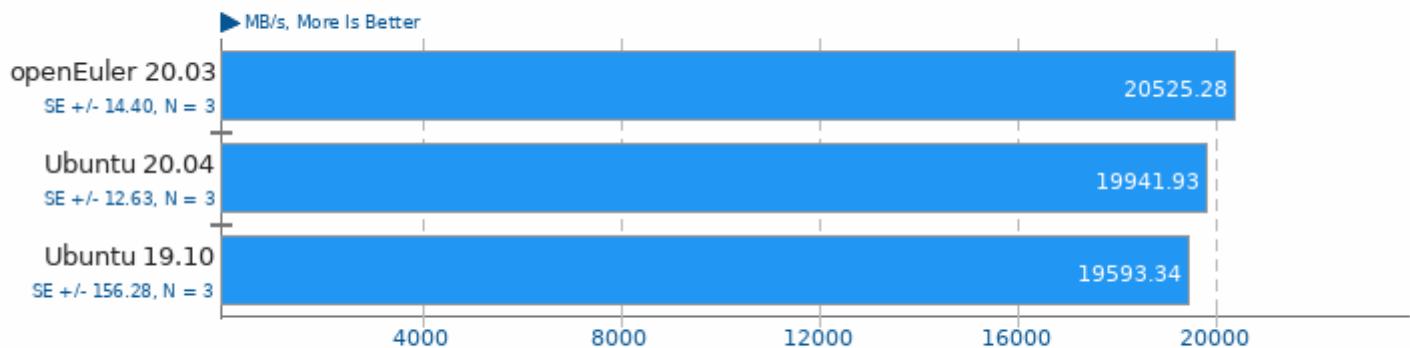
Type: Copy - Benchmark: Floating Point



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

## RAMspeed SMP v3.5.0

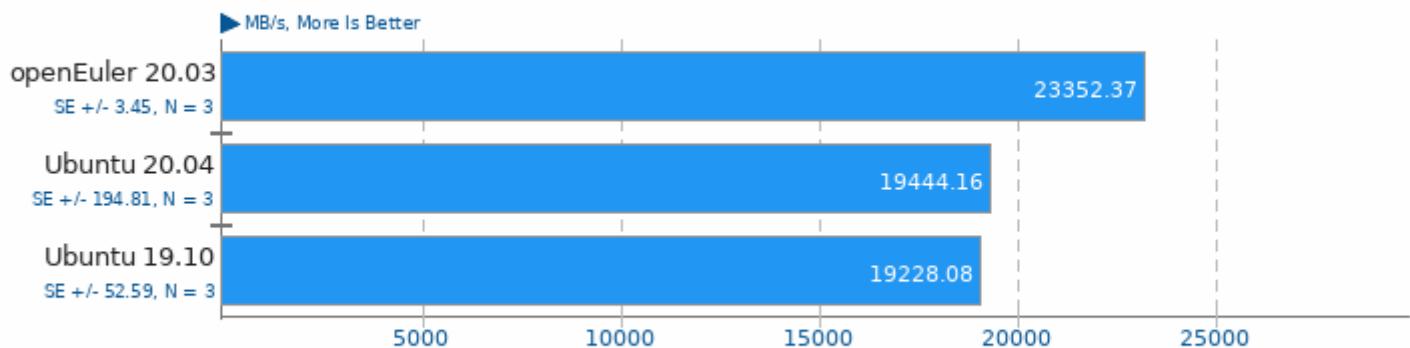
Type: Scale - Benchmark: Floating Point



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

## RAMspeed SMP v3.5.0

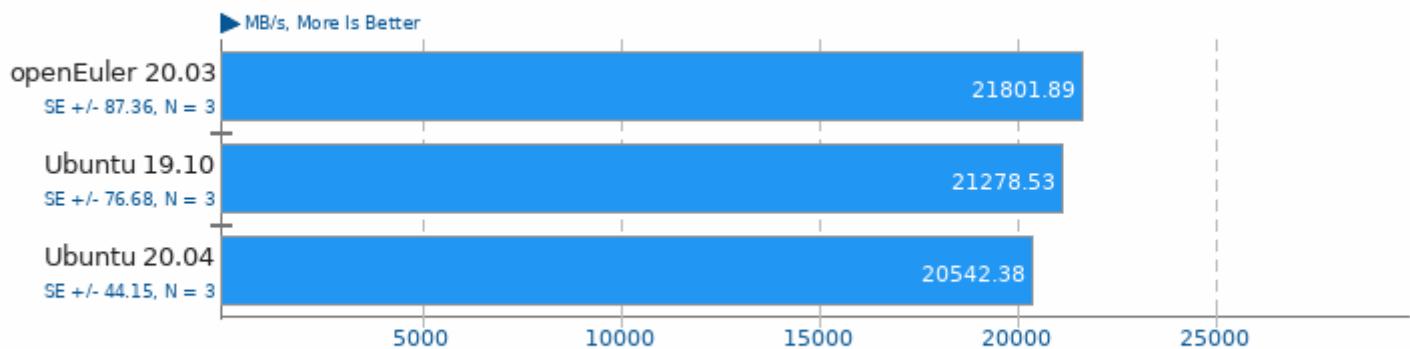
Type: Triad - Benchmark: Floating Point



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

## RAMspeed SMP v3.5.0

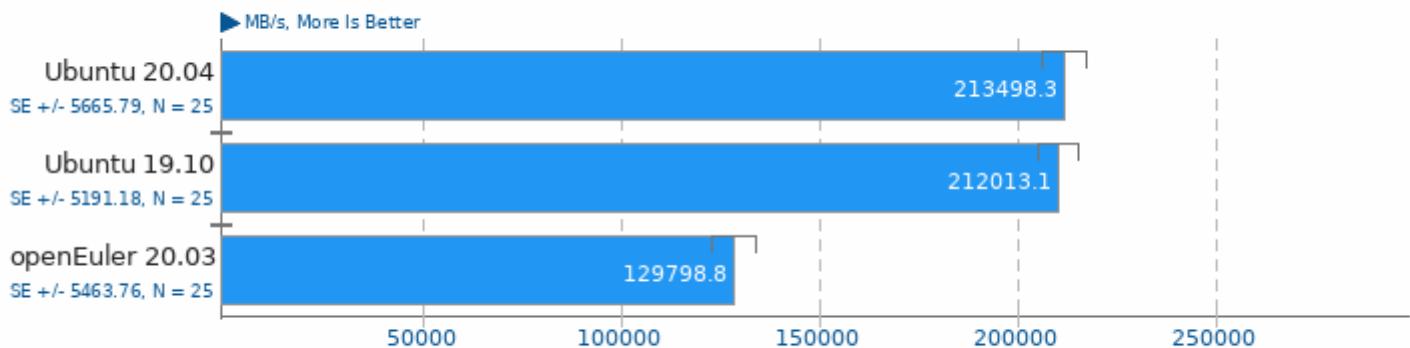
Type: Average - Benchmark: Floating Point



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

## Stream v2013-01-17

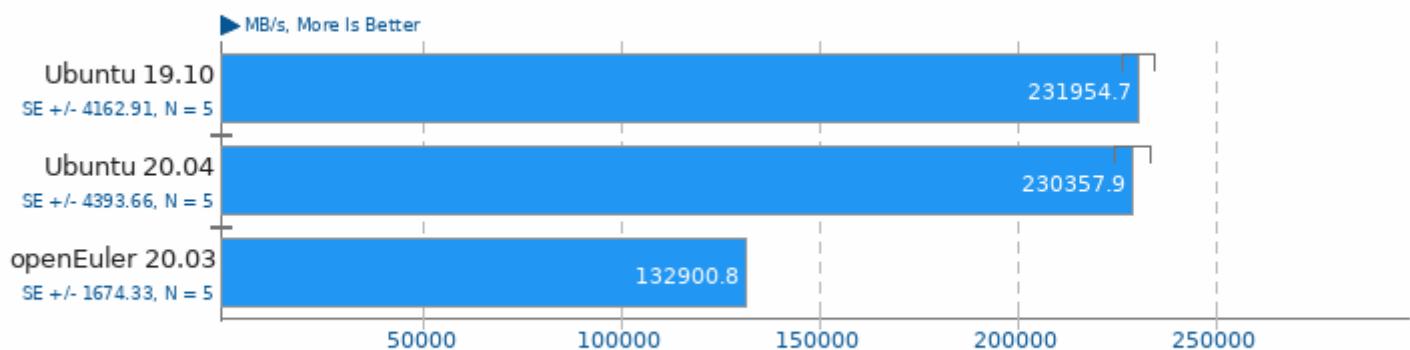
Type: Copy



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

## Stream v2013-01-17

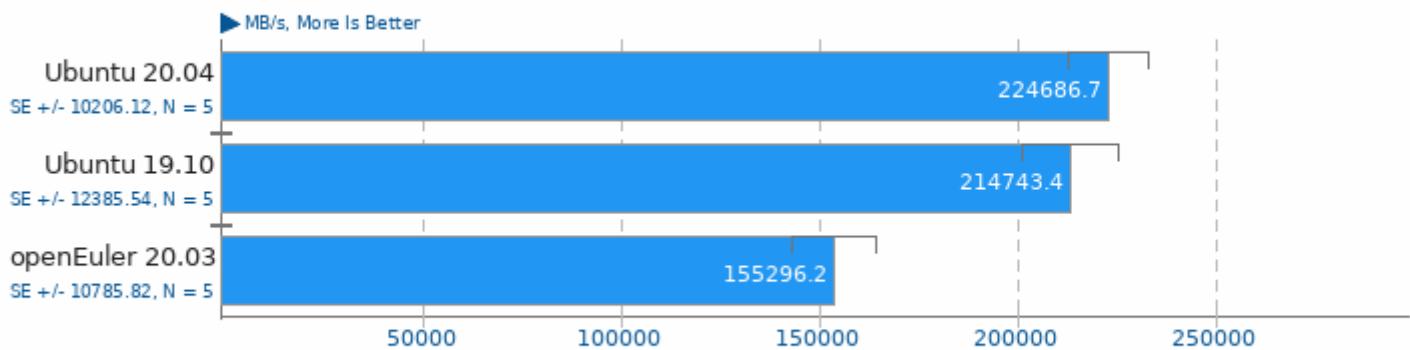
Type: Scale



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

## Stream v2013-01-17

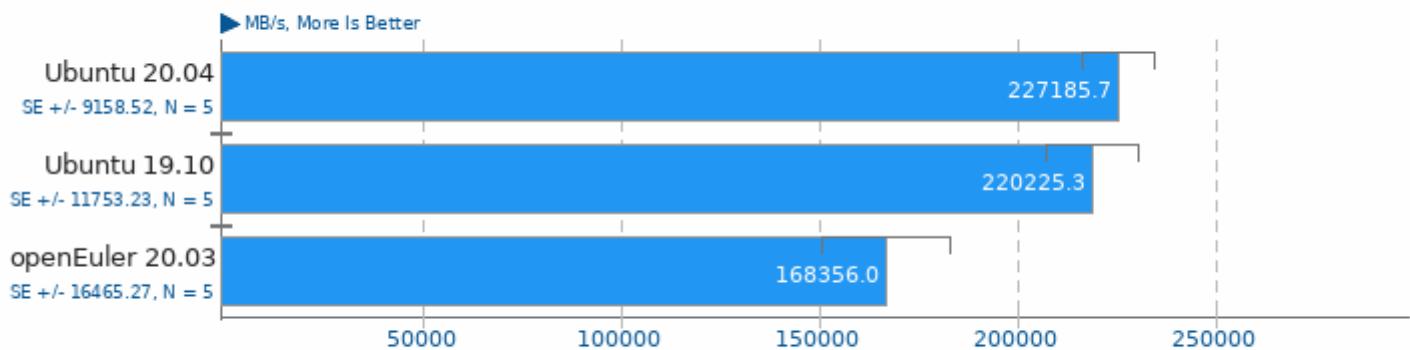
Type: Triad



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

## Stream v2013-01-17

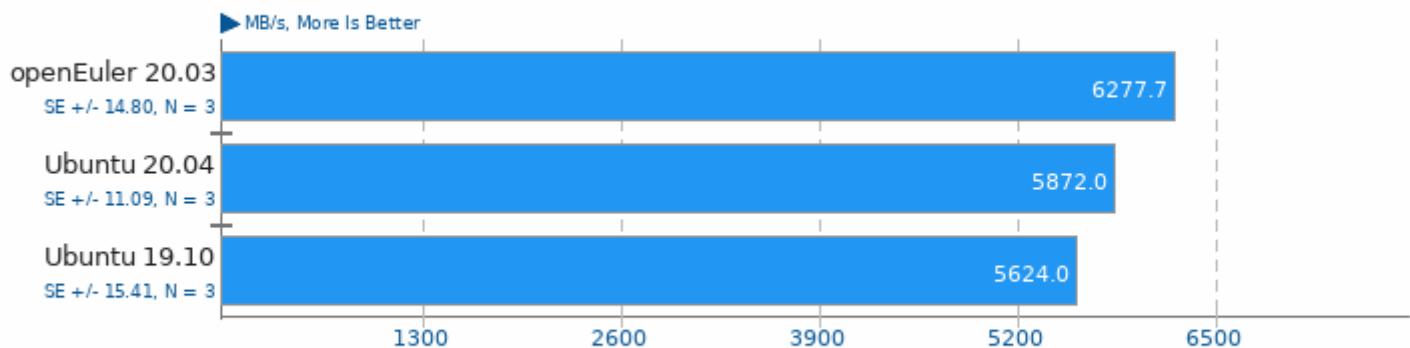
Type: Add



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

## Tinymembench v2018-05-28

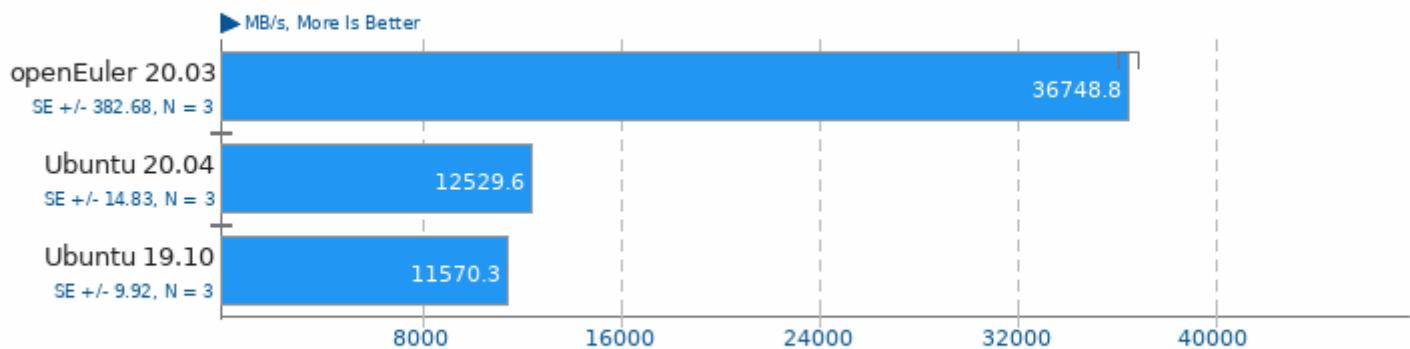
Standard Memcpy



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

## Tinymembench v2018-05-28

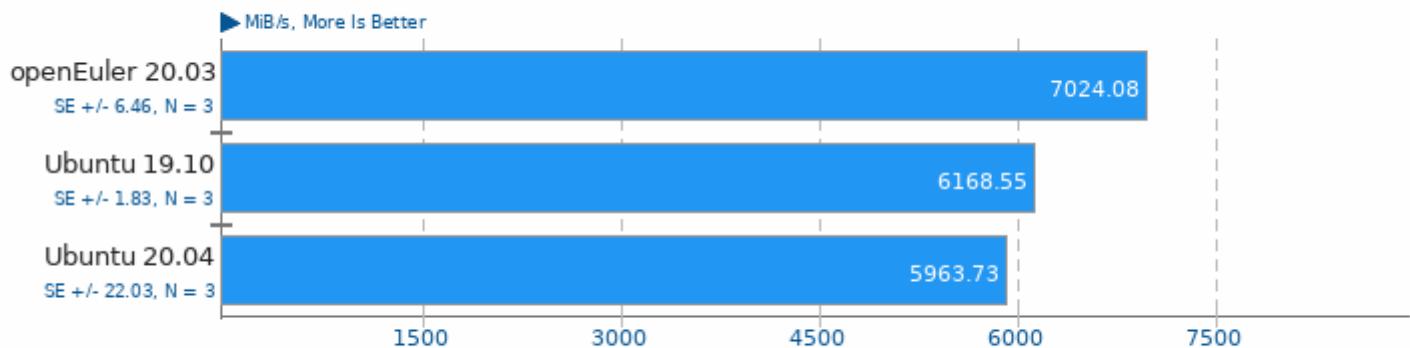
Standard Memset



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

## MBW v2018-09-08

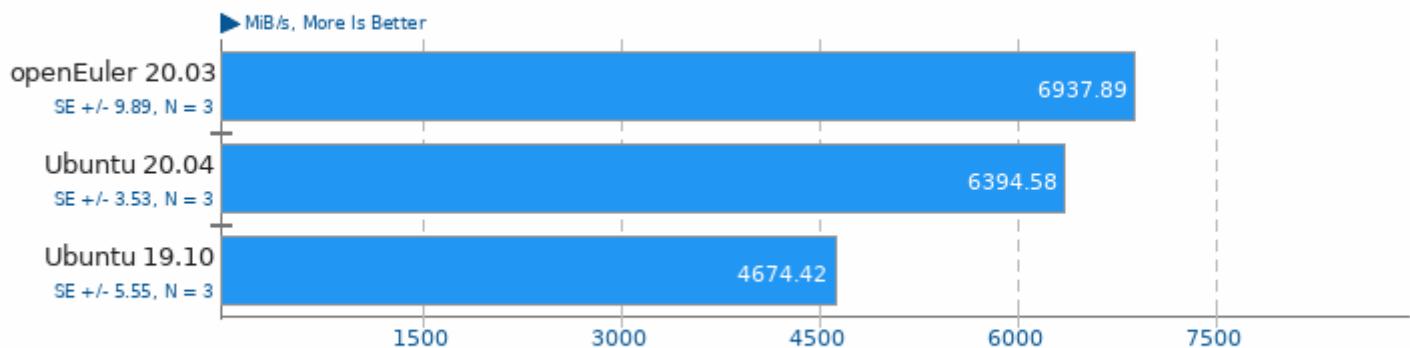
Test: Memory Copy - Array Size: 128 MiB



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

## MBW v2018-09-08

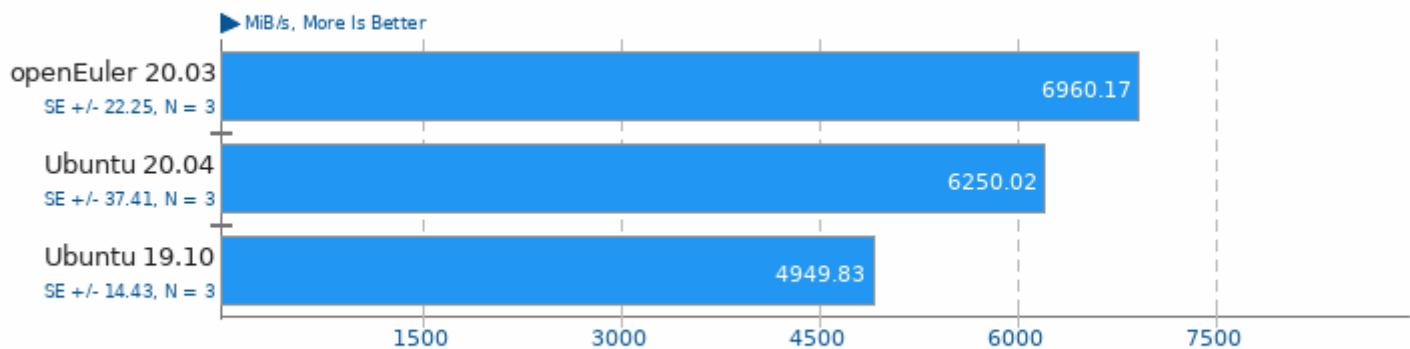
Test: Memory Copy - Array Size: 512 MiB



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

## MBW v2018-09-08

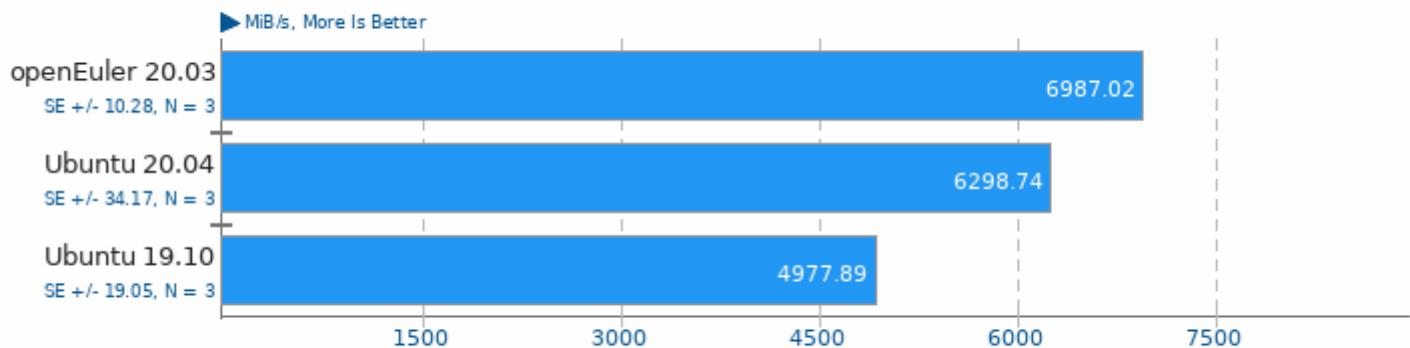
Test: Memory Copy - Array Size: 1024 MiB



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

## MBW v2018-09-08

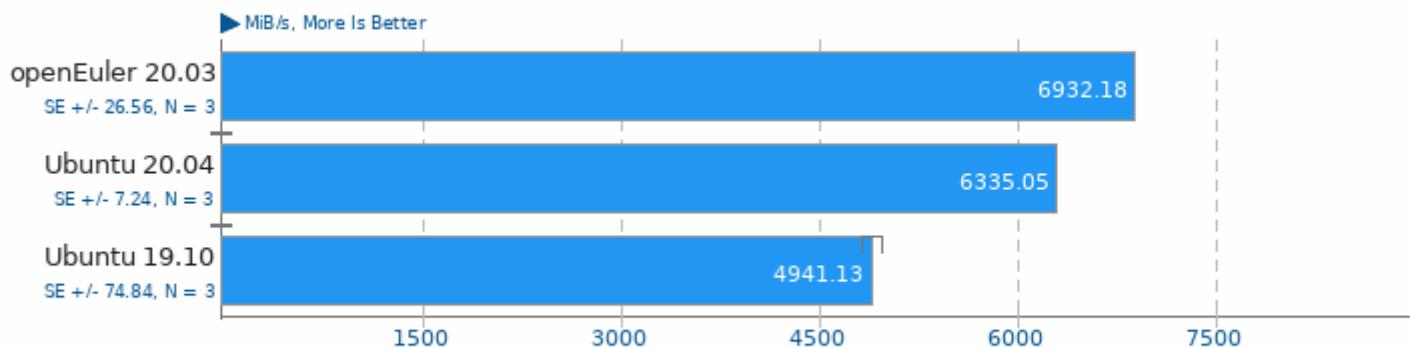
Test: Memory Copy - Array Size: 4096 MiB



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

## MBW v2018-09-08

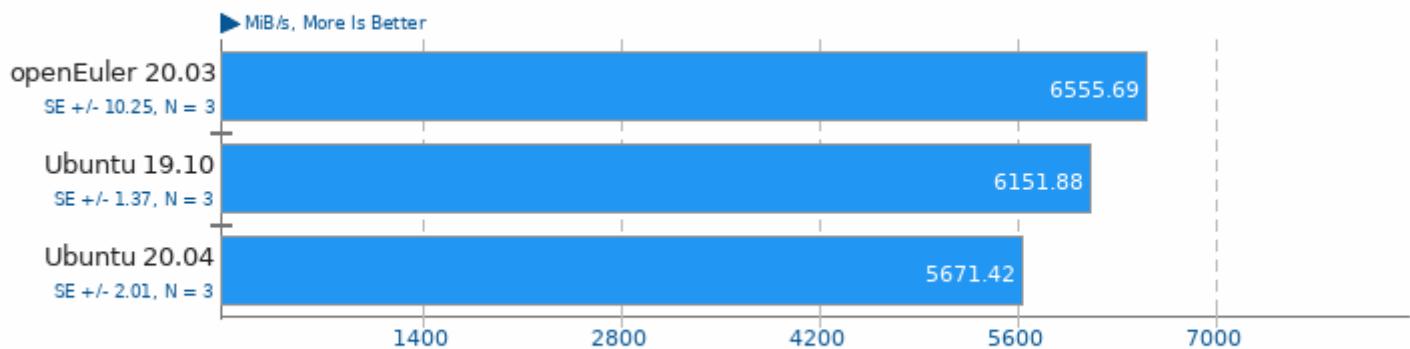
Test: Memory Copy - Array Size: 8192 MiB



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

## MBW v2018-09-08

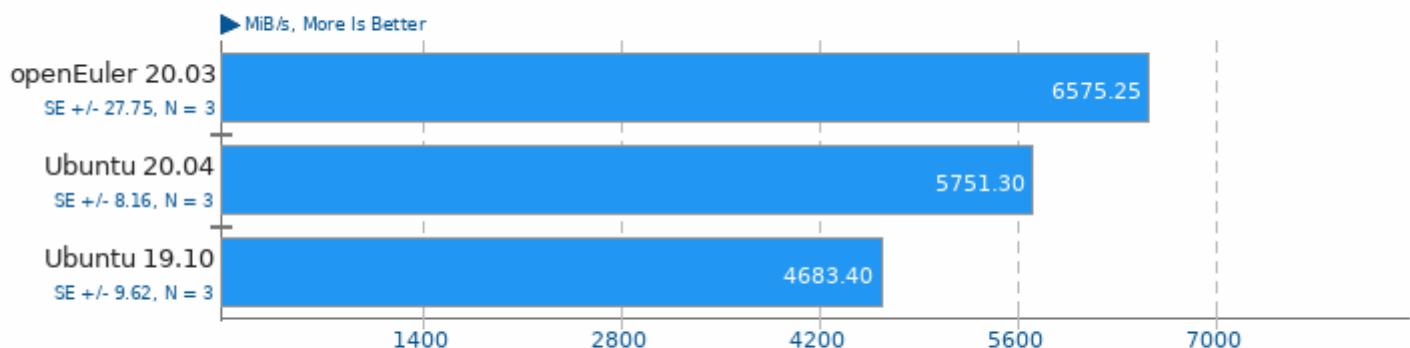
Test: Memory Copy, Fixed Block Size - Array Size: 128 MiB



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

## MBW v2018-09-08

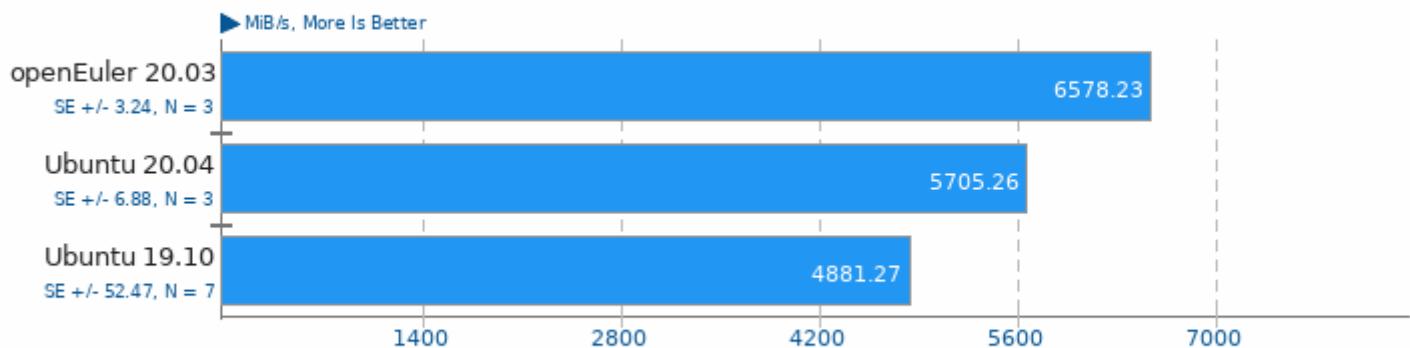
Test: Memory Copy, Fixed Block Size - Array Size: 512 MiB



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

## MBW v2018-09-08

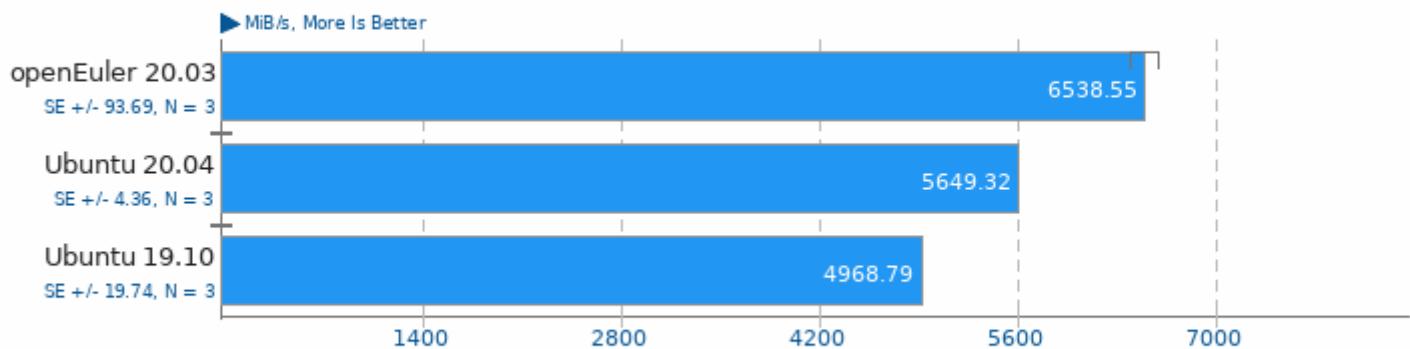
Test: Memory Copy, Fixed Block Size - Array Size: 1024 MiB



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

## MBW v2018-09-08

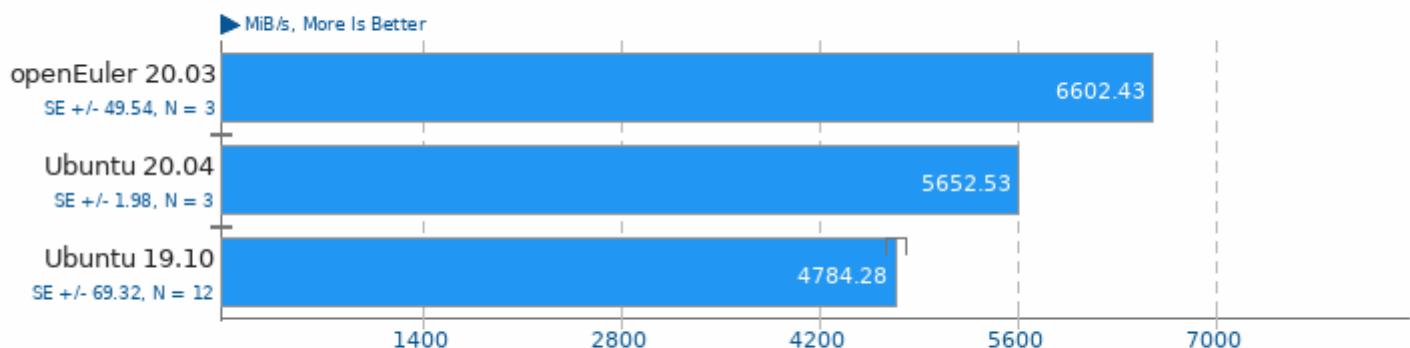
Test: Memory Copy, Fixed Block Size - Array Size: 4096 MiB



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

## MBW v2018-09-08

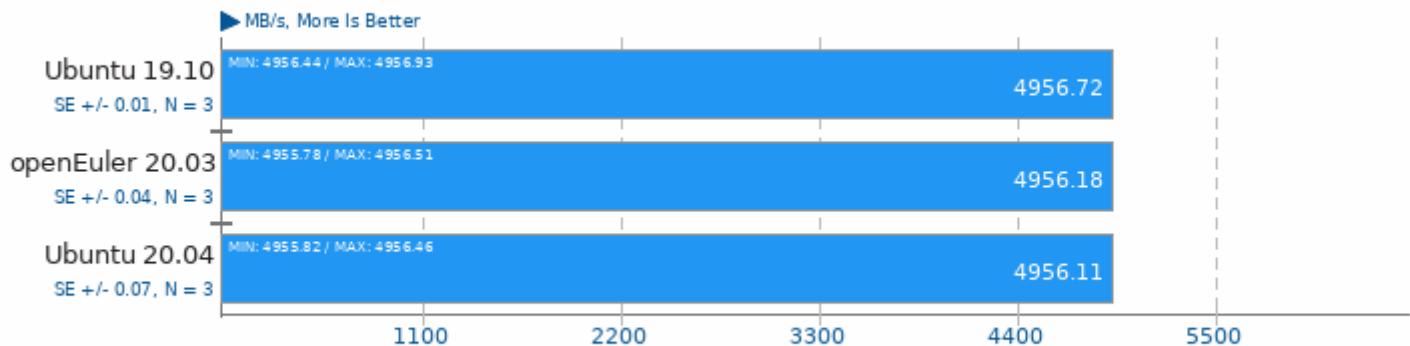
Test: Memory Copy, Fixed Block Size - Array Size: 8192 MiB



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

## CacheBench

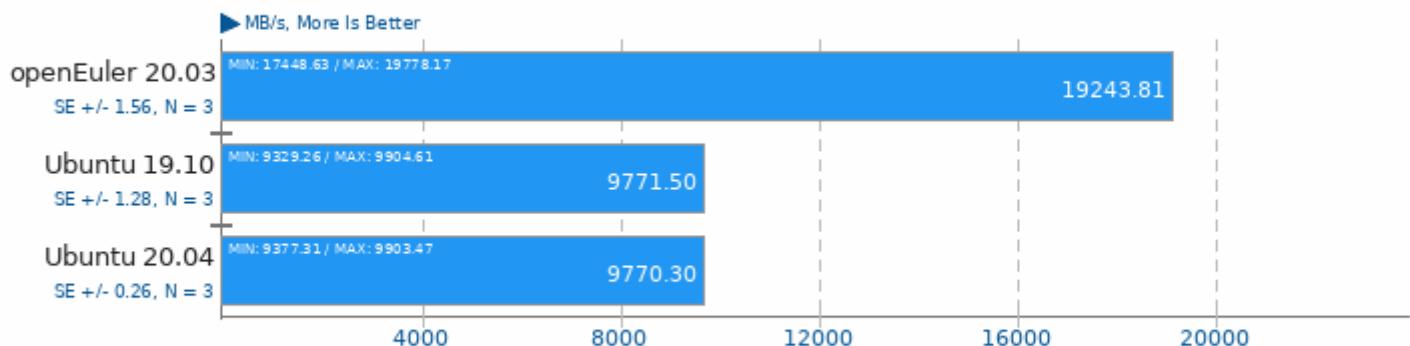
Test: Read



1. (CC) gcc options: -lrt

## CacheBench

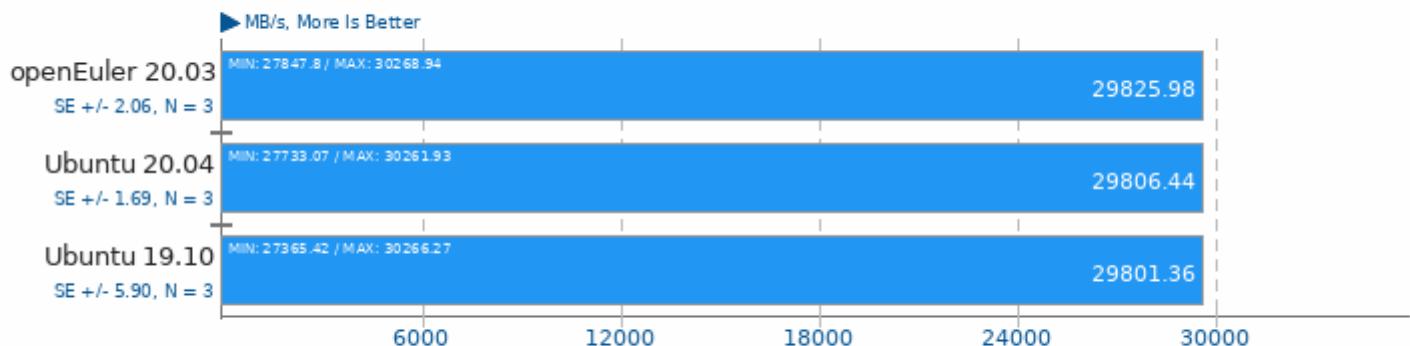
Test: Write



1. (CC) gcc options: -lrt

## CacheBench

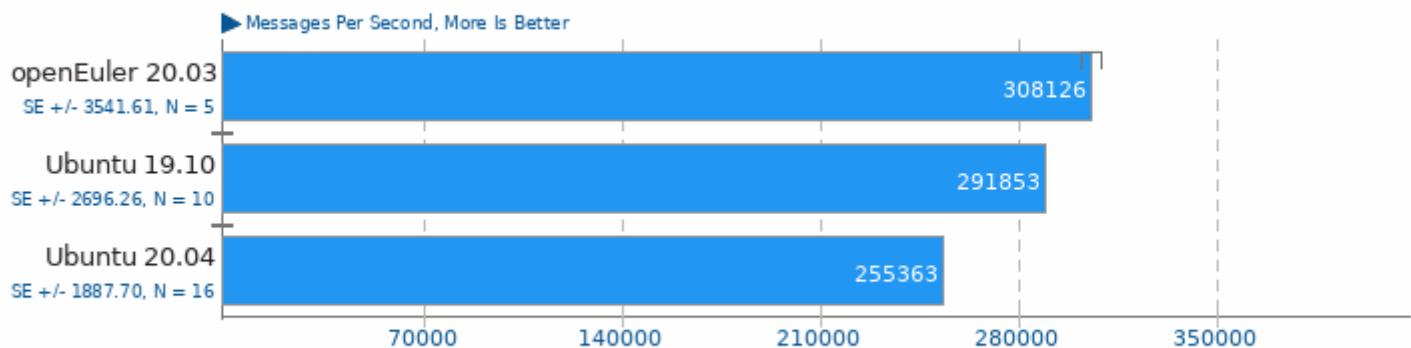
Test: Read / Modify / Write



1. (CC) gcc options: -lrt

## Sockperf v3.4

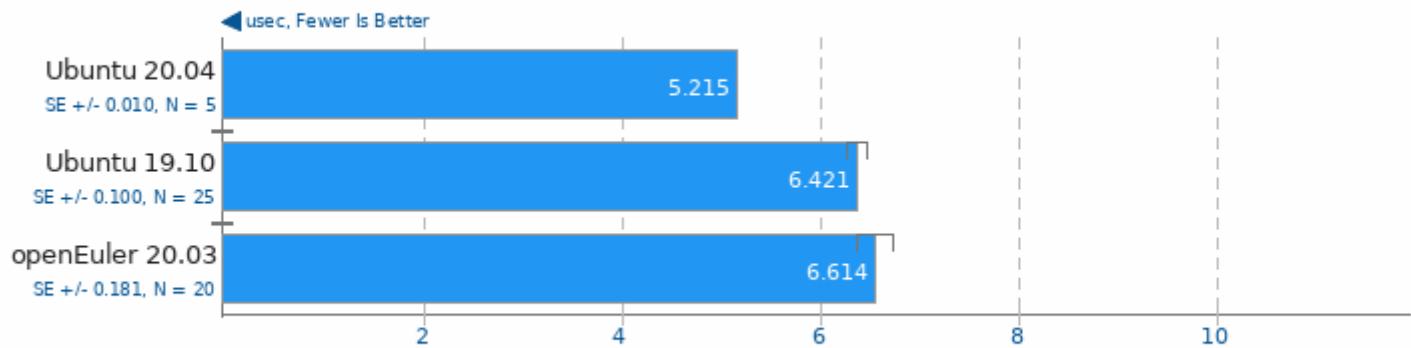
Test: Throughput



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

## Sockperf v3.4

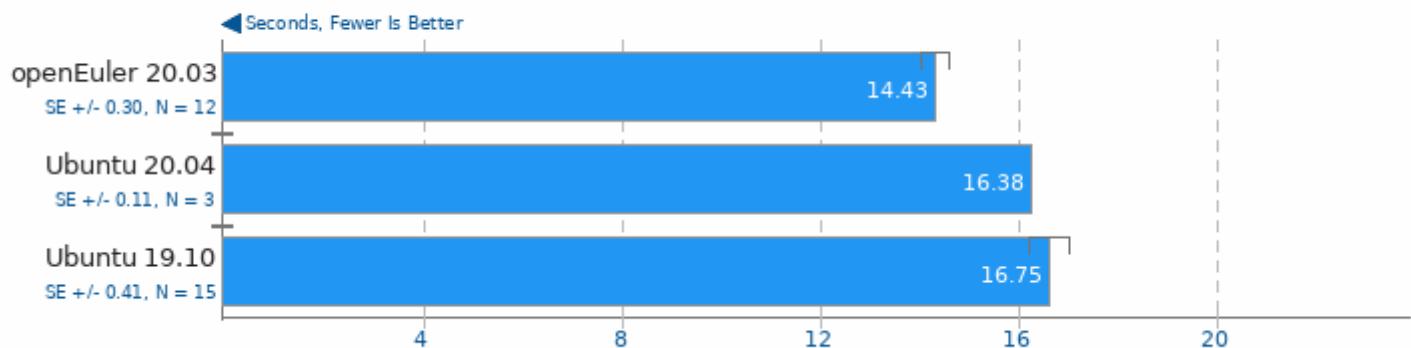
Test: Latency Ping Pong



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

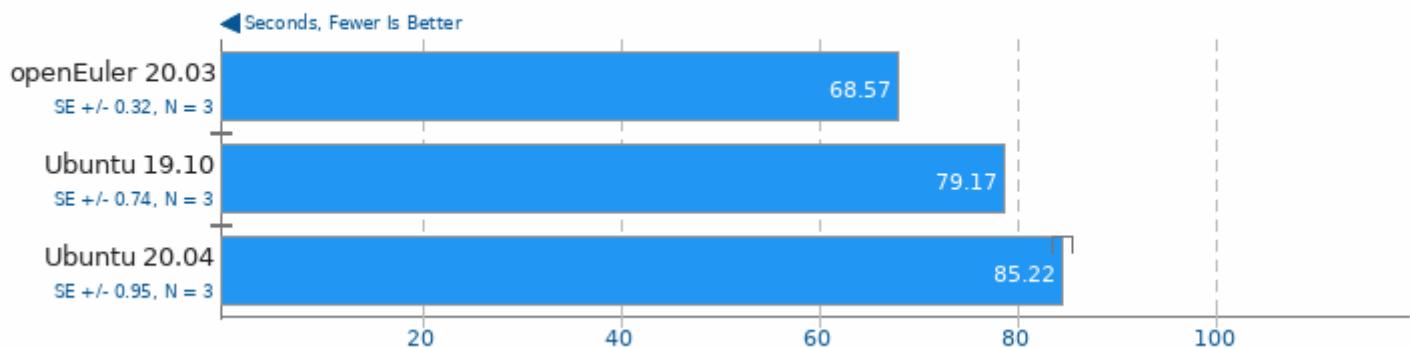
## Loopback TCP Network Performance

Time To Transfer 10GB Via Loopback



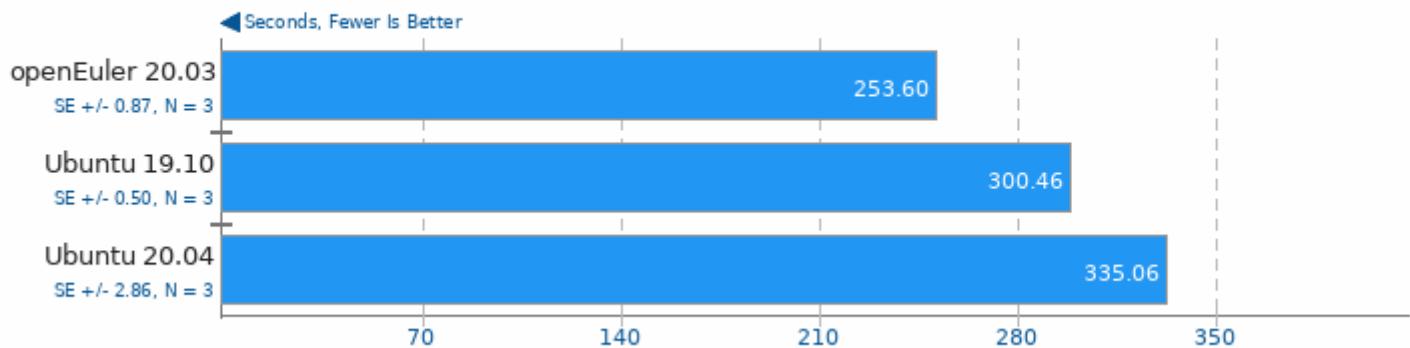
## Timed Linux Kernel Compilation v5.4

Time To Compile



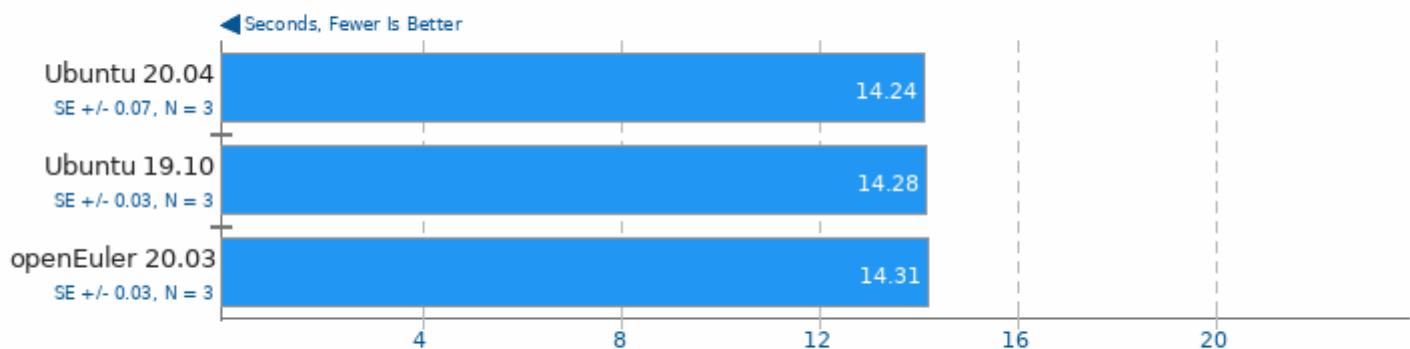
## Timed LLVM Compilation v10.0

Time To Compile



## C-Ray v1.1

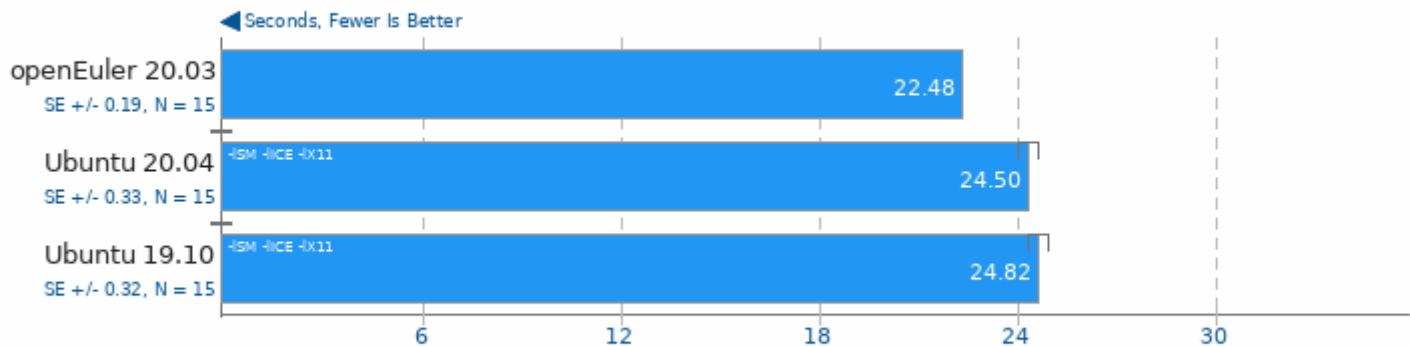
Total Time - 4K, 16 Rays Per Pixel



1. (CC) gcc options: -fno-omit-frame-pointer -O3

## POV-Ray v3.7.0.7

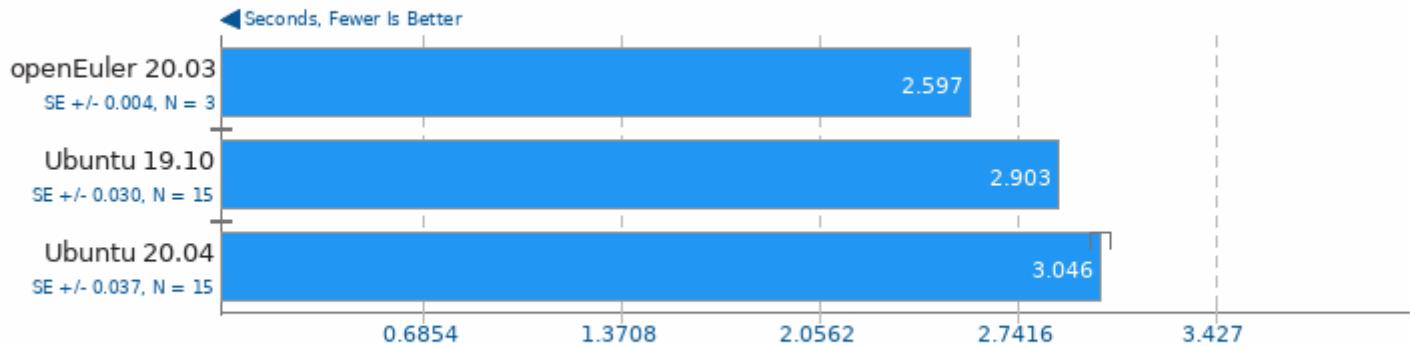
Trace Time



1. (CXX) g++ options: -pipe -O3 -ffast-math -pthread -ltiff -jpeg -lpng -lz -lrt -lm -boost\_thread -boost\_system

## Smallpt v1.0

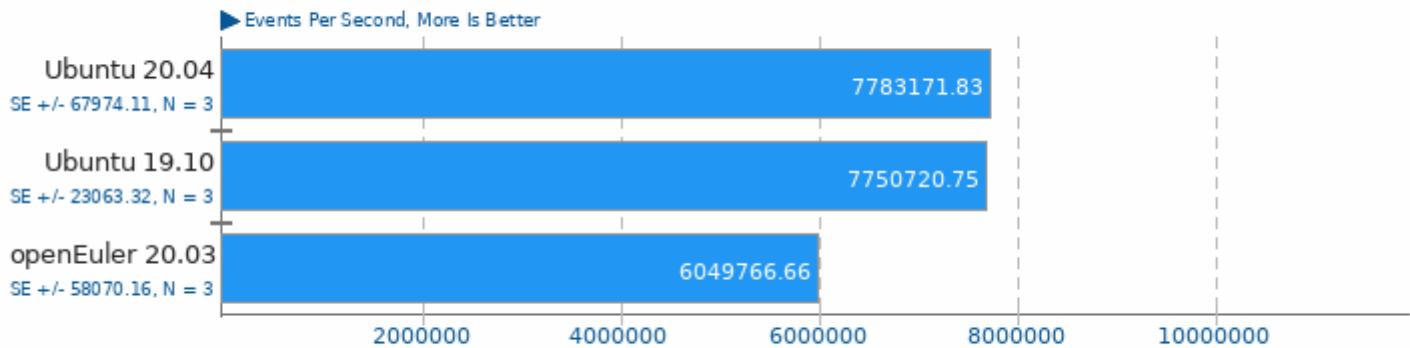
Global Illumination Renderer; 128 Samples



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

## Sysbench v2018-07-28

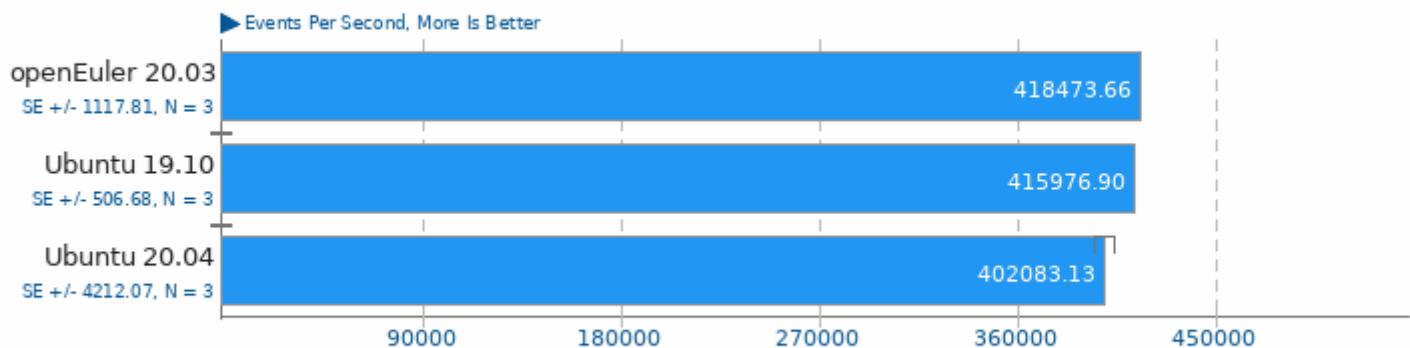
Test: Memory



1. (CC) gcc options: -pthread -O3 -funroll-loops -ggdb3 -rdynamic -ldl -laio -lm

## Sysbench v2018-07-28

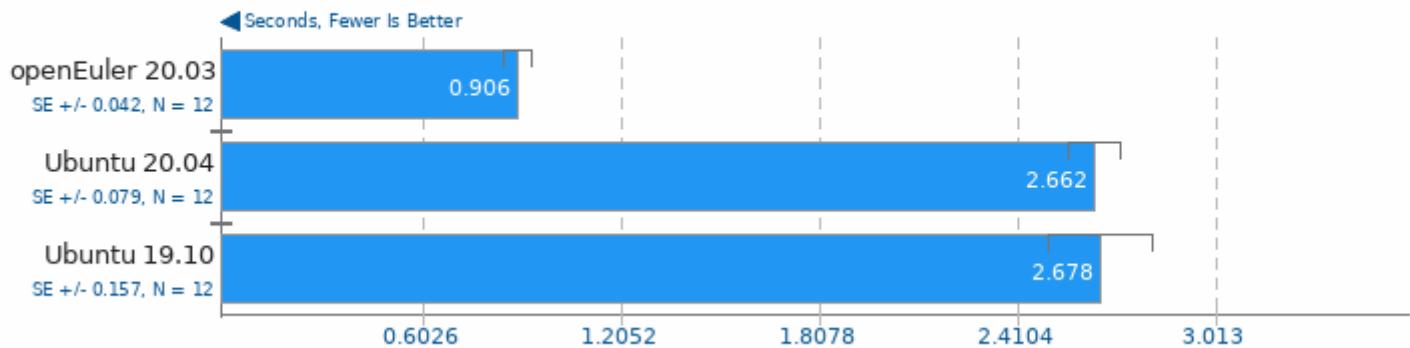
Test: CPU



1. (CC) gcc options: -pthread -O3 -funroll-loops -ggdb3 -rdynamic -ldl -laio -lm

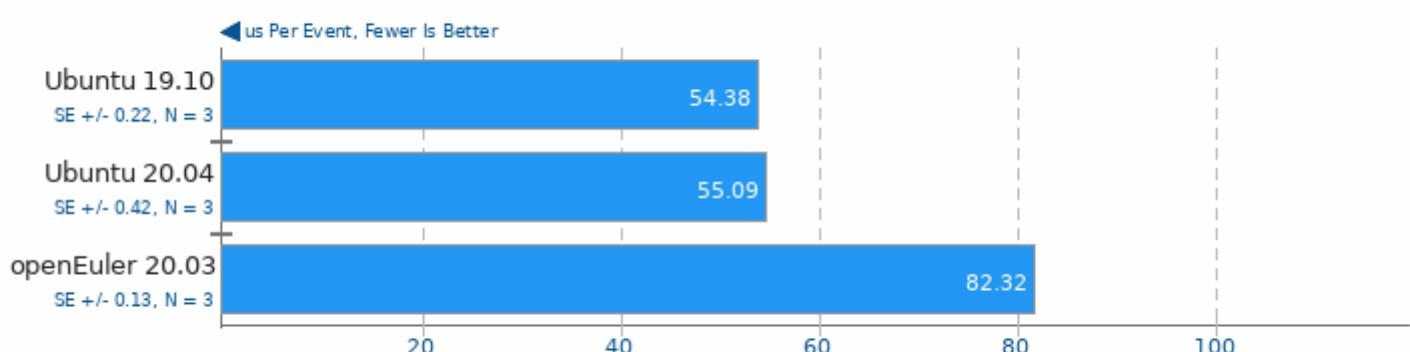
## Sunflow Rendering System v0.07.2

Global Illumination + Image Synthesis



## OSBench

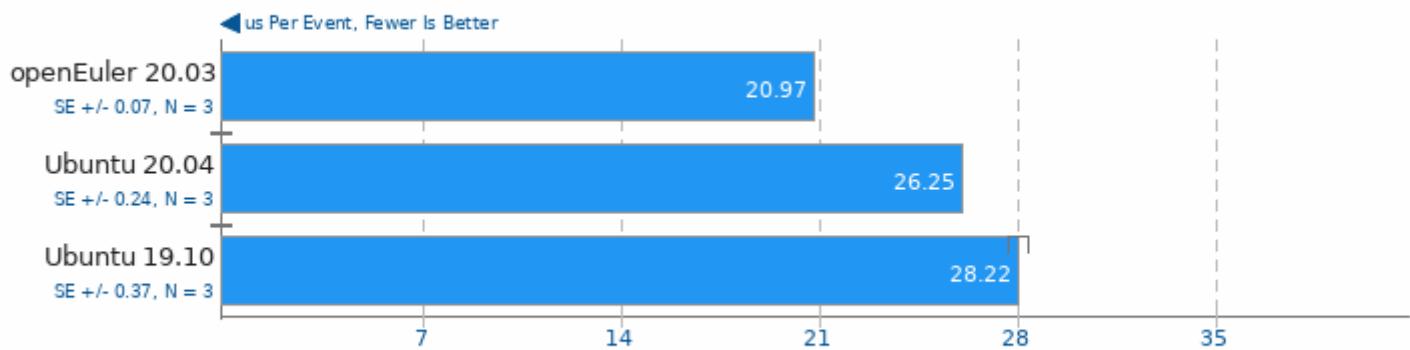
Test: Create Files



1. (CC) gcc options: -lm

## OSBench

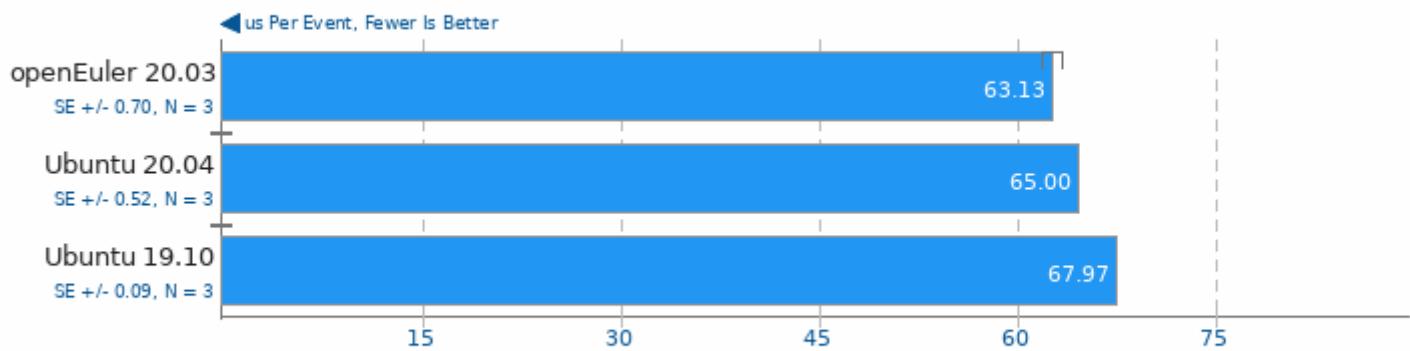
Test: Create Threads



1. (CC) gcc options: -lm

## OSBench

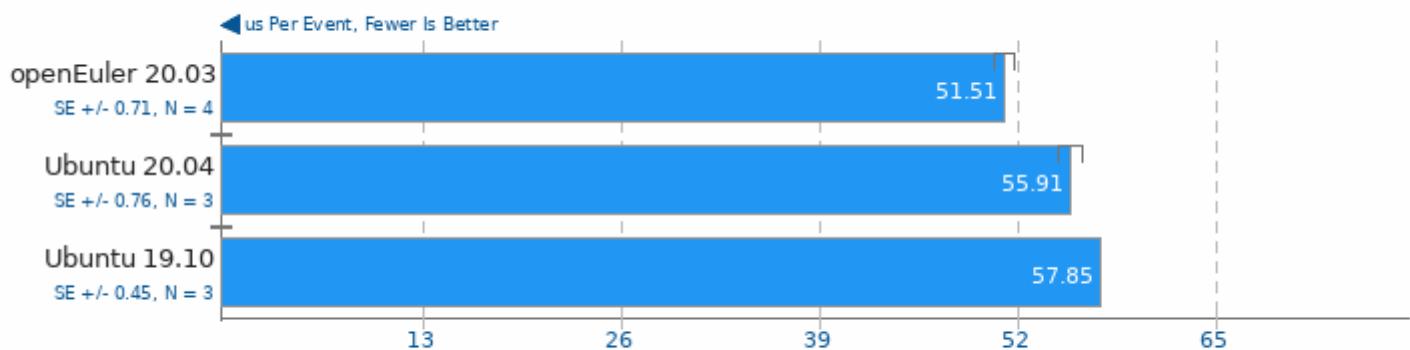
Test: Launch Programs



1. (CC) gcc options: -lm

## OSBench

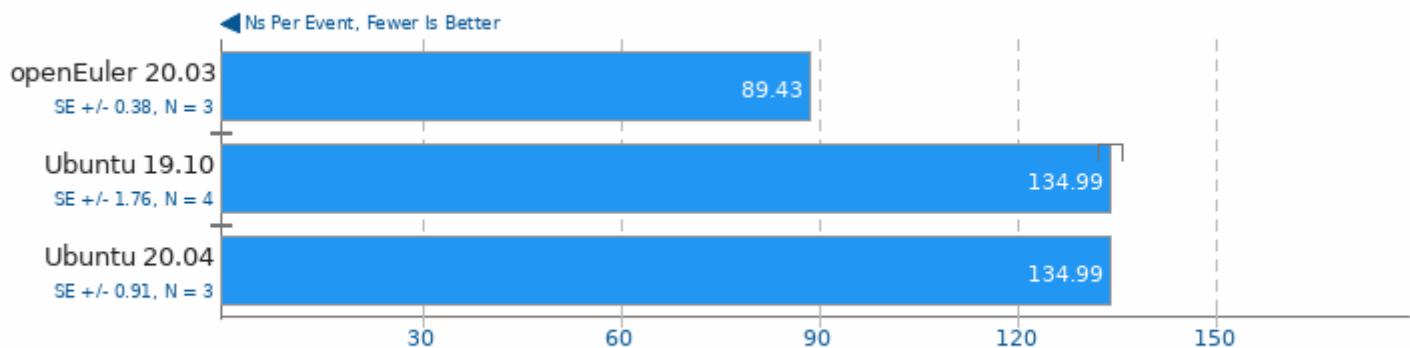
Test: Create Processes



1. (CC) gcc options: -lm

## OSBench

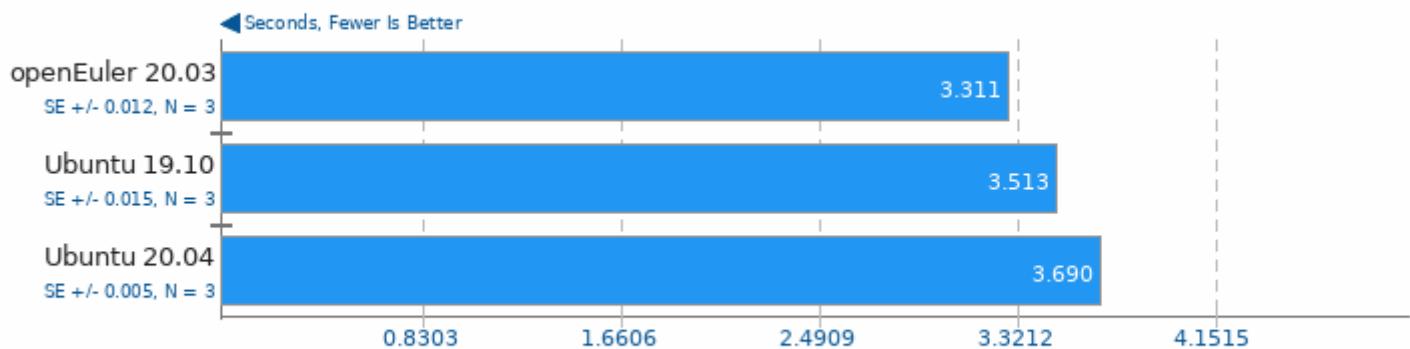
Test: Memory Allocations



1. (CC) gcc options: -lm

## SQLite v3.30.1

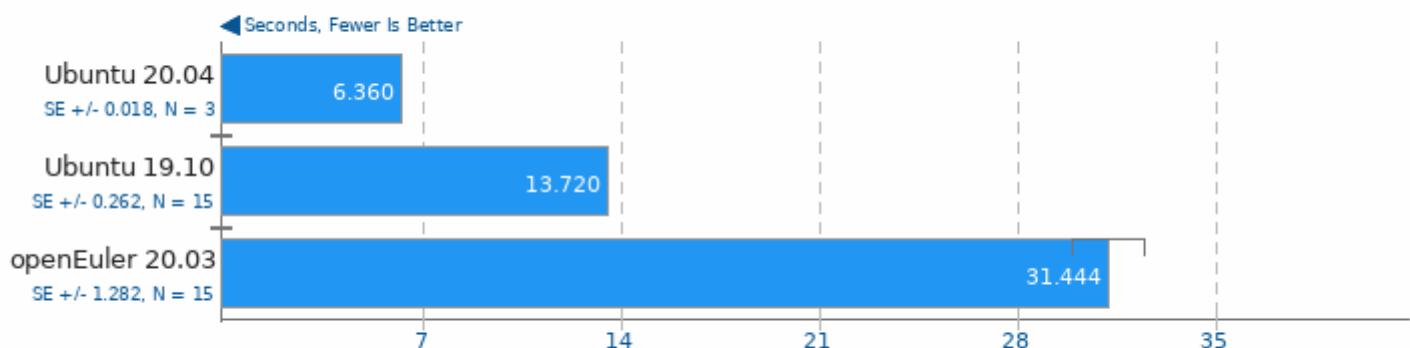
Threads / Copies: 1



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

## SQLite v3.30.1

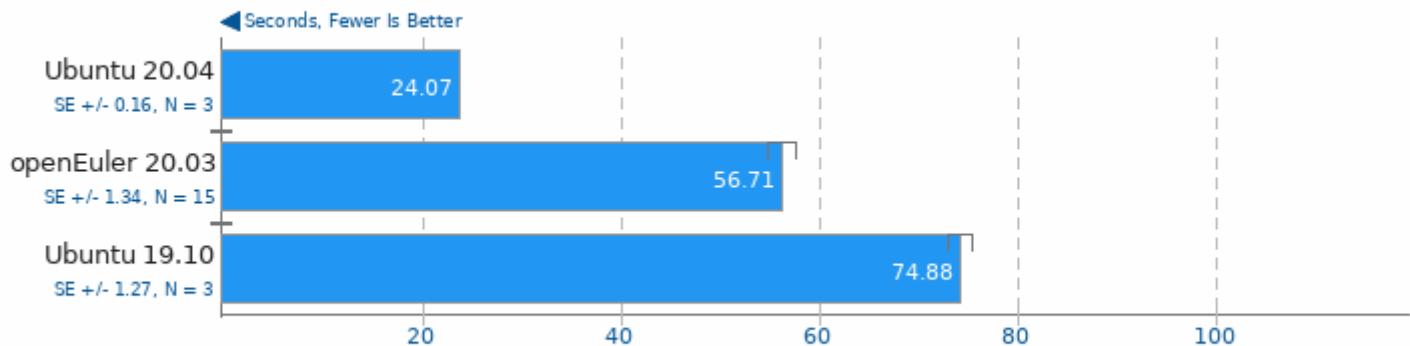
Threads / Copies: 8



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

## SQLite v3.30.1

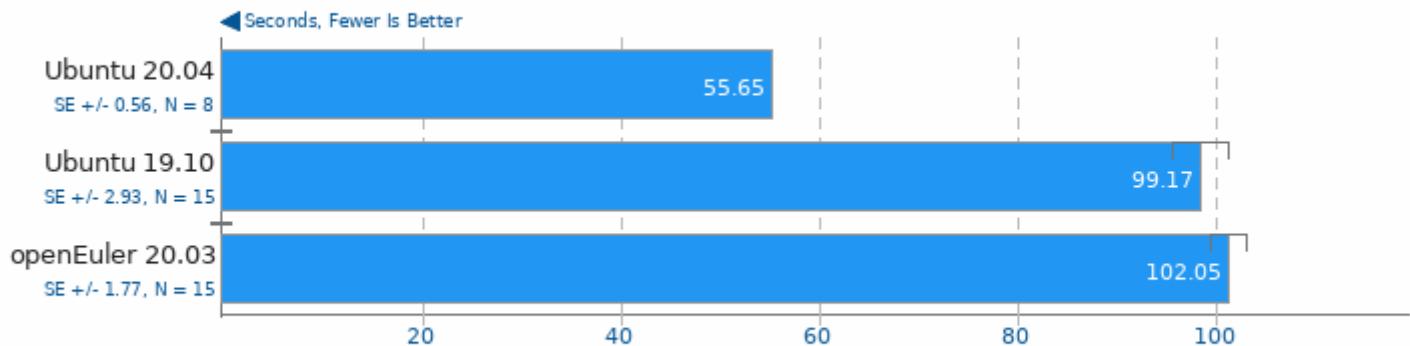
Threads / Copies: 32



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

## SQLite v3.30.1

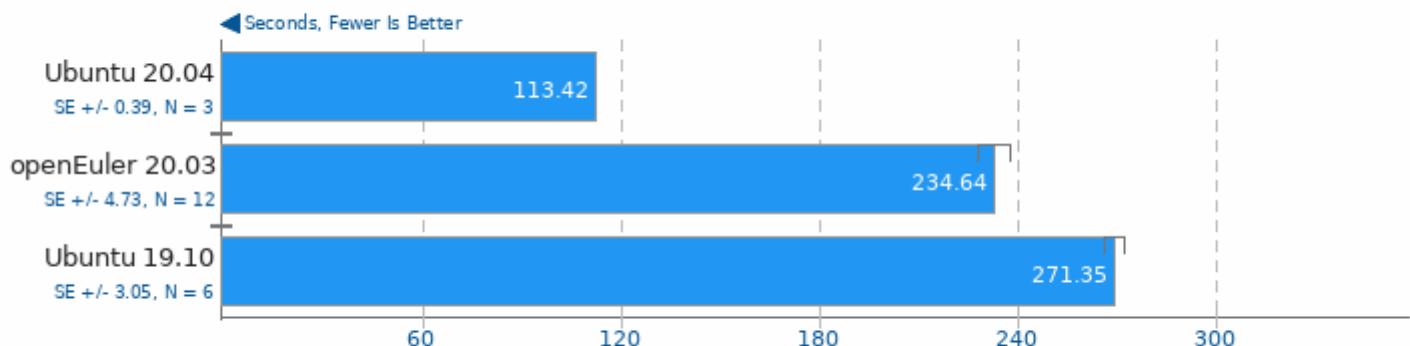
Threads / Copies: 64



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

## SQLite v3.30.1

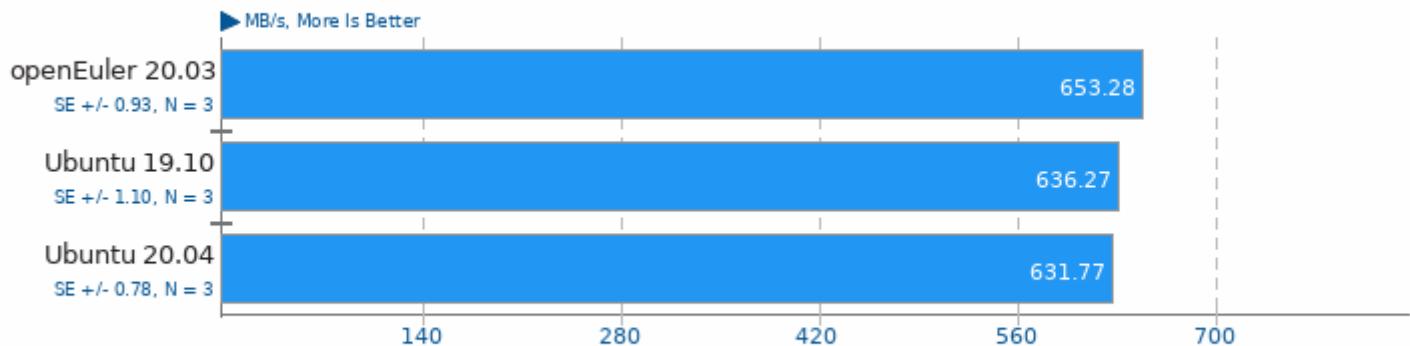
Threads / Copies: 128



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

## Dbench v4.0

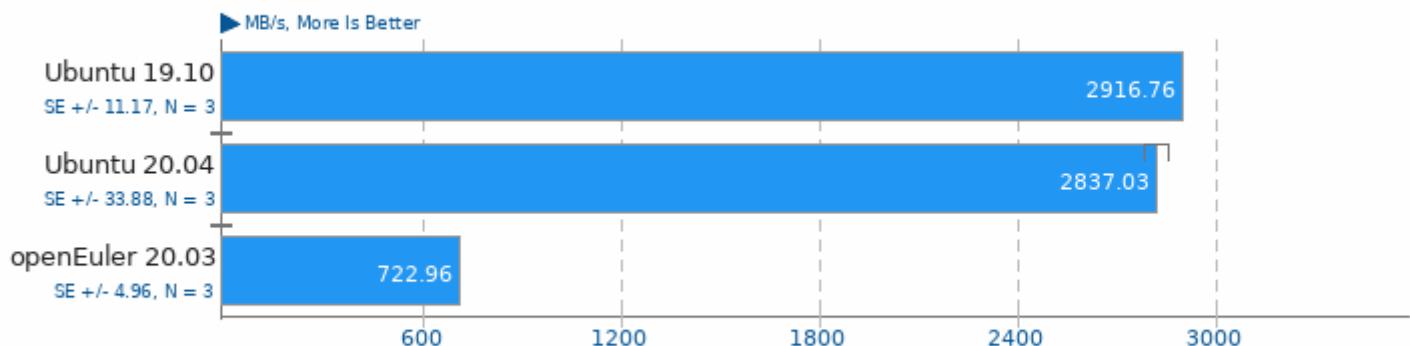
Client Count: 1



1. (CC) gcc options: -fno-optimize-sibling-calls

## Dbench v4.0

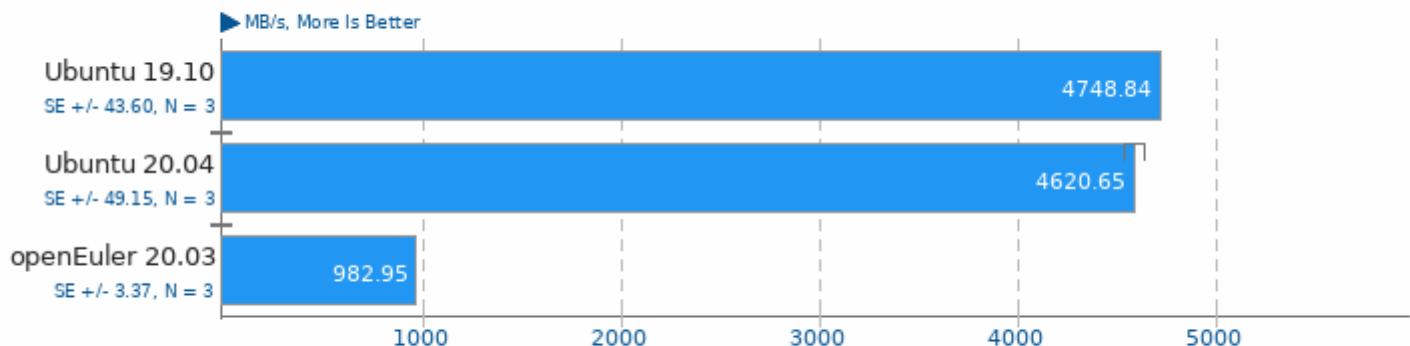
Client Count: 6



1. (CC) gcc options: -fno-optimize-sibling-calls

## Dbench v4.0

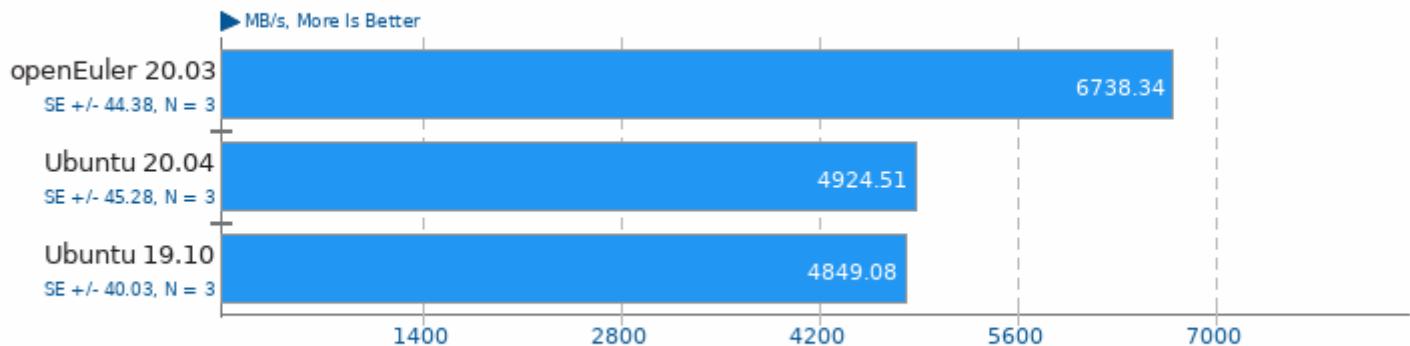
Client Count: 12



1. (CC) gcc options: -fno-optimize-sibling-calls

## Dbench v4.0

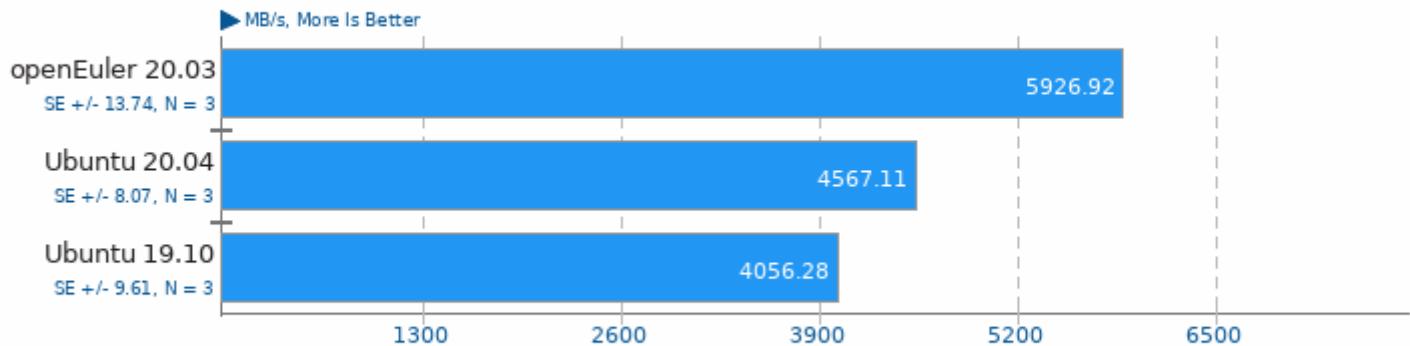
Client Count: 48



1. (CC) gcc options: -fno-pie -O2

## Dbench v4.0

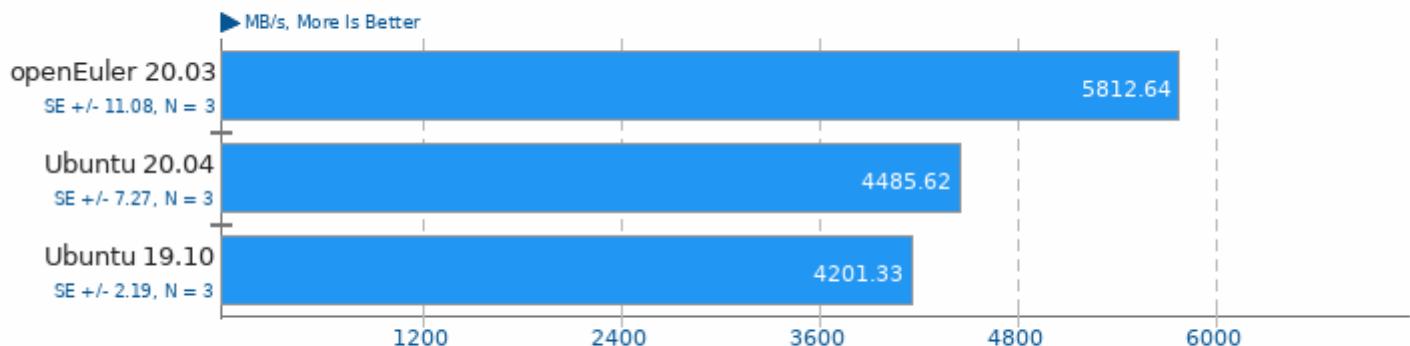
Client Count: 128



1. (CC) gcc options: -fno-pie -O2

## Dbench v4.0

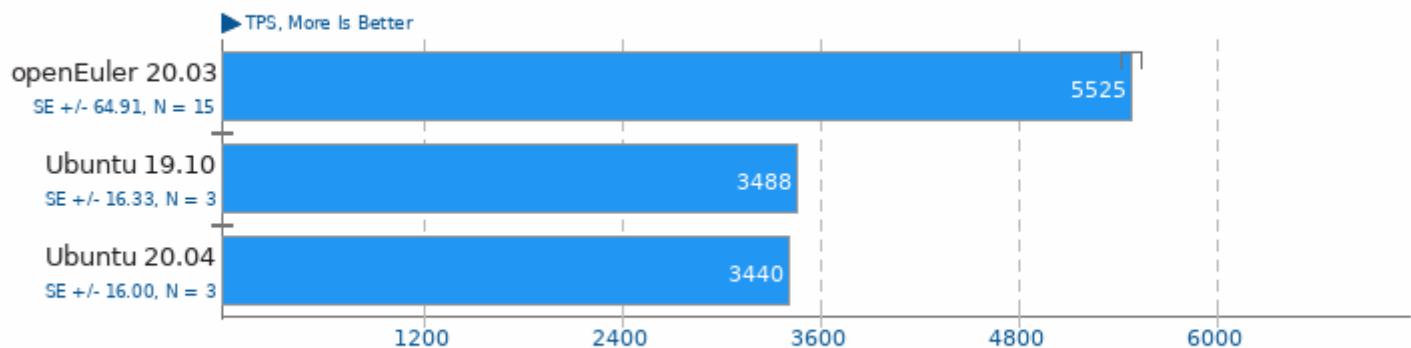
Client Count: 256



1. (CC) gcc options: -fno-pie -O2

## PostMark v1.51

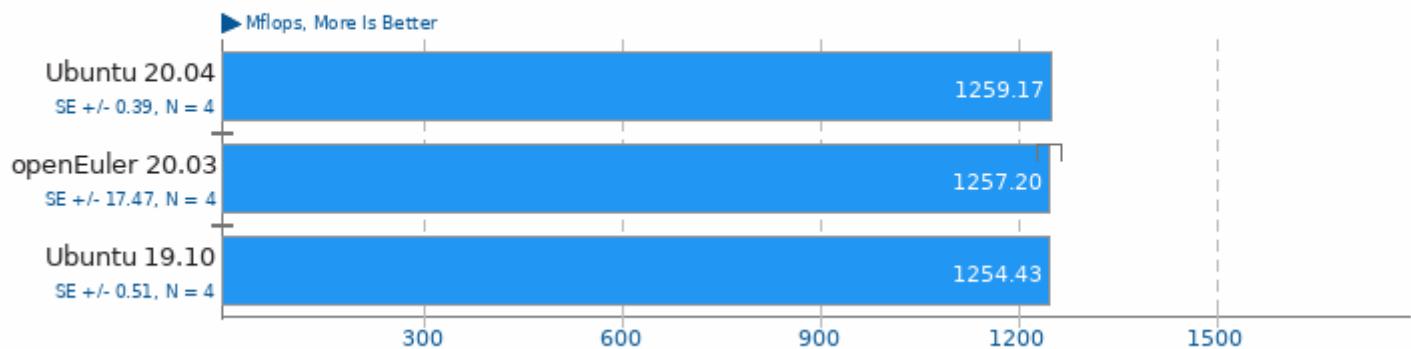
Disk Transaction Performance



1. (CC) gcc options: -O3

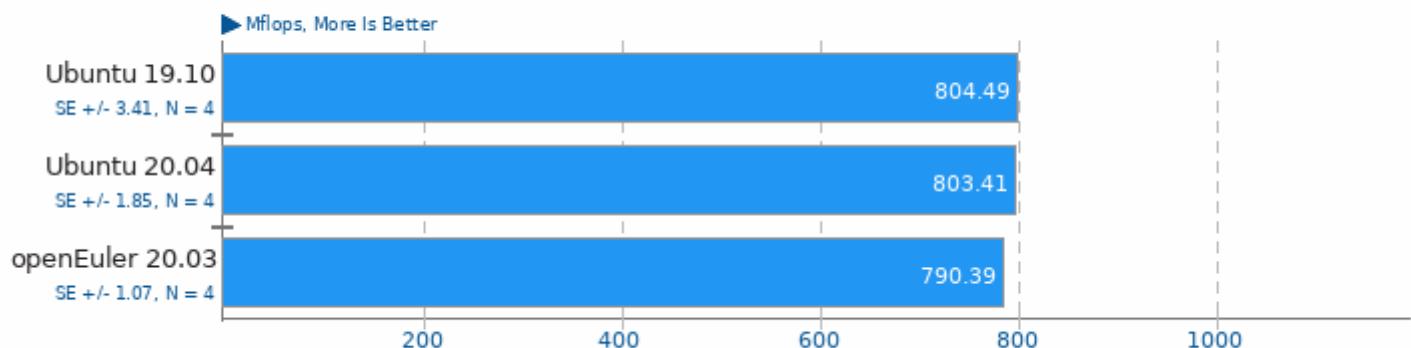
## Java SciMark v2.0

Computational Test: Composite



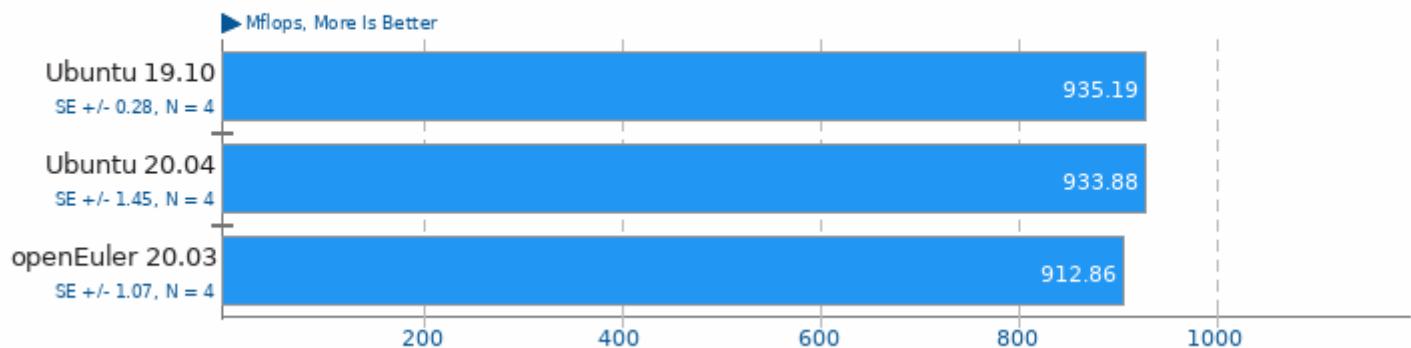
## Java SciMark v2.0

Computational Test: Monte Carlo



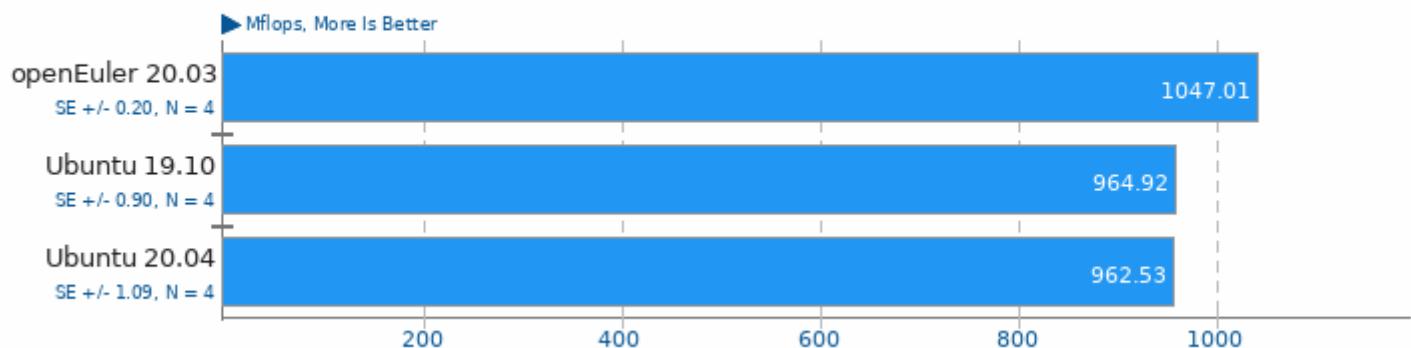
## Java SciMark v2.0

Computational Test: Fast Fourier Transform



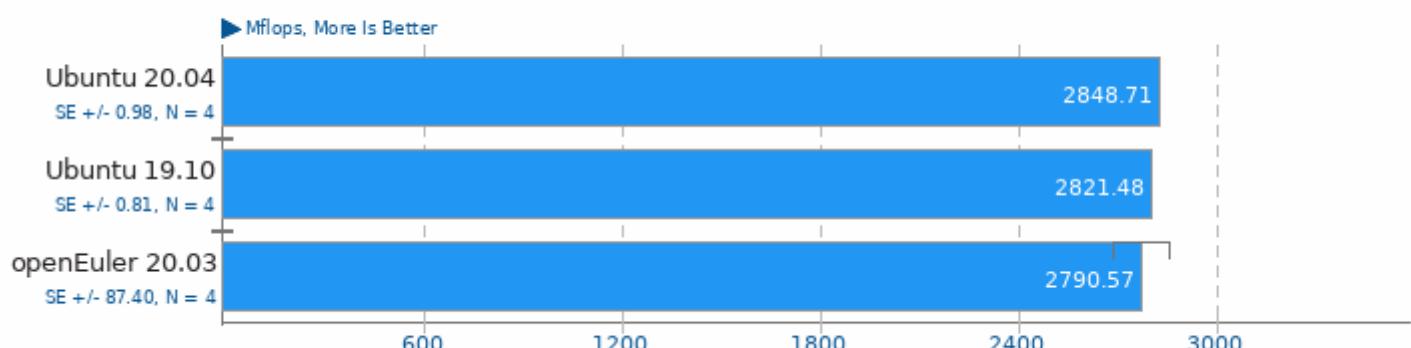
## Java SciMark v2.0

Computational Test: Sparse Matrix Multiply



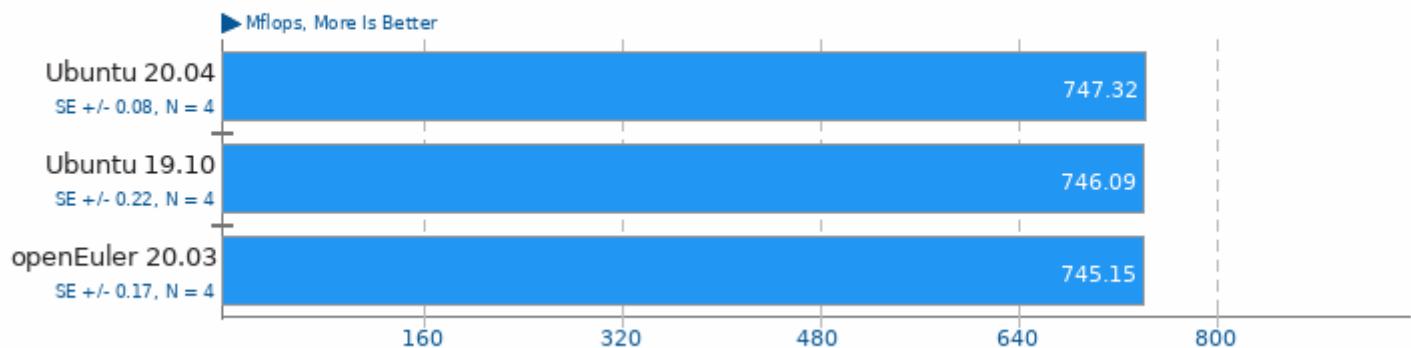
## Java SciMark v2.0

Computational Test: Dense LU Matrix Factorization



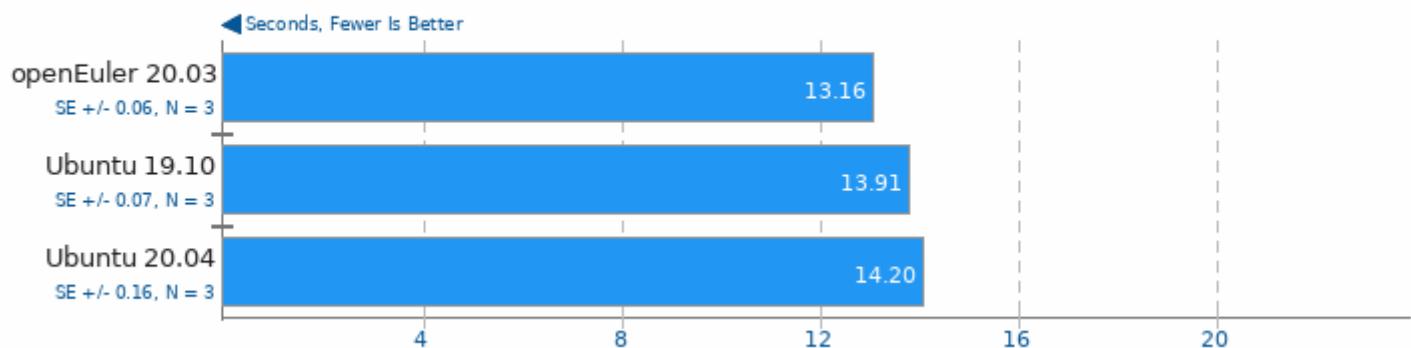
## Java SciMark v2.0

Computational Test: Jacobi Successive Over-Relaxation



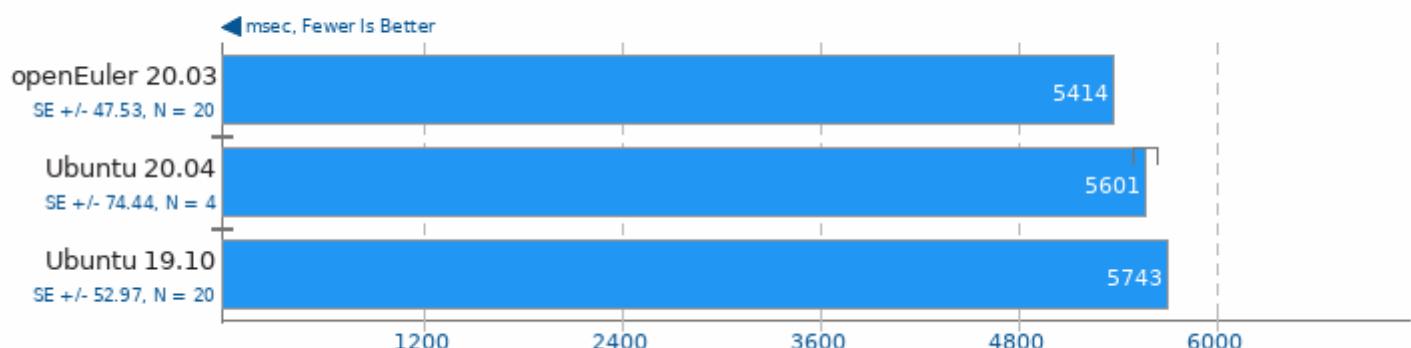
## Bork File Encrypter v1.4

File Encryption Time



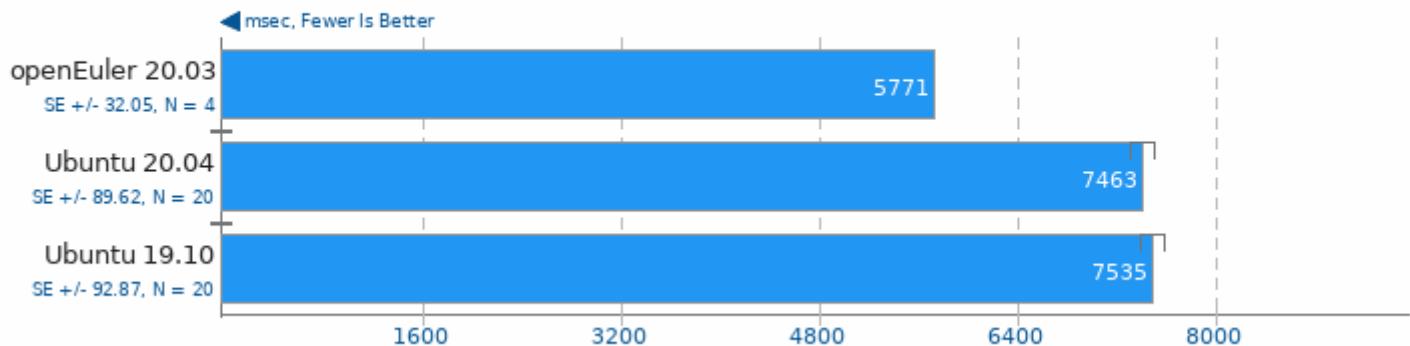
## DaCapo Benchmark v9.12-MR1

Java Test: H2



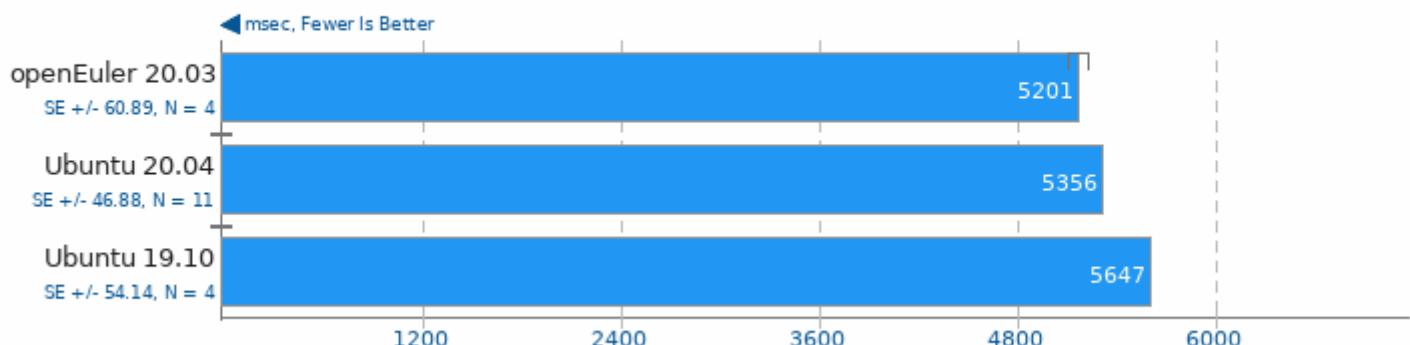
## DaCapo Benchmark v9.12-MR1

Java Test: Jython



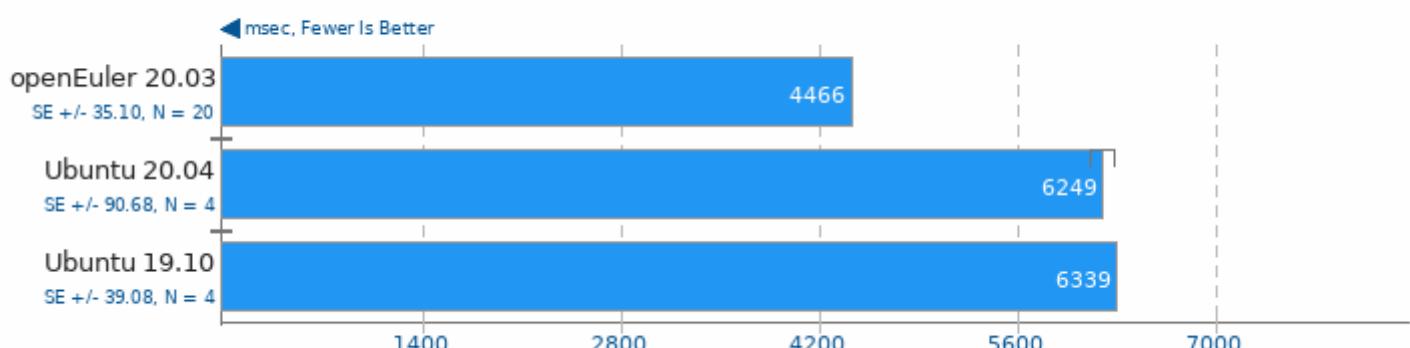
## DaCapo Benchmark v9.12-MR1

Java Test: Tradesoap



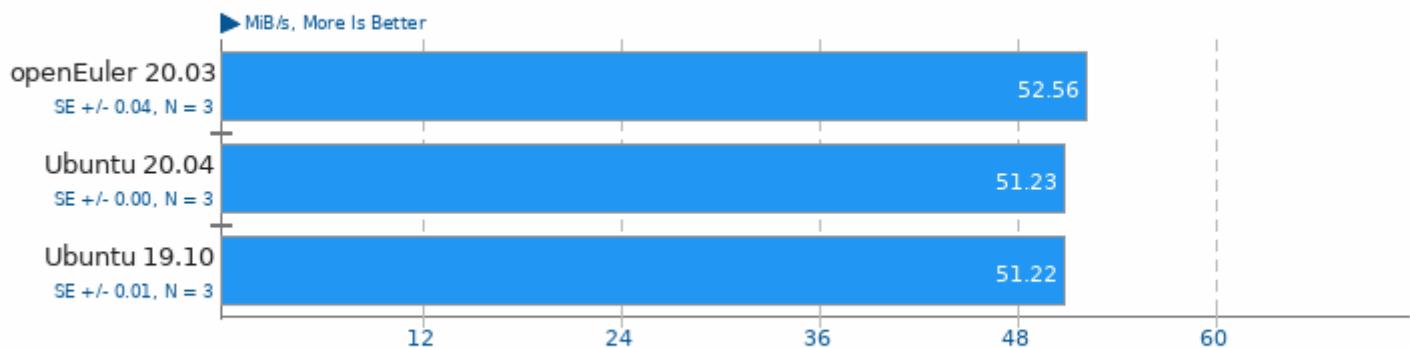
## DaCapo Benchmark v9.12-MR1

Java Test: Tradebeans



## Botan v2.13.0

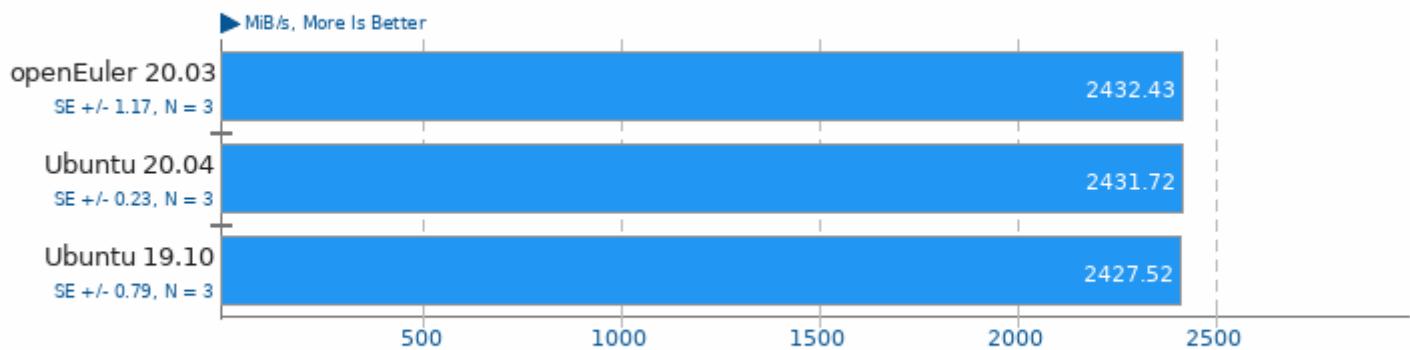
Test: KASUMI



1. (CXX) g++ options: -fstack-protector -pthread -lbotan-2 -ldl -lrt

## Botan v2.13.0

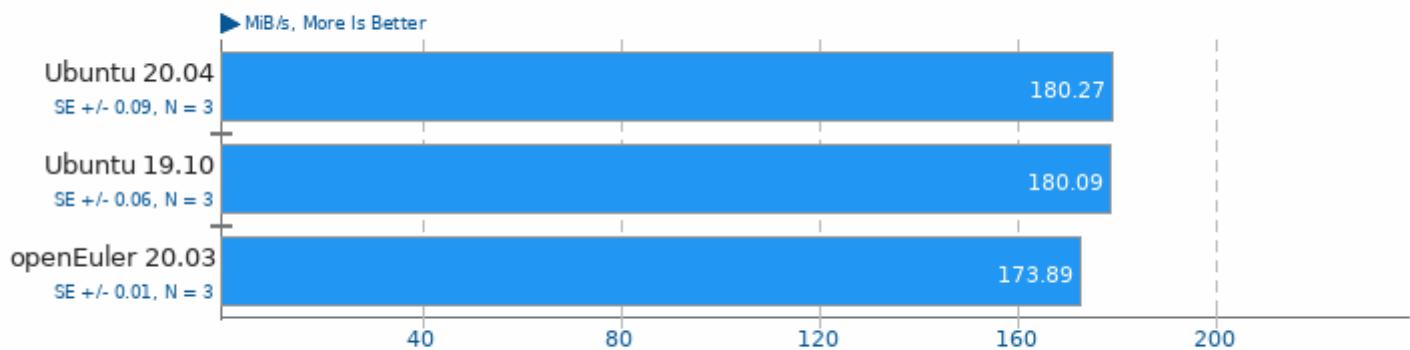
Test: AES-256



1. (CXX) g++ options: -fstack-protector -pthread -lbotan-2 -ldl -lrt

## Botan v2.13.0

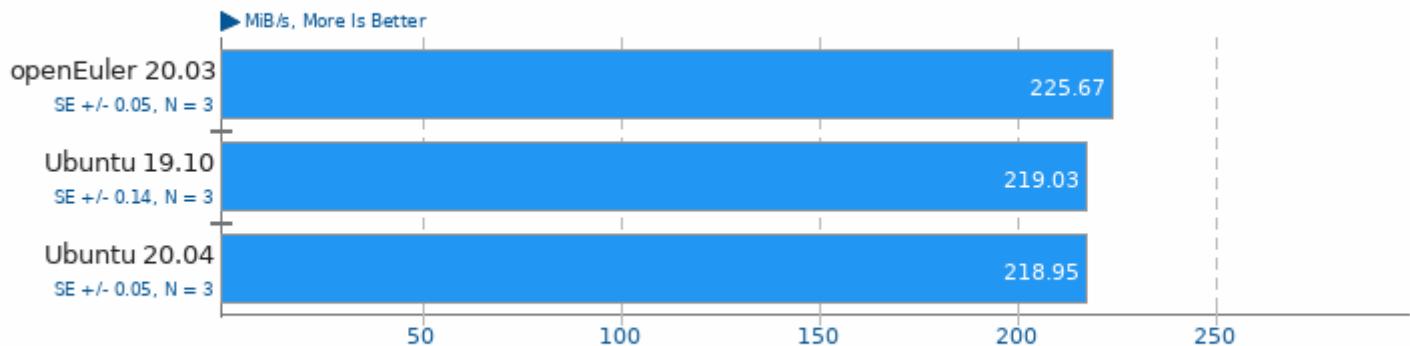
Test: Twofish



1. (CXX) g++ options: -fstack-protector -pthread -lbotan-2 -ldl -lrt

## Botan v2.13.0

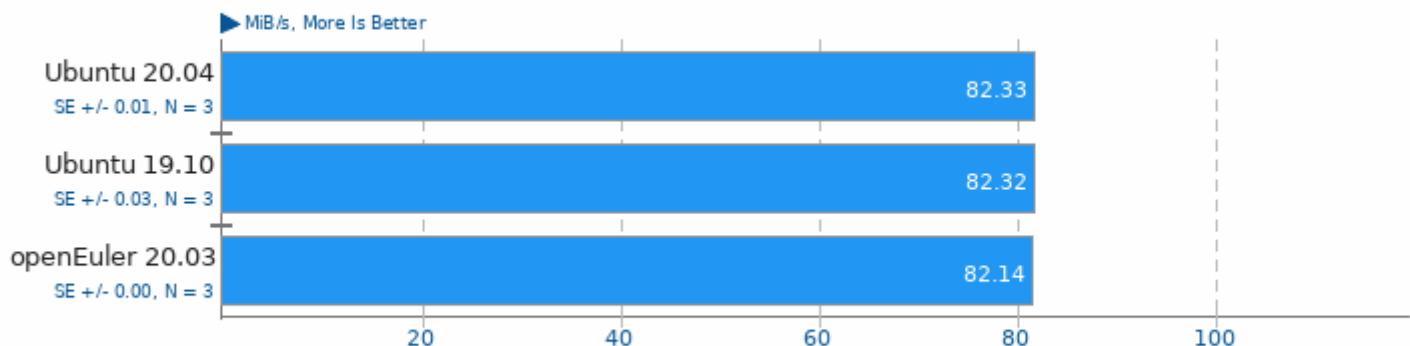
Test: Blowfish



1. (CXX) g++ options: -fstack-protector -pthread -lbotan-2 -ldl -lrt

## Botan v2.13.0

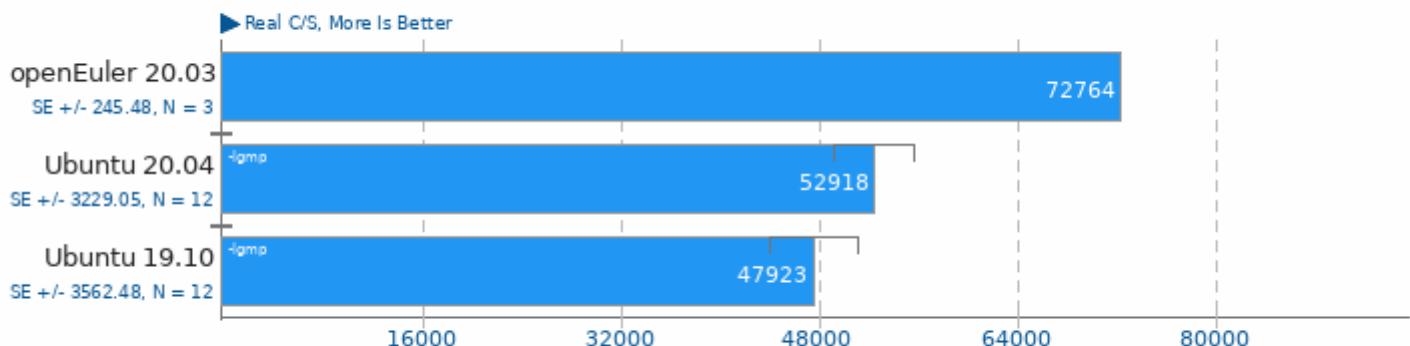
Test: CAST-256



1. (CXX) g++ options: -fstack-protector -pthread -lbotan-2 -ldl -lrt

## John The Ripper v1.9.0-jumbo-1

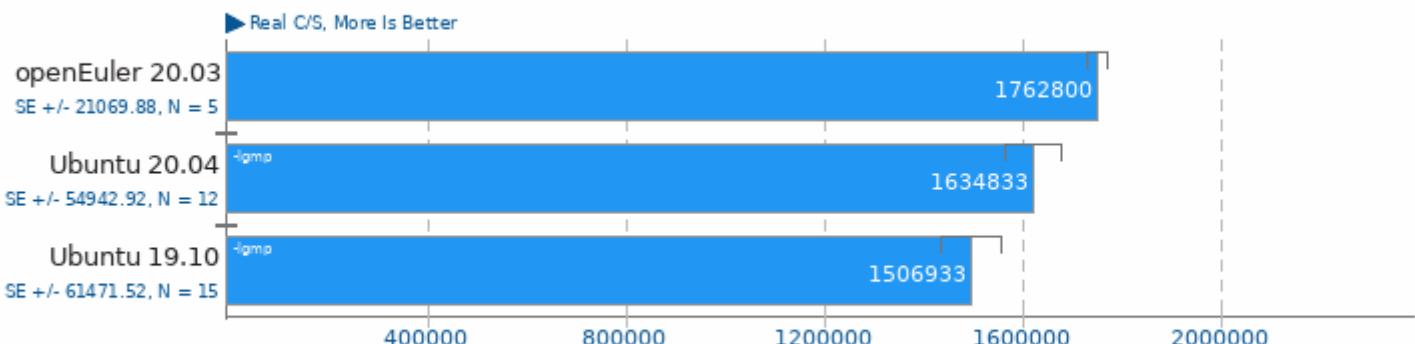
Test: Blowfish



1. (CC) gcc options: -lssl -lcrypto -fopenmp -pthread -lm -lz -ldl -lcrypt

## John The Ripper v1.9.0-jumbo-1

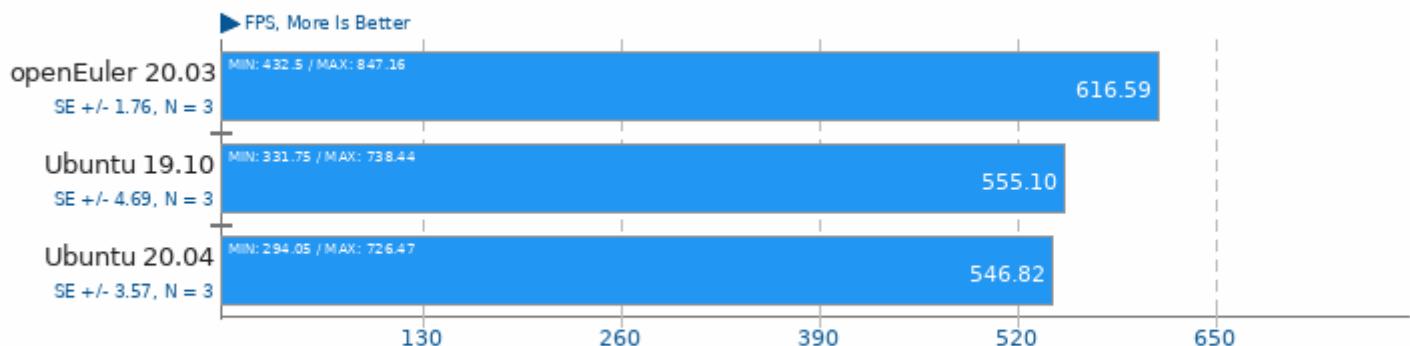
Test: MD5



1. (CC) gcc options: -lssl -lcrypto -fopenmp -pthread -lm -lz -ldl -lcrypt

## dav1d v0.6.0

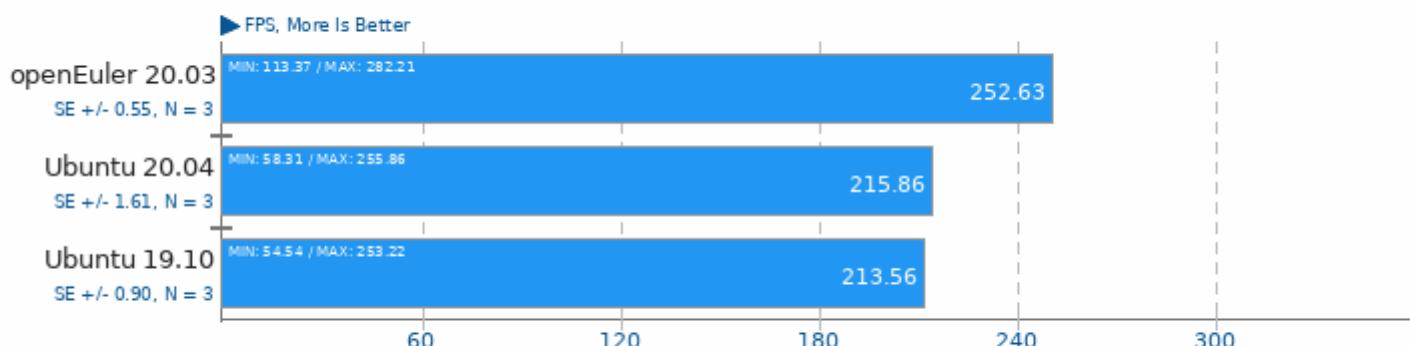
Video Input: Chimera 1080p



1. (CC) gcc options: -pthread

## dav1d v0.6.0

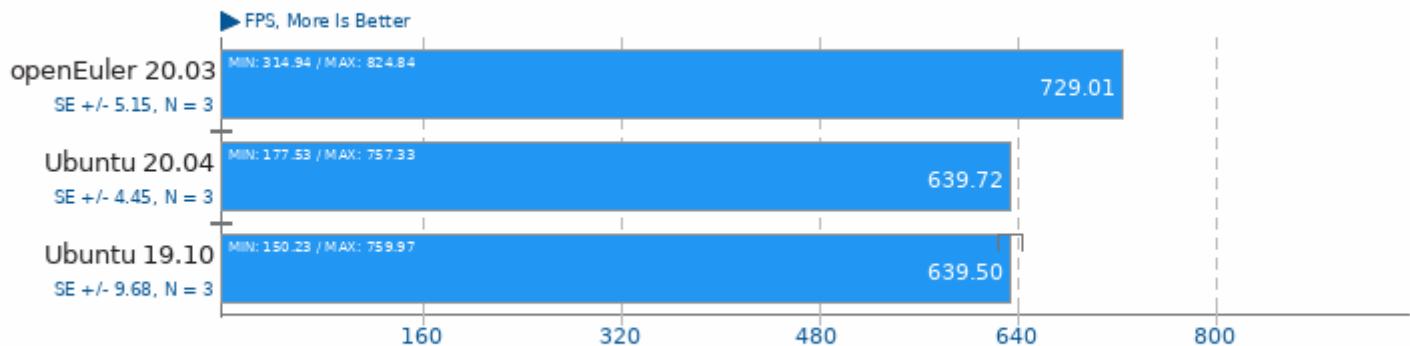
Video Input: Summer Nature 4K



1. (CC) gcc options: -pthread

## dav1d v0.6.0

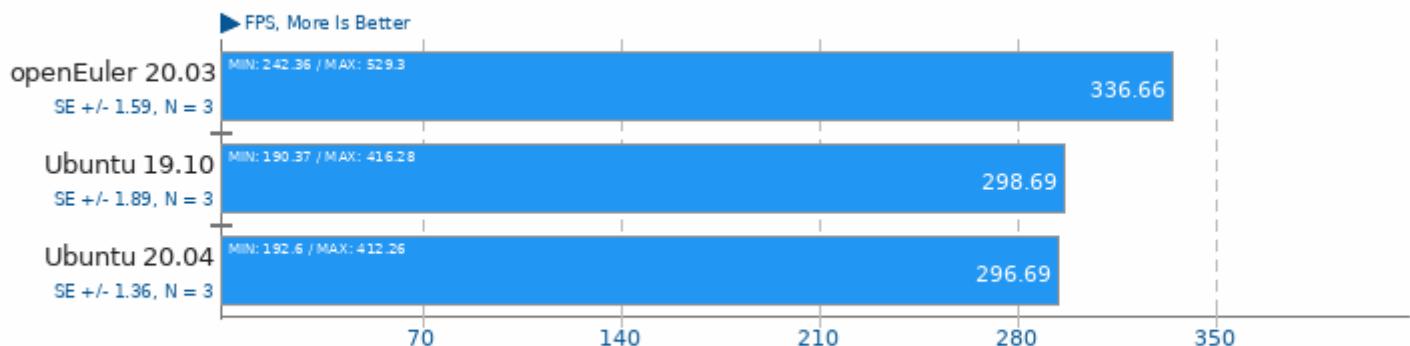
Video Input: Summer Nature 1080p



1. (CC) gcc options: -pthread

## dav1d v0.6.0

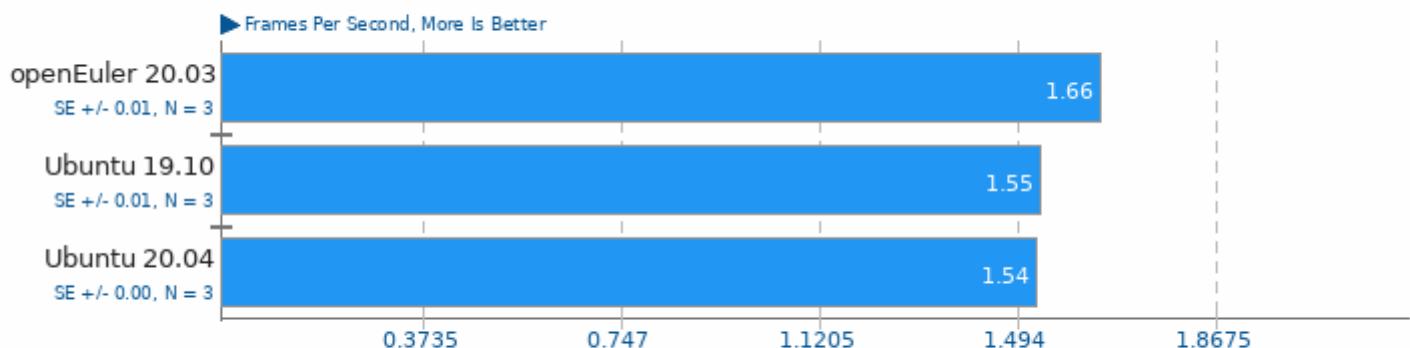
Video Input: Chimera 1080p 10-bit



1. (CC) gcc options: -pthread

## VP9 libvpx Encoding v1.8.2

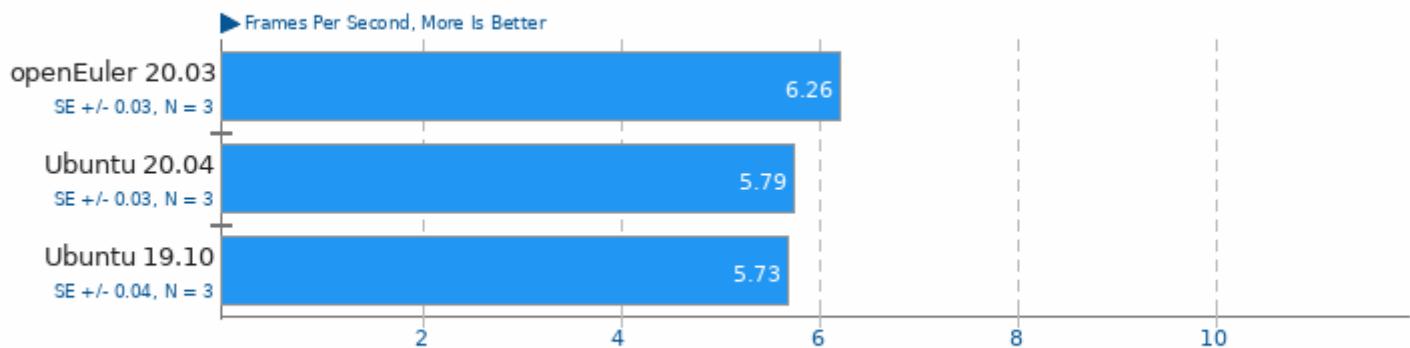
Speed: Speed 0



1. (CXX) g++ options: -lm -lpthread -march=armv8-a -O3 -fPIC -U\_FORTIFY\_SOURCE -std=c++11

## VP9 libvpx Encoding v1.8.2

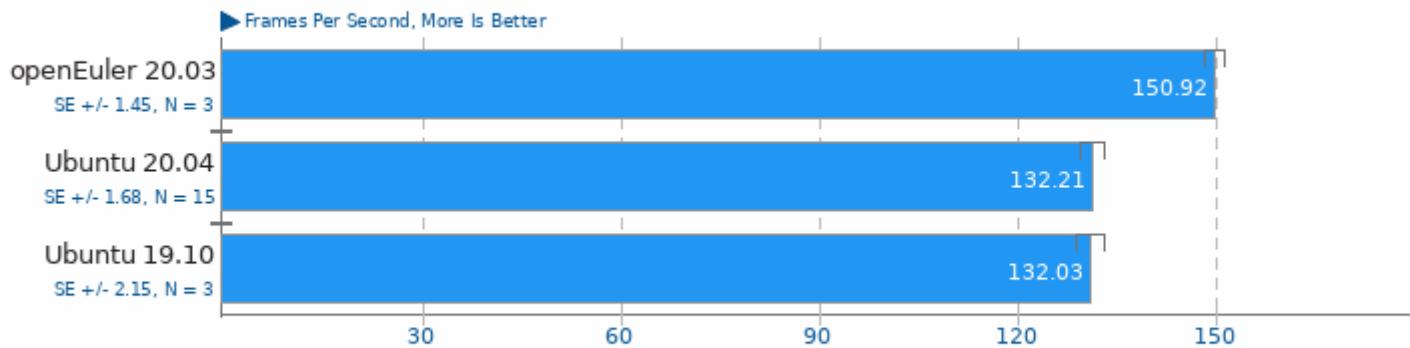
Speed: Speed 5



1. (CXX) g++ options: -lm -lpthread -march=armv8-a -O3 -fPIC -U\_FORTIFY\_SOURCE -std=c++11

## x264 v2019-12-17

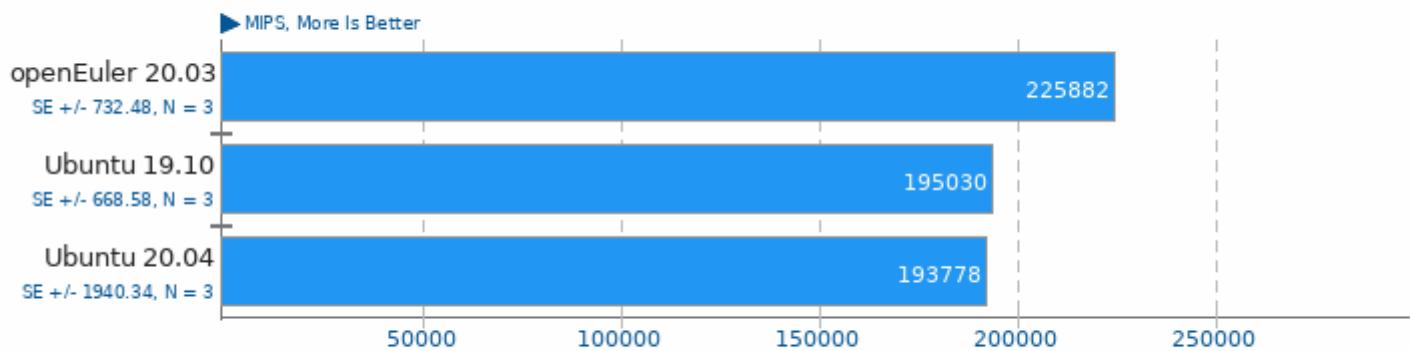
H.264 Video Encoding



1. (CC) gcc options: -ldl -lm -lpthread

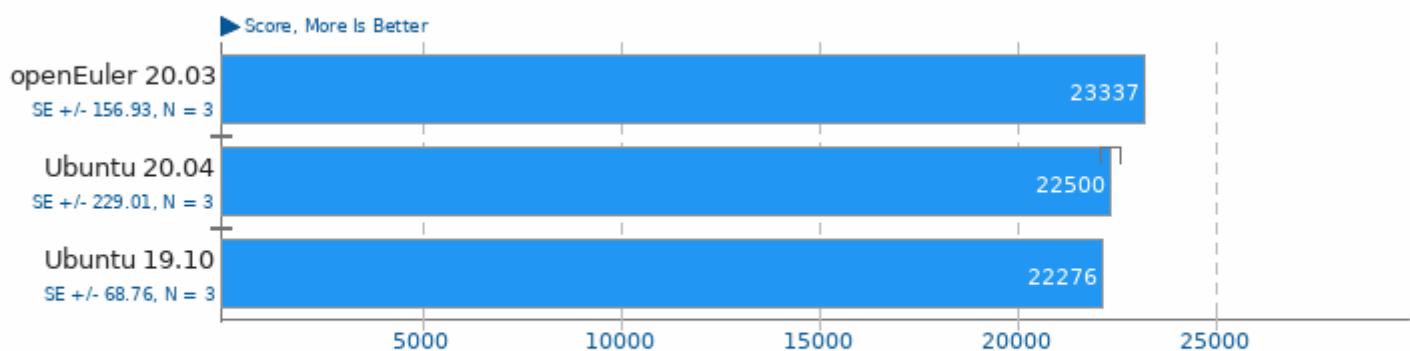
## 7-Zip Compression v16.02

Compress Speed Test



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

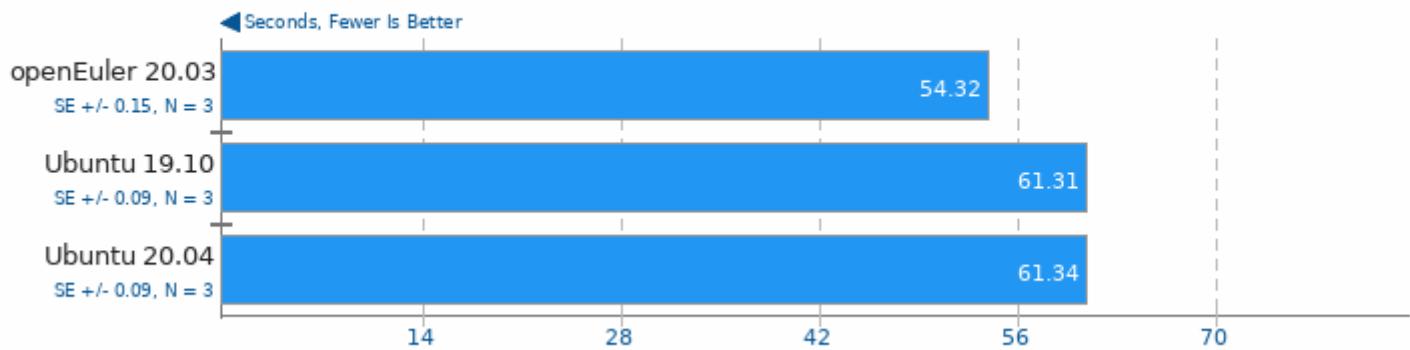
## Node.js Octane Benchmark



1. openEuler 20.03: Nodejs v10.15.2
2. Ubuntu 20.04: Nodejs v10.15.2
3. Ubuntu 19.10: Nodejs v10.15.2

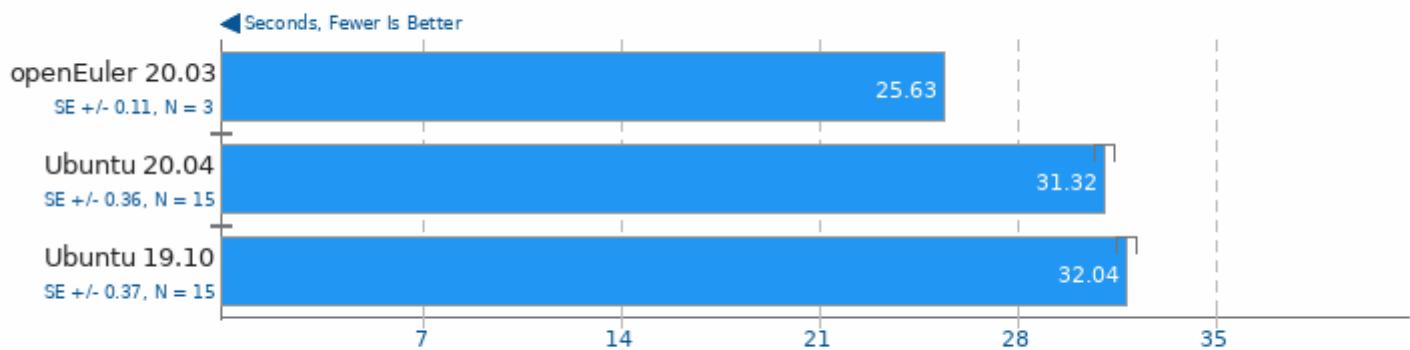
## Gzip Compression

Linux Source Tree Archiving To .tar.gz



## XZ Compression v5.2.4

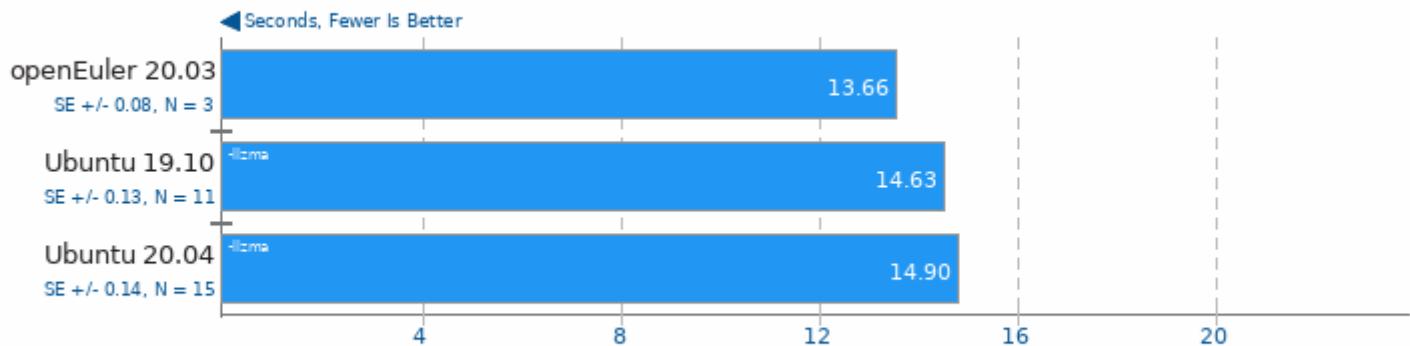
Compressing ubuntu-16.04.3-server-i386.img, Compression Level 9



1. (CC) gcc options: -pthread -fvisibility=hidden -O2

## Zstd Compression v1.3.4

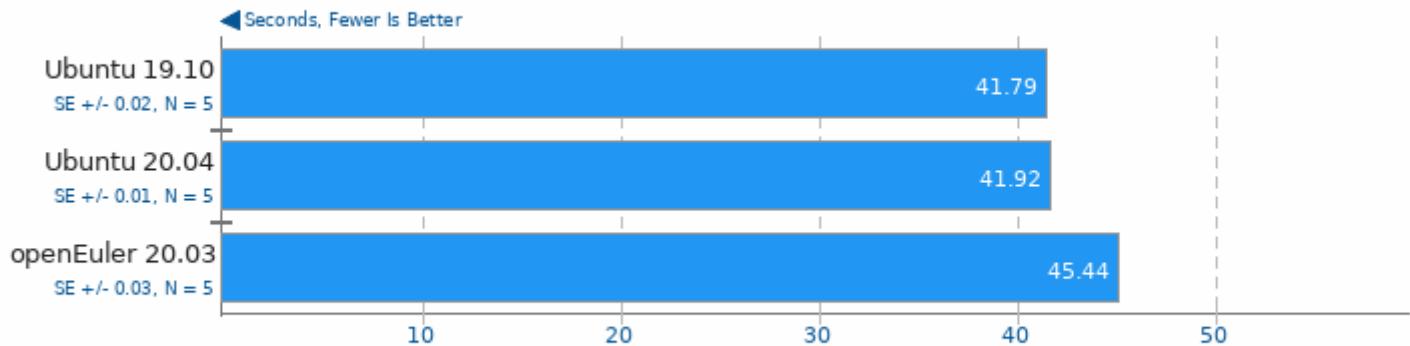
Compressing ubuntu-16.04.3-server-i386.img, Compression Level 19



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

## FLAC Audio Encoding v1.3.2

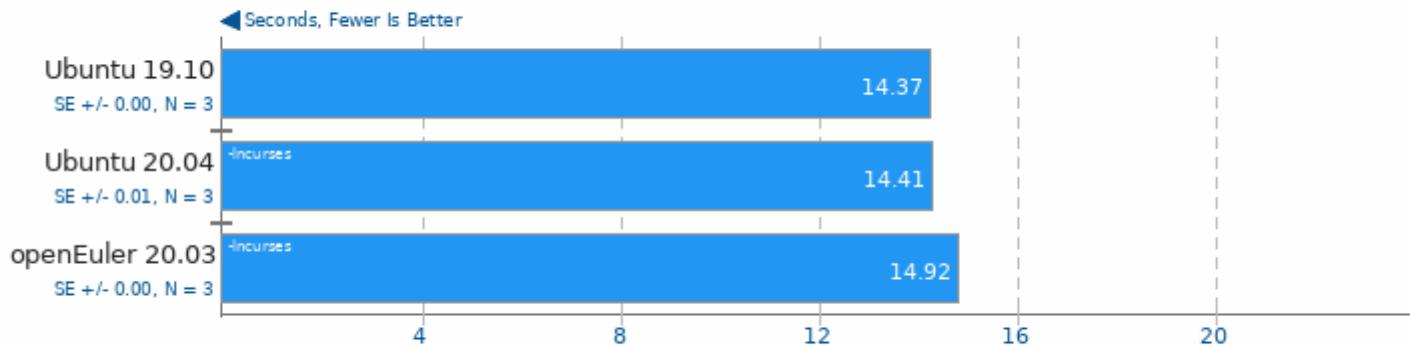
WAV To FLAC



1. (CXX) g++ options: -O2 -visibility=hidden -lm

## LAME MP3 Encoding v3.100

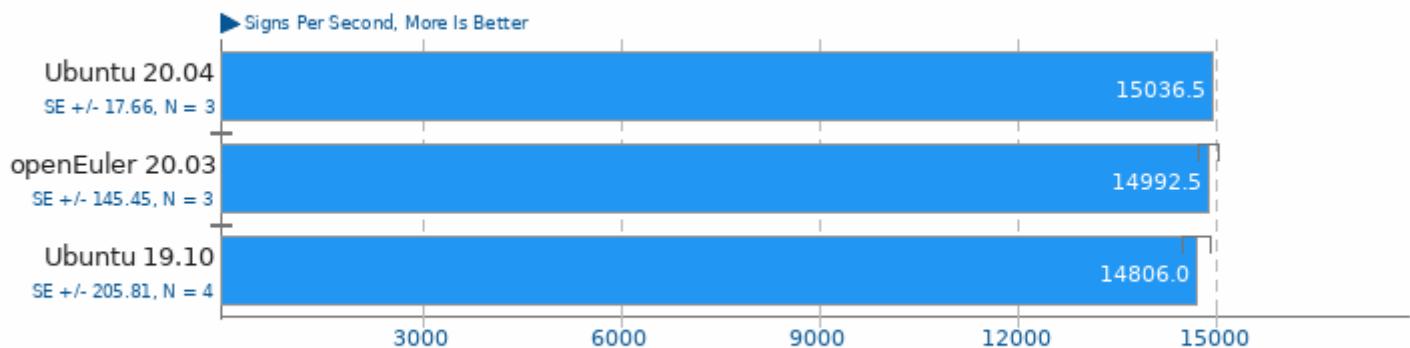
WAV To MP3



1. (CC) gcc options: -O3 -ffast-math -funroll-loops -fschedule-insns2 -fbranch-count-reg -fforce-addr -pipe -lm

## OpenSSL v1.1.1

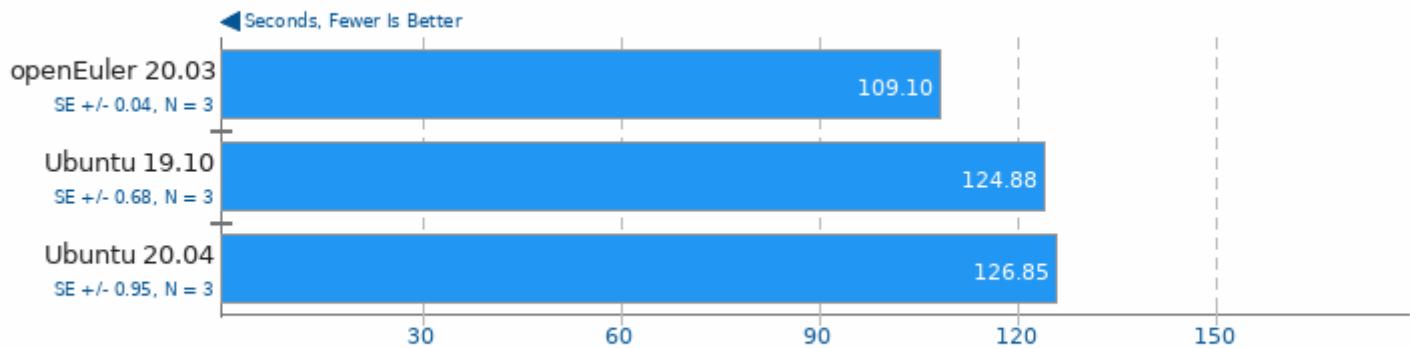
RSA 4096-bit Performance



1. (CC) gcc options: -pthread -O3 -lssl -lcrypto -ldl

## SQLite Speedtest v3.30

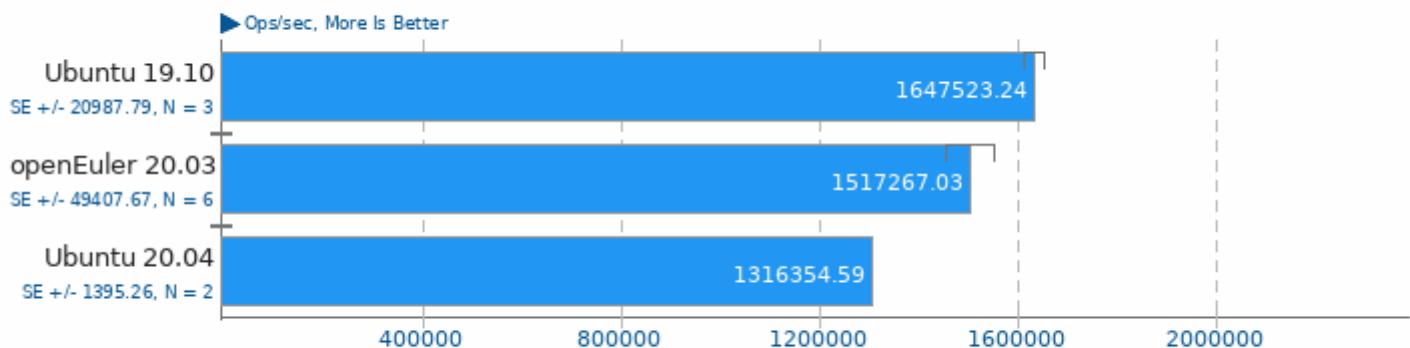
Timed Time - Size 1,000



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

## Memtier\_benchmark v1.2.17

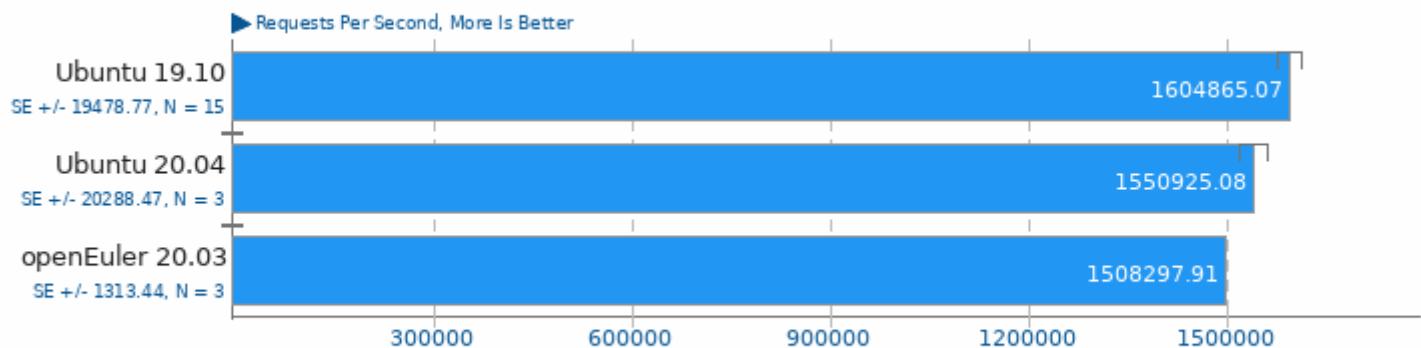
Protocol: Redis



1. (CXX) g++ options: -O2 -levent -lpthread -lz -lpcres

## Redis v5.0.5

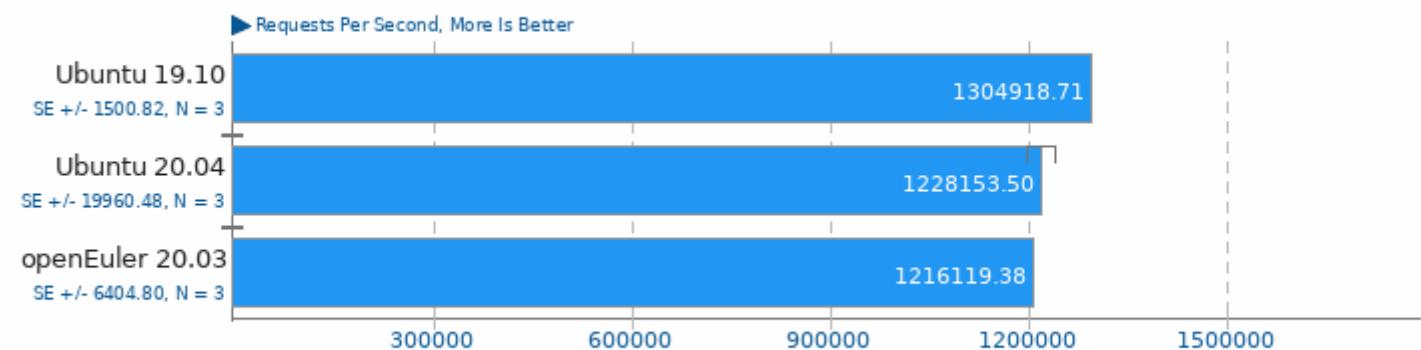
Test: LPOP



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

## Redis v5.0.5

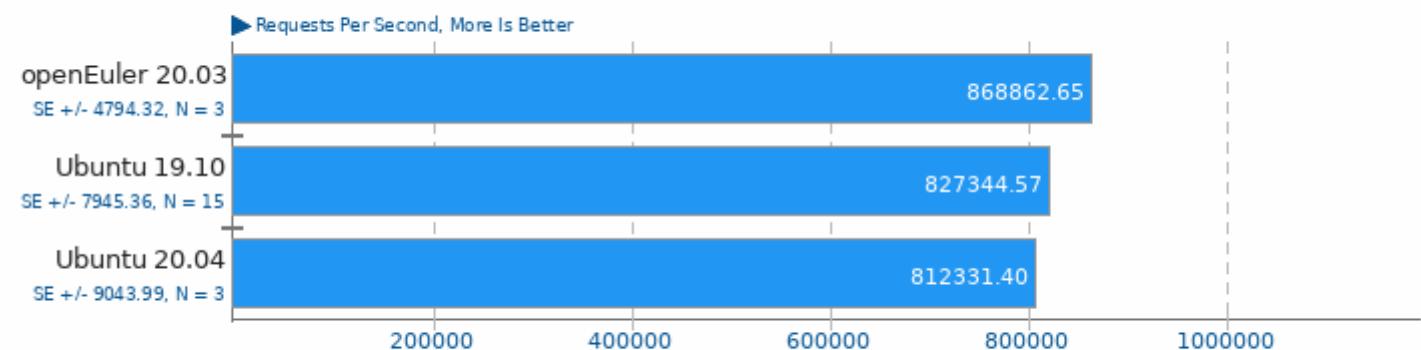
Test: SADD



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

## Redis v5.0.5

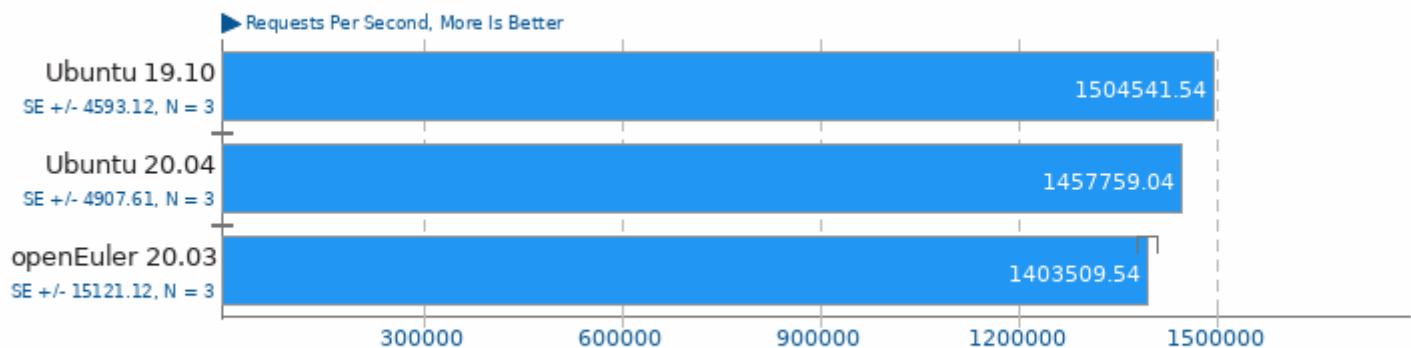
Test: LPUSH



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

## Redis v5.0.5

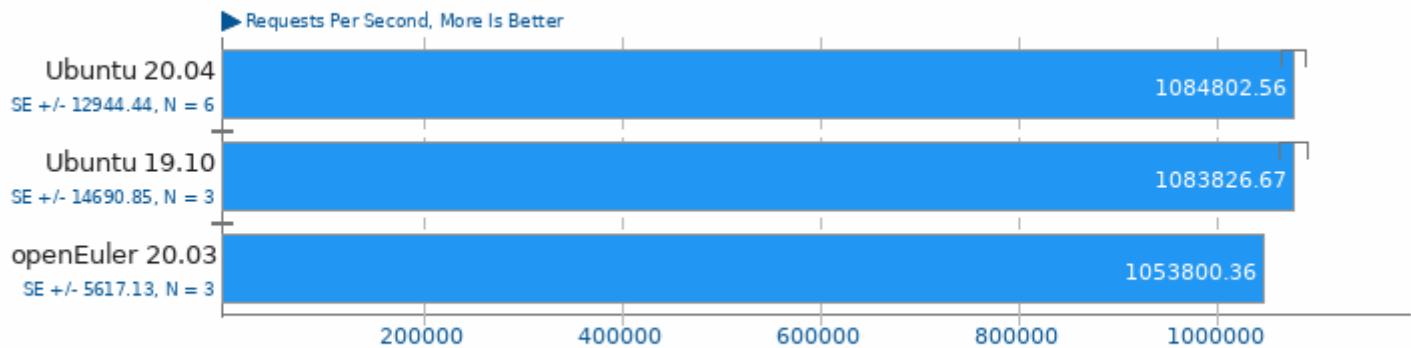
Test: GET



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

## Redis v5.0.5

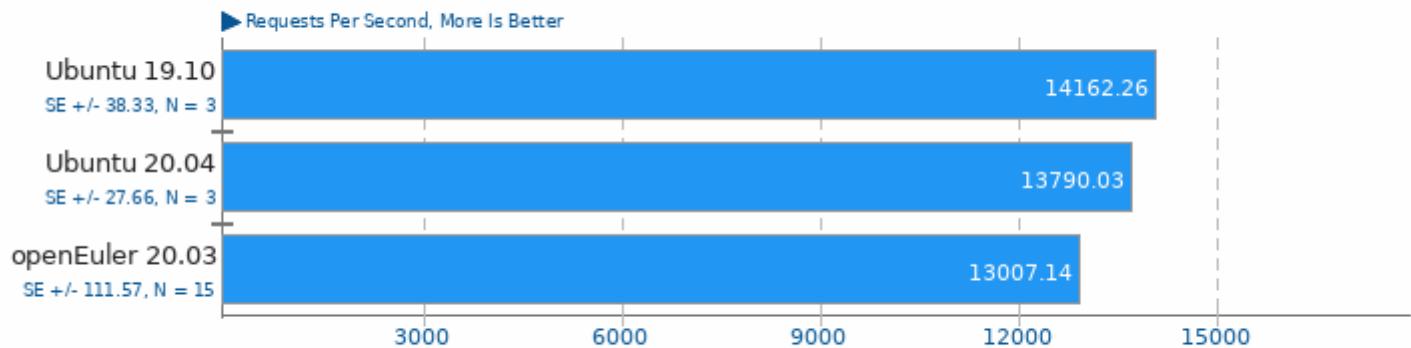
Test: SET



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

## Apache Benchmark v2.4.29

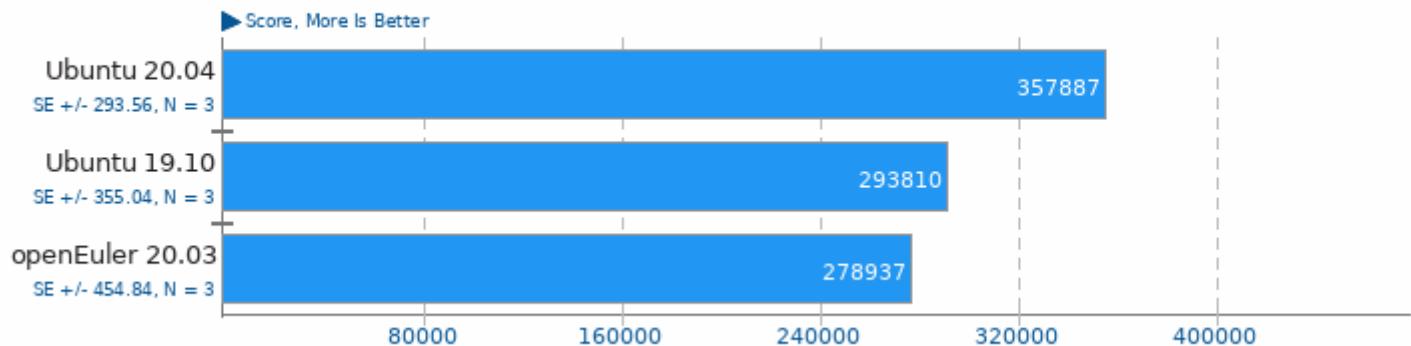
Static Web Page Serving



1. (CC) gcc options: -shared -fPIC -O2 -pthread

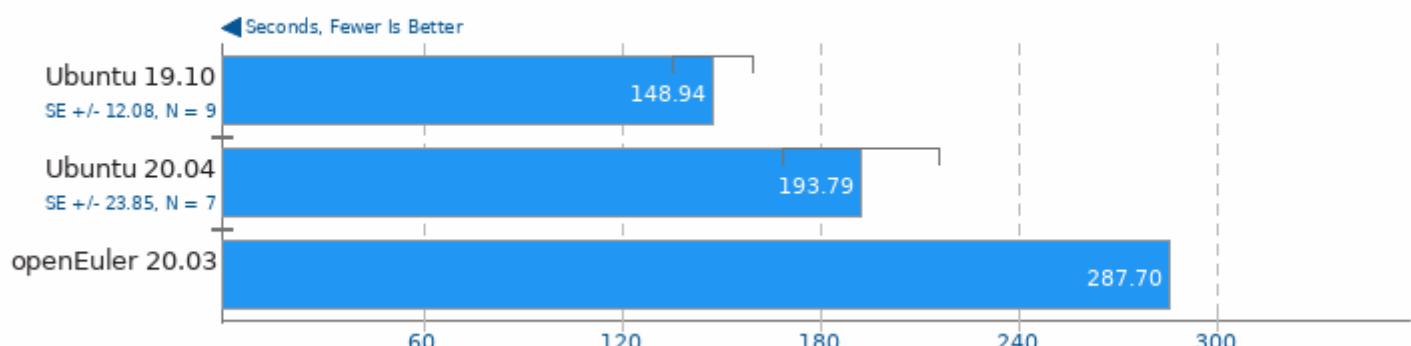
## PHPBench v0.8.1

PHP Benchmark Suite



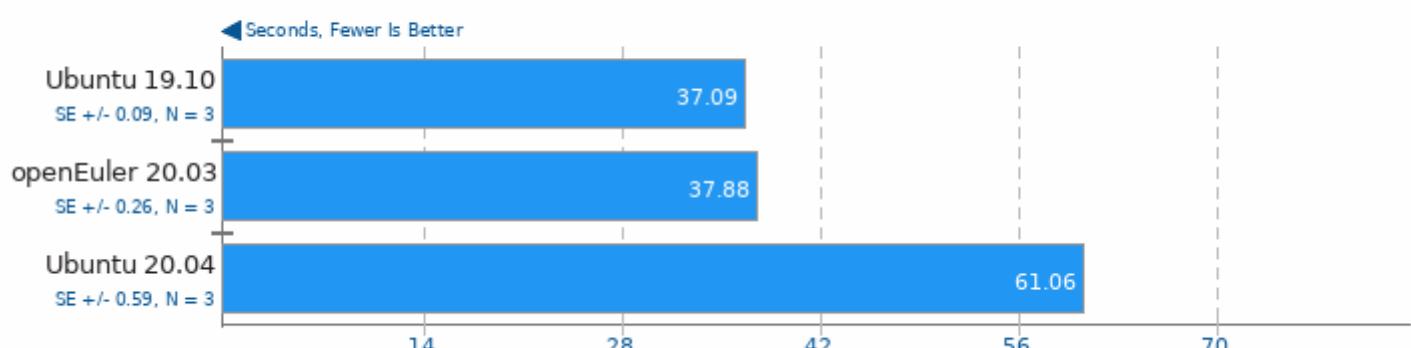
## Milpack Benchmark

Benchmark: scikit\_ica



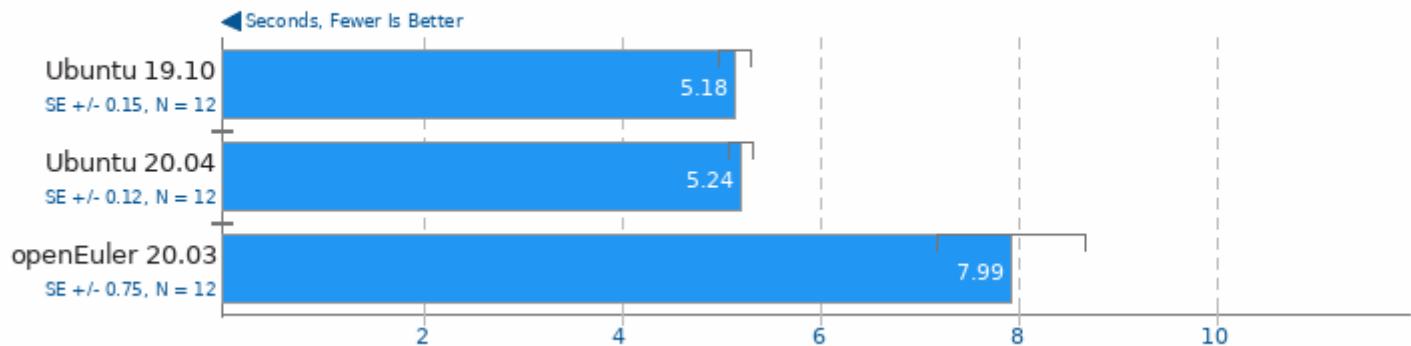
## Milpack Benchmark

Benchmark: scikit\_svm



## Mlpack Benchmark

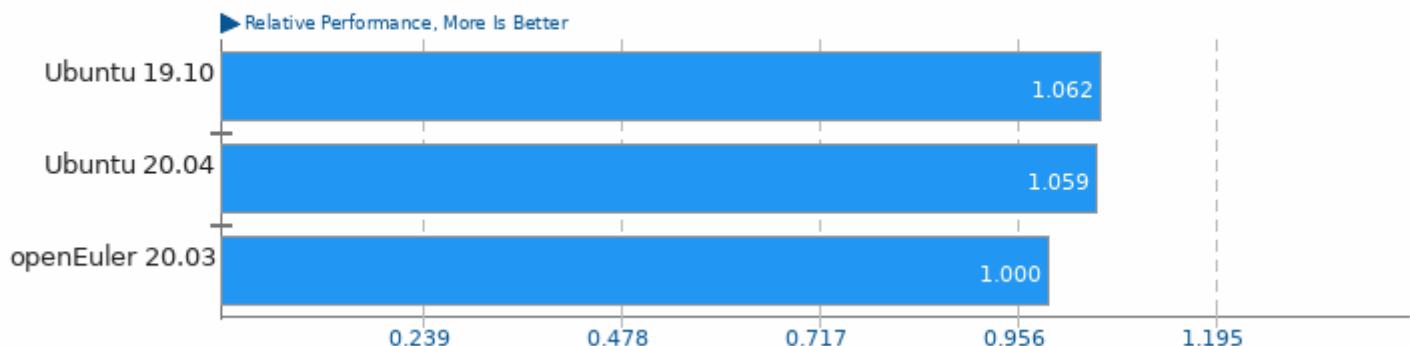
Benchmark: scikit\_linearridge\_regression



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

## Geometric Mean Of Audio Encoding Tests

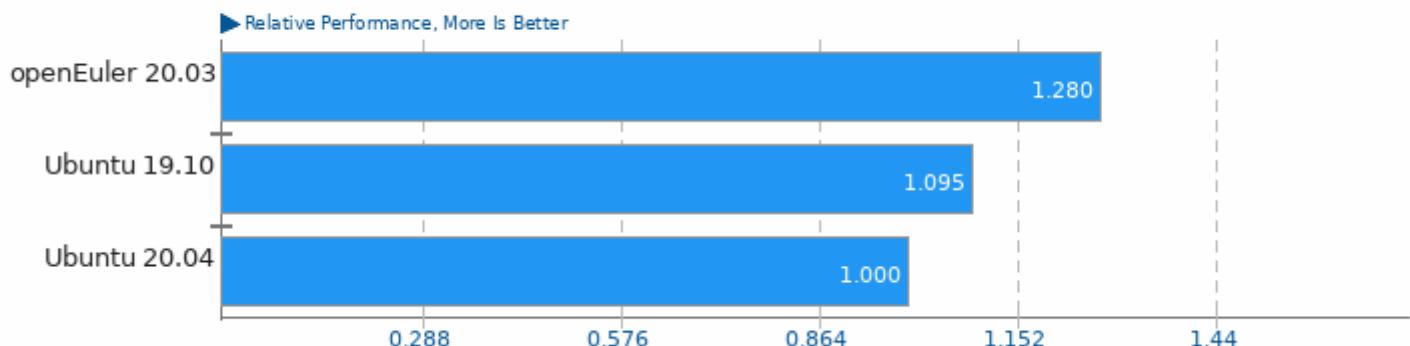
Result Composite



Geometric mean based upon tests: pts/encode-mp3 and pts/encode-flac

## Geometric Mean Of Timed Code Compilation Tests

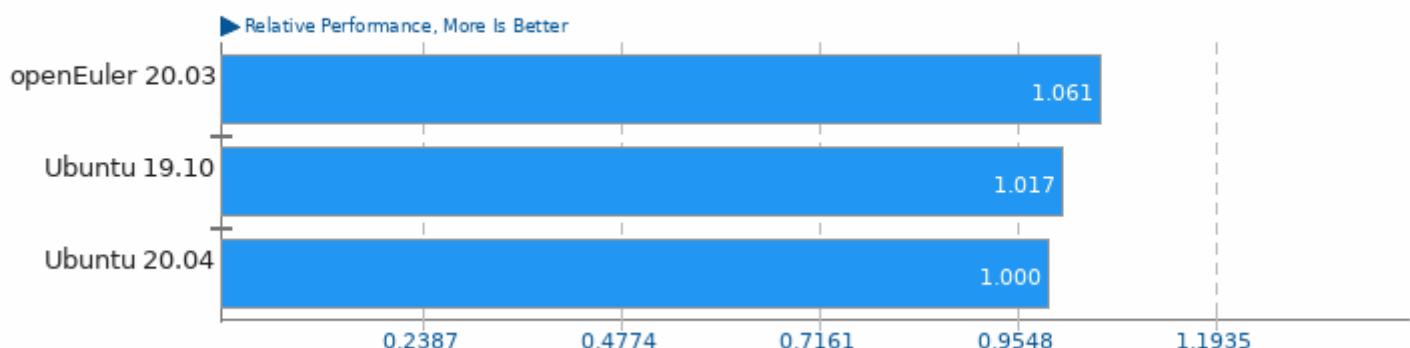
Result Composite



Geometric mean based upon tests: pts/build-linux-kernel and pts/build-llvm

## Geometric Mean Of C/C++ Compiler Tests Tests

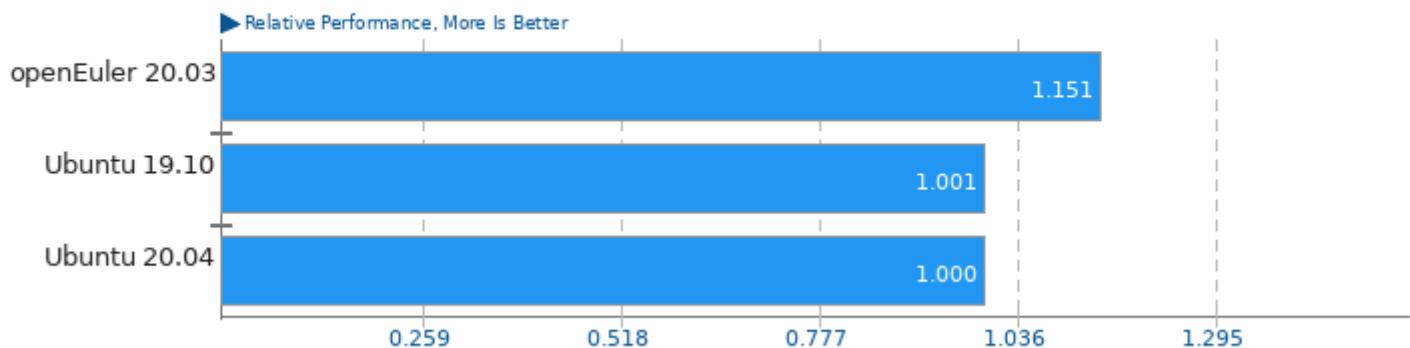
Result Composite



Geometric mean based upon tests: pts/vpxenc, pts/stockfish, pts/build-llvm, pts/c-ray, pts/compress-7zip, pts/encode-mp3, pts/encode-flac and pts/apache

## Geometric Mean Of Timed File Compression Tests

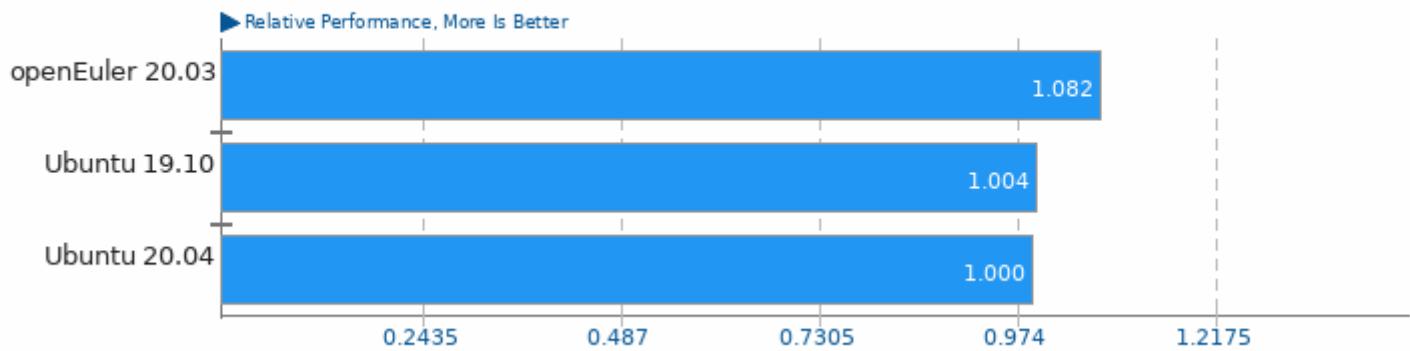
Result Composite



Geometric mean based upon tests: pts/compress-7zip, pts/compress-gzip, pts/compress-zstd and pts/compress-xz

## Geometric Mean Of Creator Workloads Tests

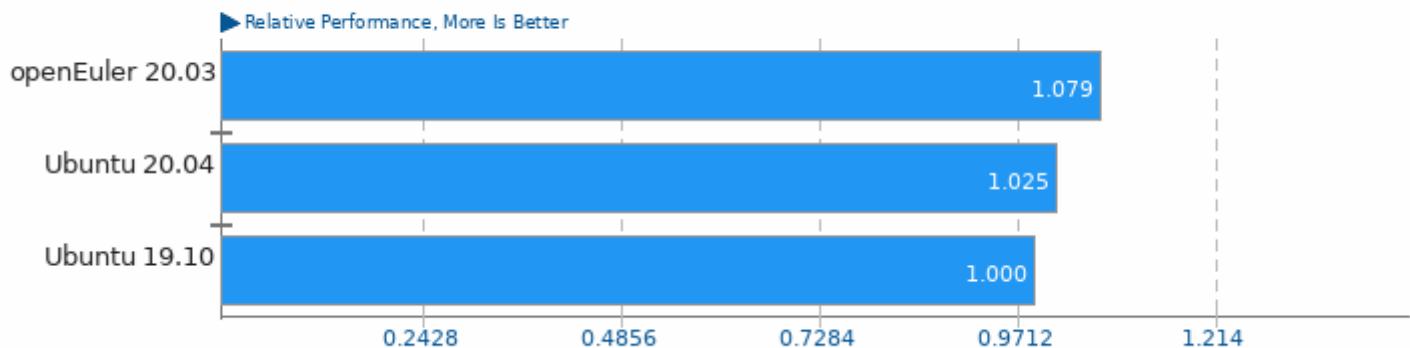
Result Composite



Geometric mean based upon tests: pts/c-ray, pts/povray, pts/smallpt, pts/x264, pts/vpxenc, pts/dav1d, pts/encode-mp3 and pts/encode-flac

## Geometric Mean Of Cryptography Tests

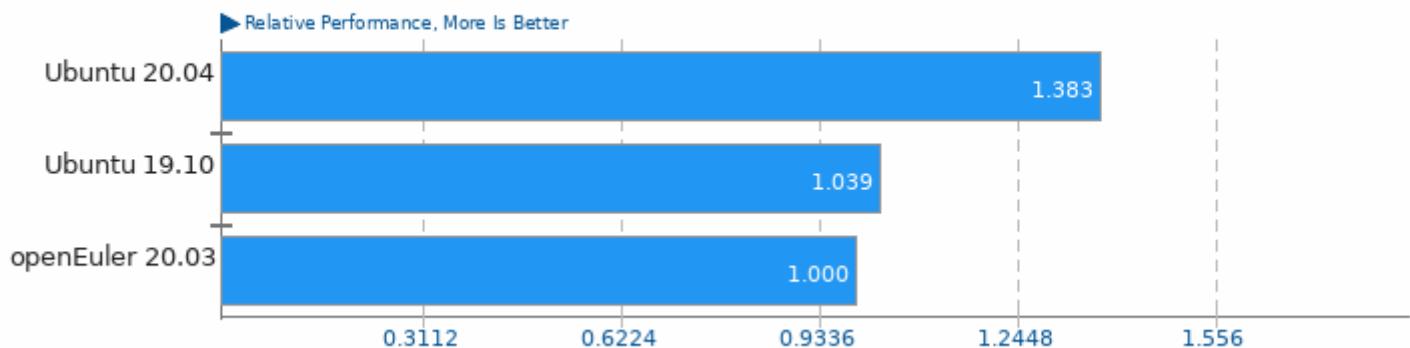
Result Composite



Geometric mean based upon tests: pts/openssl, pts/john-the-ripper and pts/botan

## Geometric Mean Of Database Test Suite Tests

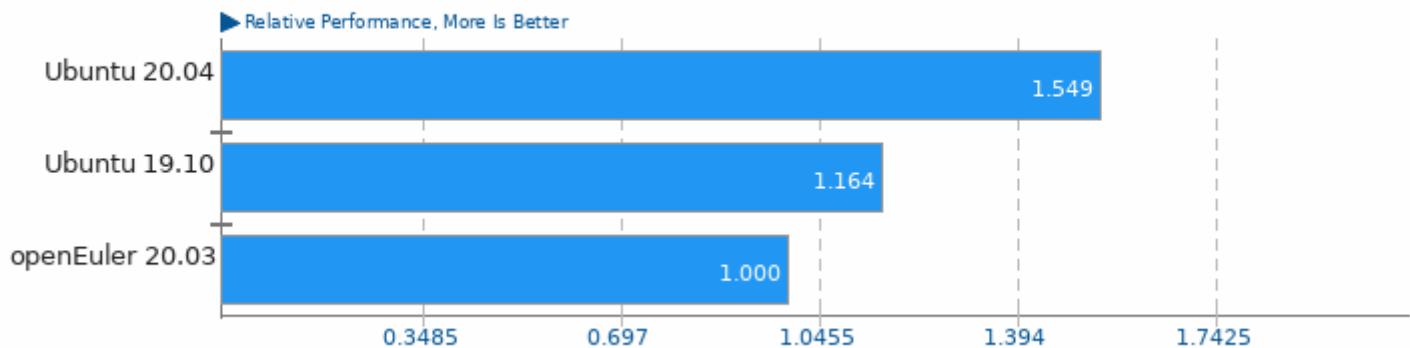
Result Composite



Geometric mean based upon tests: pts/sqlite, pts/sqlite-speedtest and pts/redis

## Geometric Mean Of Disk Test Suite Tests

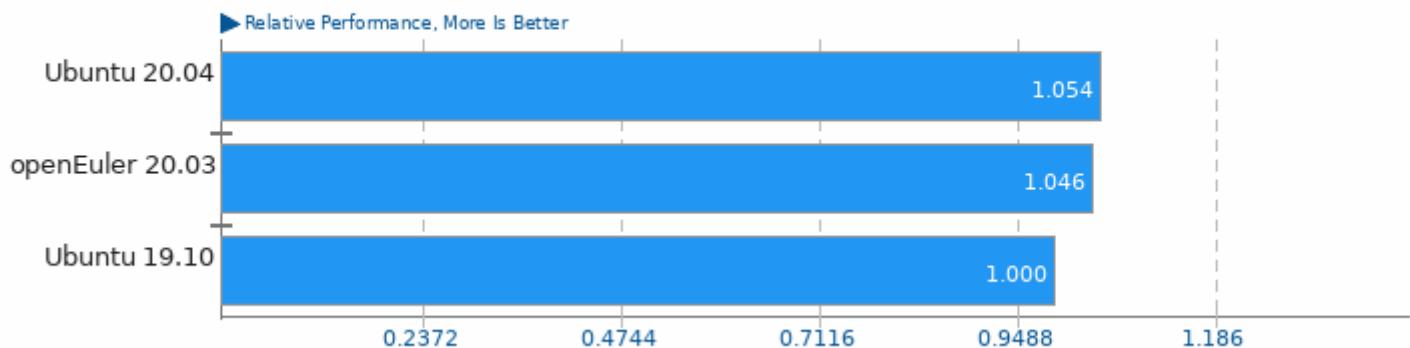
Result Composite



Geometric mean based upon tests: pts/sqlite, pts/dbench and pts/postmark

## Geometric Mean Of HPC - High Performance Computing Tests

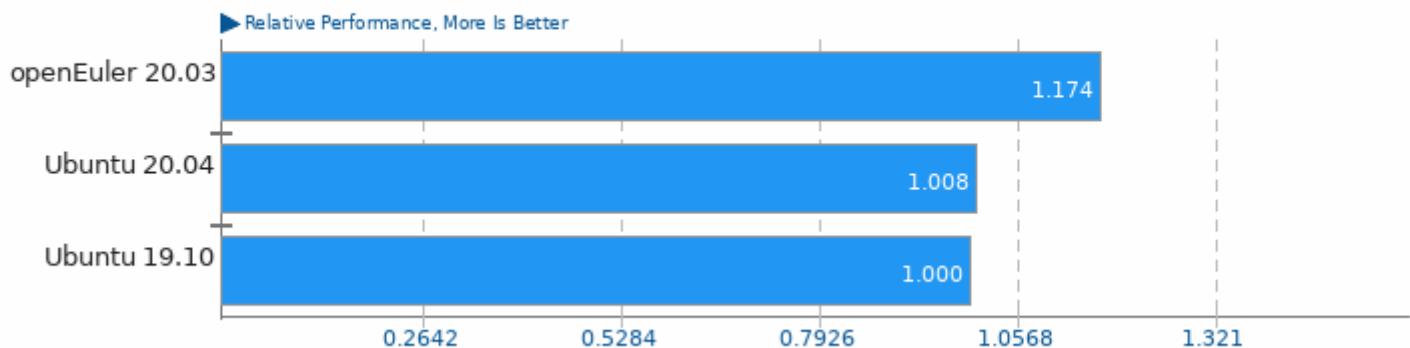
Result Composite



Geometric mean based upon tests: pts/rodinia and pts/hpcg

## Geometric Mean Of Java Tests

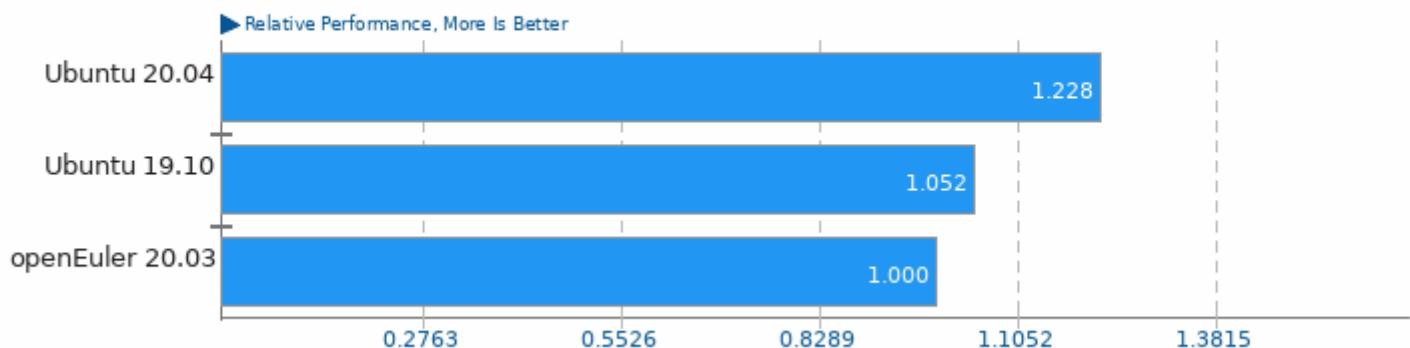
Result Composite



Geometric mean based upon tests: pts/sunflow, pts/bork, pts/java-scimark2 and pts/dacapobench

## Geometric Mean Of Kernel Tests

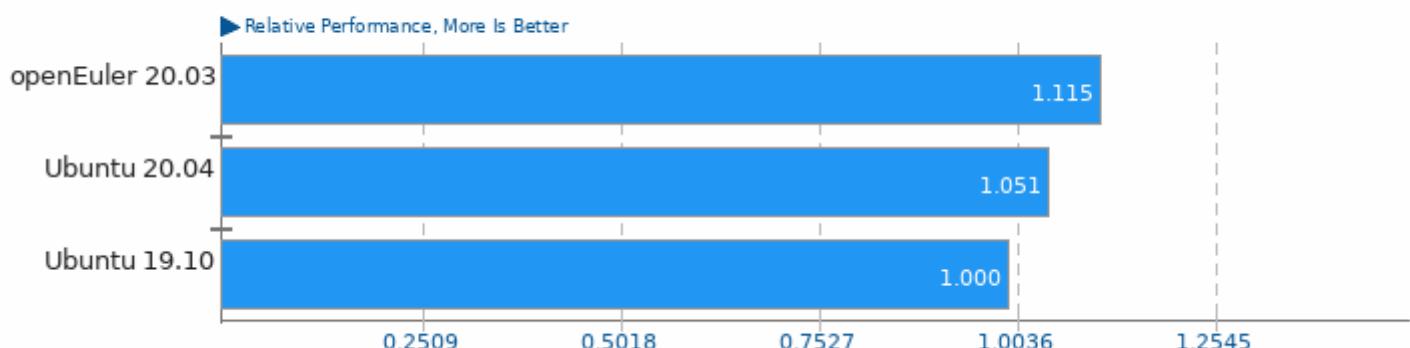
Result Composite



Geometric mean based upon tests: pts/apache, pts/compress-7zip, pts/encode-mp3, pts/x264, pts/openssl, pts/c-ray, pts/sqlite, pts/dbench, pts/postmark and pts/osbench

## Geometric Mean Of Memory Test Suite Tests

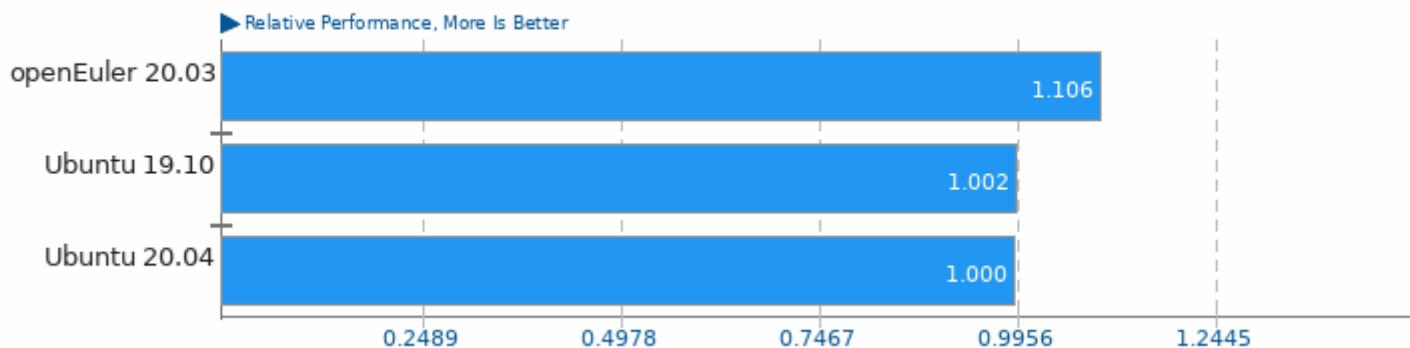
Result Composite



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

## Geometric Mean Of Multi-Core Tests

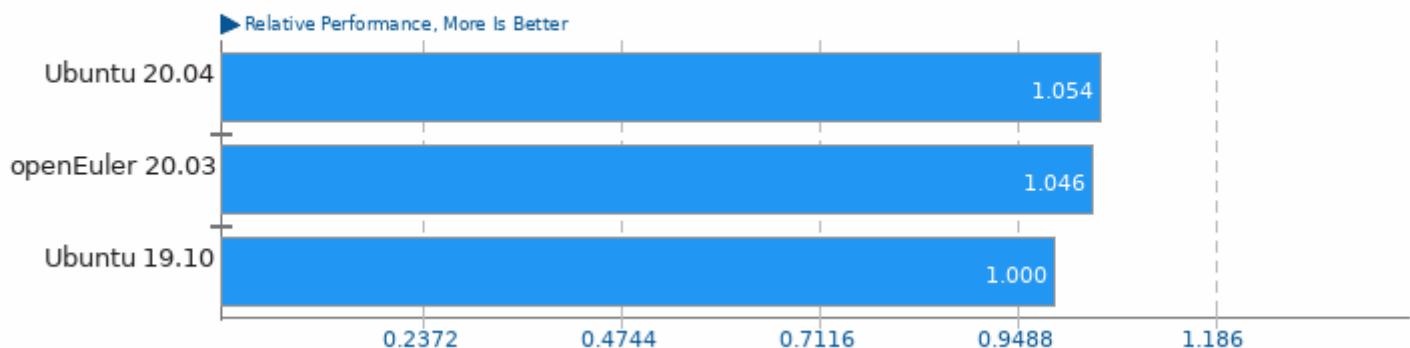
Result Composite



Geometric mean based upon tests: pts/c-ray, pts/stockfish, pts/rodinia, pts/povray, pts/smallpt, pts/compress-7zip, pts/build-linux-kernel and pts/build-llvm

## Geometric Mean Of OpenMPI Tests

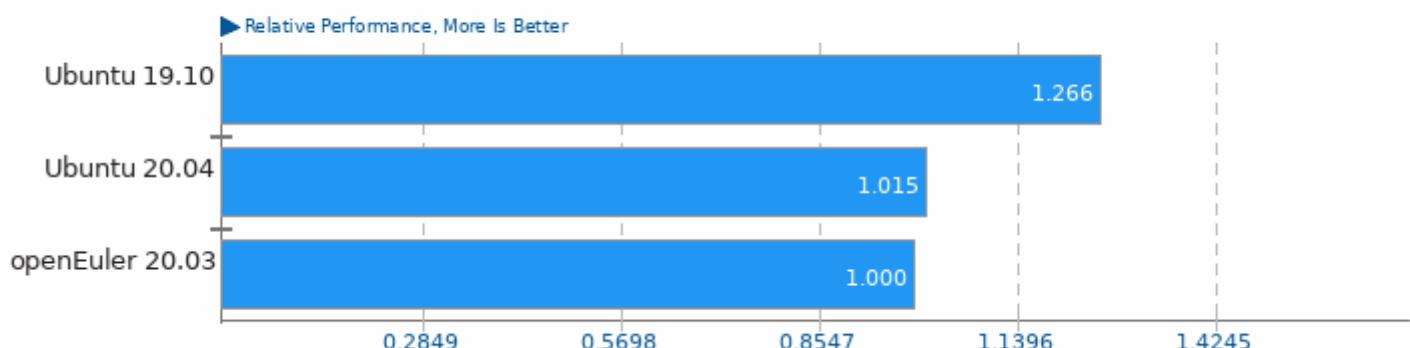
Result Composite



Geometric mean based upon tests: pts/hpcg and pts/rodinia

## Geometric Mean Of Python Tests

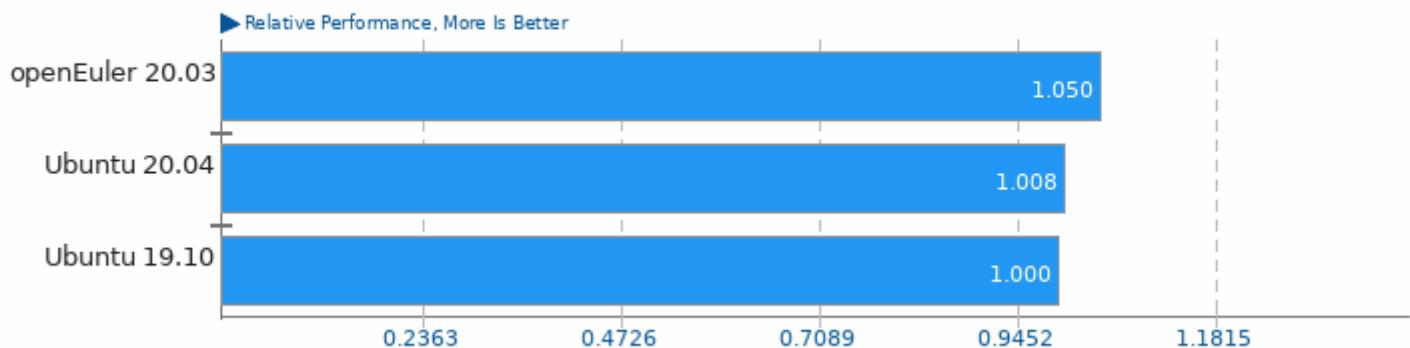
Result Composite



Geometric mean based upon tests: pts/build-llvm and pts/mlpack

## Geometric Mean Of Raytracing Tests

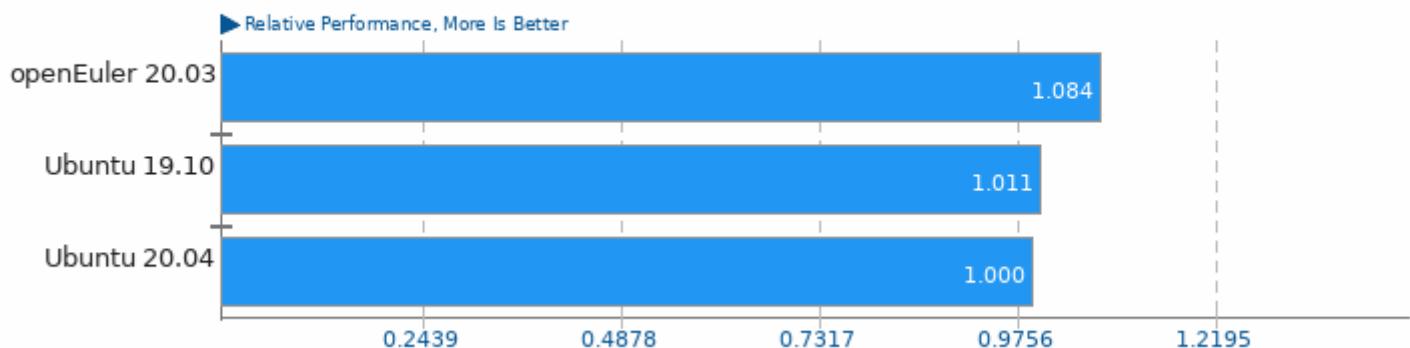
Result Composite



Geometric mean based upon tests: pts/c-ray and pts/povray

## Geometric Mean Of Renderers Tests

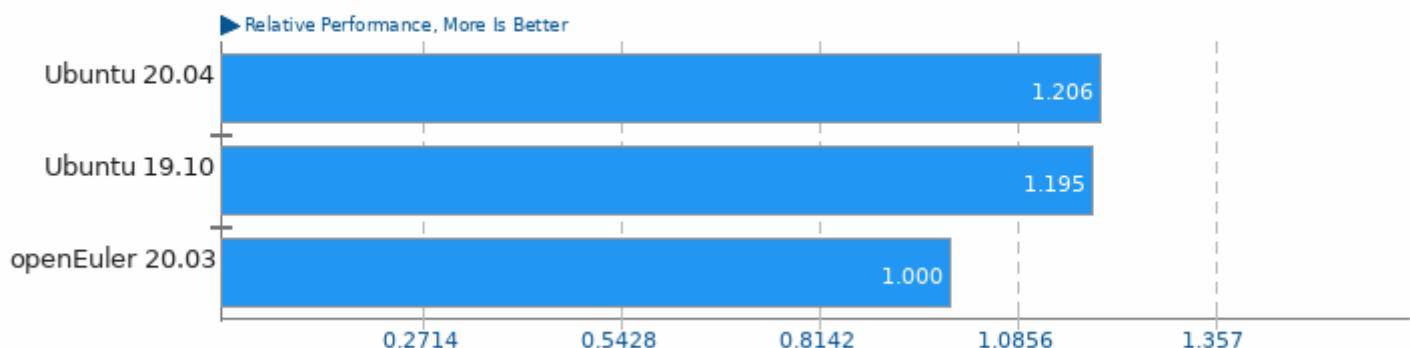
Result Composite



Geometric mean based upon tests: pts/c-ray, pts/povray and pts/smallpt

## Geometric Mean Of Server Motherboard Tests

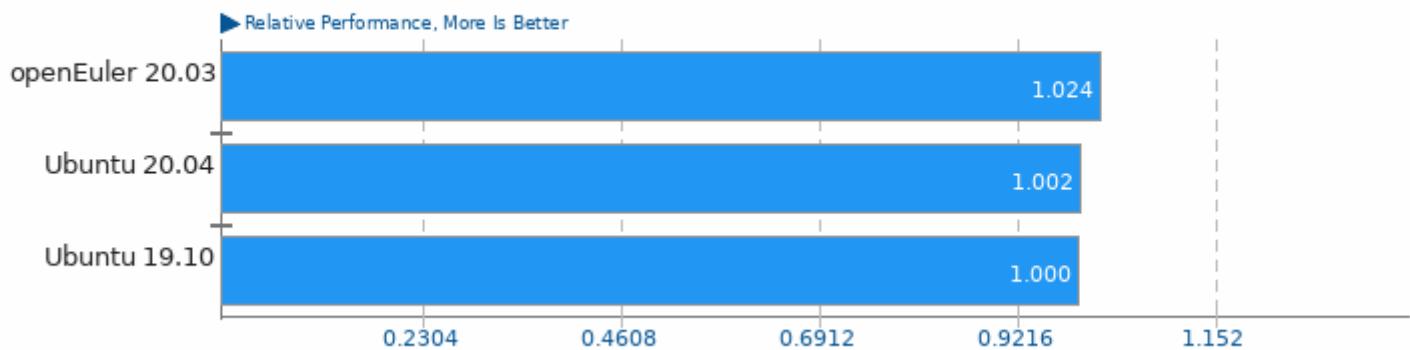
Result Composite



Geometric mean based upon tests: pts/apache, pts/redis, pts/phpbench and pts/stream

## Geometric Mean Of Single-Threaded Tests

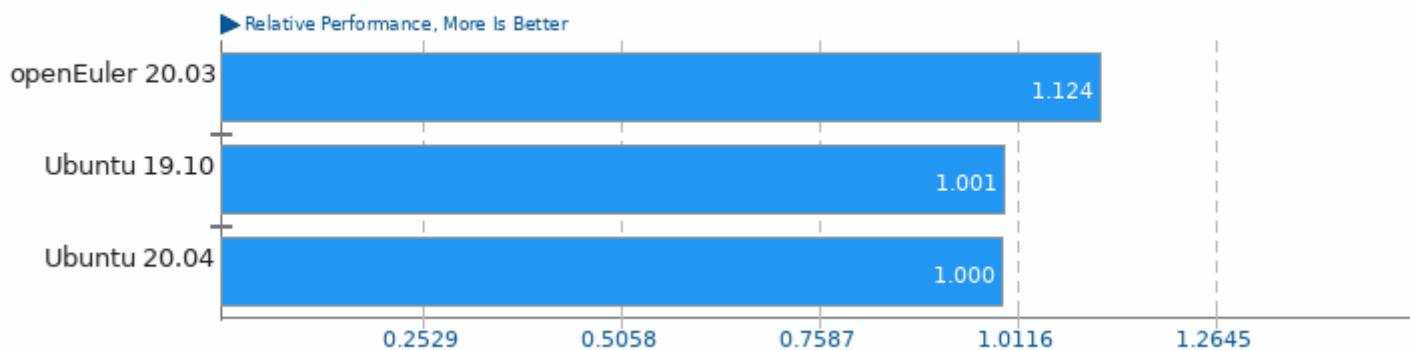
Result Composite



Geometric mean based upon tests: pts/java-scimark2, pts/bork, pts/cachebench, pts/botan, pts/node-octane, pts/compress-gzip, pts/encode-flac, pts/encode-mp3, pts/redis and pts/phpbench

## Geometric Mean Of Video Encoding Tests

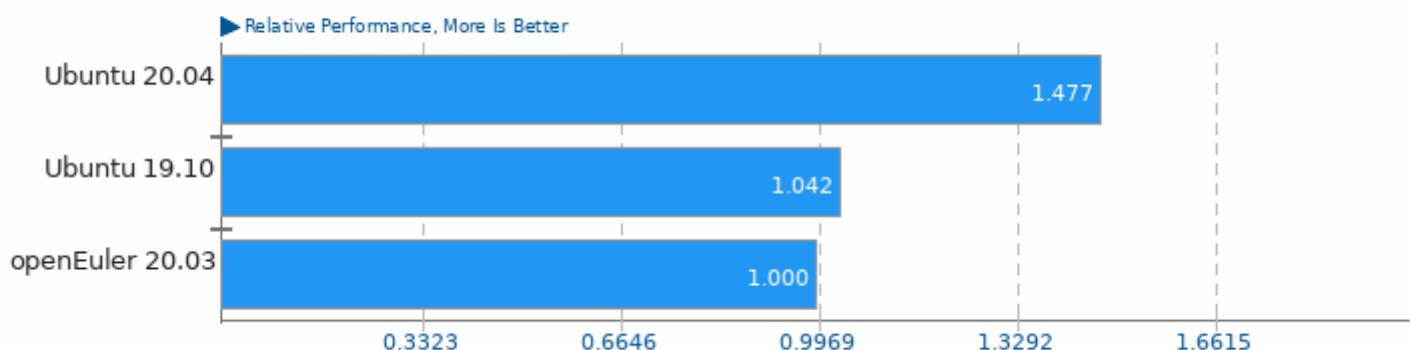
Result Composite



Geometric mean based upon tests: pts/x264, pts/vpxenc and pts/dav1d

## Geometric Mean Of Workstation Tests

Result Composite



Geometric mean based upon tests: pts/rodinia, pts/sysbench and pts/sqlite

*This file was automatically generated via the Phoronix Test Suite benchmarking software on Monday, 11 May 2020 22:48.*