



## ryzen

Intel Core i7-8650U testing with a Intel NUC7i7DNB (DNKBLi7v.86A.0073.2021.0414.1709 BIOS) and Intel UHD 620 KBL GT2 3GB on ManjaroLinux 21.0.6 via the Phoronix Test Suite.

### Automated Executive Summary

*ryzen manjaro had the most wins, coming in first place for 68% of the tests.*

*Based on the geometric mean of all complete results, the fastest (ryzen manjaro) was 1.577x the speed of the slowest (kabybomb). ryzen vmware was 0.889x the speed of ryzen manjaro, tigerbomb was 0.9x the speed of ryzen vmware, kabybomb was 0.793x the speed of tigerbomb.*

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

*Aircrack-ng at 5.421x*

*Coremark (CoreMark Size 666 - Iterations Per Second) at 5.21x*

*x264 (H.264 Video Encoding) at 5.095x*

*7-Zip Compression (Compress Speed Test) at 4.766x*

*Parallel BZIP2 Compression (256MB File Compression) at 4.136x*

*x265 (Video Input: Bosphorus 1080p) at 3.164x*

*dav1d (Video Input: Summer Nature 1080p) at 3.14x*

*FFTE (N=256, 3D Complex FFT Routine) at 2.819x*

Cython Benchmark (Test: N-Queens) at 2.119x

Java SciMark (Computational Test: Sparse Matrix Multiply) at 2.092x.

## Test Systems:

### ryzen vmware

Processor: AMD Ryzen 9 5900X 12-Core (24 Cores), Motherboard: Intel 440BX (VMW71.00V.16722896.B64.2008100651 BIOS), Chipset: Intel 440BX/ZX/DX, Memory: 64GB, Disk: 86GB VMware Virtual S, Graphics: VMware SVGA II, Audio: Ensoniq ES1371/ES1373, Network: Intel 82545EM

OS: Arch rolling, Kernel: 5.12.10-arch1-1 (x86\_64), Display Server: X Server 1.20.9, Compiler: GCC 11.1.0, File-System: btrfs, Screen Resolution: 1176x885, System Layer: VMware

Kernel Notes: Transparent Huge Pages: madvise

Compiler Notes: --disable-libssp --disable-libstdcxx-pch --disable-libunwind-exceptions --disable-werror --enable-\_\_cxa\_atexit --enable-cet=auto --enable-checking=release --enable-clocale=gnu --enable-default-pie --enable-default-ssp --enable-gnu-indirect-function --enable-gnu-unique-object --enable-install-liberty --enable-languages=c,c++,ada,fortran,go,lto,objc,obj-c++,d --enable-lto --enable-multilib --enable-plugin --enable-shared --enable-threads=posix --mandir=/usr/share/man --with-isl --with-linker-hash-style=gnu

Processor Notes: CPU Microcode: 0xa201016

Java Notes: OpenJDK Runtime Environment (build 11.0.11+9)

Python Notes: Python 3.9.5

Security Notes: itlb\_multihit: Not affected + l1tf: Not affected + mds: Not affected + meltdown: Not affected + spec\_store\_bypass: Mitigation of SSB disabled via prctl and seccomp + spectre\_v1: Mitigation of usercopy/swaps barriers and \_\_user pointer sanitization + spectre\_v2: Mitigation of Full AMD retpoline IBPB: conditional STIBP: disabled RSB filling + srbds: Not affected + tsx\_async\_abort: Not affected

### ryzen manjaro

Processor: AMD Ryzen 9 5900X 12-Core @ 3.70GHz (12 Cores / 24 Threads), Motherboard: MSI MAG B550 TOMAHAWK (MS-7C91) v2.0 (A.71 BIOS), Chipset: AMD Starship/Matisse, Memory: 126GB, Disk: 1024GB ADATA SX8200PNP + 1024GB INTEL SSDPEKNW010T8 + 4001GB Seagate ST4000DM000-1F21 + 4001GB Seagate ST4000DM005-2DP1 + 960GB Patriot Burst + 2 x 2000GB CT2000MX500SSD1, Graphics: eVGA NVIDIA GeForce RTX 3080 10GB, Audio: NVIDIA GA102 HD Audio, Monitor: BenQ GW2765, Network: Realtek RTL8111/8168/8411 + Realtek RTL8125 2.5GbE

OS: ManjaroLinux 21.0.6, Kernel: 5.12.11-1-MANJARO (x86\_64), Desktop: Cinnamon 5.0.2, Display Server: X Server 1.20.11, Display Driver: NVIDIA 465.31, OpenGL: 4.6.0, Compiler: GCC 11.1.0 + Clang 12.0.0, File-System: btrfs, Screen Resolution: 2560x1440

Kernel Notes: Transparent Huge Pages: madvise

Compiler Notes: --disable-libssp --disable-libstdcxx-pch --disable-libunwind-exceptions --disable-werror --enable-\_\_cxa\_atexit --enable-cet=auto --enable-checking=release --enable-clocale=gnu --enable-default-pie --enable-default-ssp --enable-gnu-indirect-function --enable-gnu-unique-object --enable-install-liberty --enable-languages=c,c++,ada,fortran,go,lto,objc,obj-c++,d --enable-lto --enable-multilib --enable-plugin --enable-shared --enable-threads=posix --mandir=/usr/share/man --with-isl --with-linker-hash-style=gnu

Processor Notes: Scaling Governor: acpi-cpufreq performance (Boost: Enabled) - CPU Microcode: 0xa201016

Java Notes: OpenJDK Runtime Environment (build 1.8.0\_292-b10)

Python Notes: Python 3.9.5

Security Notes: itlb\_multihit: Not affected + l1tf: Not affected + mds: Not affected + meltdown: Not affected + spec\_store\_bypass: Vulnerable + spectre\_v1: Vulnerable: \_\_user pointer sanitization and usercopy barriers only; no swaps barriers + spectre\_v2: Vulnerable IBPB: disabled STIBP: disabled + srbds: Not affected + tsx\_async\_abort: Not affected

### tigerbomb

Processor: Intel Core i7-1165G7 @ 4.70GHz (4 Cores / 8 Threads), Motherboard: Intel NUC11TNBi7

(TNTGL357.0056.2021.0513.1618 BIOS), Chipset: Intel Tiger Lake-LP, Memory: 64GB, Disk: 512GB SAMSUNG MZVLB512HAJQ-000H1, Graphics: Intel Xe TGL GT2 3GB (1300MHz), Audio: Intel Tiger Lake-LP Smart Sound Audio, Monitor: SAMSUNG, Network: Intel I225-LM + Intel Wi-Fi 6 AX201

OS: ManjaroLinux 21.0.6, Kernel: 5.12.11-1-MANJARO (x86\_64), Desktop: Cinnamon 5.0.2, Display Server: X Server 1.20.11, OpenGL: 4.6 Mesa 21.1.2, Compiler: GCC 11.1.0, File-System: btrfs, Screen Resolution: 1920x1080

Kernel Notes: Transparent Huge Pages: madvise  
 Compiler Notes: --disable-libssp --disable-libstdc++-pch --disable-libunwind-exceptions --disable-werror --enable-\_\_cxa\_atexit --enable-cet=auto --enable-checking=release --enable-clocale=gnu --enable-default-pie --enable-default-ssp --enable-gnu-indirect-function --enable-gnu-unique-object --enable-install-liberty --enable-languages=c,c++,ada,fortran,go,lto,objc,obj-c++,d --enable-lto --enable-multilib --enable-plugin --enable-shared --enable-threads=posix --mandir=/usr/share/man --with-isl --with-linker-hash-style=gnu  
 Processor Notes: Scaling Governor: intel\_pstate performance - CPU Microcode: 0x88  
 Java Notes: OpenJDK Runtime Environment (build 1.8.0\_292-b10)  
 Python Notes: Python 3.9.5  
 Security Notes: itlb\_multihit: Not affected + l1tf: Not affected + mds: Not affected + meltdown: Not affected + spec\_store\_bypass: Mitigation of SSB disabled via prctl and seccomp + spectre\_v1: Mitigation of usercopy/swaps barriers and \_\_user pointer sanitization + spectre\_v2: Mitigation of Enhanced IBRS IBPB: conditional RSB filling + srbds: Not affected + tsx\_async\_abort: Not affected

## kabybomb

Processor: Intel Core i7-8650U @ 4.20GHz (4 Cores / 8 Threads), Motherboard: Intel NUC7i7DNB (DNKBLi7v.86A.0073.2021.0414.1709 BIOS), Chipset: Intel Xeon E3-1200 v6/7th, Memory: 32GB, Disk: 525GB Crucial CT525MX3, Graphics: Intel UHD 620 KBL GT2 3GB (1150MHz), Audio: Intel Sunrise Point-LP HD Audio, Monitor: SAMSUNG, Network: Intel I219-LM + Intel 8265 / 8275

OS: ManjaroLinux 21.0.6, Kernel: 5.12.12-1-MANJARO (x86\_64), Desktop: Cinnamon 5.0.2, Display Server: X Server 1.20.11, OpenGL: 4.6 Mesa 21.1.2, Compiler: GCC 11.1.0 + Clang 12.0.0, File-System: btrfs, Screen Resolution: 1920x1080

Kernel Notes: Transparent Huge Pages: madvise  
 Compiler Notes: --disable-libssp --disable-libstdc++-pch --disable-libunwind-exceptions --disable-werror --enable-\_\_cxa\_atexit --enable-cet=auto --enable-checking=release --enable-clocale=gnu --enable-default-pie --enable-default-ssp --enable-gnu-indirect-function --enable-gnu-unique-object --enable-install-liberty --enable-languages=c,c++,ada,fortran,go,lto,objc,obj-c++,d --enable-lto --enable-multilib --enable-plugin --enable-shared --enable-threads=posix --mandir=/usr/share/man --with-isl --with-linker-hash-style=gnu  
 Processor Notes: Scaling Governor: intel\_pstate powersave - CPU Microcode: 0xea  
 Java Notes: OpenJDK Runtime Environment (build 1.8.0\_292-b10)  
 Python Notes: Python 3.9.5  
 Security Notes: itlb\_multihit: KVM: Mitigation of VMX disabled + l1tf: Mitigation of PTE Inversion; VMX: vulnerable + mds: Vulnerable; SMT vulnerable + meltdown: Vulnerable + spec\_store\_bypass: Vulnerable + spectre\_v1: Vulnerable; \_\_user pointer sanitization and usercopy barriers only; no swaps barriers + spectre\_v2: Vulnerable IBPB: disabled STIBP: disabled + srbds: Vulnerable + tsx\_async\_abort: Vulnerable

	ryzen vmware	ryzen manjaro	tigerbomb	kabybomb
<b>CloverLeaf - L.E.H (sec)</b>	107.39	<b>90.94</b>	154.94	<b>201.19</b>
Normalized	84.68%	100%	58.69%	45.2%
Standard Deviation	10.7%	0.1%	3.1%	0.3%
<b>PolyBench-C - C.C (sec)</b>	<b>5.773</b>		<b>3.145</b>	5.535
Normalized	54.48%		100%	56.82%
Standard Deviation	0.5%		4.8%	0.1%
<b>PolyBench-C - C.C (sec)</b>	<b>5.666</b>		<b>3.080</b>	5.561
Normalized	54.36%		100%	55.39%
Standard Deviation	2.2%		6.5%	0.2%
<b>Izbench - XZ 0 - Compression (MB/s)</b>	50	<b>55</b>	51	<b>40</b>
Normalized	90.91%	100%	92.73%	72.73%
Standard Deviation	13.4%	1.1%	1.1%	0%

Izbench - XZ 0 - Decompression	175	177	139	113
Normalized	98.87%	100%	78.53%	63.84%
Standard Deviation	0.3%	0.9%	0.4%	0.5%
Izbench - Zstd 1 - Compression (MB/s)	686	688	608	487
Normalized	99.71%	100%	88.37%	70.78%
Standard Deviation	0.3%	0.5%	0.3%	0.2%
Izbench - Zstd 1 - Decompression	2341	2338	2107	1650
Normalized	100%	99.87%	90%	70.48%
Standard Deviation	0.2%	0.7%	0.2%	0.5%
Izbench - Zstd 8 - Compression (MB/s)	109	128	122	86
Normalized	85.16%	100%	95.31%	67.19%
Standard Deviation	14.2%	1.6%	0.5%	0.7%
Izbench - Zstd 8 - Decompression	2591	2567	2217	1775
Normalized	100%	99.07%	85.57%	68.51%
Standard Deviation	0.2%	0.5%	0.1%	0.4%
Izbench - Crush 0 - Compression	131	169	120	94
Normalized	77.51%	100%	71.01%	55.62%
Standard Deviation	14.5%	1.7%	0.5%	1.8%
Izbench - Crush 0 - Decompression (MB/s)	714	714	562	492
Normalized	100%	100%	78.71%	68.91%
Standard Deviation	0.2%	0.4%	0.2%	0.3%
Izbench - Brotli 0 - Compression	494	640	567	436
Normalized	77.19%	100%	88.59%	68.13%
Standard Deviation	23.5%	0.1%	0.4%	0.2%
Izbench - Brotli 0 - Decompression (MB/s)	760	832	727	598
Normalized	91.35%	100%	87.38%	71.88%
Standard Deviation	9.5%	0.4%	0.8%	0.2%
Izbench - Brotli 2 - Compression	193	273	240	183
Normalized	70.7%	100%	87.91%	67.03%
Standard Deviation	25.5%	0.4%	0.2%	0%
Izbench - Brotli 2 - Decompression (MB/s)	878	998	881	691
Normalized	87.98%	100%	88.28%	69.24%
Standard Deviation	0.1%	0.1%	1%	0.2%
Izbench - Libdeflate 1 - Compression (MB/s)	187	341	287	227
Normalized	54.84%	100%	84.16%	66.57%
Standard Deviation	11.9%	0.2%	0.2%	0.3%
Izbench - Libdeflate 1 - Decompression (MB/s)	1542	1638	1301	1143
Normalized	94.14%	100%	79.43%	69.78%
Standard Deviation	0%	0.1%	0.1%	0.4%
FFTE - N.2.3.C.F.R (MFLOPS)	55961	58152	30071	20631
Normalized	96.23%	100%	51.71%	35.48%
Standard Deviation	0.5%	0.2%	0.4%	0.8%
Timed HMMer Search - P.D.S (sec)	128.328	88.315	119.011	156.032
Normalized	68.82%	100%	74.21%	56.6%
Standard Deviation	1.4%	0.3%	0.1%	0.7%
Timed MAFFT Alignment - M.S.A - LSU RNA (sec)	14.151	6.850	8.709	12.370
Normalized	48.41%	100%	78.65%	55.38%
Standard Deviation	17.5%	0.2%	0.6%	0.3%

<b>Java SciMark - Composite (Mflops)</b>	3081	<b>4109</b>	3012	<b>2247</b>
Normalized	74.98%	100%	73.3%	54.67%
Standard Deviation	5.4%	0.9%	1.2%	2.3%
<b>Java SciMark - Monte Carlo (Mflops)</b>	<b>1992</b>	1974	1138	<b>1019</b>
Normalized	100%	99.1%	57.13%	51.17%
Standard Deviation	0.2%	2.8%	0.3%	0.2%
<b>Java SciMark - F.F.T (Mflops)</b>	2000	<b>2413</b>	1937	<b>1399</b>
Normalized	82.88%	100%	80.25%	57.98%
Standard Deviation	47.7%	6.9%	0.7%	0.1%
<b>Java SciMark - S.M.M (Mflops)</b>	2792	<b>4095</b>	3002	<b>1958</b>
Normalized	68.2%	100%	73.32%	47.81%
Standard Deviation	4.7%	1.3%	0.2%	0.2%
<b>Java SciMark - D.L.M.F (Mflops)</b>	6297	<b>9997</b>	7171	<b>5396</b>
Normalized	62.99%	100%	71.73%	53.97%
Standard Deviation	10.1%	0.5%	2.5%	5.4%
<b>Java SciMark - J.S.O.R (Mflops)</b>	<b>2325</b>	2069	1813	<b>1503</b>
Normalized	100%	88.99%	77.96%	64.66%
Standard Deviation	0.1%	0%	1.7%	0.3%
<b>Bork File Encrypter - F.E.T (sec)</b>	8.876	<b>6.610</b>	<b>9.261</b>	7.086
Normalized	74.47%	100%	71.37%	93.28%
Standard Deviation	38.5%	0.2%	1.6%	1.6%
<b>LuaJIT - Composite (Mflops)</b>	1892	<b>2013</b>	1749	<b>1397</b>
Normalized	93.99%	100%	86.89%	69.41%
Standard Deviation	0.2%	0.1%	1.4%	0.2%
<b>LuaJIT - Monte Carlo (Mflops)</b>	586.97	<b>610.69</b>	604.42	<b>496.02</b>
Normalized	96.12%	100%	98.97%	81.22%
Standard Deviation	0.2%	0.5%	0.4%	0.5%
<b>LuaJIT - F.F.T (Mflops)</b>	496.79	<b>528.62</b>	522.00	<b>360.85</b>
Normalized	93.98%	100%	98.75%	68.26%
Standard Deviation	0.3%	0.2%	0.3%	1%
<b>LuaJIT - S.M.M (Mflops)</b>	1488	<b>1770</b>	1732	<b>1180</b>
Normalized	84.09%	100%	97.84%	66.65%
Standard Deviation	0.3%	0.1%	4.4%	0.9%
<b>LuaJIT - D.L.M.F (Mflops)</b>	4416	<b>4658</b>	4158	<b>3416</b>
Normalized	94.81%	100%	89.28%	73.33%
Standard Deviation	0.6%	0.2%	1.3%	0.2%
<b>LuaJIT - J.S.O.R (Mflops)</b>	2473	<b>2498</b>	1729	<b>1534</b>
Normalized	98.97%	100%	69.21%	61.41%
Standard Deviation	0%	0%	0.1%	0.8%
<b>SciMark - Composite (Mflops)</b>	734.74	<b>906.49</b>	697.75	<b>618.15</b>
Normalized	81.05%	100%	76.97%	68.19%
Standard Deviation	15.1%	0.2%	5.3%	0.2%
<b>SciMark - Monte Carlo (Mflops)</b>	<b>116.96</b>	199.22	<b>219.71</b>	140.37
Normalized	53.23%	90.67%	100%	63.89%
Standard Deviation	0.2%	0.3%	0.1%	0.2%
<b>SciMark - F.F.T (Mflops)</b>	447.21	<b>546.71</b>	289.04	<b>191.30</b>
Normalized	81.8%	100%	52.87%	34.99%
Standard Deviation	28.3%	0.4%	2.5%	0.8%
<b>SciMark - S.M.M (Mflops)</b>	<b>850.51</b>	848.94	829.52	<b>692.60</b>
Normalized	100%	99.82%	97.53%	81.43%
Standard Deviation	0.1%	0.4%	0.2%	0.3%
<b>SciMark - D.L.M.F (Mflops)</b>	1028	<b>1537</b>	<b>845.79</b>	911.69
Normalized	66.86%	100%	55.01%	59.3%
Standard Deviation	52.4%	0.3%	21.5%	0%
<b>SciMark - J.S.O.R (Mflops)</b>	1231	<b>1400</b>	1305	<b>1155</b>

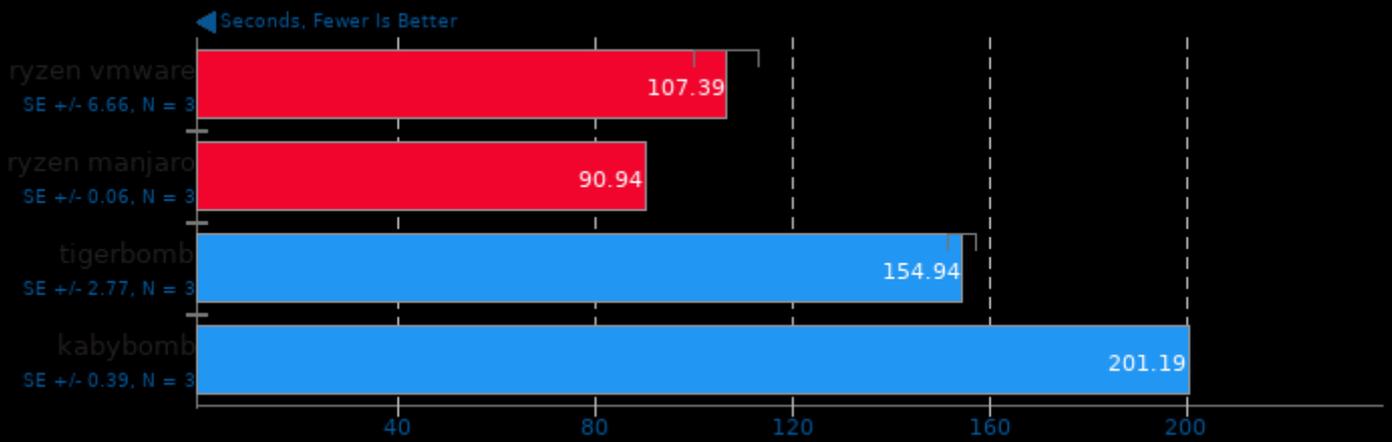
	Normalized	87.93%	100%	93.18%	82.48%
	Standard Deviation	1.6%	0%	0.9%	0.2%
<b>Botan - KASUMI (MiB/s)</b>		117.572	<b>117.670</b>	94.678	<b>91.364</b>
	Normalized	99.92%	100%	80.46%	77.64%
	Standard Deviation	0.1%	0.3%	0.4%	0.1%
<b>Botan - KASUMI - Decrypt (MiB/s)</b>		<b>113.299</b>	112.997	91.702	<b>89.182</b>
	Normalized	100%	99.73%	80.94%	78.71%
	Standard Deviation	0.2%	0.2%	0.4%	0.1%
<b>Botan - AES-256 (MiB/s)</b>		7562	7581	<b>7634</b>	<b>3978</b>
	Normalized	99.06%	99.3%	100%	52.11%
	Standard Deviation	0.4%	0.1%	0.4%	0.1%
<b>Botan - AES-256 - Decrypt (MiB/s)</b>		7574	7636	<b>7690</b>	<b>3973</b>
	Normalized	98.5%	99.31%	100%	51.67%
	Standard Deviation	0.2%	0%	0.4%	0.1%
<b>Botan - Twofish (MiB/s)</b>		430.959	<b>431.321</b>	<b>329.534</b>	330.414
	Normalized	99.92%	100%	76.4%	76.61%
	Standard Deviation	1.8%	0.8%	0.8%	0.1%
<b>Botan - Twofish - Decrypt (MiB/s)</b>		428.197	<b>436.124</b>	330.629	<b>329.009</b>
	Normalized	98.18%	100%	75.81%	75.44%
	Standard Deviation	2.8%	0.8%	0.7%	0.2%
<b>Botan - Blowfish (MiB/s)</b>		502.878	<b>528.601</b>	<b>396.719</b>	400.058
	Normalized	95.13%	100%	75.05%	75.68%
	Standard Deviation	3.6%	0.1%	0.2%	0.2%
<b>Botan - Blowfish - Decrypt (MiB/s)</b>		497.884	<b>526.450</b>	<b>392.132</b>	393.315
	Normalized	94.57%	100%	74.49%	74.71%
	Standard Deviation	2.3%	0.1%	1%	0.2%
<b>Botan - CAST-256 (MiB/s)</b>		173.852	<b>174.874</b>	145.370	<b>137.089</b>
	Normalized	99.42%	100%	83.13%	78.39%
	Standard Deviation	0.2%	0.4%	0.4%	0.4%
<b>Botan - CAST-256 - Decrypt (MiB/s)</b>		174.056	<b>174.777</b>	145.636	<b>137.212</b>
	Normalized	99.59%	100%	83.33%	78.51%
	Standard Deviation	0.3%	0.4%	0.6%	0.1%
<b>Botan - ChaCha20Poly1305 (MiB/s)</b>		<b>1000</b>	990.266	956.974	<b>756.501</b>
	Normalized	100%	99%	95.67%	75.63%
	Standard Deviation	0.1%	0.5%	0.3%	1.1%
<b>Botan - ChaCha20Poly1305 - Decrypt (MiB/s)</b>		<b>983.754</b>	971.160	948.275	<b>745.194</b>
	Normalized	100%	98.72%	96.39%	75.75%
	Standard Deviation	0.1%	0.4%	0.2%	2.2%
<b>TSCP - A.C.P (Nodes/s)</b>		<b>2020523</b>	2013083	1652006	<b>1229427</b>
	Normalized	100%	99.63%	81.76%	60.85%
	Standard Deviation	0.6%	0.2%	0.2%	0.4%
<b>dav1d - S.N.1 (FPS)</b>		813.99	<b>926.17</b>	386.16	<b>294.93</b>
	Normalized	87.89%	100%	41.69%	31.84%
	Standard Deviation	0.4%	0.4%	0.2%	0.3%
<b>x264 - H.2.V.E (FPS)</b>		170.63	<b>187.36</b>	51.17	<b>36.77</b>
	Normalized	91.07%	100%	27.31%	19.63%
	Standard Deviation	0.1%	1.1%	1.9%	0.8%
<b>x265 - Bosphorus 1080p (FPS)</b>		85.22	<b>87.35</b>	38.37	<b>27.61</b>
	Normalized	97.56%	100%	43.93%	31.61%
	Standard Deviation	0.7%	1.1%	3.9%	0.1%
<b>Coremark - CoreMark Size 666 - I.P.S (Iterations/Sec)</b>		<b>680863</b>	679031	174008	<b>130694</b>
	Normalized	100%	99.73%	25.56%	19.2%
	Standard Deviation	0.2%	0.5%	0.2%	0.9%

<b>Himeno Benchmark - P.P.S (MFLOPS)</b>	4216	<b>5370</b>	5345	<b>3273</b>
Normalized	78.51%	100%	99.55%	60.96%
Standard Deviation	4.9%		0.6%	0.4%
<b>7-Zip Compression - C.S.T (MIPS)</b>	87943	<b>102080</b>	29272	<b>21418</b>
Normalized	86.15%	100%	28.68%	20.98%
Standard Deviation	0.2%	0.2%	0.6%	2.1%
<b>Swet - Average (Operations/sec)</b>	867185556	<b>1137690354</b>	793416169	<b>715106400</b>
Normalized	76.22%	100%	69.74%	62.86%
Standard Deviation	1%	0.6%	0.3%	1.8%
<b>Parallel BZIP2 Compression - 2.F.C (sec)</b>	<b>2.257</b>		7.108	<b>9.335</b>
Normalized	100%		31.75%	24.18%
Standard Deviation	0.3%		1.7%	1.7%
<b>Bullet Physics Engine - 3000 Fall (sec)</b>	2.86143	<b>2.674918</b>	<b>3.123245</b>	
Normalized	93.48%	100%	85.65%	
Standard Deviation	5.4%	0%	0.8%	
<b>Bullet Physics Engine - 1000 Stack</b>	3.397069	<b>3.283612</b>	<b>3.646767</b>	
Normalized	96.66%	100%	90.04%	
Standard Deviation	3.5%	0.1%	0.5%	
<b>Bullet Physics Engine - 1000 Convex (sec)</b>	3.046435	<b>2.923287</b>	<b>3.494797</b>	
Normalized	95.96%	100%	83.65%	
Standard Deviation	3.1%	0.2%	0.6%	
<b>Bullet Physics Engine - 136 Ragdolls (sec)</b>	1.842695	<b>1.748024</b>	<b>1.957212</b>	
Normalized	94.86%	100%	89.31%	
Standard Deviation	5.5%	0%	0.3%	
<b>Bullet Physics Engine - Prim Trimesh (sec)</b>	0.690433	<b>0.658662</b>	<b>0.714912</b>	
Normalized	95.4%	100%	92.13%	
Standard Deviation	4.8%	0%	0.8%	
<b>Bullet Physics Engine - Convex Trimesh (sec)</b>	0.785933	<b>0.755915</b>	<b>0.878443</b>	
Normalized	96.18%	100%	86.05%	
Standard Deviation	3.8%	0.3%	2.8%	
<b>Cython Benchmark - N-Queens (sec)</b>	<b>33.731</b>	<b>15.915</b>	18.584	22.987
Normalized	47.18%	100%	85.64%	69.23%
Standard Deviation	3.9%	0.1%	0.1%	0.6%
<b>eSpeak-NG Speech Engine - T.T.S.S (sec)</b>	23.183	<b>20.437</b>	27.319	<b>28.372</b>
Normalized	88.16%	100%	74.81%	72.03%
Standard Deviation	4.9%	1.3%	2.1%	1.1%
<b>FFmpeg - H.2.H.T.N.D (sec)</b>	4.408	<b>4.057</b>	5.038	<b>6.442</b>
Normalized	92.04%	100%	80.53%	62.98%
Standard Deviation	0.3%	1.2%	6%	1.2%
<b>Mencoder - AVI To LAVC (sec)</b>	12.653			
Standard Deviation	5%			
<b>Aircrack-ng (k/s)</b>	66544	<b>67286</b>	16316	<b>12412</b>
Normalized	98.9%	100%	24.25%	18.45%
Standard Deviation	0.2%	0.2%	0.1%	1.1%
<b>glibc bench - cos (nanoseconds)</b>	34.7727	34.7368	<b>33.9884</b>	<b>48.0597</b>
Normalized	97.74%	97.85%	100%	70.72%
Standard Deviation	0%	0.1%	0.8%	0.8%
<b>glibc bench - exp (nanoseconds)</b>	4.29539	4.47529	<b>3.94797</b>	<b>5.40897</b>

	Normalized	91.91%	88.22%	100%	72.99%
	Standard Deviation	0.6%	1.3%	1.2%	0.1%
<b>glibc bench - ffs (nanoseconds)</b>		<b>2.39543</b>	2.34778	<b>1.44789</b>	1.84618
	Normalized	60.44%	61.67%	100%	78.43%
	Standard Deviation	0.1%	0.2%	0.5%	0.2%
<b>glibc bench - sin (nanoseconds)</b>		34.2985	34.3210	<b>34.0024</b>	<b>47.0639</b>
	Normalized	99.14%	99.07%	100%	72.25%
	Standard Deviation	0.1%	0%	2.4%	0%
<b>glibc bench - log2 (nanoseconds)</b>		4.79505	<b>4.39004</b>	6.07831	<b>7.41363</b>
	Normalized	91.55%	100%	72.22%	59.22%
	Standard Deviation	0.1%	0.4%	0.5%	0.1%
<b>glibc bench - modf (nanoseconds)</b>		<b>2.39672</b>	2.35093	<b>1.53472</b>	2.08685
	Normalized	64.03%	65.28%	100%	73.54%
	Standard Deviation	0.1%	0.3%	0.6%	0.1%
<b>glibc bench - sinh (nanoseconds)</b>		6.91209	6.84033	<b>6.39559</b>	<b>9.10574</b>
	Normalized	92.53%	93.5%	100%	70.24%
	Standard Deviation	1%	0.1%	1.9%	0.3%
<b>glibc bench - sqrt (nanoseconds)</b>		<b>2.33712</b>	2.32521	<b>1.44816</b>	1.86096
	Normalized	61.96%	62.28%	100%	77.82%
	Standard Deviation	4.2%	0%	0.4%	1%
<b>glibc bench - tanh (nanoseconds)</b>		8.35672	<b>8.26959</b>	8.64333	<b>12.8875</b>
	Normalized	98.96%	100%	95.68%	64.17%
	Standard Deviation	0.1%	0.6%	0.4%	0.2%
<b>glibc bench - asinh (nanoseconds)</b>		<b>6.09428</b>	6.12683	6.97796	<b>10.0409</b>
	Normalized	100%	99.47%	87.34%	60.69%
	Standard Deviation	0.7%	0.6%	0.1%	0%
<b>glibc bench - atanh (nanoseconds)</b>		7.87058	<b>7.77575</b>	9.56644	<b>11.6555</b>
	Normalized	98.8%	100%	81.28%	66.71%
	Standard Deviation	0.5%	0.8%	0.1%	0%
<b>glibc bench - ffsll (nanoseconds)</b>		2.14519	<b>2.14599</b>	<b>1.23069</b>	1.60875
	Normalized	57.37%	57.35%	100%	76.5%
	Standard Deviation	3.7%	0%	0.5%	0.1%
<b>glibc bench - sincos (nanoseconds)</b>		9.83196	<b>9.75163</b>	10.7141	<b>15.4406</b>
	Normalized	99.18%	100%	91.02%	63.16%
	Standard Deviation	0.1%	0.3%	0.6%	1.9%
<b>glibc bench - pthread_once (nanoseconds)</b>		2.12701	<b>2.14446</b>	<b>1.22615</b>	1.61140
	Normalized	57.65%	57.18%	100%	76.09%
	Standard Deviation	4.1%	0%	0.6%	0.2%
<b>Multichase Pointer Chaser - 2.A.2.B.S (ns)</b>		64.404	<b>54.082</b>	<b>78.174</b>	60.937
	Normalized	83.97%	100%	69.18%	88.75%
	Standard Deviation	4.2%	2.3%	0.1%	3.1%

## CloverLeaf

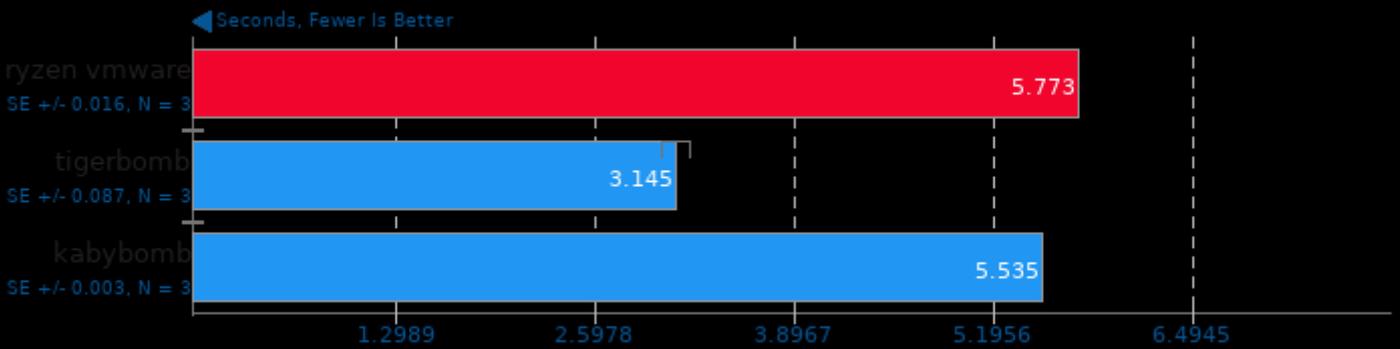
Lagrangian-Eulerian Hydrodynamics



1. (F9X) gfortran options: -O3 -march=native -funroll-loops -fopenmp

## PolyBench-C 4.2

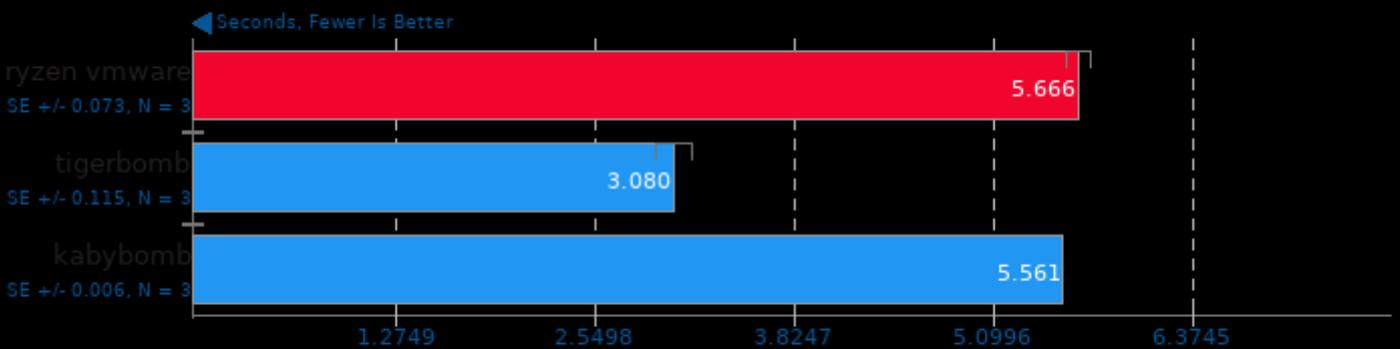
Test: Covariance Computation



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

## PolyBench-C 4.2

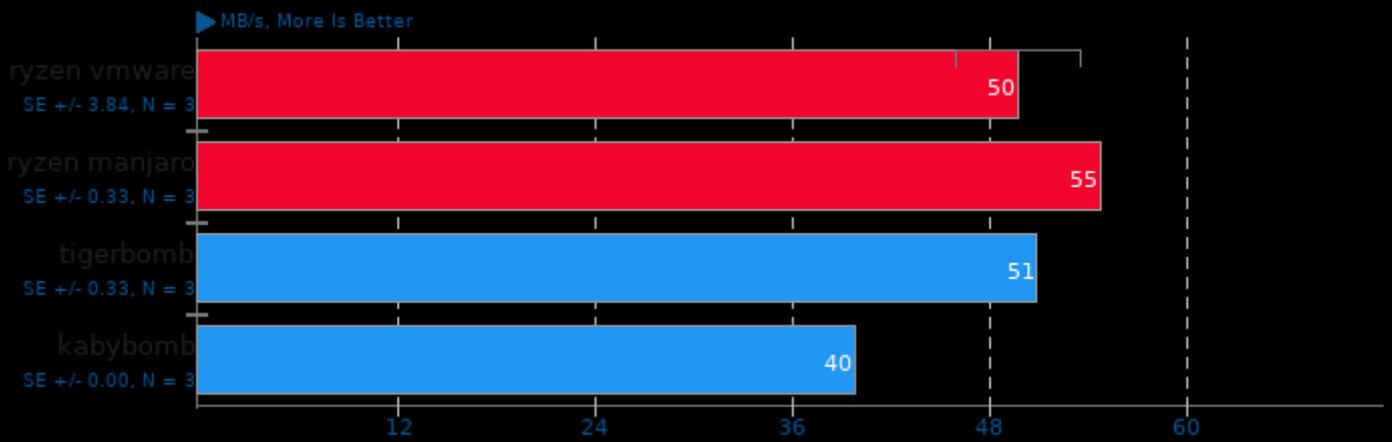
Test: Correlation Computation



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

## Izbench 1.8

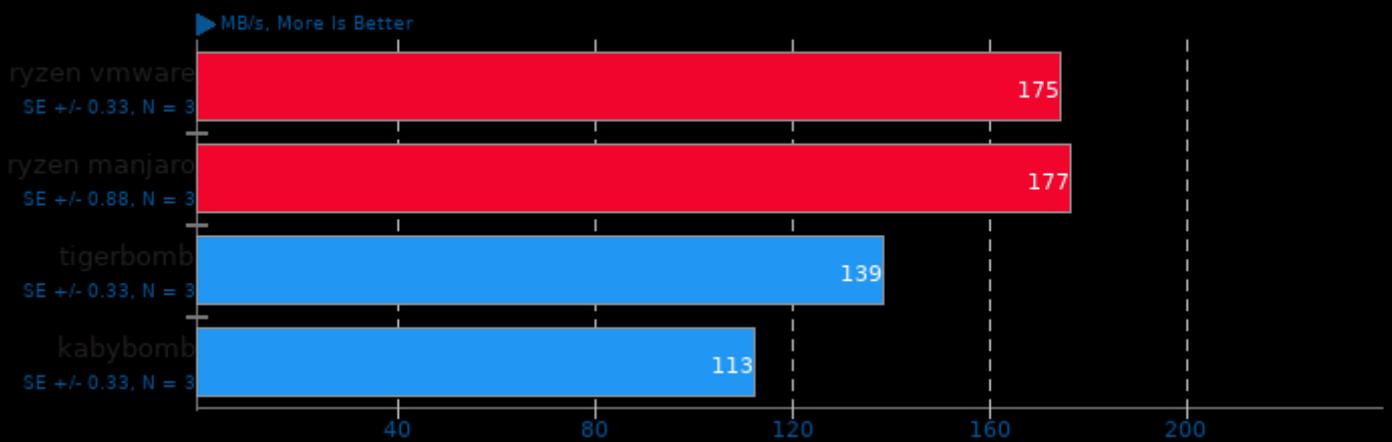
Test: XZ 0 - Process: Compression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

## Izbench 1.8

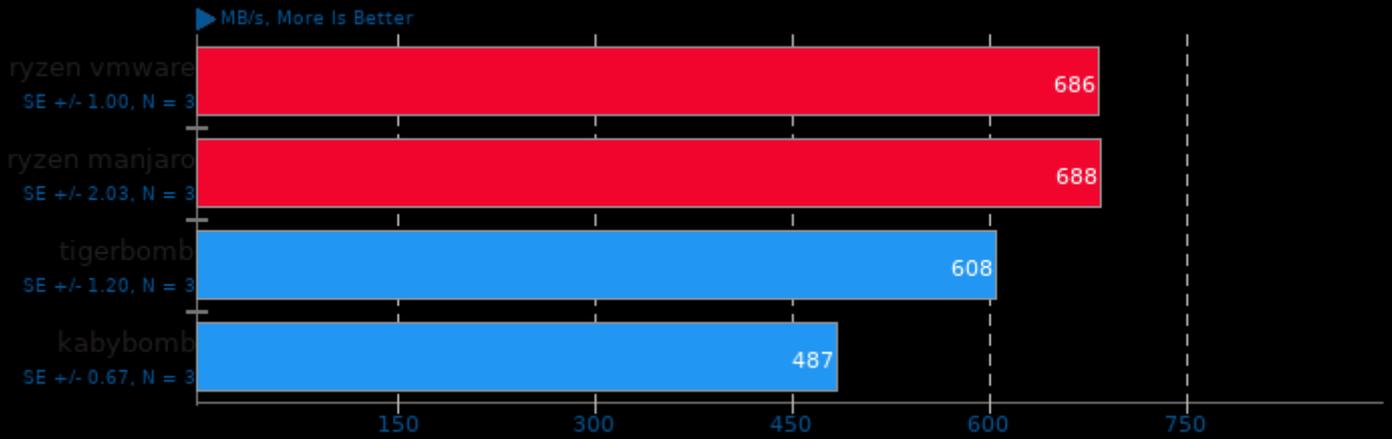
Test: XZ 0 - Process: Decompression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

## Izbench 1.8

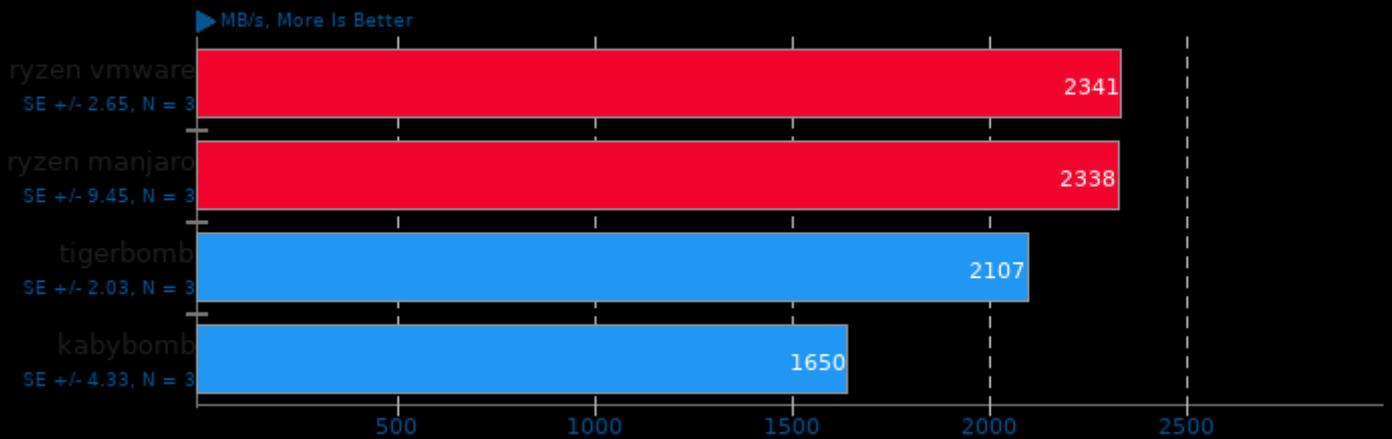
Test: Zstd 1 - Process: Compression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

## Izbench 1.8

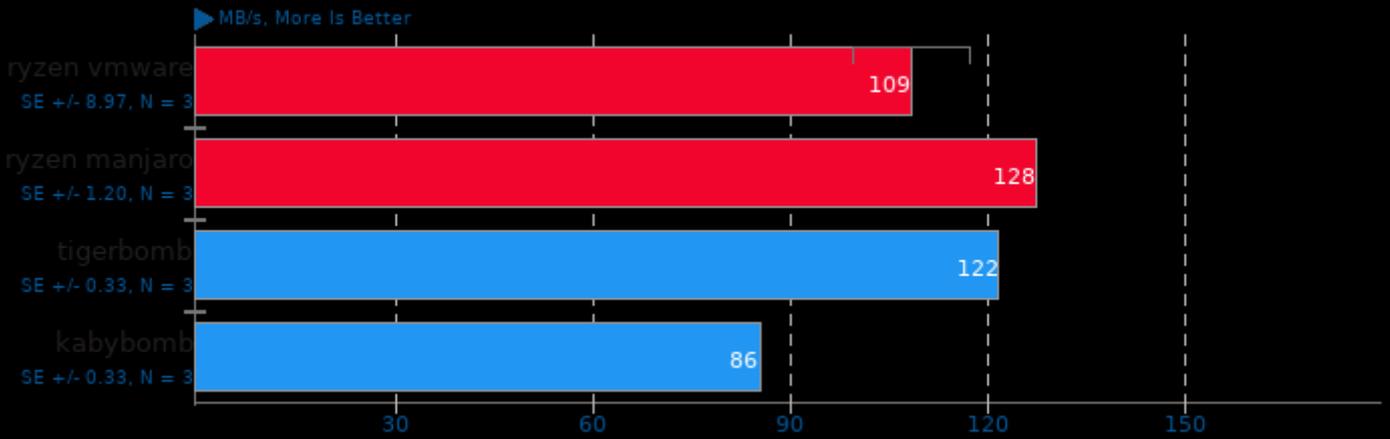
Test: Zstd 1 - Process: Decompression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

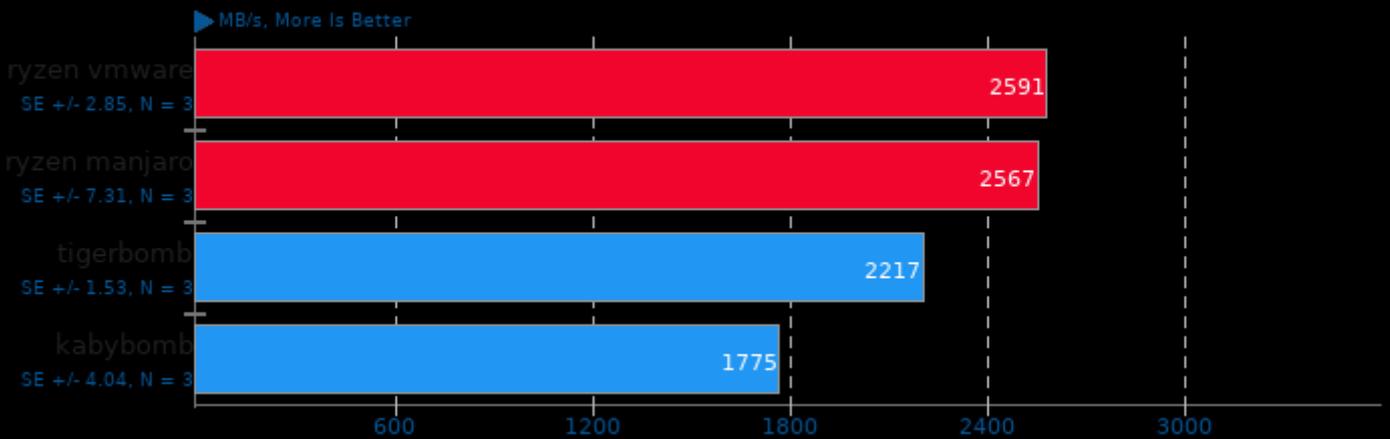
Test: Zstd 8 - Process: Compression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

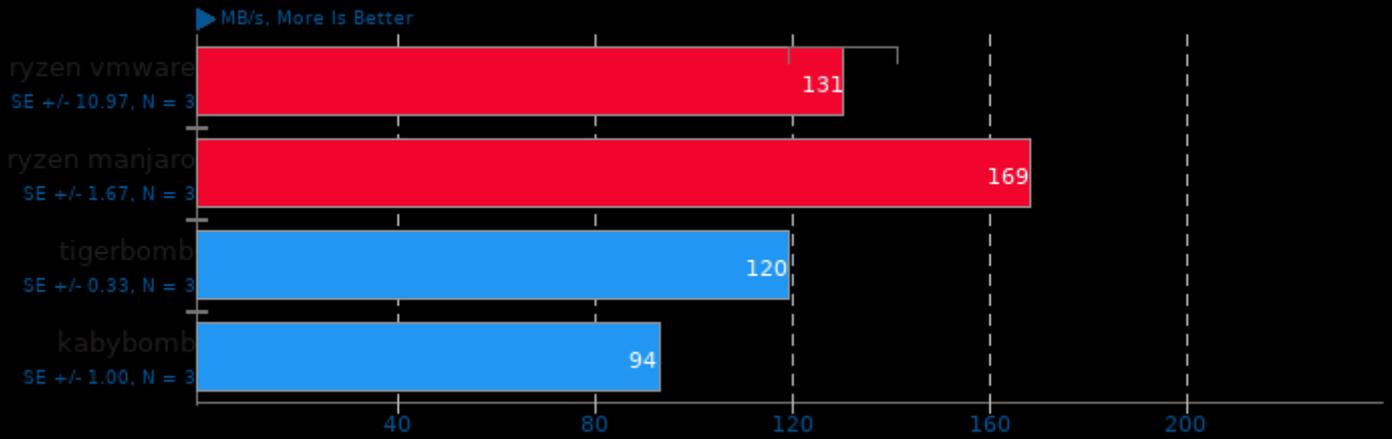
Test: Zstd 8 - Process: Decompression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

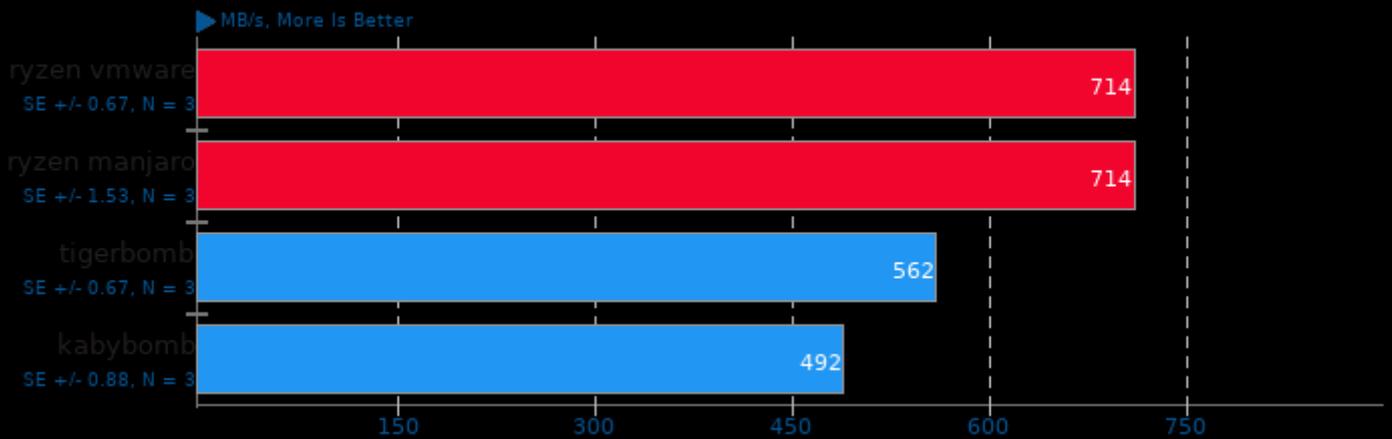
Test: Crush 0 - Process: Compression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

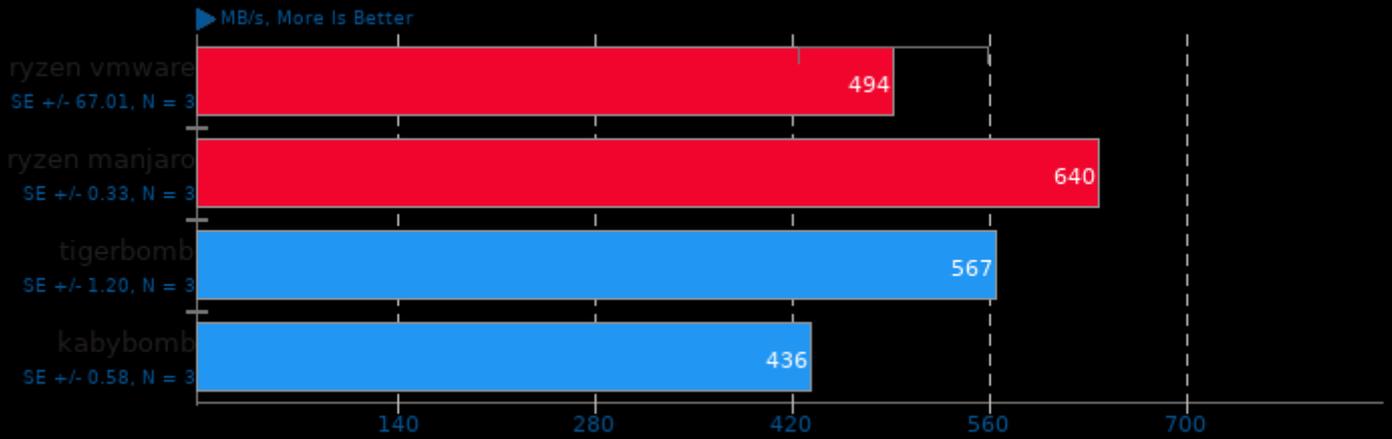
Test: Crush 0 - Process: Decompression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

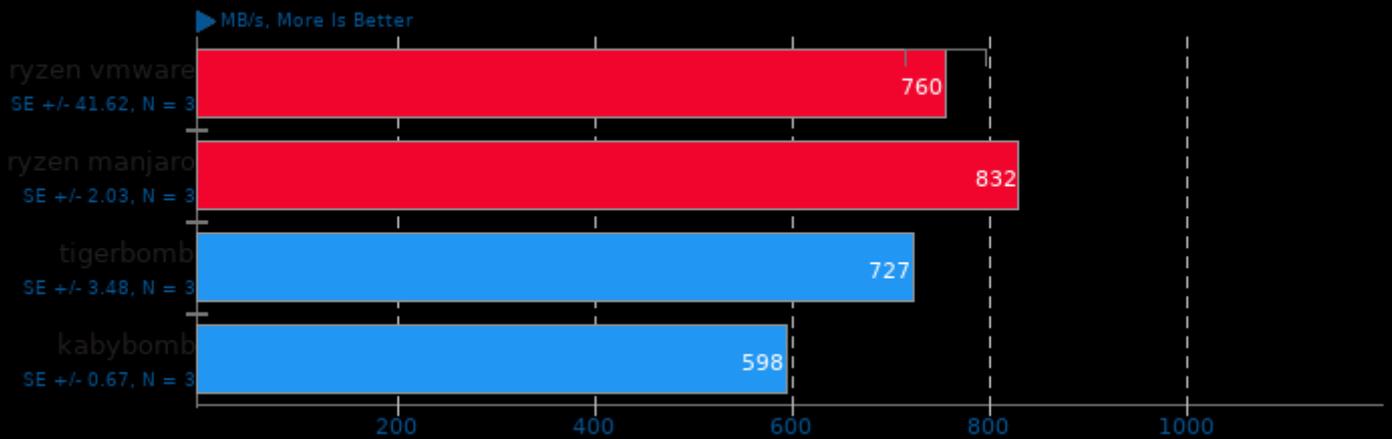
Test: Brotli 0 - Process: Compression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

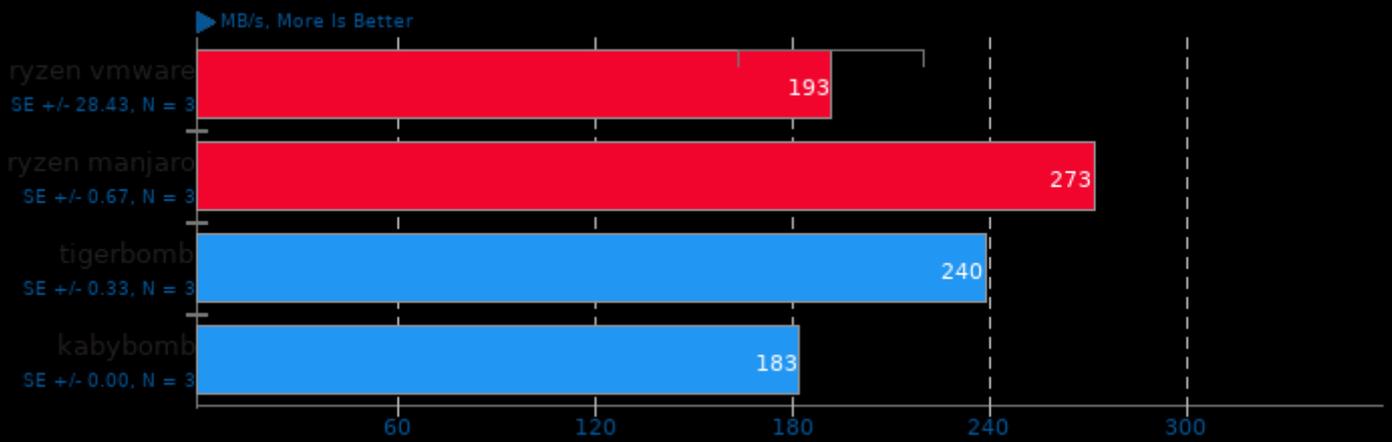
Test: Brotli 0 - Process: Decompression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

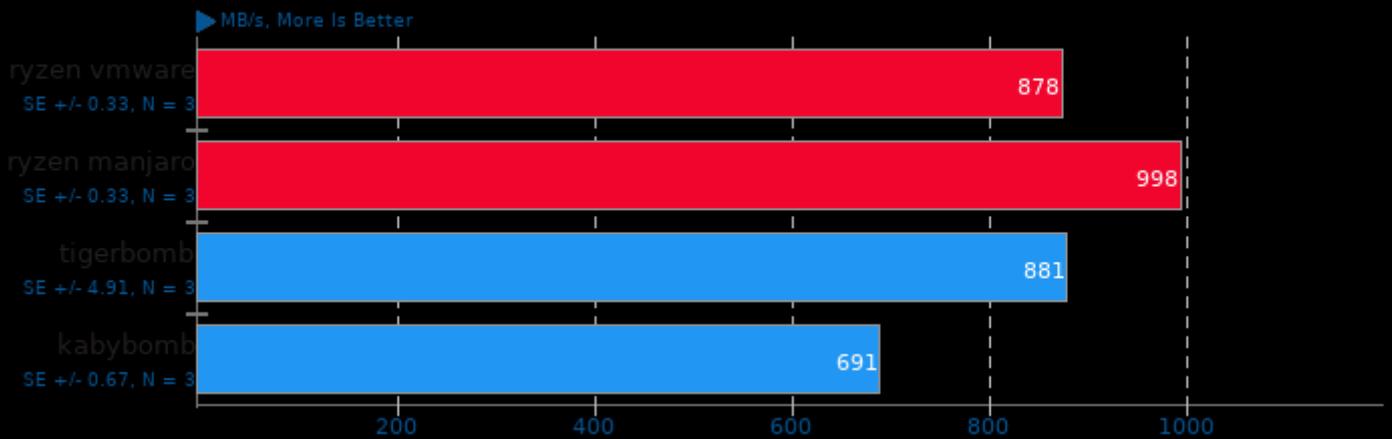
Test: Brotli 2 - Process: Compression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

### Izbench 1.8

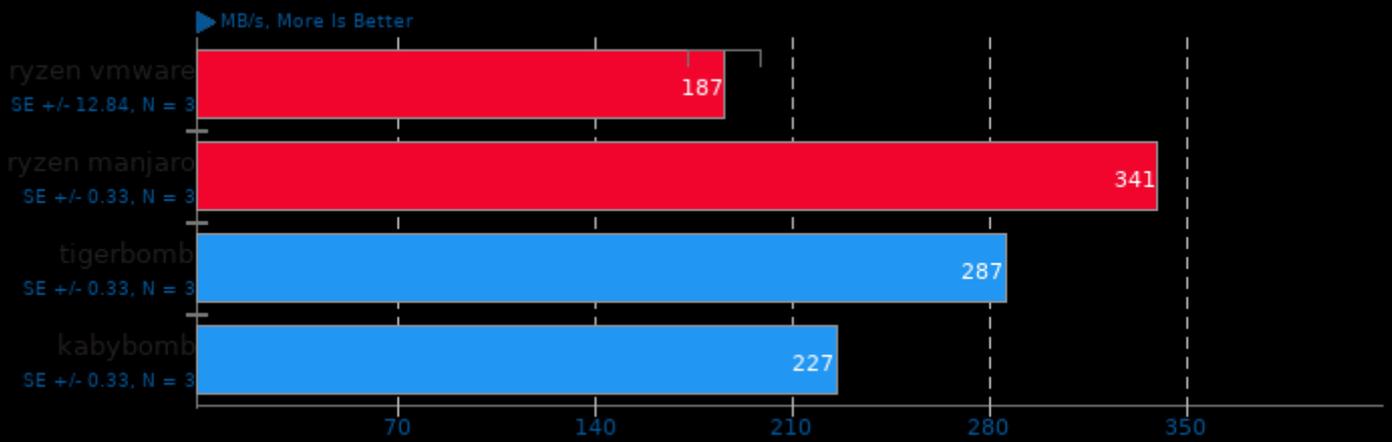
Test: Brotli 2 - Process: Decompression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

## Izbench 1.8

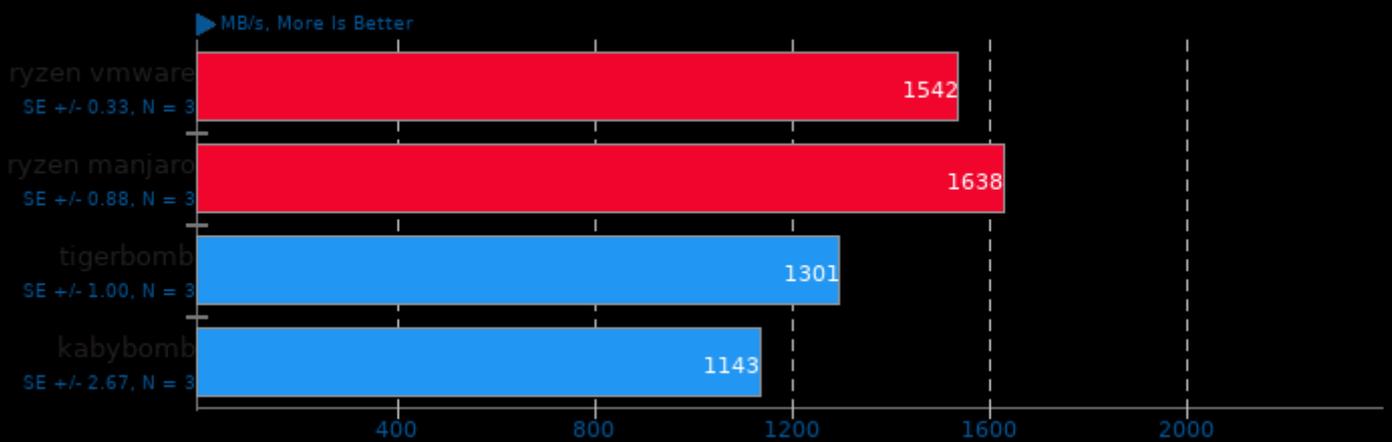
Test: Libdeflate 1 - Process: Compression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

## Izbench 1.8

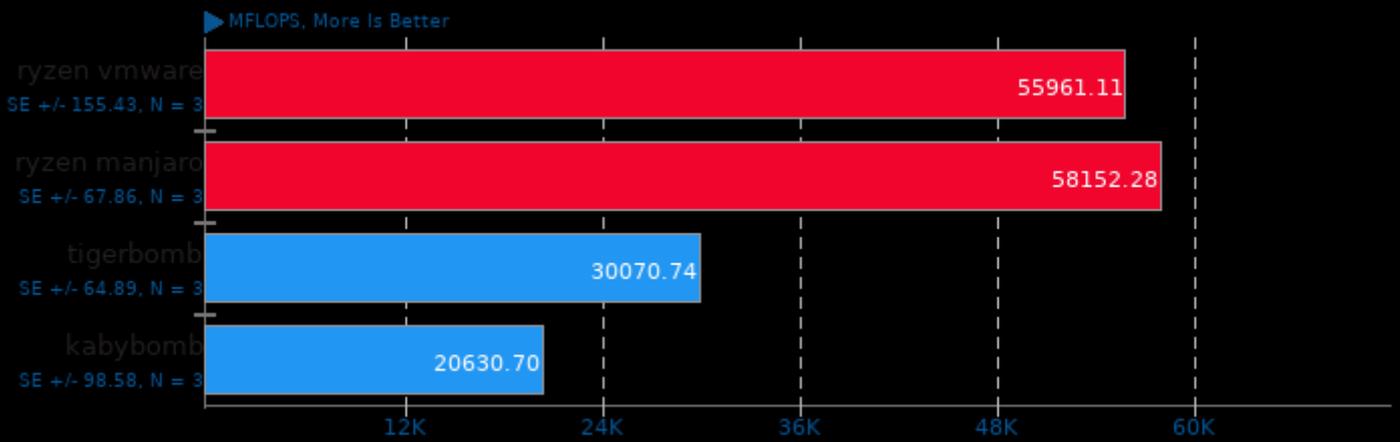
Test: Libdeflate 1 - Process: Decompression



1. (CXX) g++ options: -pthread -fomit-frame-pointer -fstrict-aliasing -ffast-math -O3

## FFTE 7.0

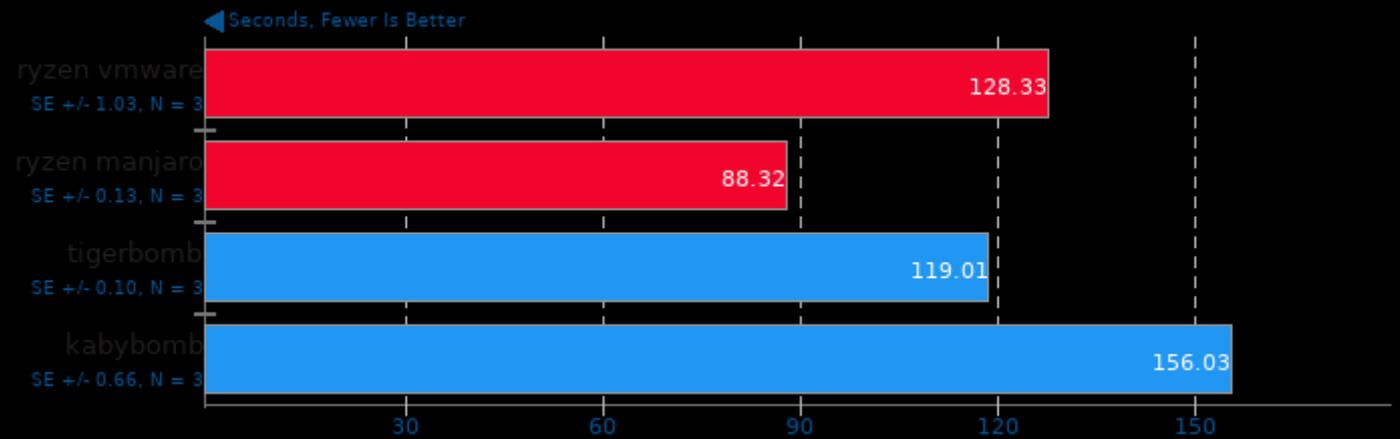
N=256, 3D Complex FFT Routine



1. (F9X) gfortran options: -O3 -fomit-frame-pointer -fopenmp

## Timed HMMer Search 3.3.2

Pfam Database Search

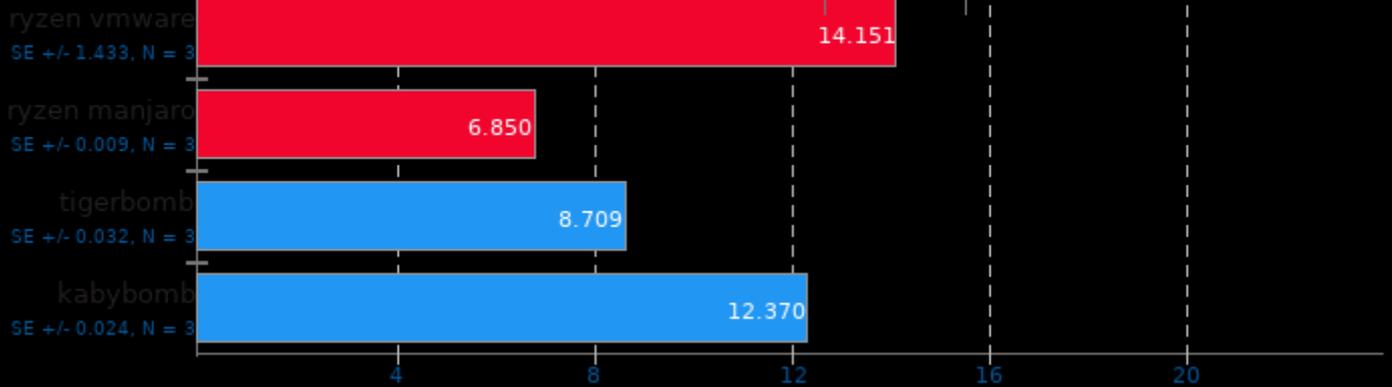


1. (CC) gcc options: -O3 -pthread -lhmmmer -leasel -lm -lmpi

### Timed MAFFT Alignment 7.471

Multiple Sequence Alignment - LSU RNA

← Seconds, Fewer Is Better

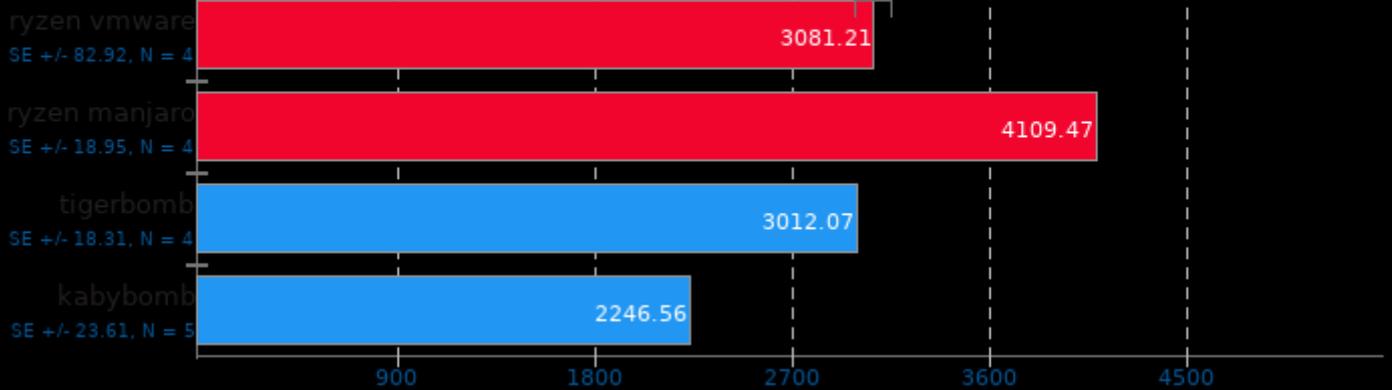


1. (CC) gcc options: -std=c99 -O3 -lm -lpthread

### Java SciMark 2.0

Computational Test: Composite

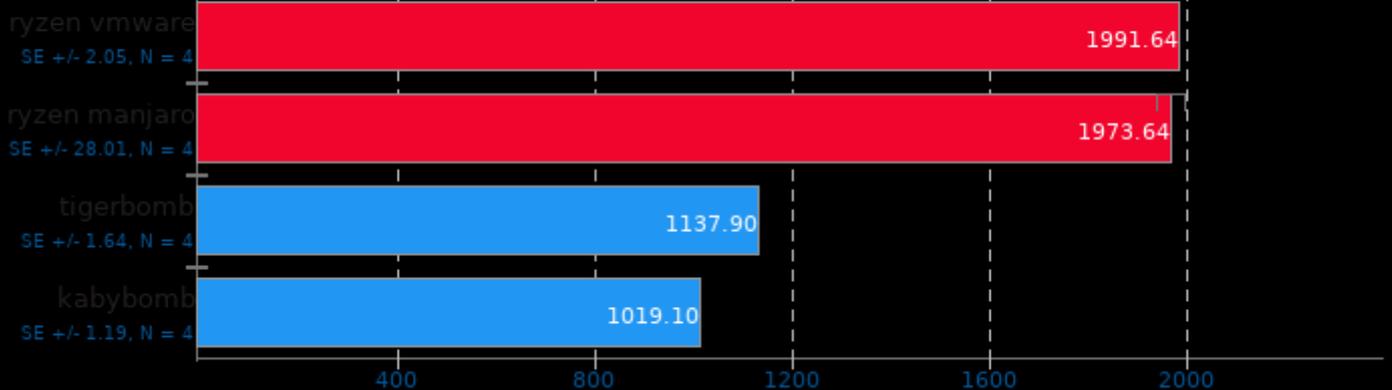
▶ Mflops, More Is Better



### Java SciMark 2.0

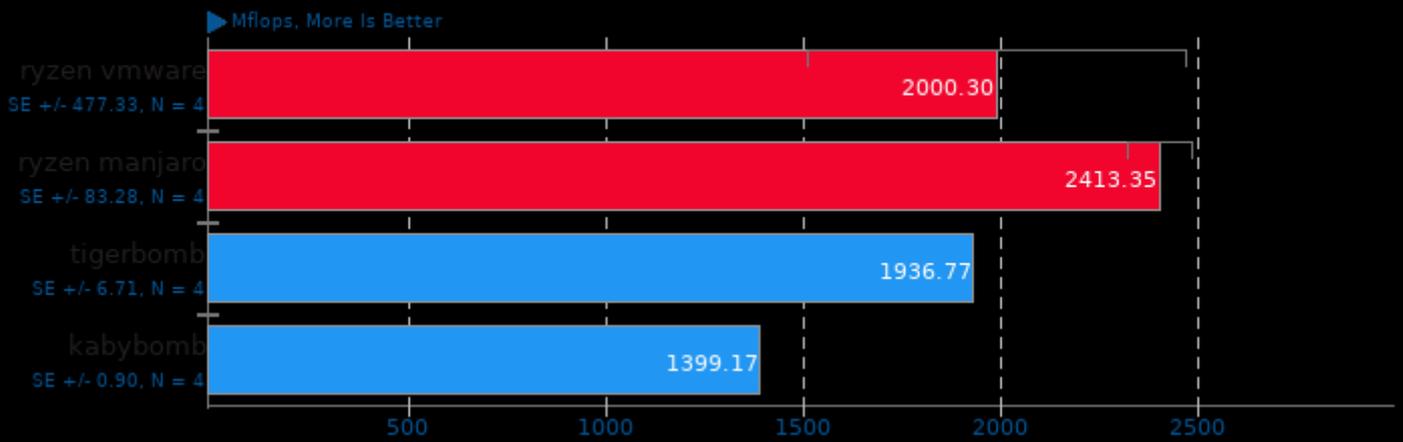
Computational Test: Monte Carlo

▶ Mflops, More Is Better



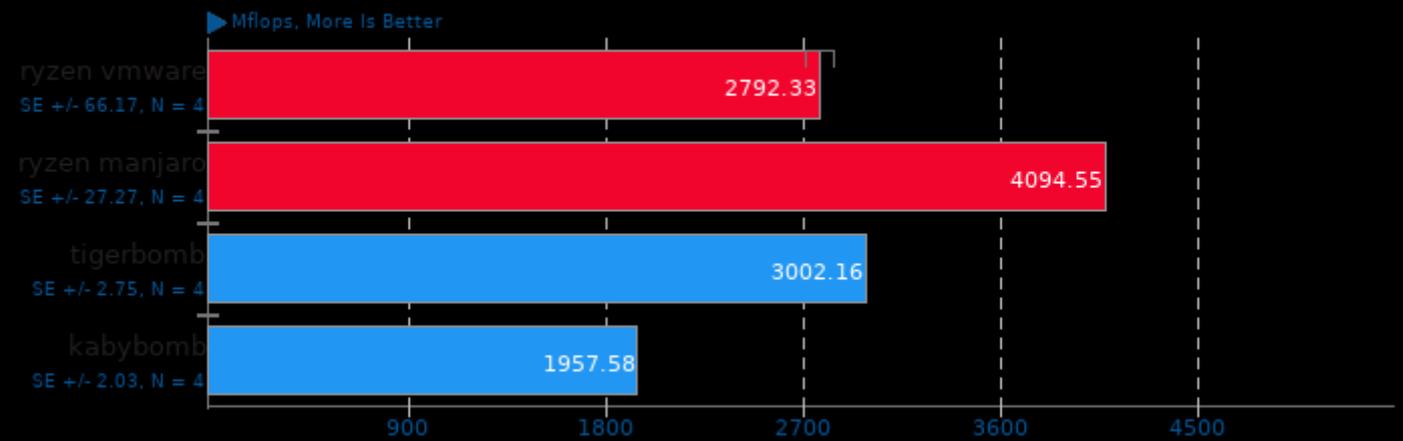
### Java SciMark 2.0

Computational Test: Fast Fourier Transform



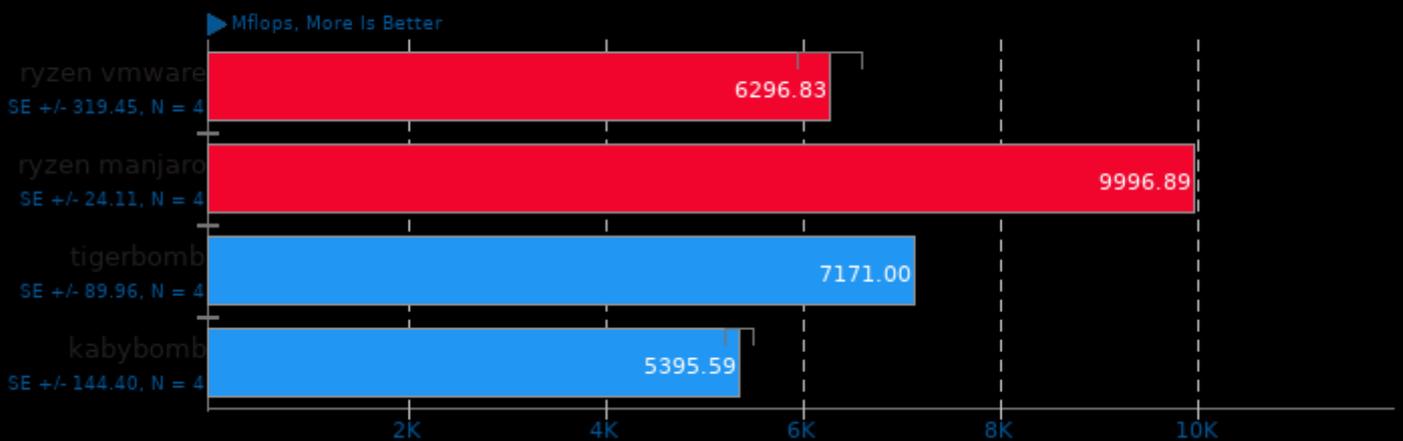
### Java SciMark 2.0

Computational Test: Sparse Matrix Multiply



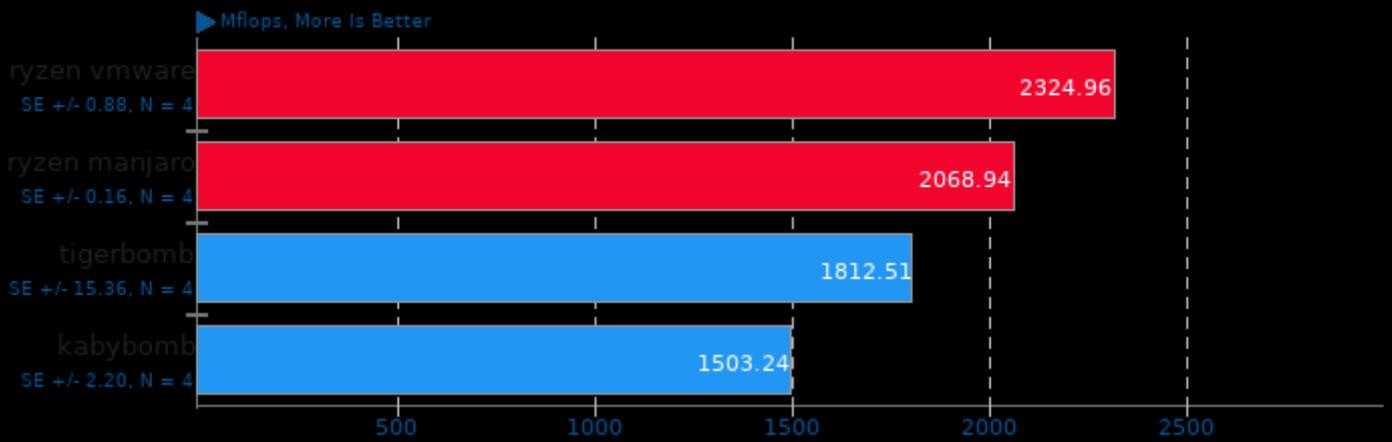
### Java SciMark 2.0

Computational Test: Dense LU Matrix Factorization



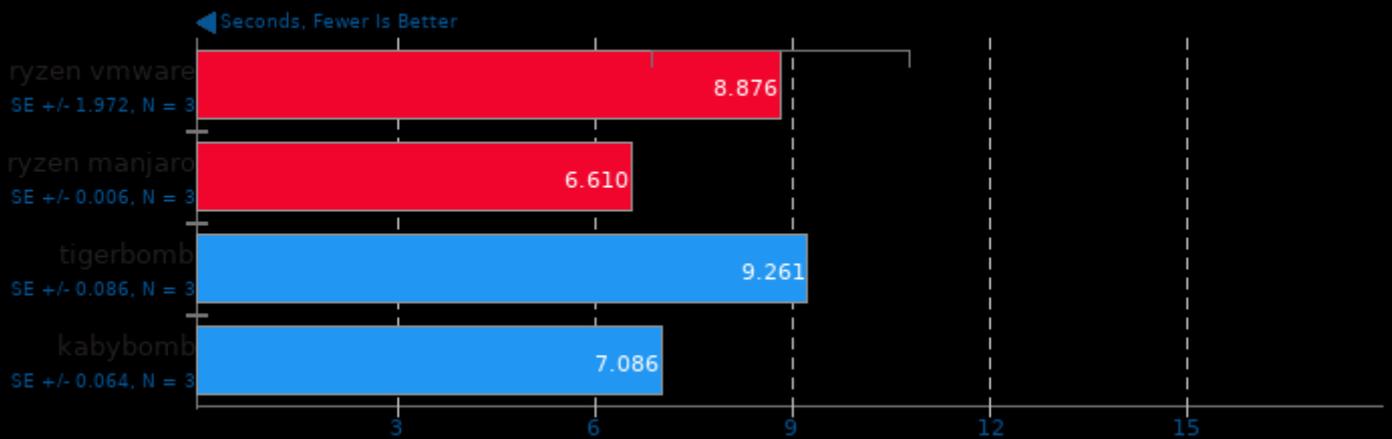
## Java SciMark 2.0

Computational Test: Jacobi Successive Over-Relaxation



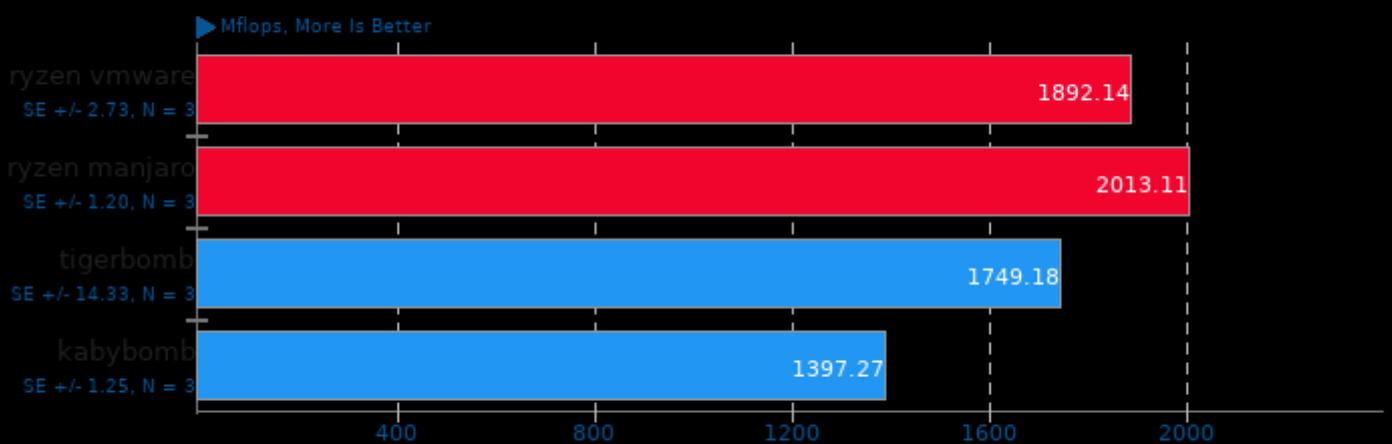
## Bork File Encrypter 1.4

File Encryption Time



## LuajIT 2.1-git

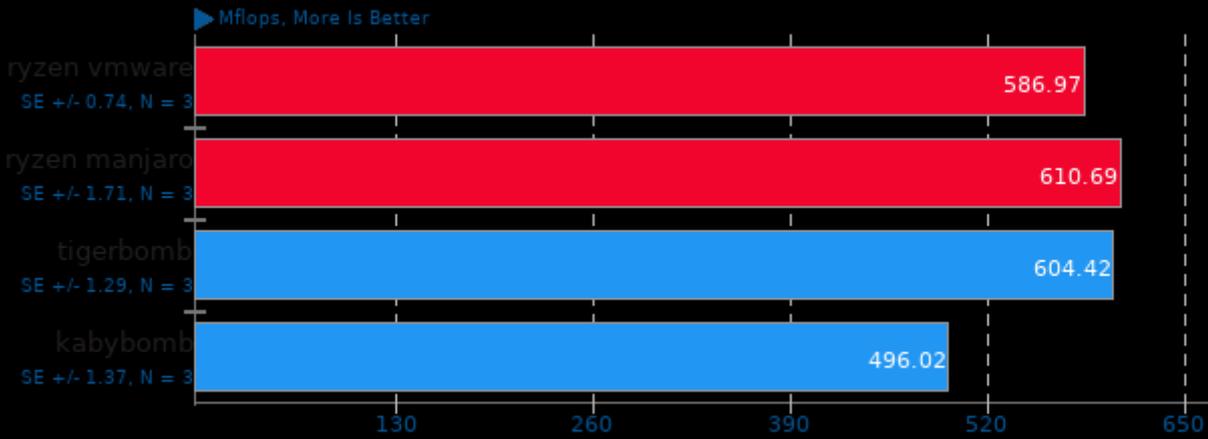
Test: Composite



1. (CO) gcc options: -lm -ldl -O2 -fomit-frame-pointer -U\_FORTIFY\_SOURCE -fno-stack-protector

### LuajIT 2.1-git

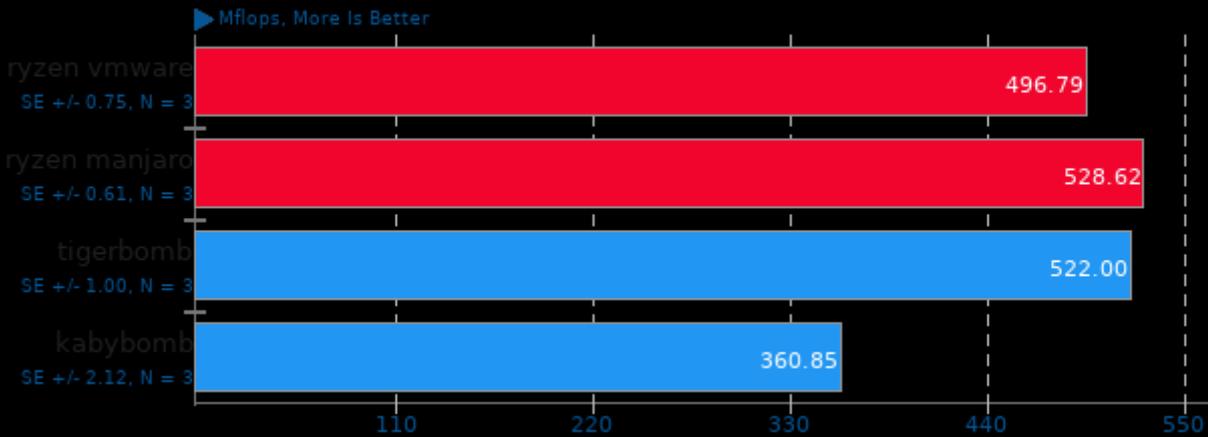
Test: Monte Carlo



1, (CC) gcc options: -lm -ldl -O2 -fomit-frame-pointer -U\_FORTIFY\_SOURCE -fno-stack-protector

### LuajIT 2.1-git

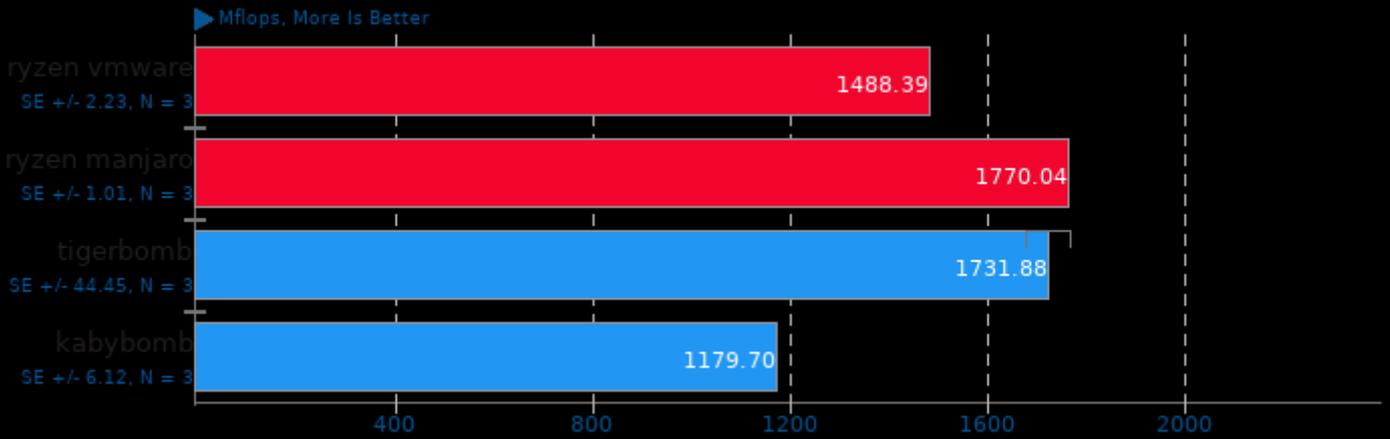
Test: Fast Fourier Transform



1, (CC) gcc options: -lm -ldl -O2 -fomit-frame-pointer -U\_FORTIFY\_SOURCE -fno-stack-protector

### LuajIT 2.1-git

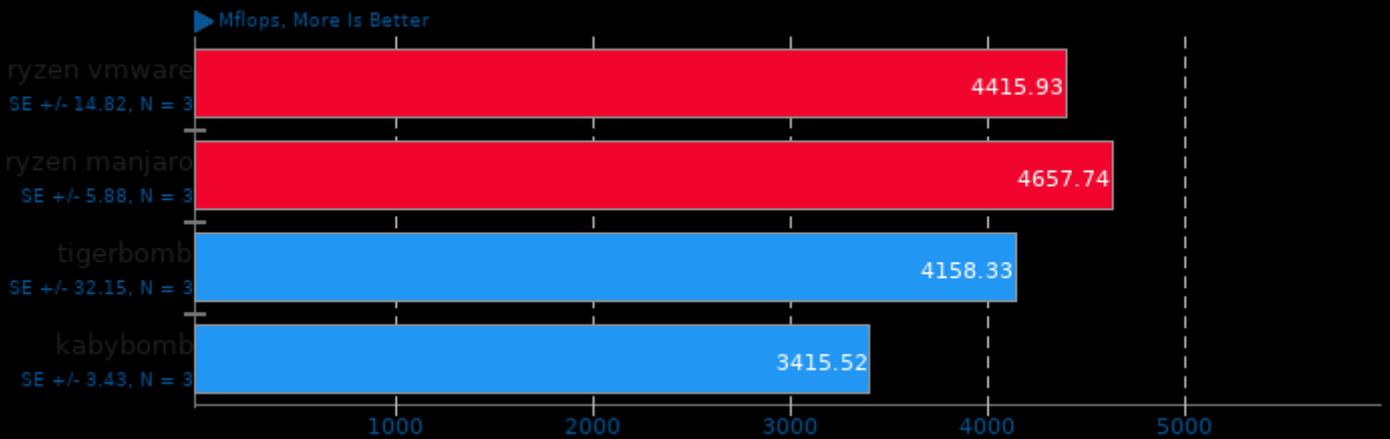
Test: Sparse Matrix Multiply



1. (CC) gcc options: -lm -ldl -O2 -fomit-frame-pointer -U\_FORTIFY\_SOURCE -fno-stack-protector

### LuajIT 2.1-git

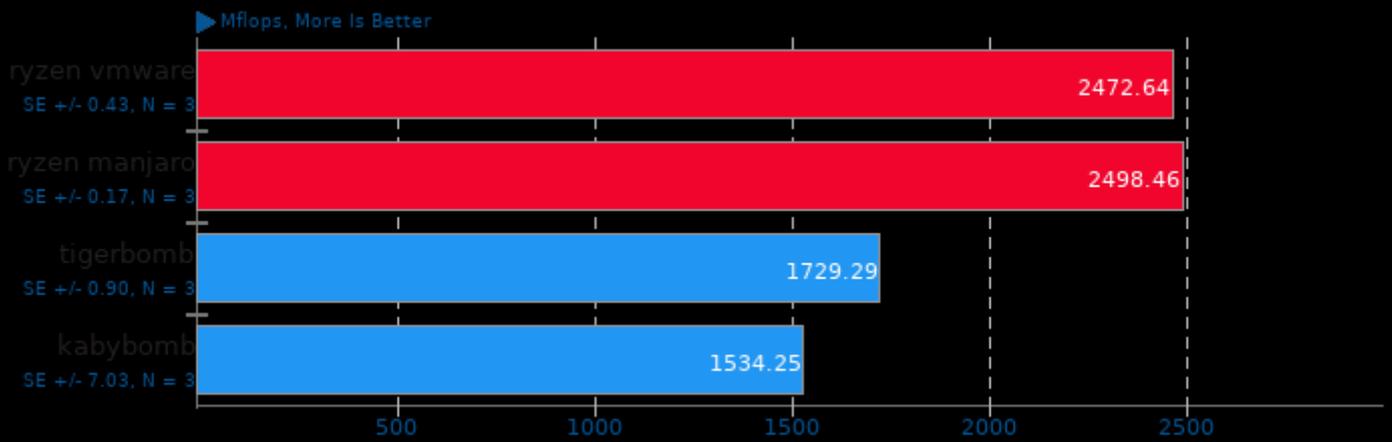
Test: Dense LU Matrix Factorization



1. (CC) gcc options: -lm -ldl -O2 -fomit-frame-pointer -U\_FORTIFY\_SOURCE -fno-stack-protector

## LuajIT 2.1-git

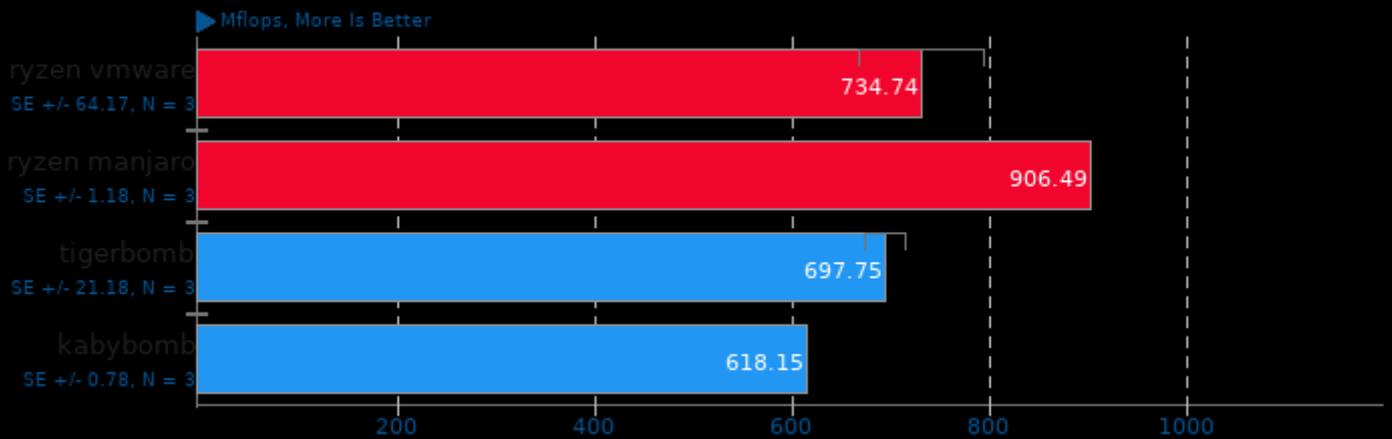
Test: Jacobi Successive Over-Relaxation



1. (CC) gcc options: -lm -ldl -O2 -fomit-frame-pointer -U\_FORTIFY\_SOURCE -fno-stack-protector

## SciMark 2.0

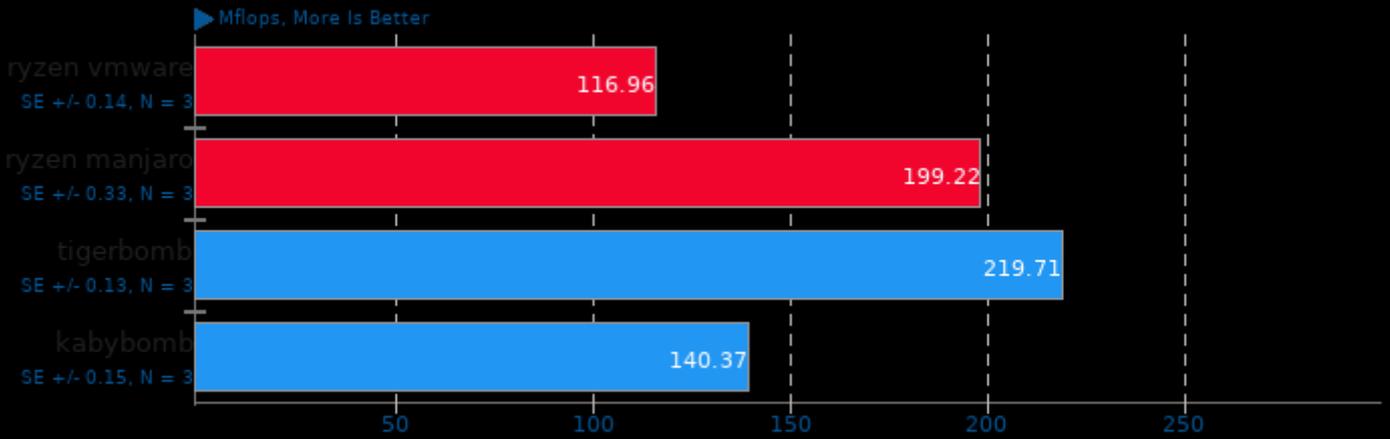
Computational Test: Composite



1. (CC) gcc options: -lm

### SciMark 2.0

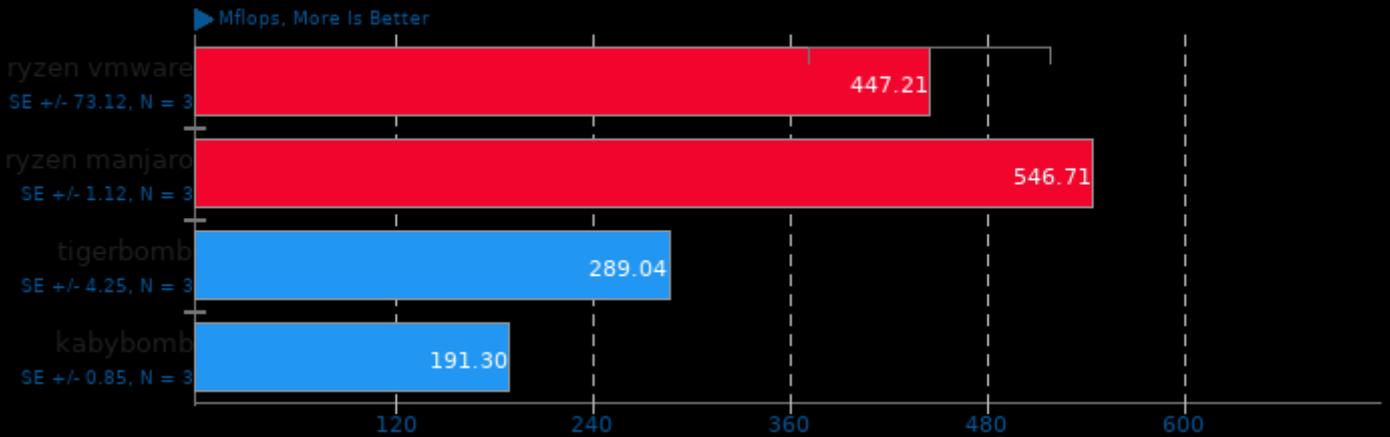
Computational Test: Monte Carlo



1. (CC) gcc options: -lm

### SciMark 2.0

Computational Test: Fast Fourier Transform

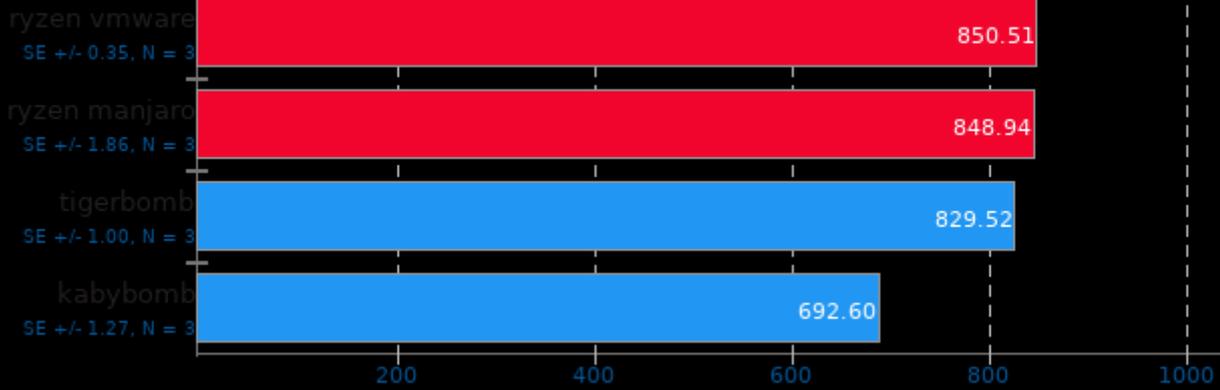


1. (CC) gcc options: -lm

### SciMark 2.0

Computational Test: Sparse Matrix Multiply

Mflops, More Is Better

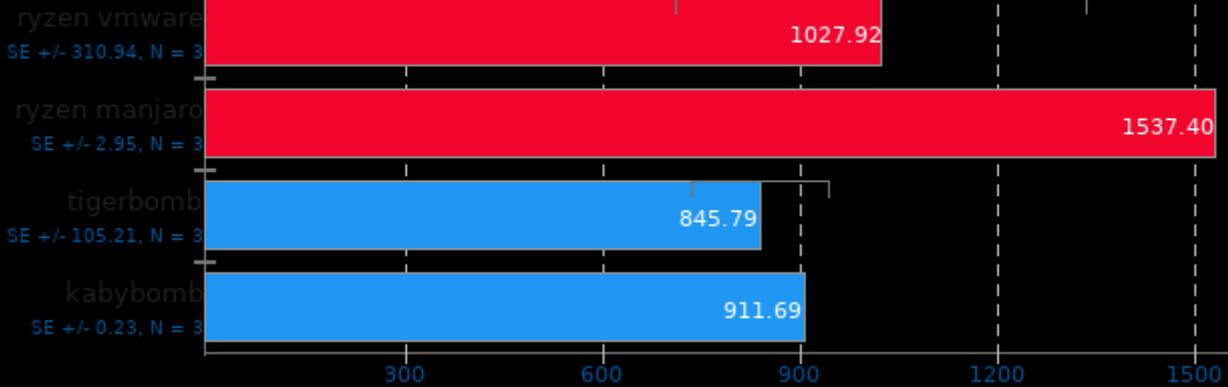


1, (CC) gcc options: -lm

### SciMark 2.0

Computational Test: Dense LU Matrix Factorization

Mflops, More Is Better

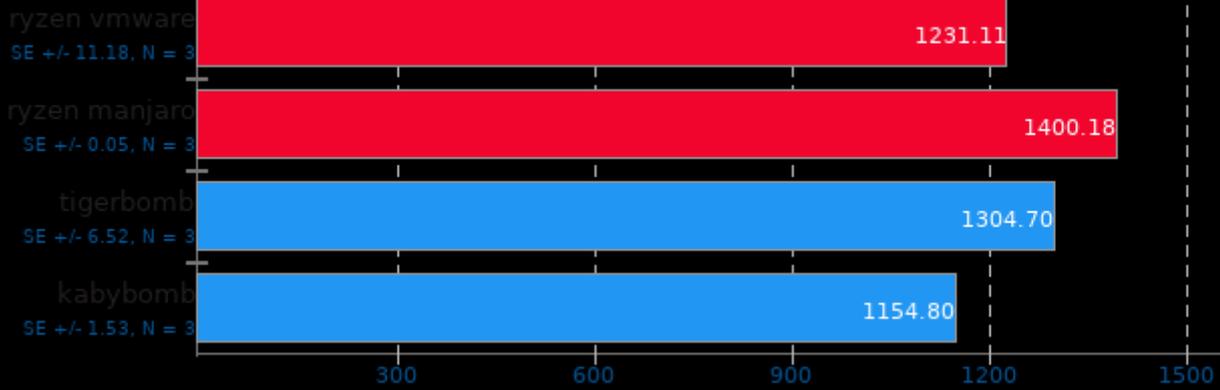


1, (CC) gcc options: -lm

## SciMark 2.0

Computational Test: Jacobi Successive Over-Relaxation

Mflops, More Is Better

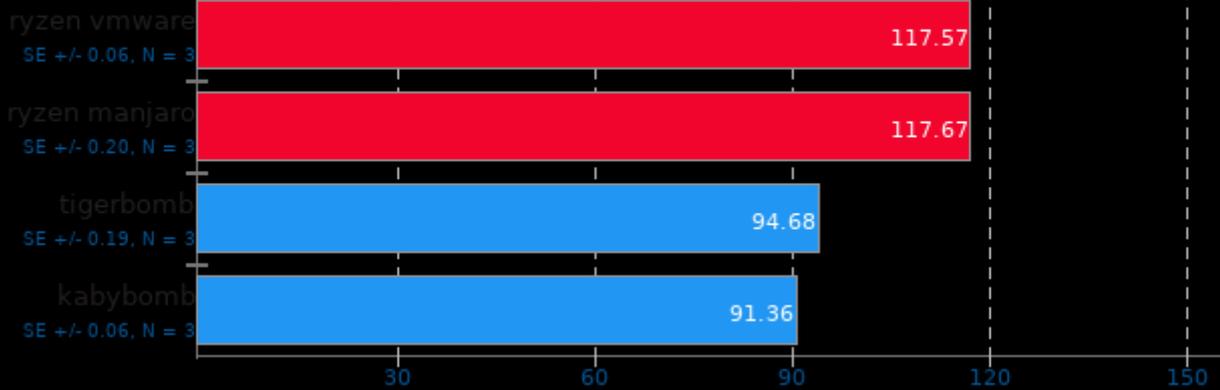


1. (CC) gcc options: -lm

## Botan 2.17.3

Test: KASUMI

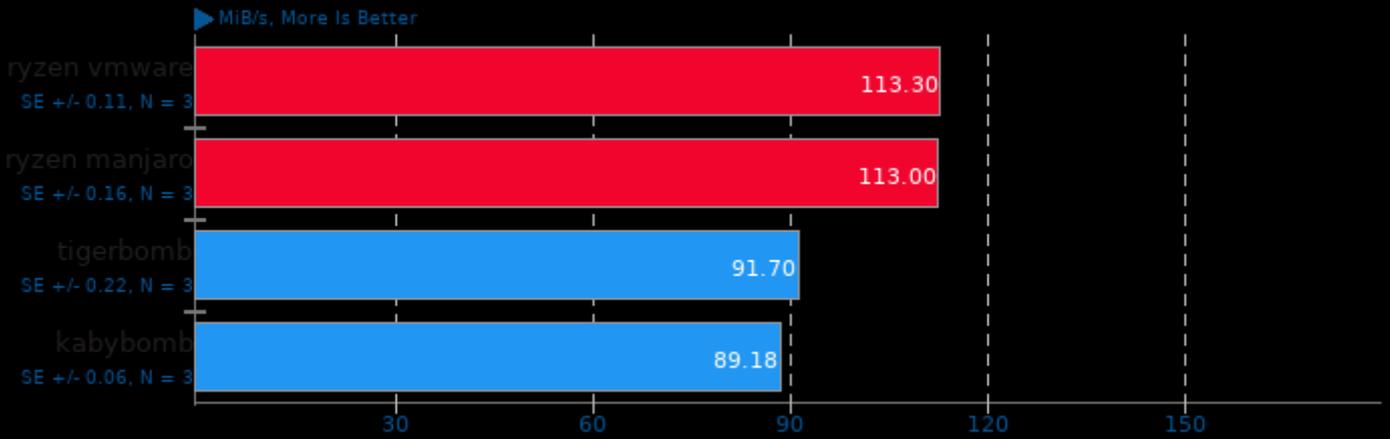
MiB/s, More Is Better



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

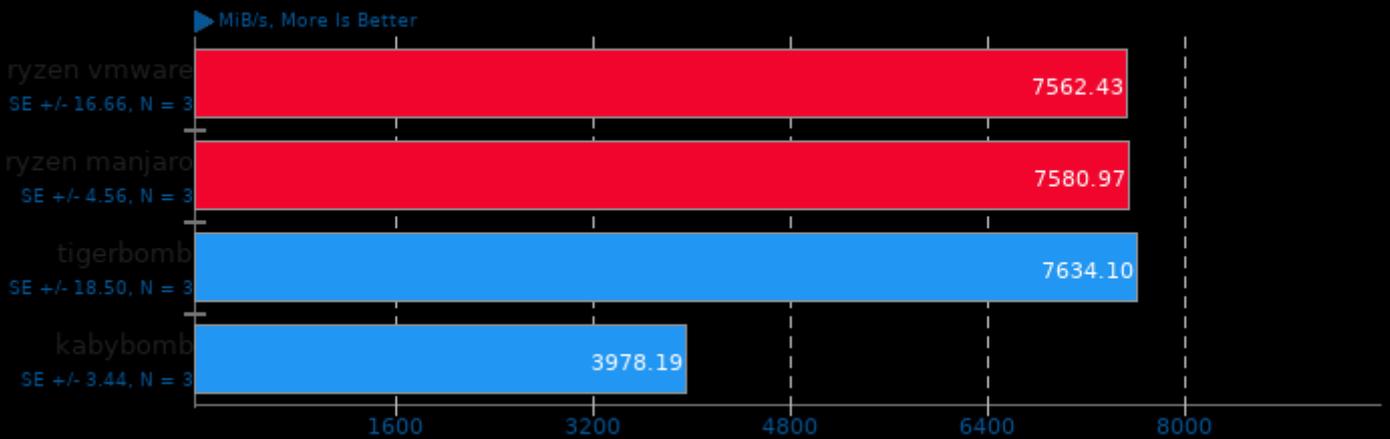
Test: KASUMI - Decrypt



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

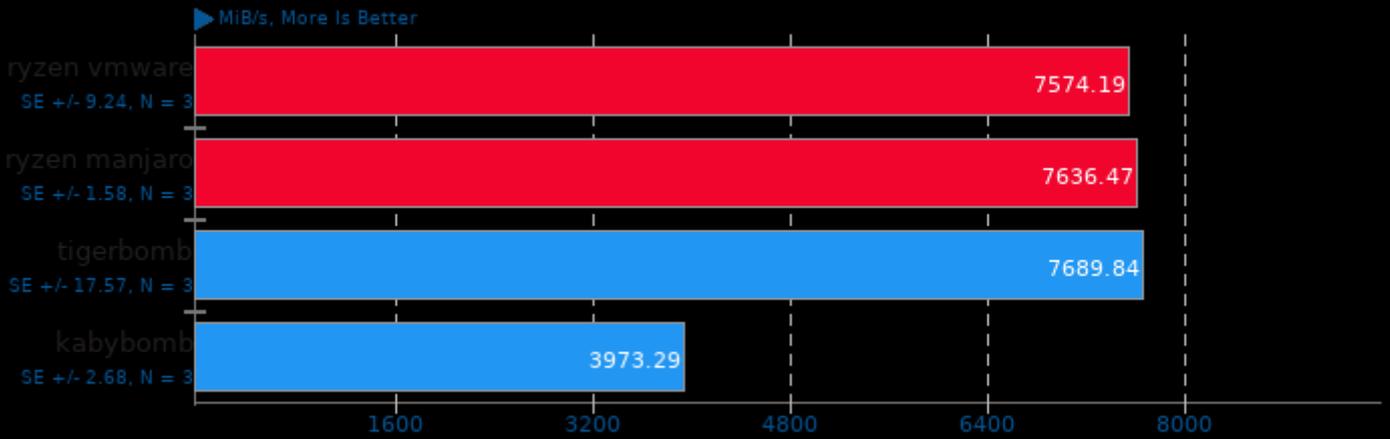
Test: AES-256



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

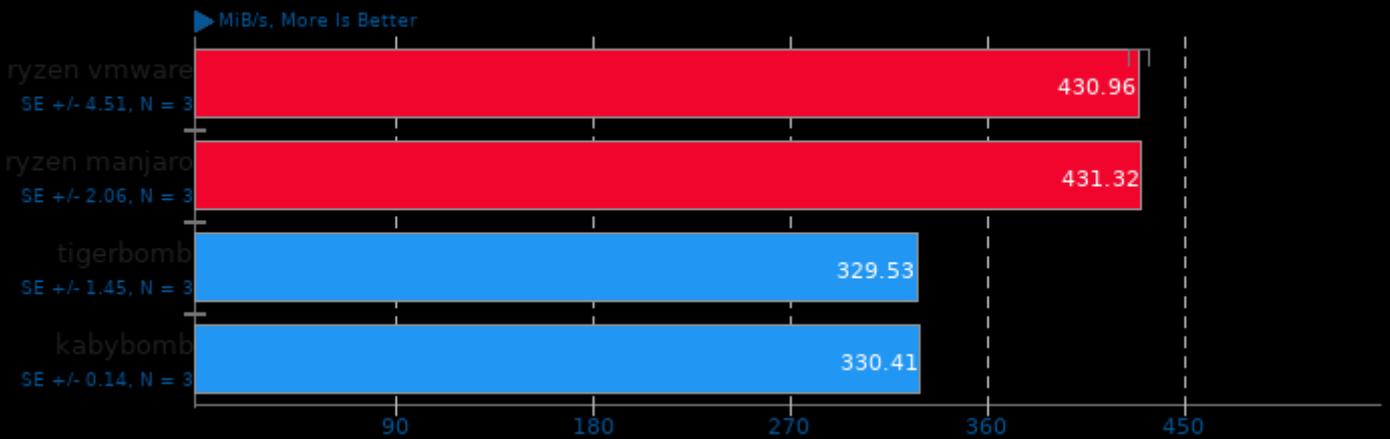
Test: AES-256 - Decrypt



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

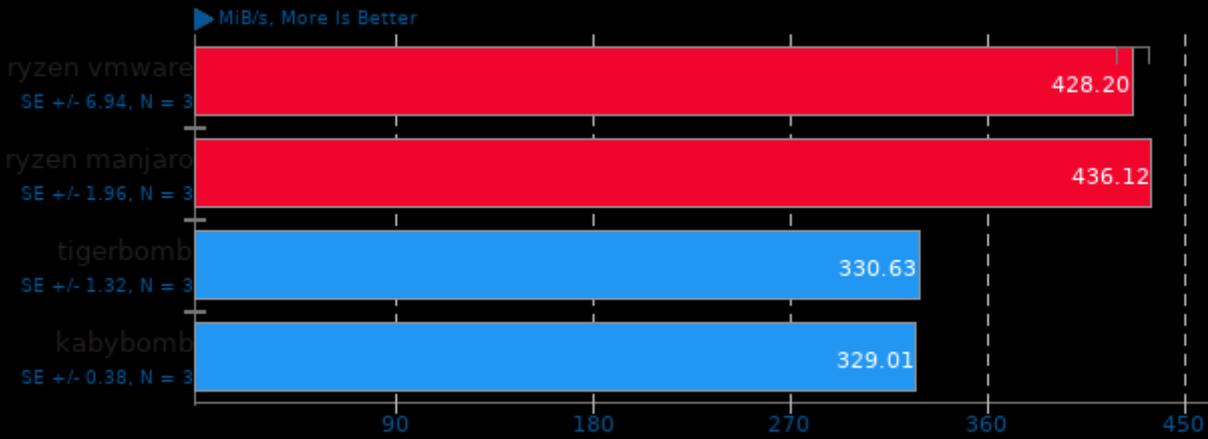
Test: Twofish



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

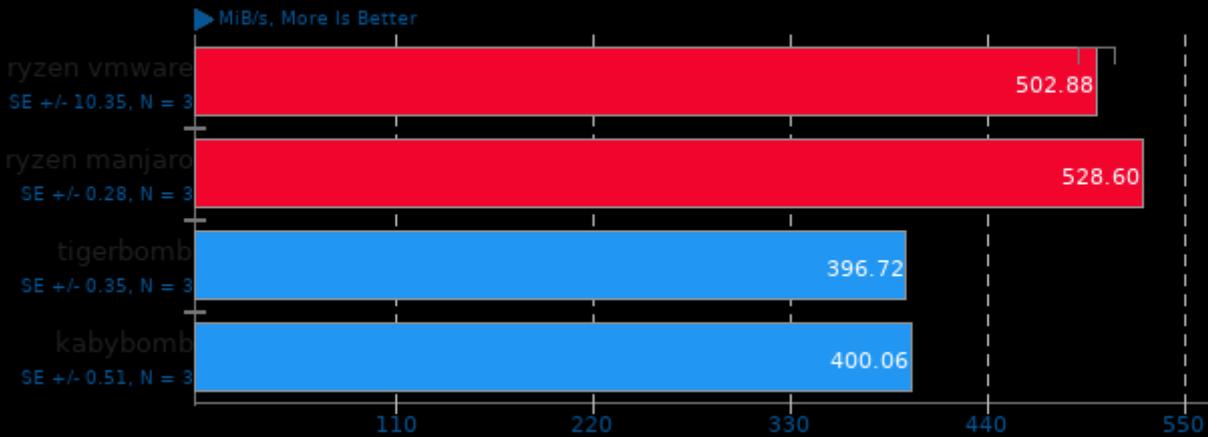
Test: Twofish - Decrypt



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

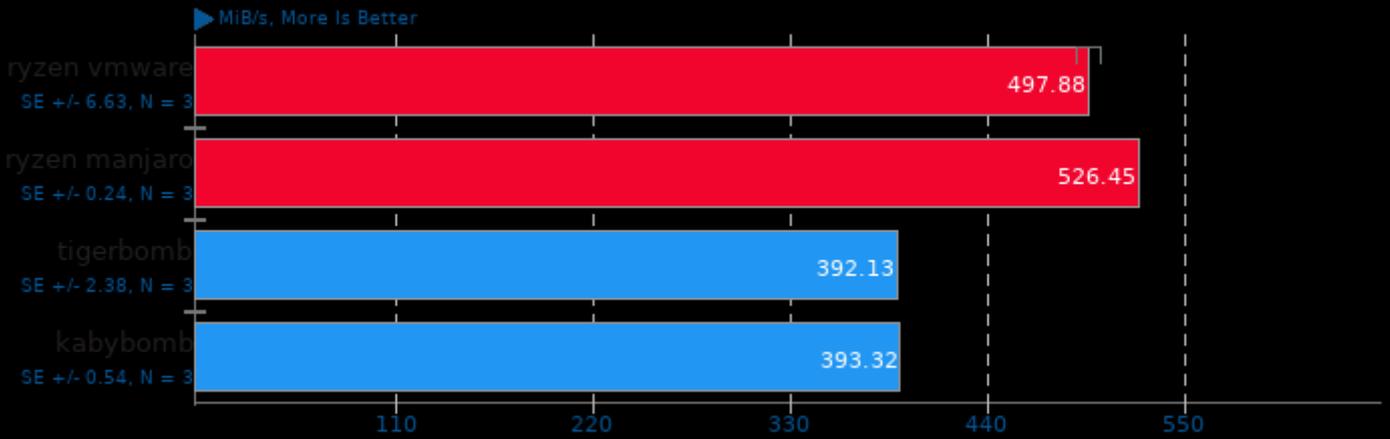
Test: Blowfish



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

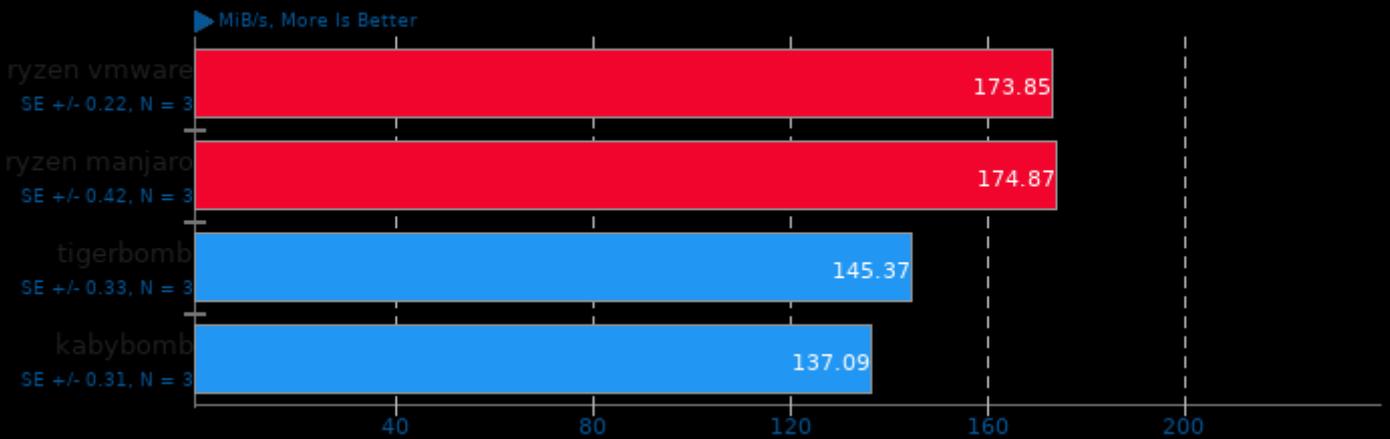
Test: Blowfish - Decrypt



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

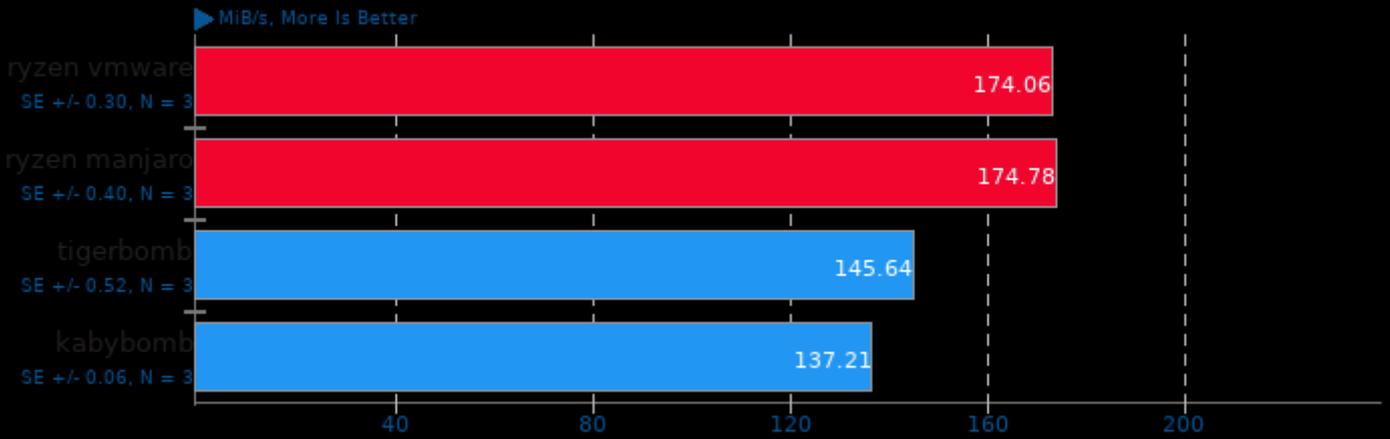
Test: CAST-256



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

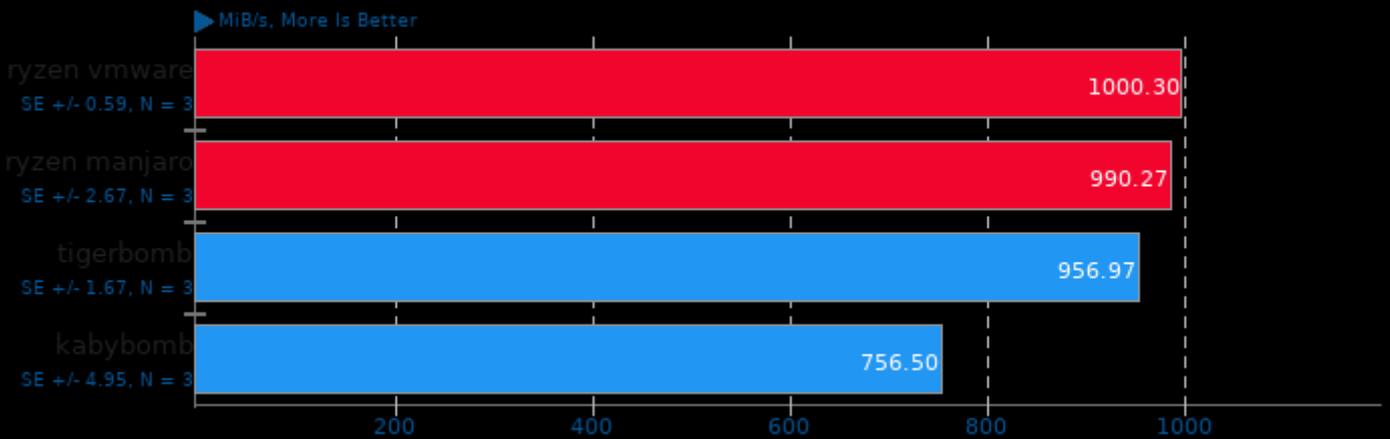
Test: CAST-256 - Decrypt



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

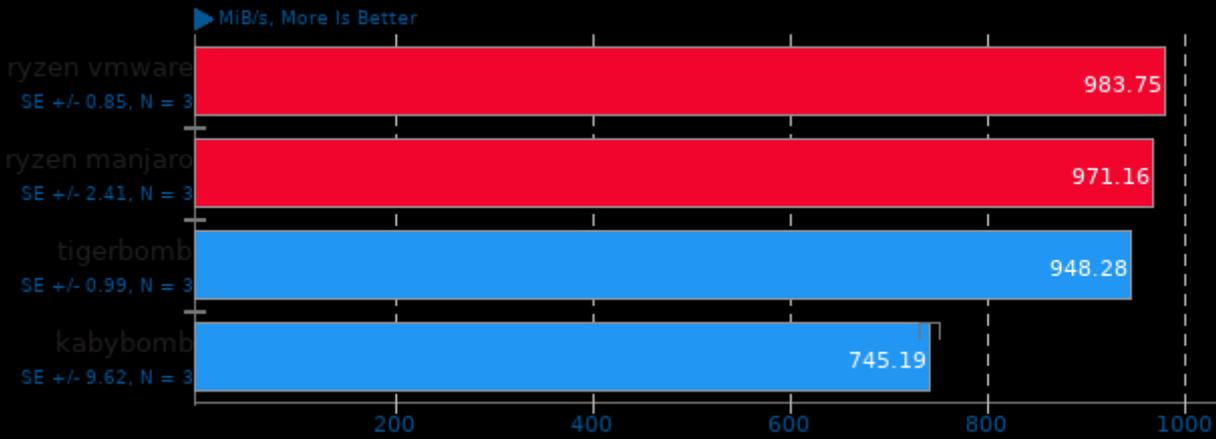
Test: ChaCha20Poly1305



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### Botan 2.17.3

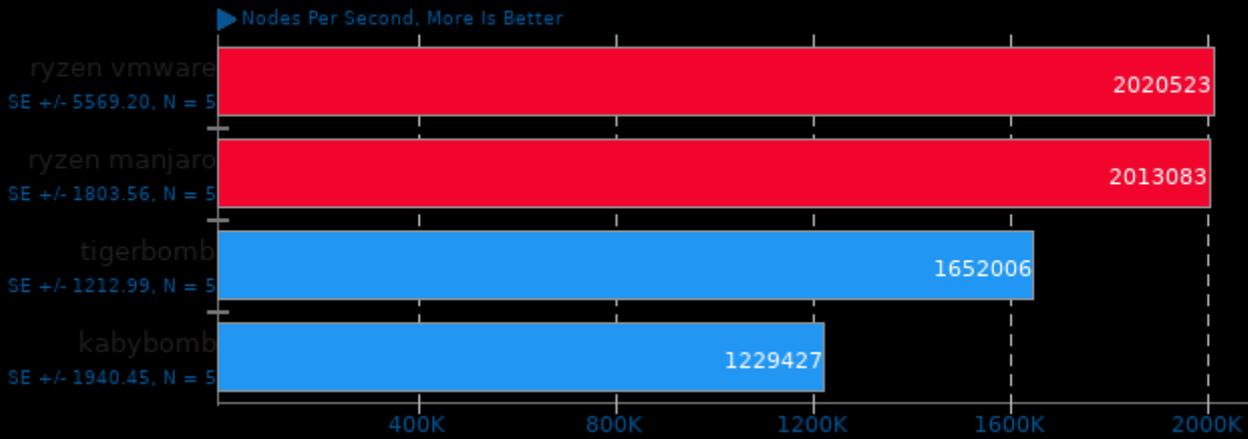
Test: ChaCha20Poly1305 - Decrypt



1. (CXX) g++ options: -fstack-protector -m64 -pthread -lbotan-2 -ldl -lrt

### TSCP 1.81

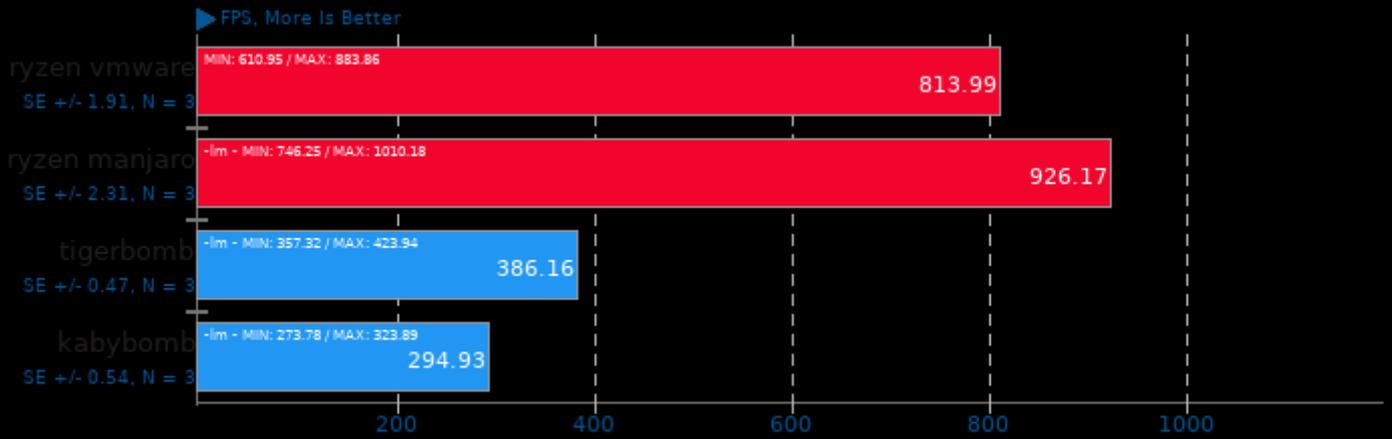
AI Chess Performance



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

### dav1d 0.9.0

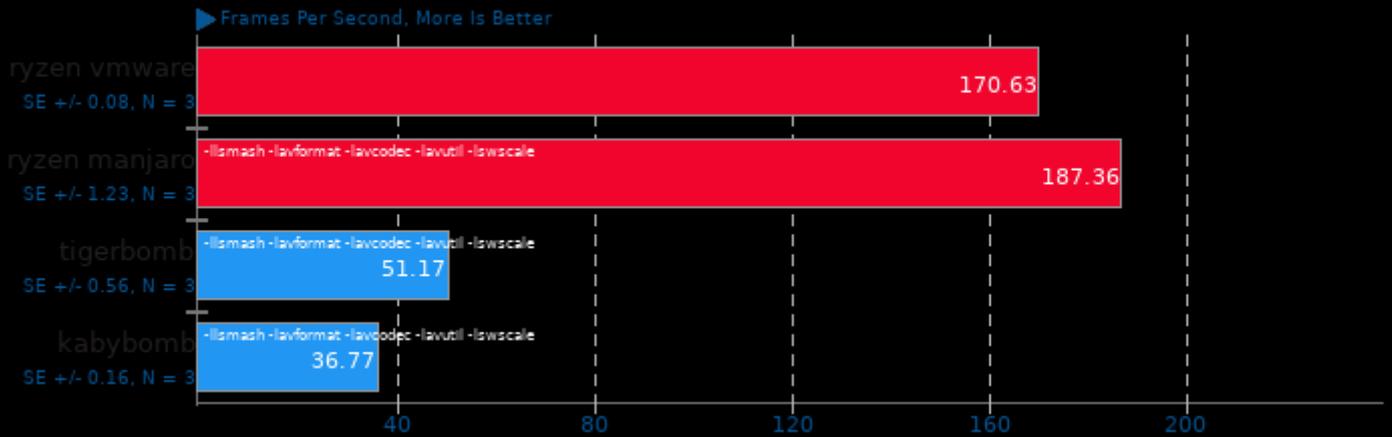
Video Input: Summer Nature 1080p



1. (CC) gcc options: -pthread

### x264 2019-12-17

H.264 Video Encoding

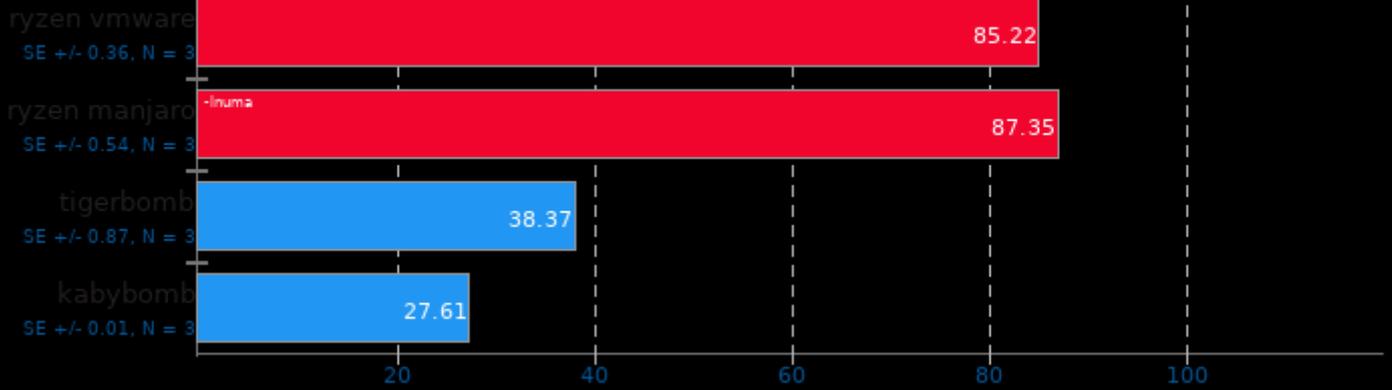


1. (CC) gcc options: -ldl -m64 -lm -lpthread -O3 -ffast-math -std=gnu99 -fPIC -fomit-frame-pointer -fno-tree-vectorize

### x265 3.4

Video Input: Bosphorus 1080p

▶ Frames Per Second, More Is Better

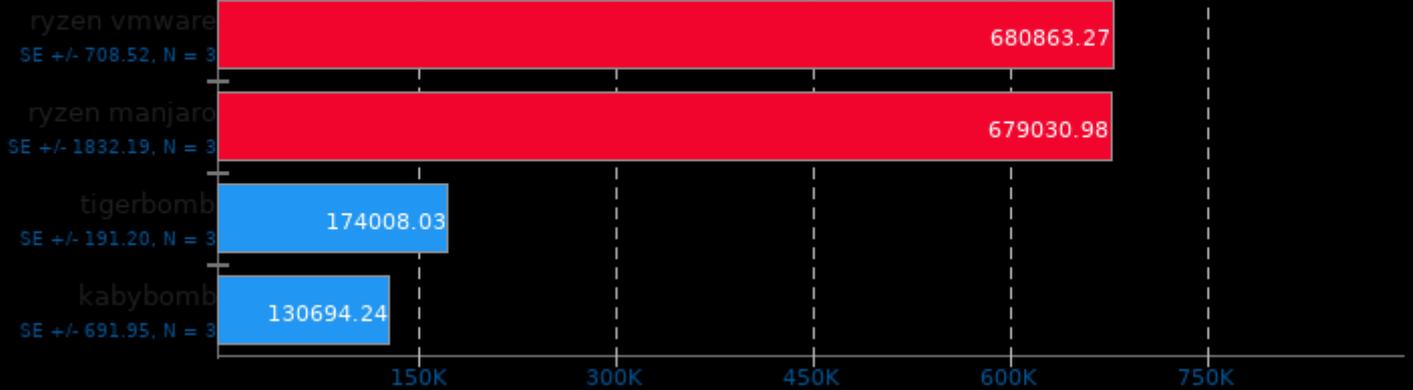


1. (CXX) g++ options: -O3 -rdynamic -lpthread -lrt -ldl

### Coremark 1.0

CoreMark Size 666 - Iterations Per Second

▶ Iterations/Sec, More Is Better

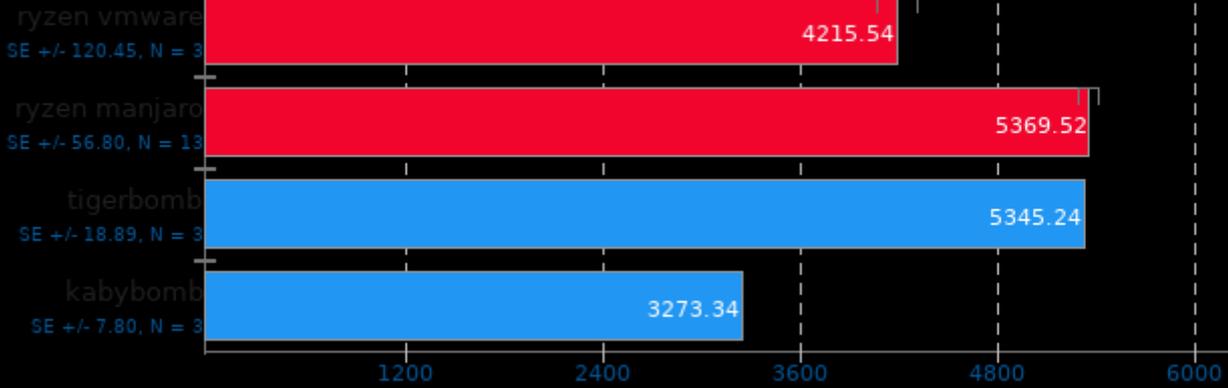


1. (CC) gcc options: -O2 -lrt -lrt

## Himeno Benchmark 3.0

Poisson Pressure Solver

► MFLOPS, More Is Better

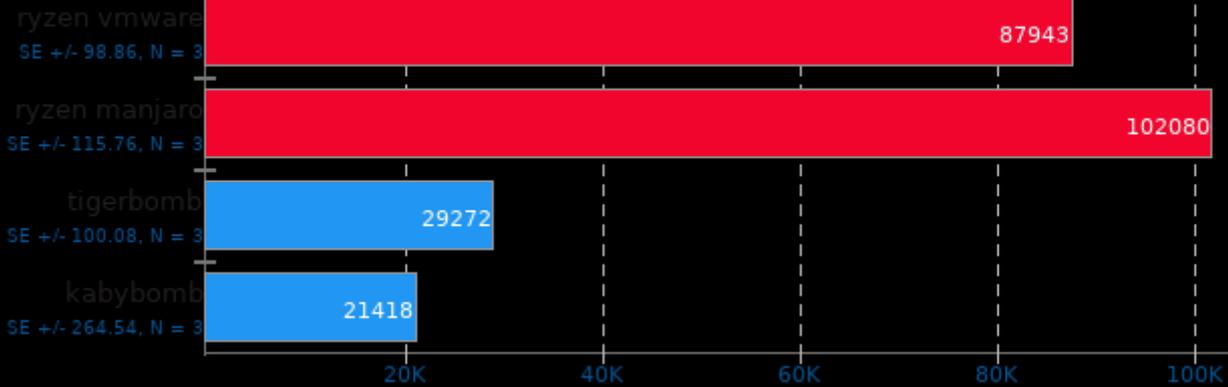


1. (CC) gcc options: -O3 -mavx2

## 7-Zip Compression 16.02

Compress Speed Test

► MIPS, More Is Better

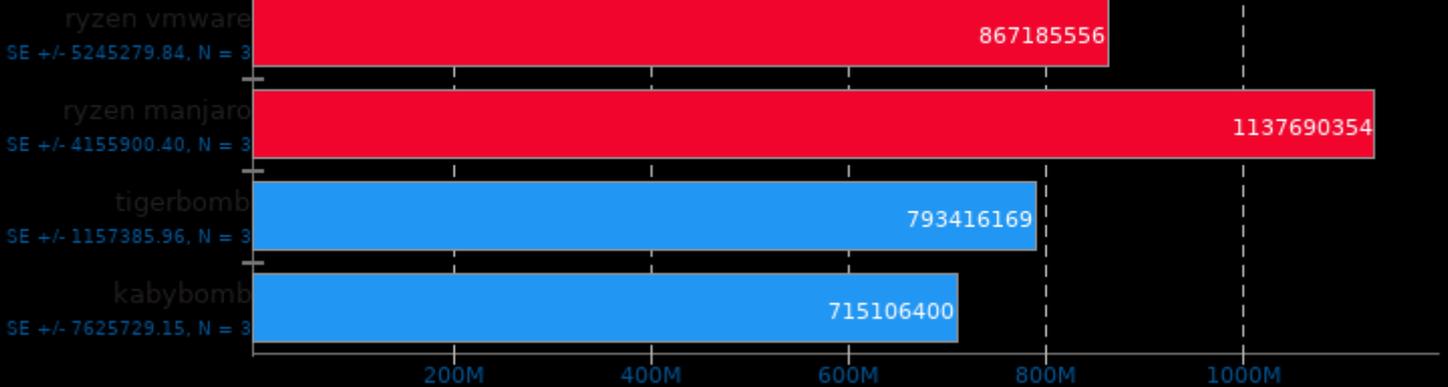


1. (CXX) g++ options: -pipe -lpthread

## Swet 1.5.16

Average

Operations Per Second, More Is Better

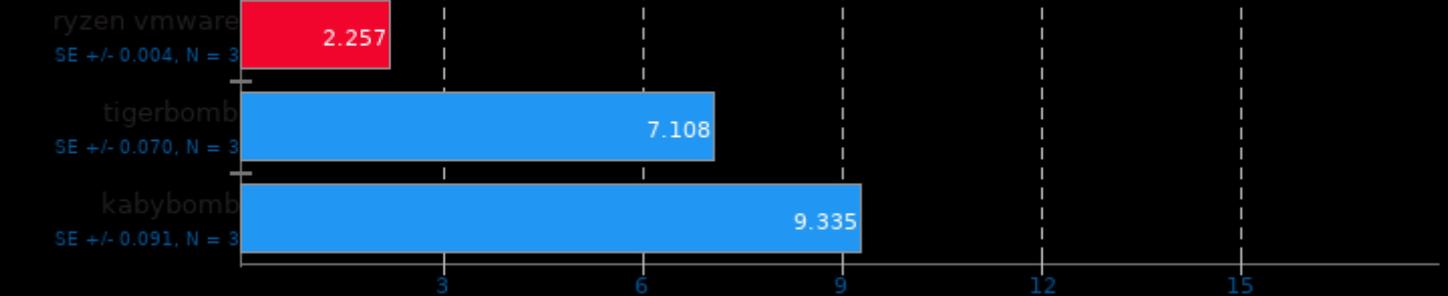


1. (GCC) gcc options: -lm -lpthread -lcurses -lrt

## Parallel BZIP2 Compression 1.1.12

256MB File Compression

Seconds, Fewer Is Better

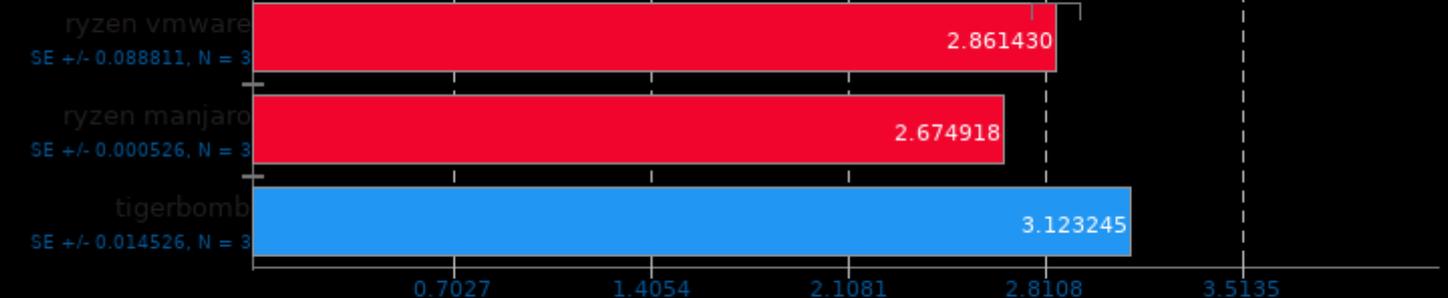


1. (CXX) g++ options: -O2 -pthread -lbz2 -lpthread

## Bullet Physics Engine 2.81

Test: 3000 Fall

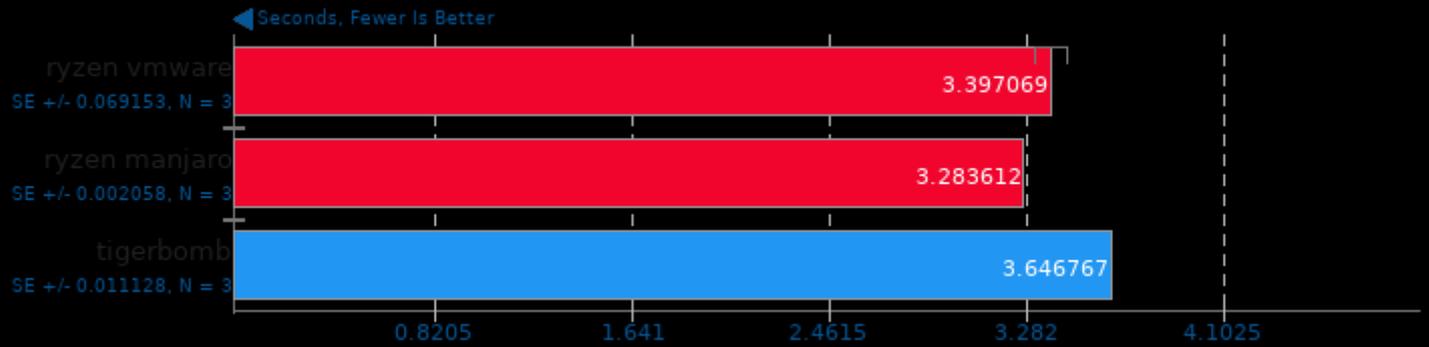
Seconds, Fewer Is Better



1. (CXX) g++ options: -O3 -rdynamic -lglut -lGL -lGLU

### Bullet Physics Engine 2.81

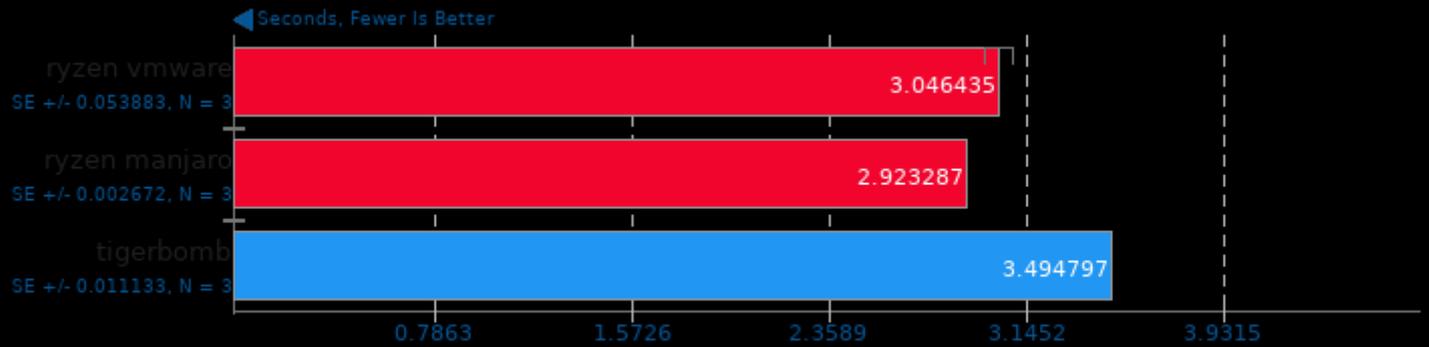
Test: 1000 Stack



1. (CXX) g++ options: -O3 -rdynamic -lglut -lGL -lGLU

### Bullet Physics Engine 2.81

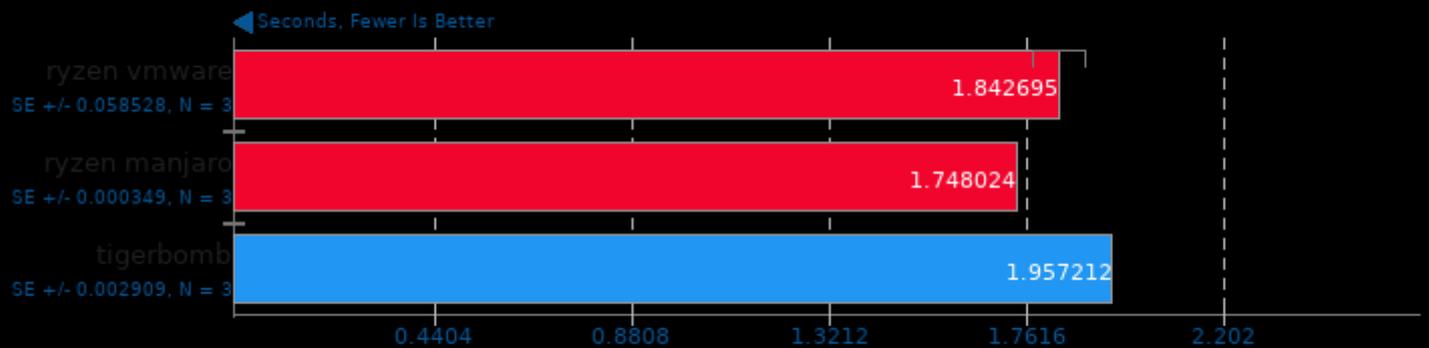
Test: 1000 Convex



1. (CXX) g++ options: -O3 -rdynamic -lglut -lGL -lGLU

### Bullet Physics Engine 2.81

Test: 136 Ragdolls

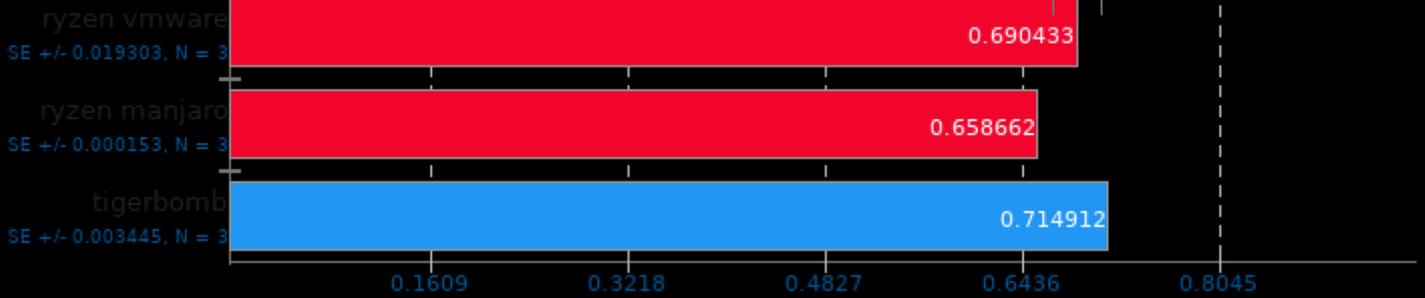


1. (CXX) g++ options: -O3 -rdynamic -lglut -lGL -lGLU

### Bullet Physics Engine 2.81

Test: Prim Trimesh

← Seconds, Fewer Is Better

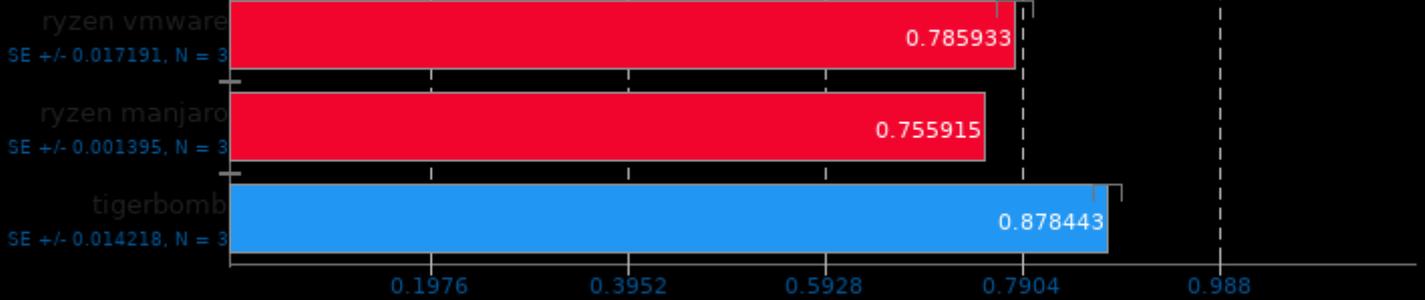


1. (CXX) g++ options: -O3 -rdynamic -lglut -lGL -lGLU

### Bullet Physics Engine 2.81

Test: Convex Trimesh

← Seconds, Fewer Is Better

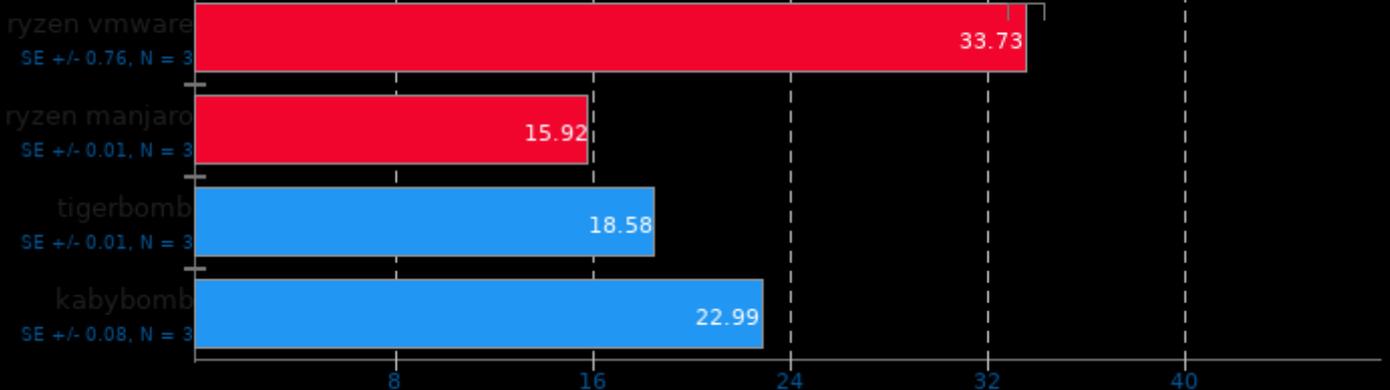


1. (CXX) g++ options: -O3 -rdynamic -lglut -lGL -lGLU

### Cython Benchmark 0.29.21

Test: N-Queens

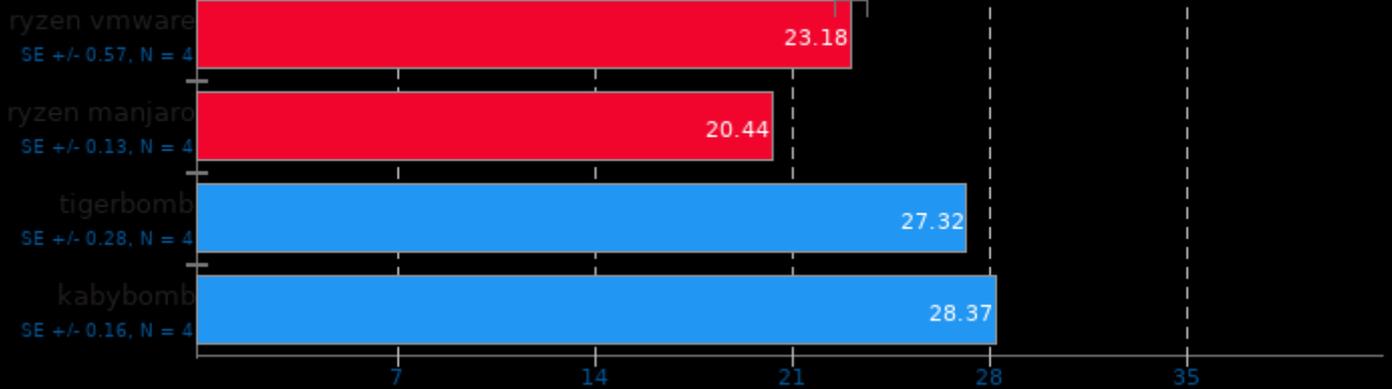
← Seconds, Fewer Is Better



## eSpeak-NG Speech Engine 20200907

Text-To-Speech Synthesis

← Seconds, Fewer Is Better

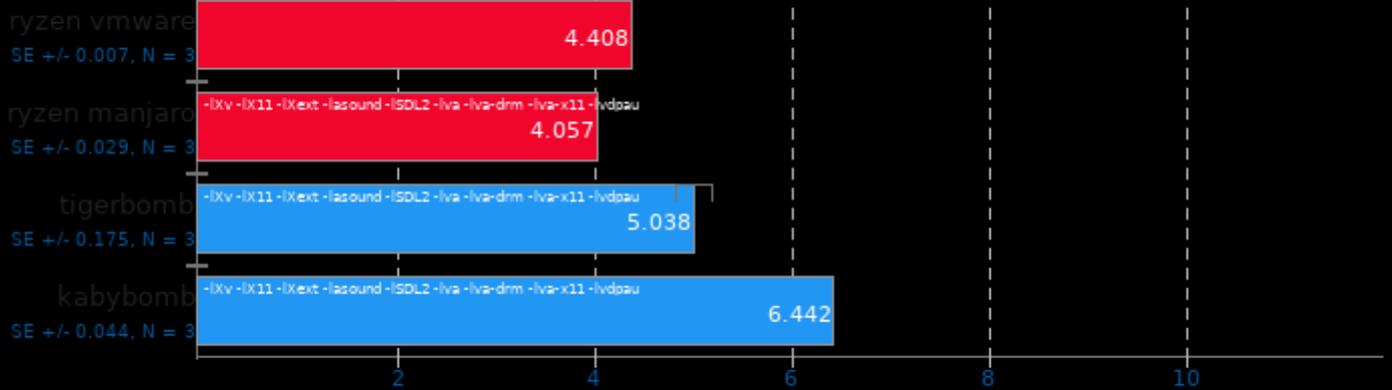


1. (CC) gcc options: -O2 -std=c99 -lpthread -lm

## FFmpeg 4.0.2

H.264 HD To NTSC DV

← Seconds, Fewer Is Better



1. (CC) gcc options: -lavdevice -lavfilter -lavformat -lavcodec -lswresample -lswscale -lavutil -lm -lxcb -lxcb-shm -lxcb-shape -lxcb-xfixes -pthread -lbz2 -lz

## Mencoder 1.3.0

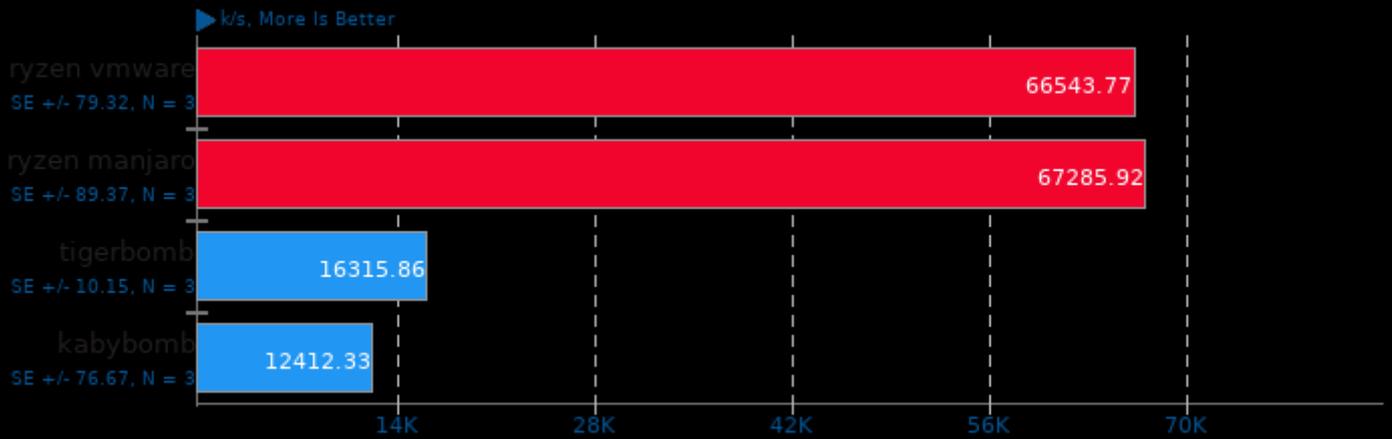
AVI To LAVC

← Seconds, Fewer Is Better



1. (CC) gcc options: -ffast-math -fpie -pie -lncurses -lrt -lgnutls -lpng -lz -ljpeg -lbz2 -lzo2 -lpthread -ldl -rdynamic -lm

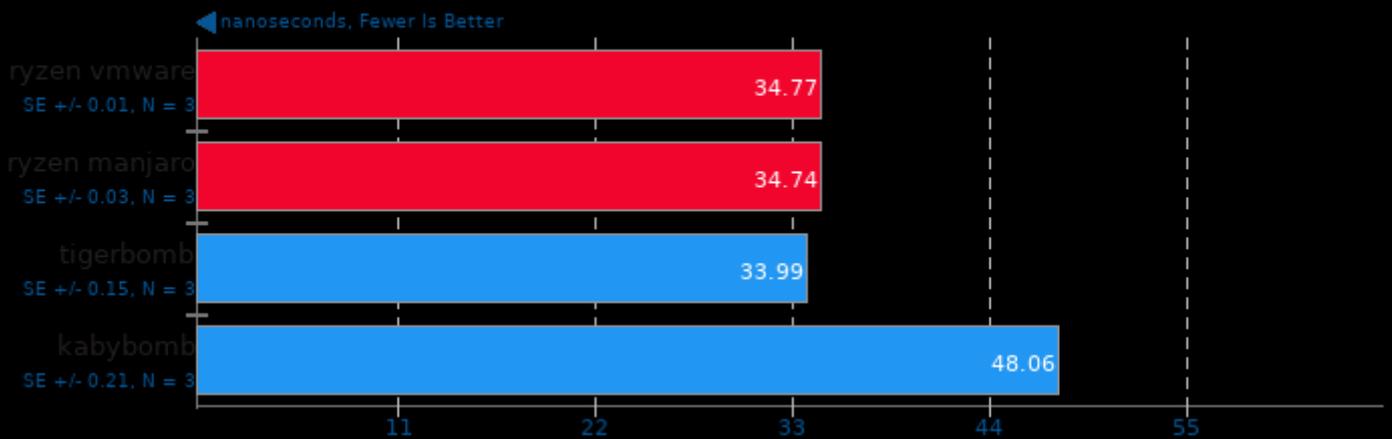
## Aircrack-ng 1.5.2



1. (CXX) g++ options: -O3 -fvisibility=hidden -masm=intel -fcommon -rdynamic -lsqlite3 -lpthread -lz -lcrypto -lhwloc -ldl -lm -pthread

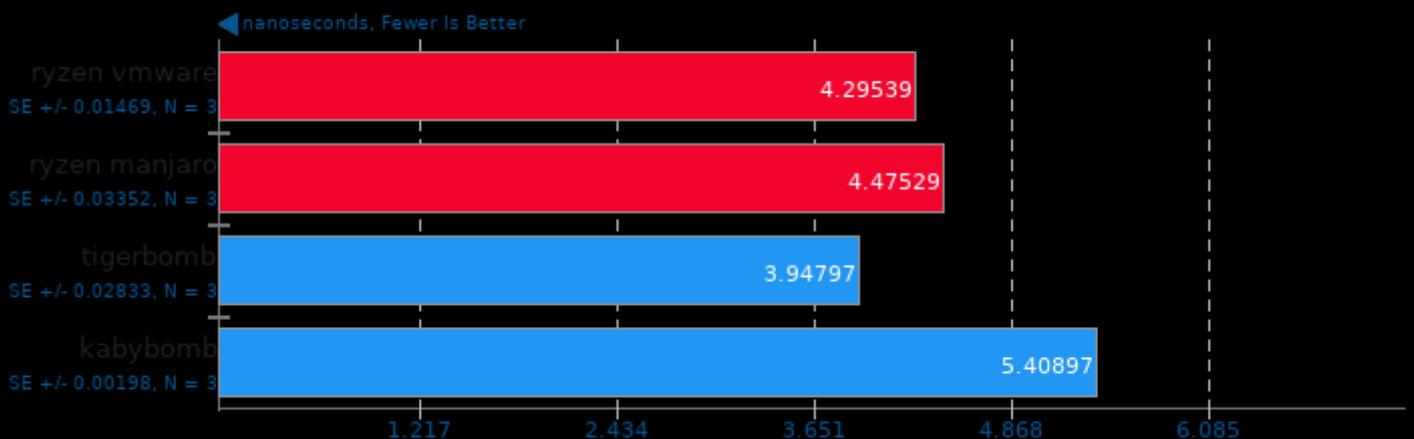
## glibc bench 1.0

Benchmark: cos



## glibc bench 1.0

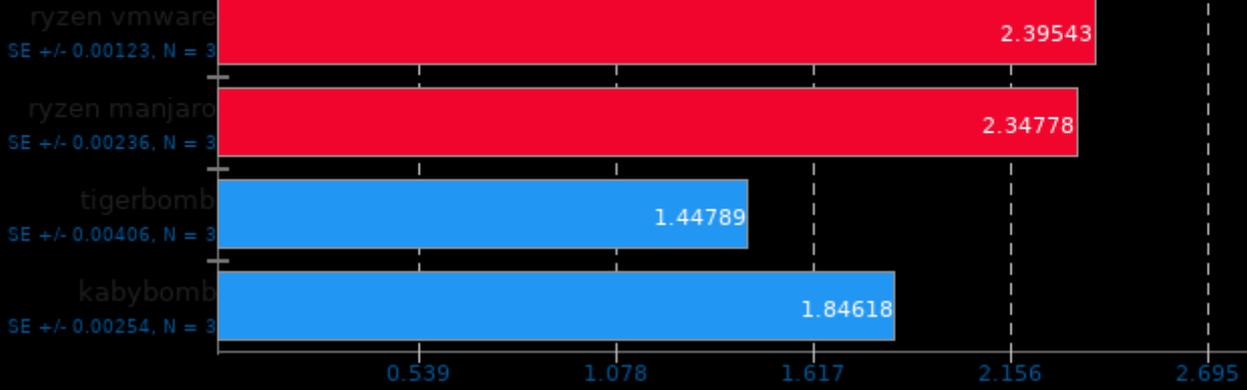
Benchmark: exp



### glibc bench 1.0

Benchmark: ffs

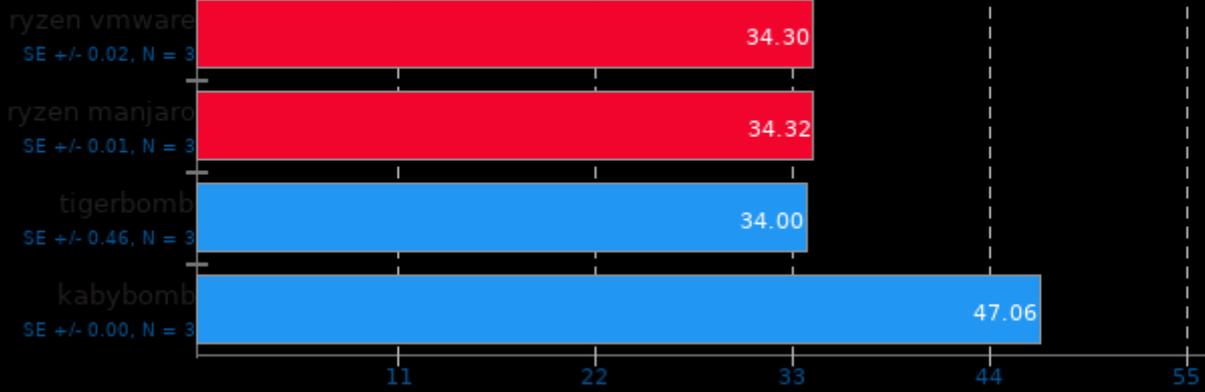
nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: sin

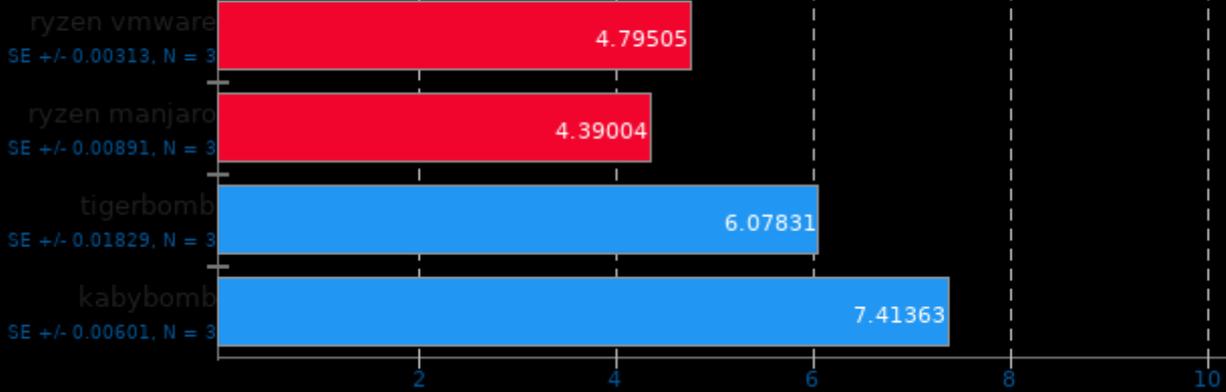
nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: log2

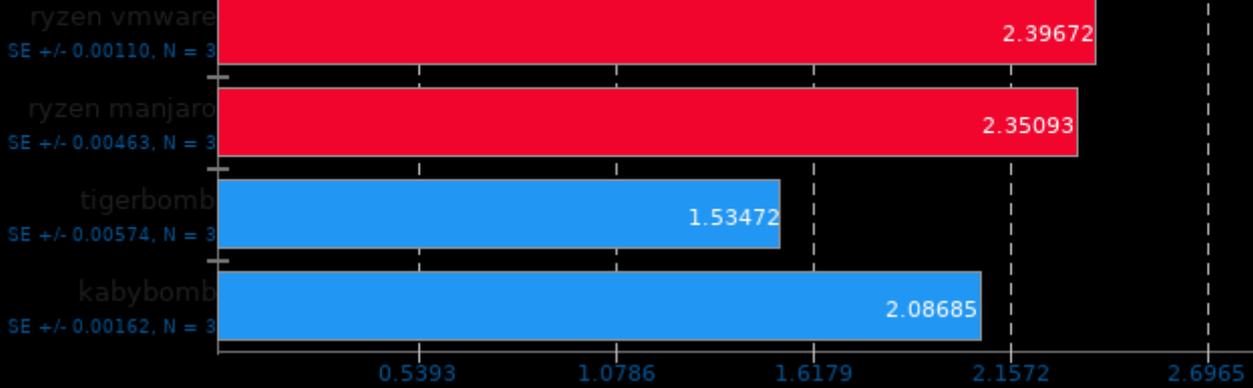
nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: modf

nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: sinh

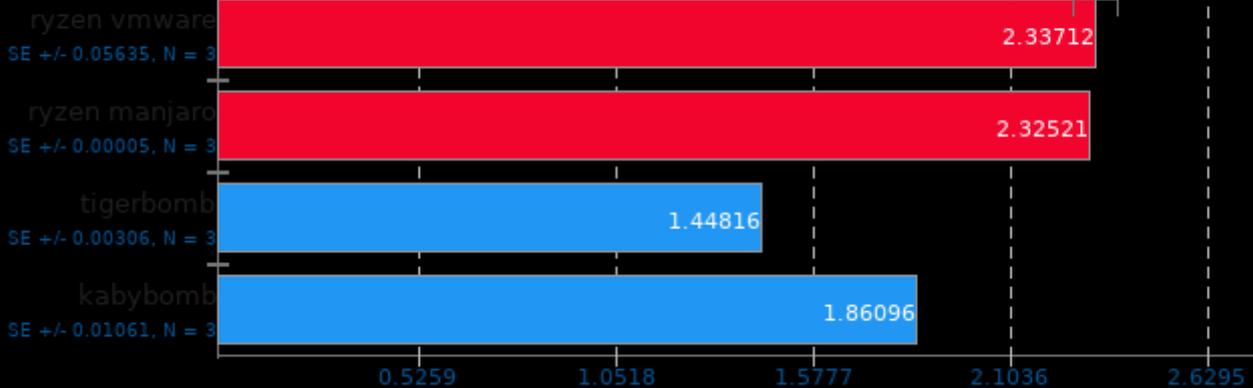
nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: sqrt

nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: tanh

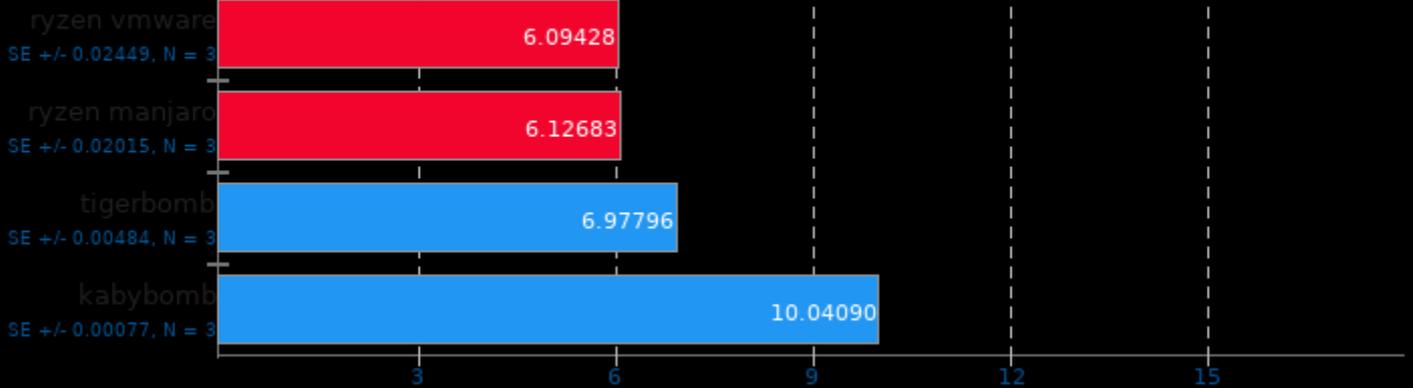
nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: asinh

nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: atanh

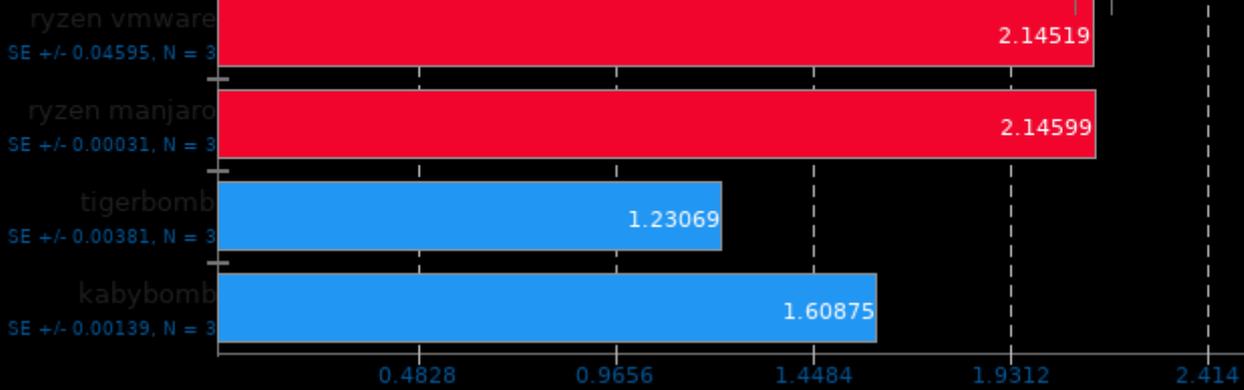
nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: ffsll

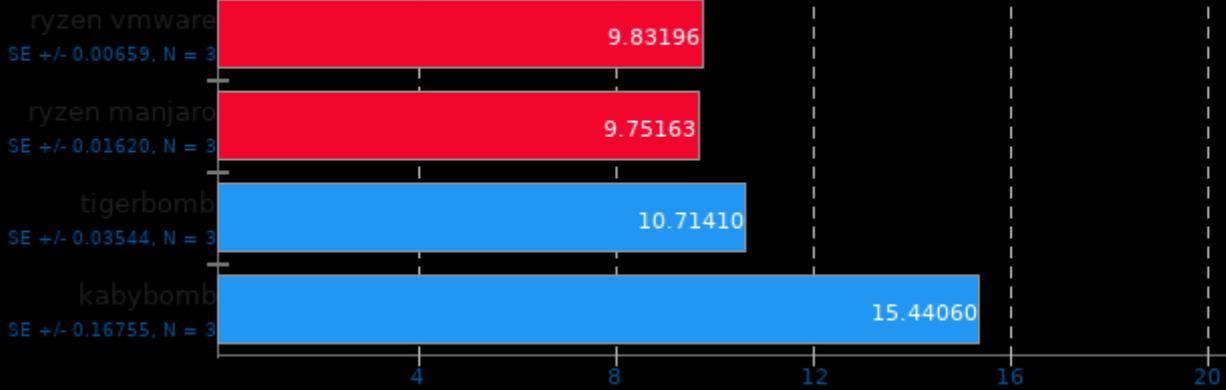
nanoseconds, Fewer Is Better



### glibc bench 1.0

Benchmark: sincos

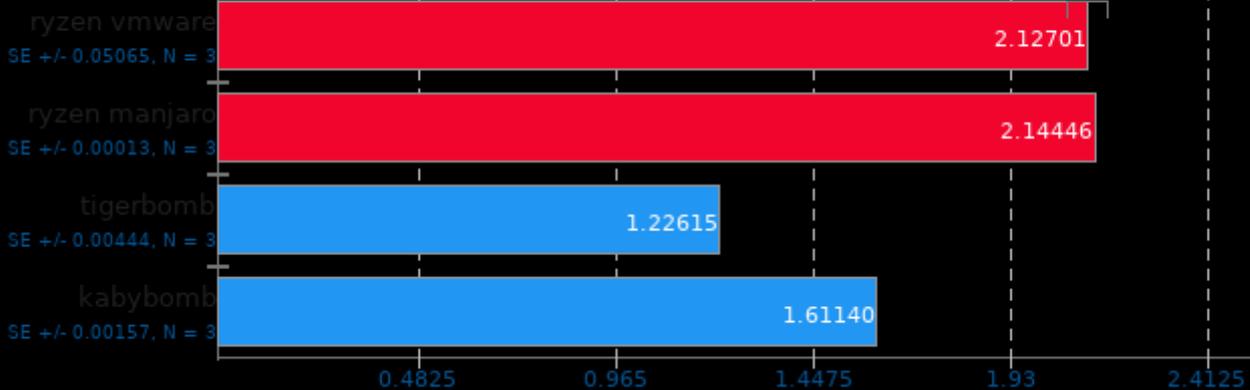
nanoseconds, Fewer Is Better

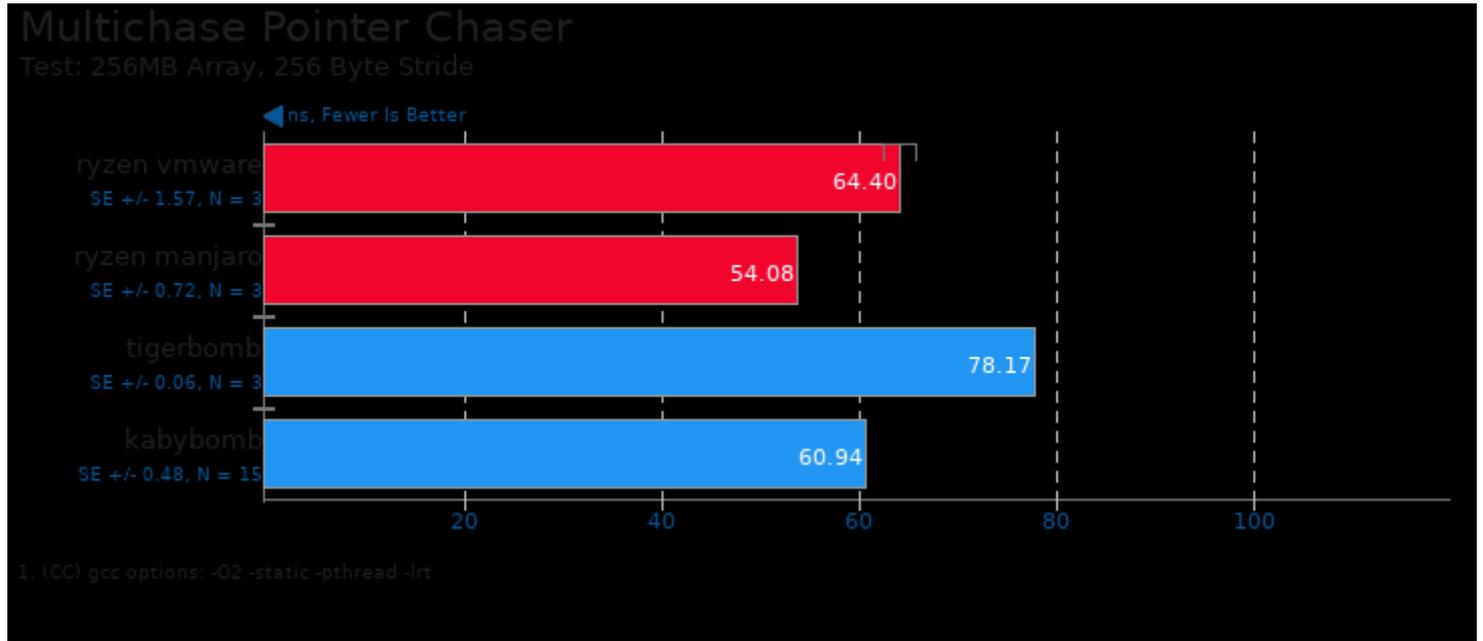


### glibc bench 1.0

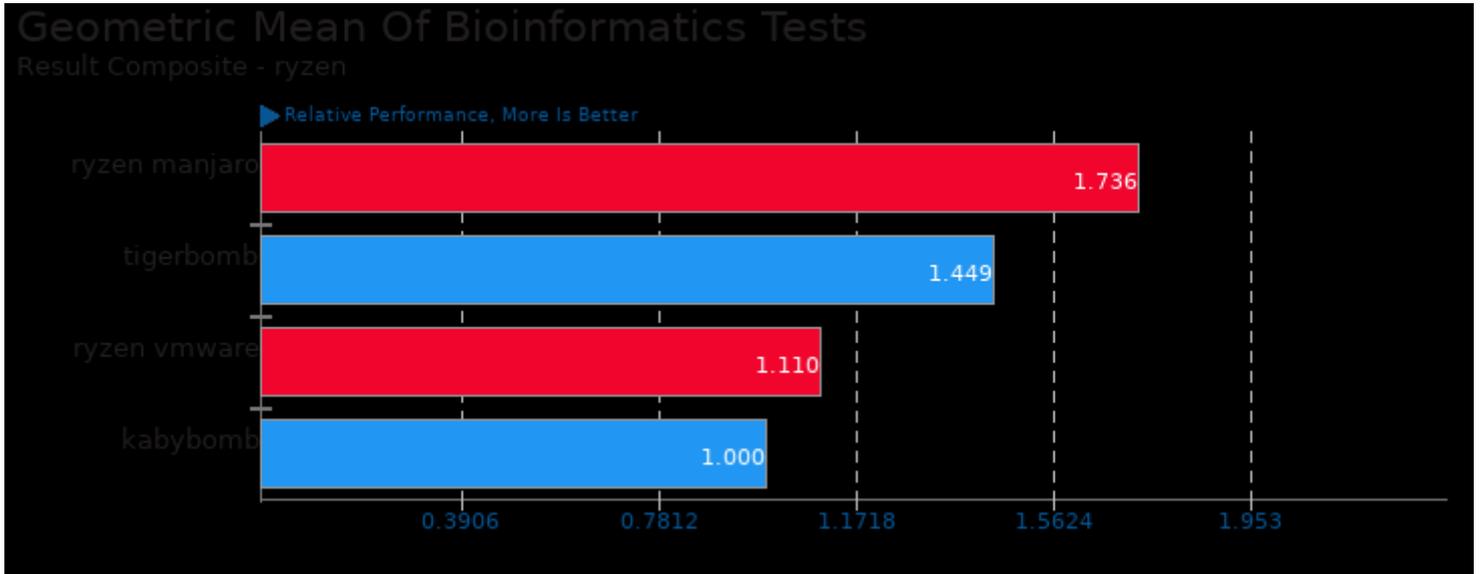
Benchmark: pthread\_once

nanoseconds, Fewer Is Better

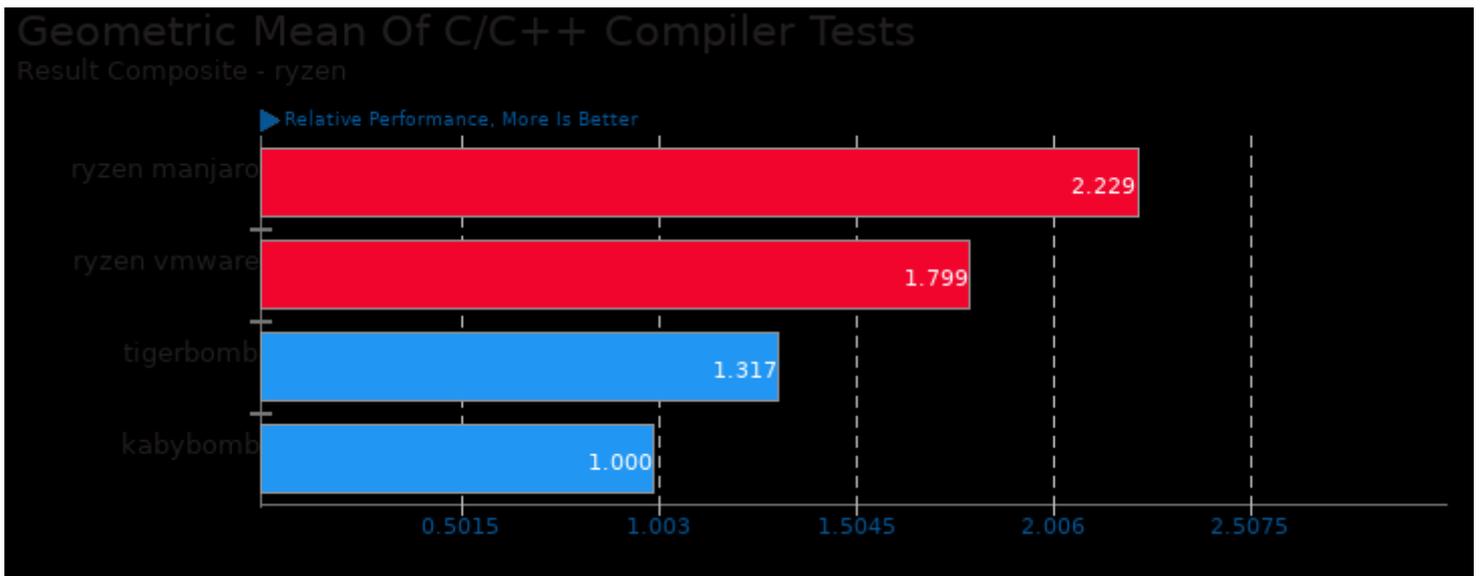




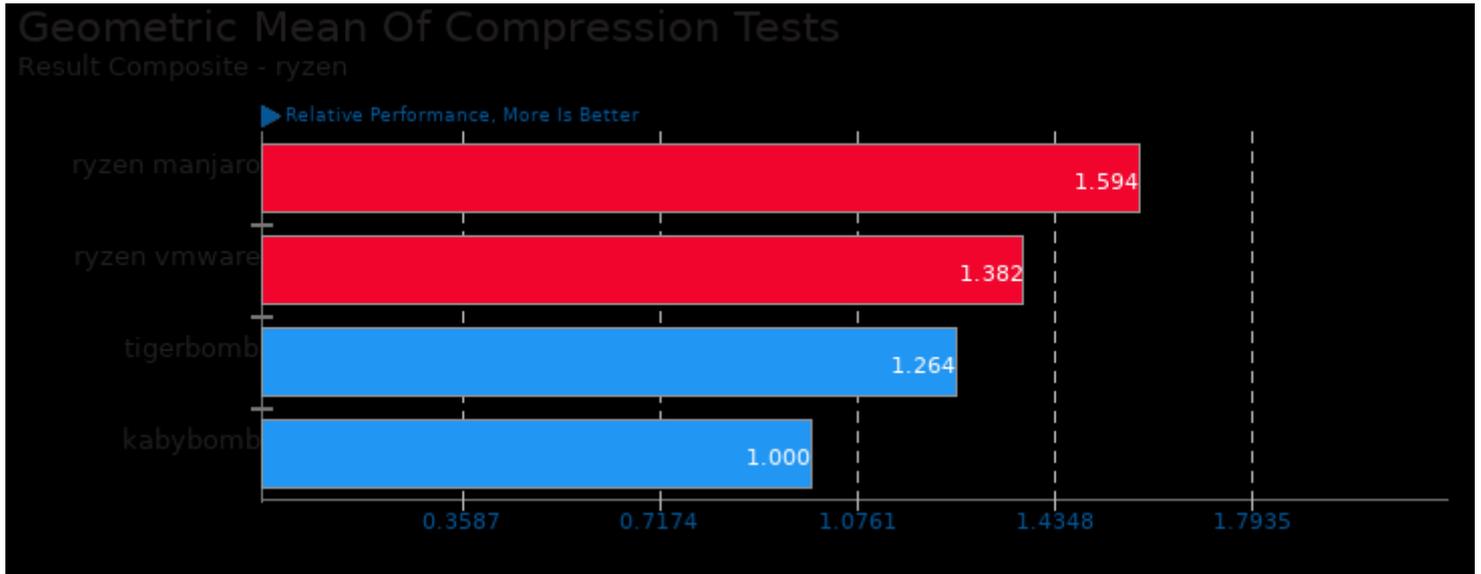
These geometric means are based upon test groupings / test suites for this result file.



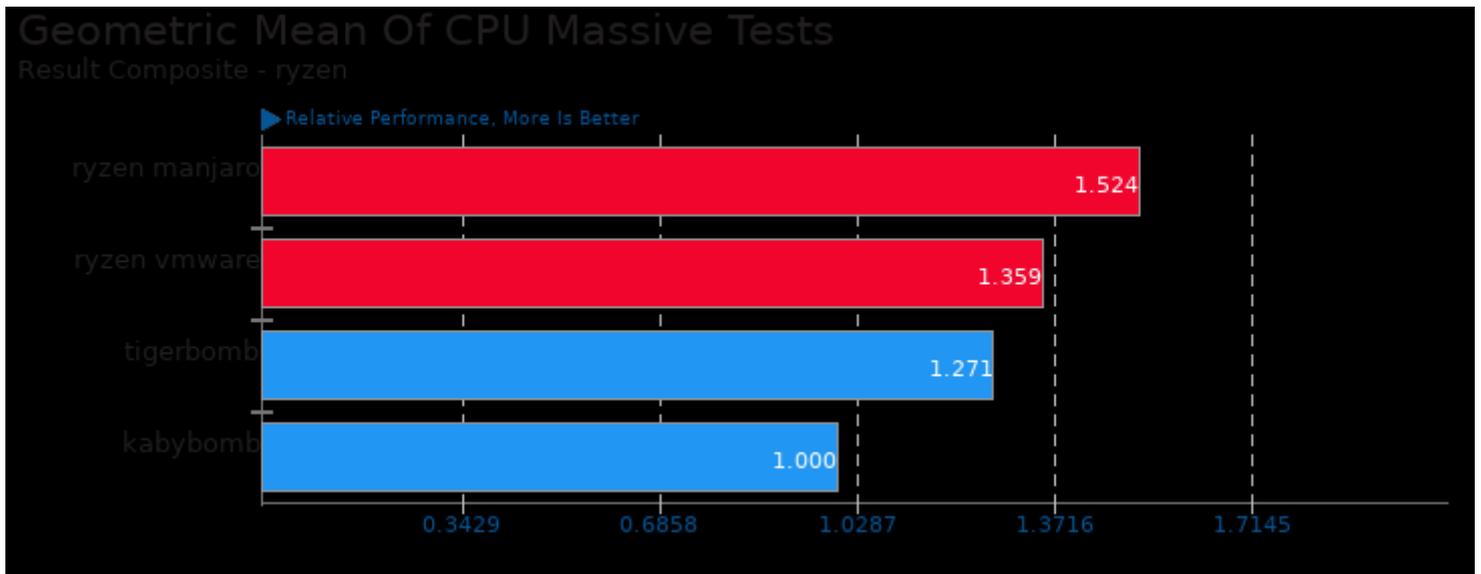
Geometric mean based upon tests: pts/himeno, pts/hmmer and pts/mafft



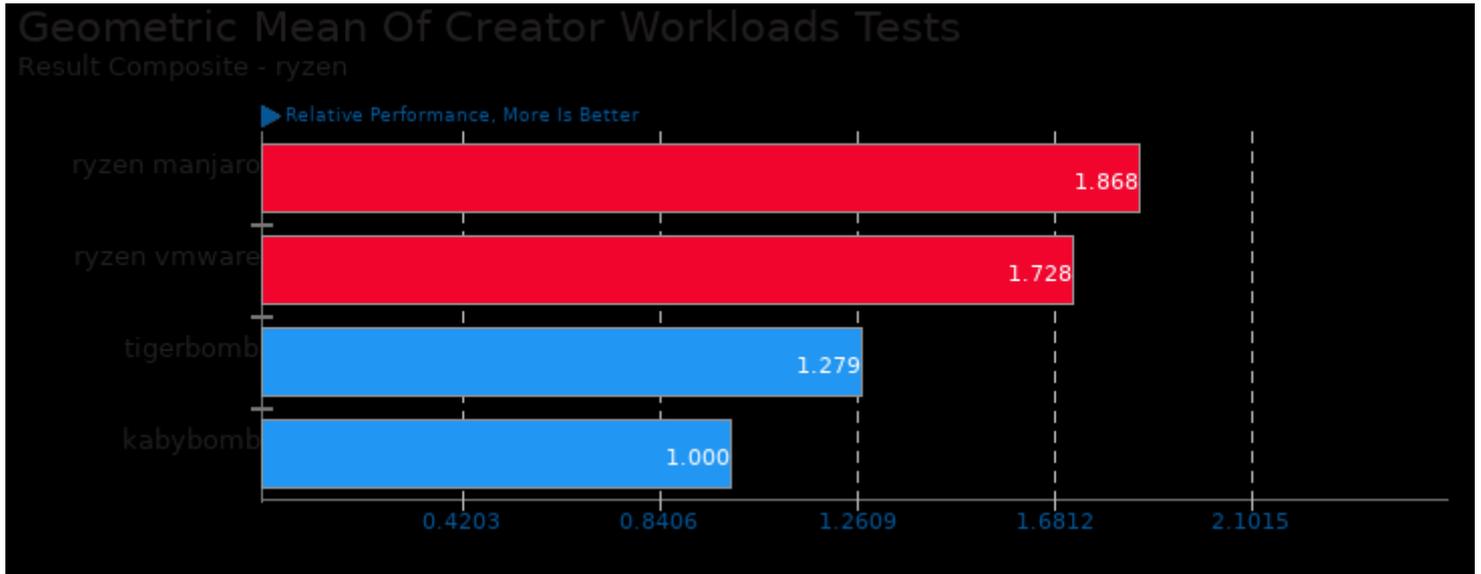
Geometric mean based upon tests: pts/mafft, pts/scimark2, pts/tscp, pts/himeno, pts/hmmer, pts/bullet, pts/compress-7zip, pts/dav1d, pts/x264, pts/x265 and pts/aircrack-ng



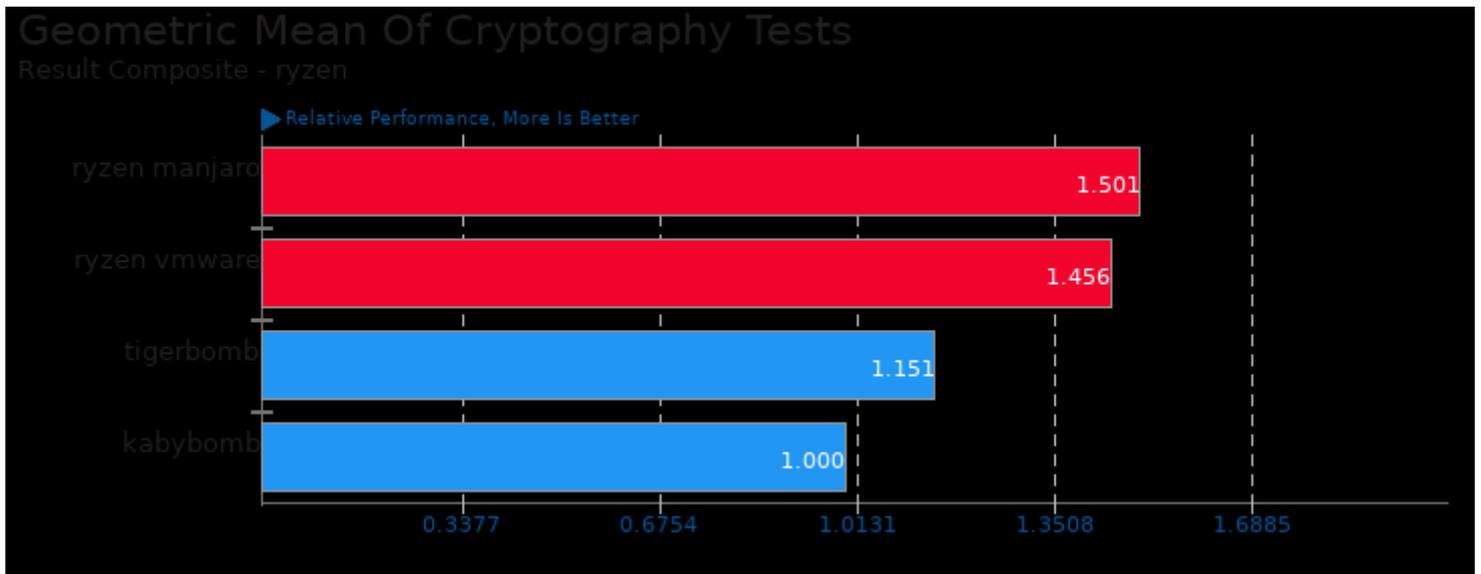
Geometric mean based upon tests: pts/compress-7zip, pts/compress-pbzip2 and pts/lzbench



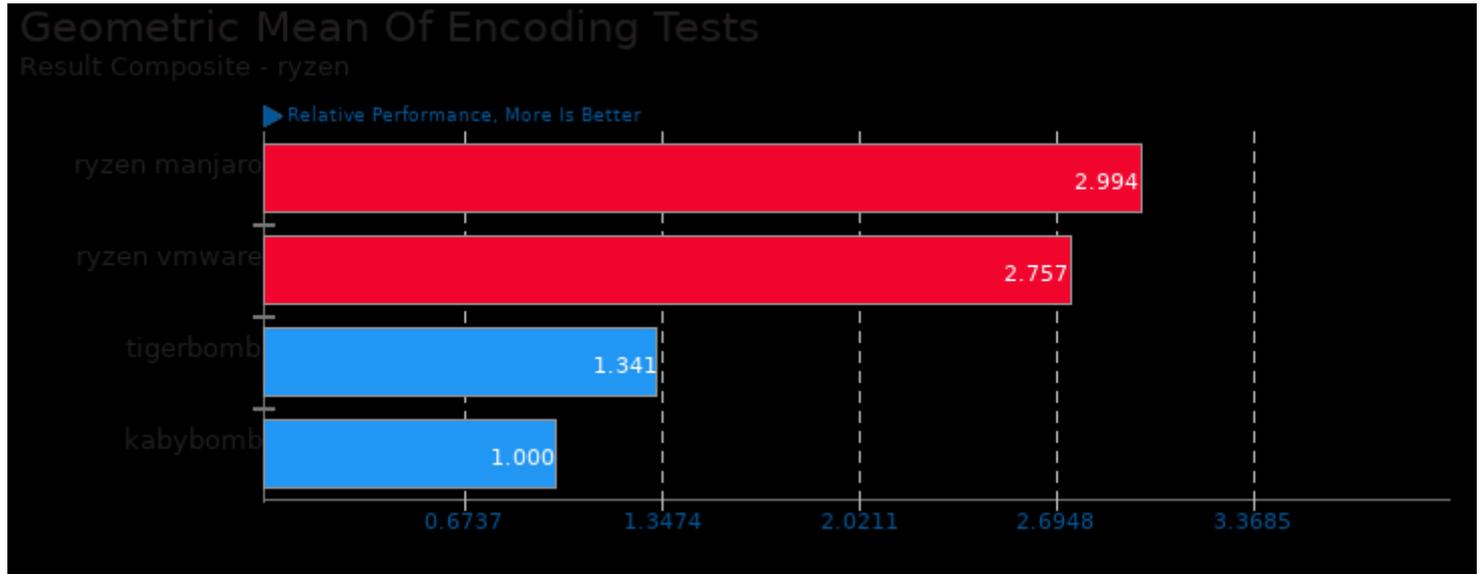
Geometric mean based upon tests: pts/cloverleaf, pts/compress-7zip, pts/compress-pbzip2, pts/cython-bench, pts/dav1d, pts/x264, pts/x265, pts/glibc-bench, pts/himeno, pts/hmmer, pts/java-scimark2, pts/lzbench, pts/mafft, pts/multichase, pts/swet and pts/botan



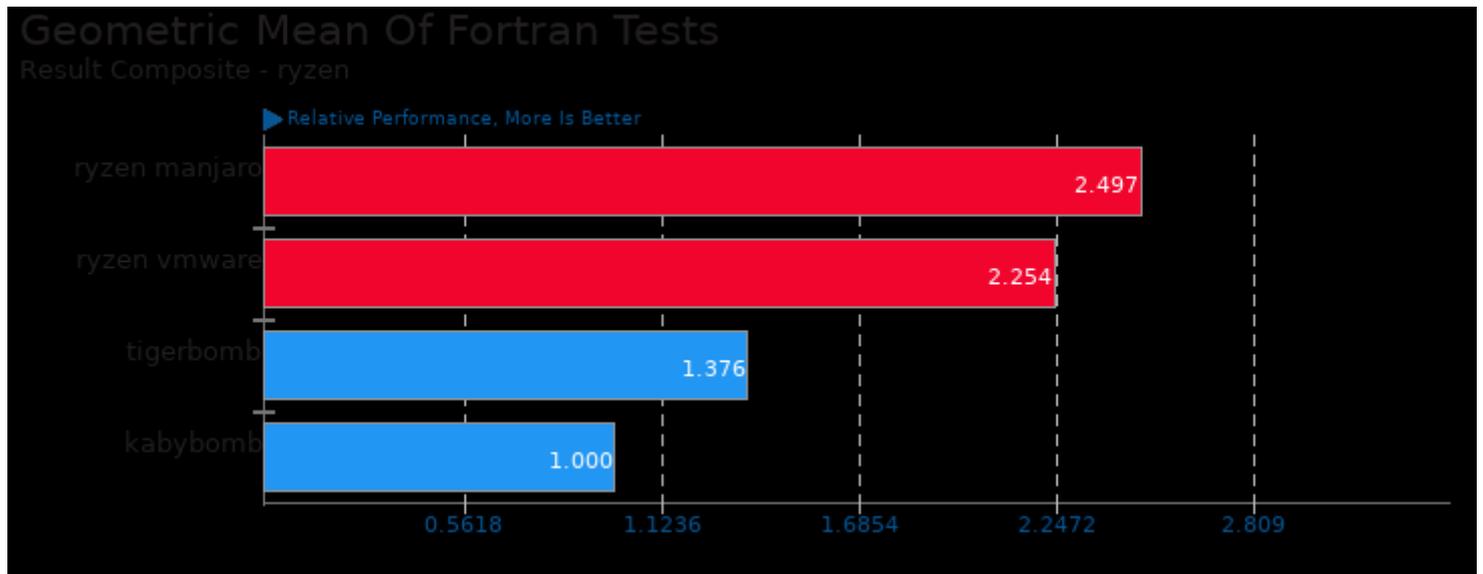
Geometric mean based upon tests: pts/x264, pts/x265, pts/ffmpeg, pts/dav1d, pts/luajit and pts/espeak



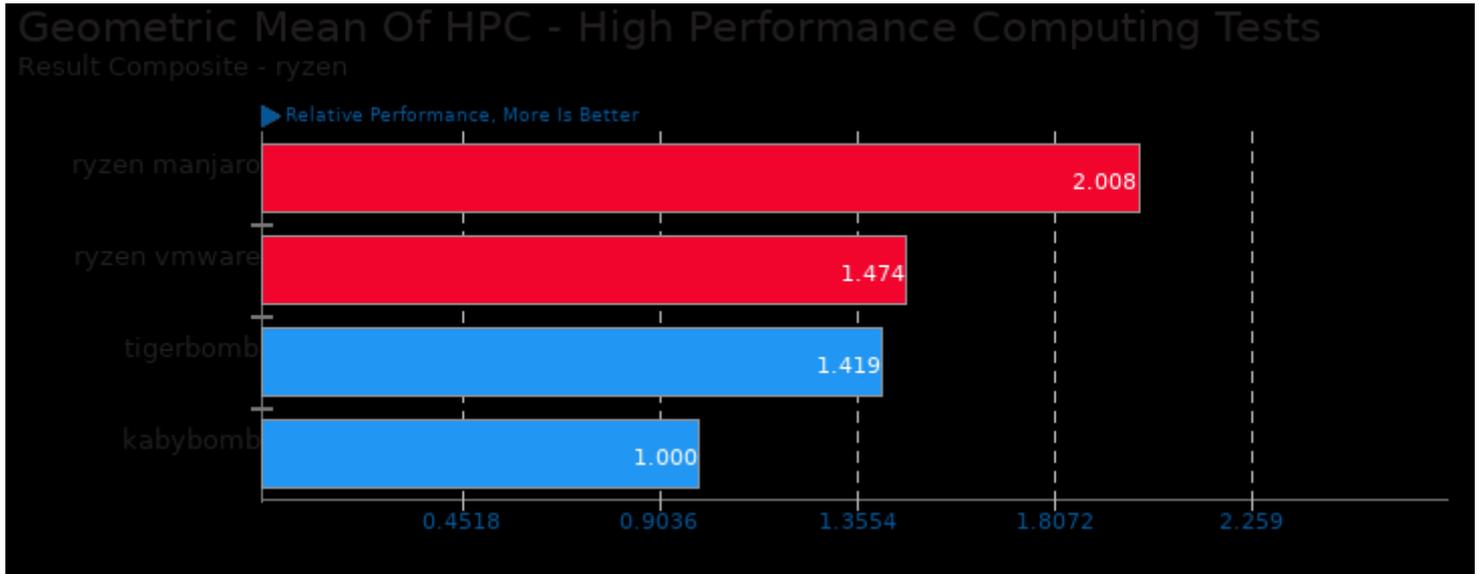
Geometric mean based upon tests: pts/botan, pts/bork and pts/aircrack-ng



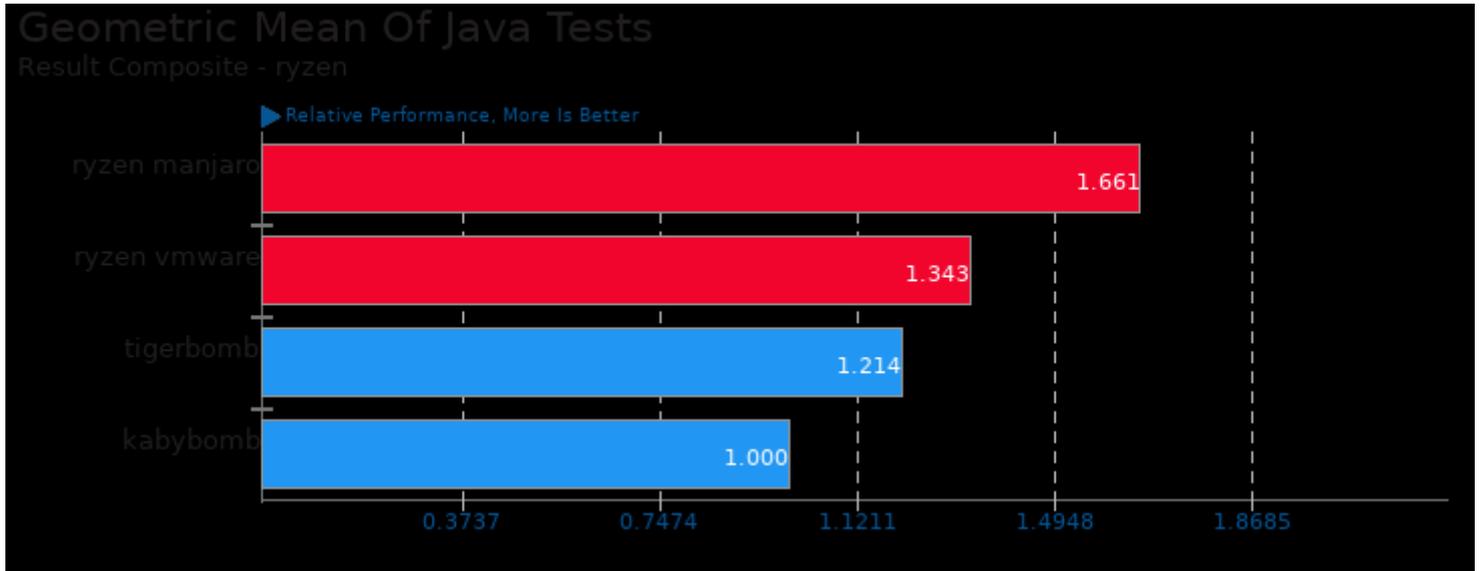
Geometric mean based upon tests: pts/x264, pts/x265, pts/ffmpeg and pts/dav1d



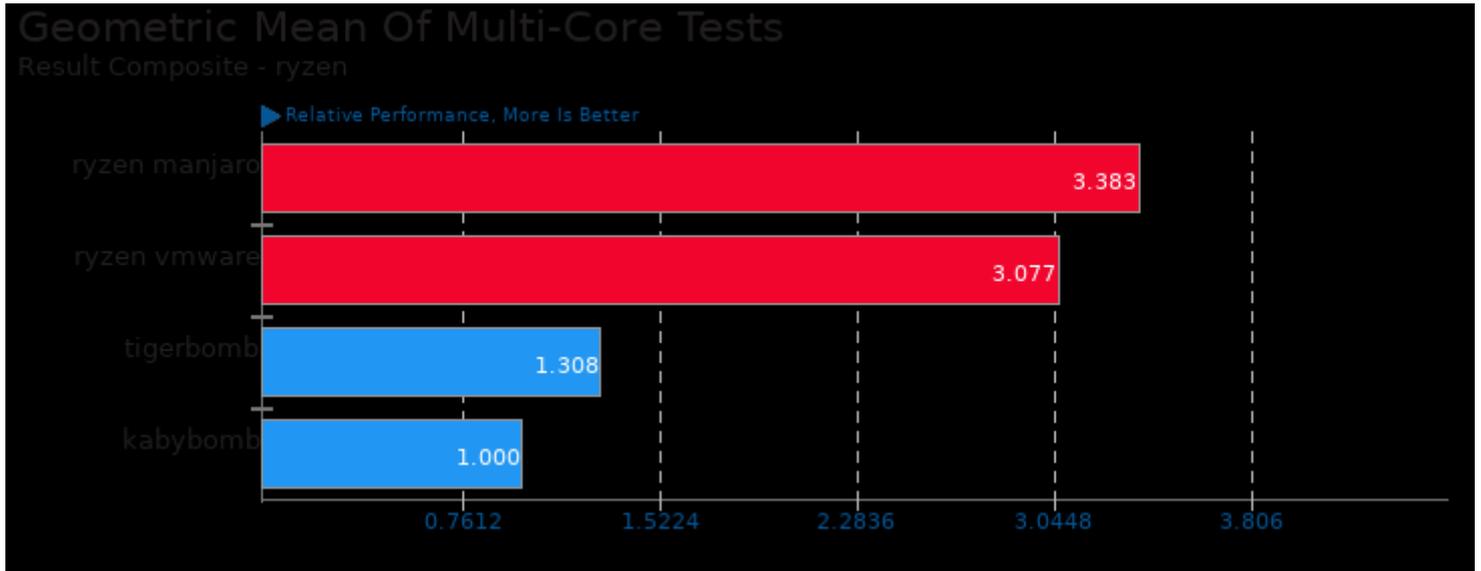
Geometric mean based upon tests: pts/cloverleaf and pts/ffte



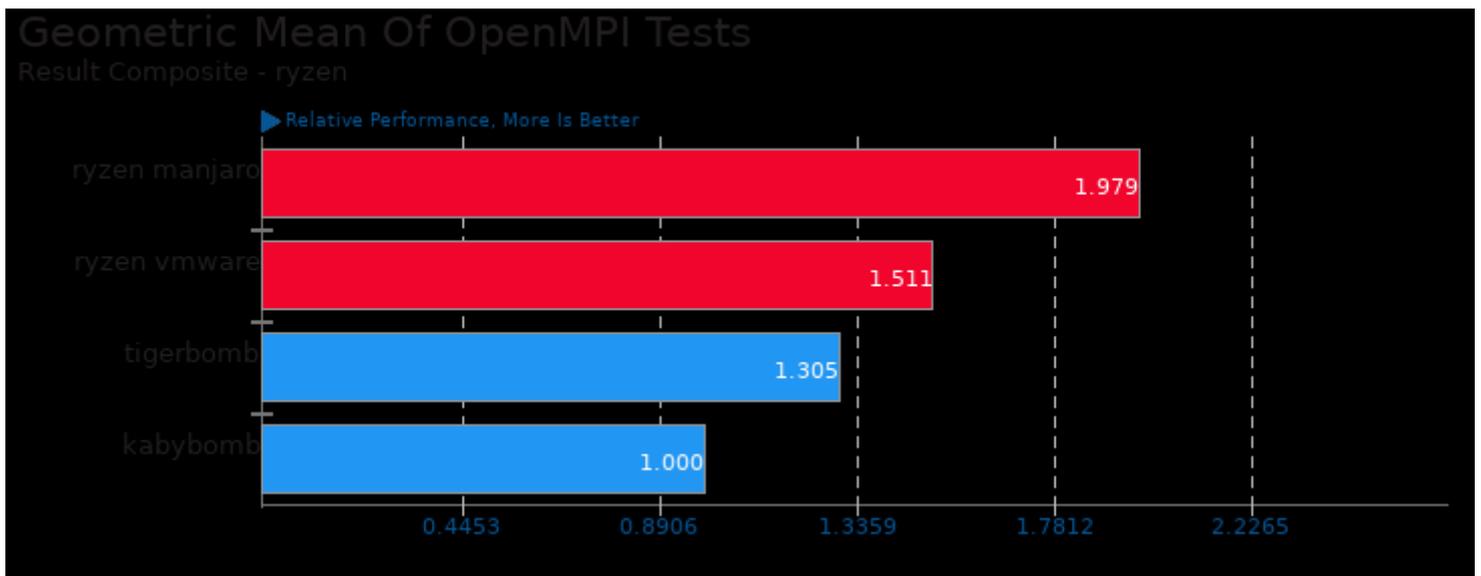
Geometric mean based upon tests: pts/ffte, pts/cloverleaf, pts/himeno, pts/hmmer and pts/mafft



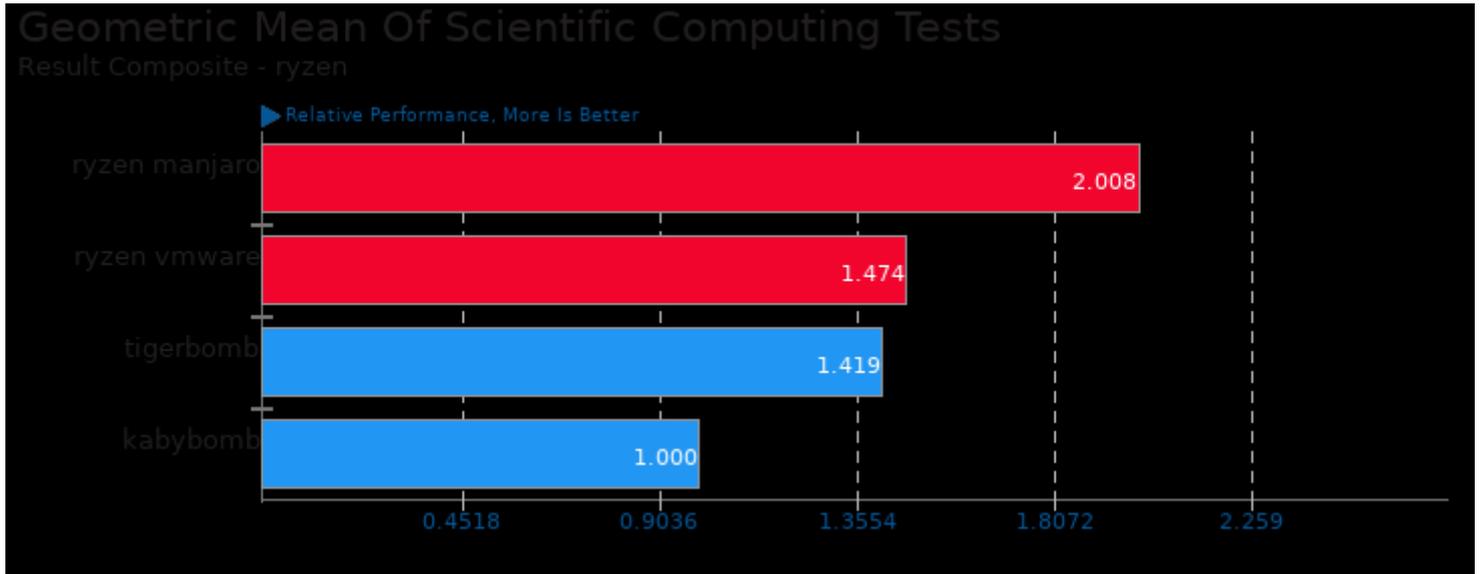
Geometric mean based upon tests: pts/bork and pts/java-scimark2



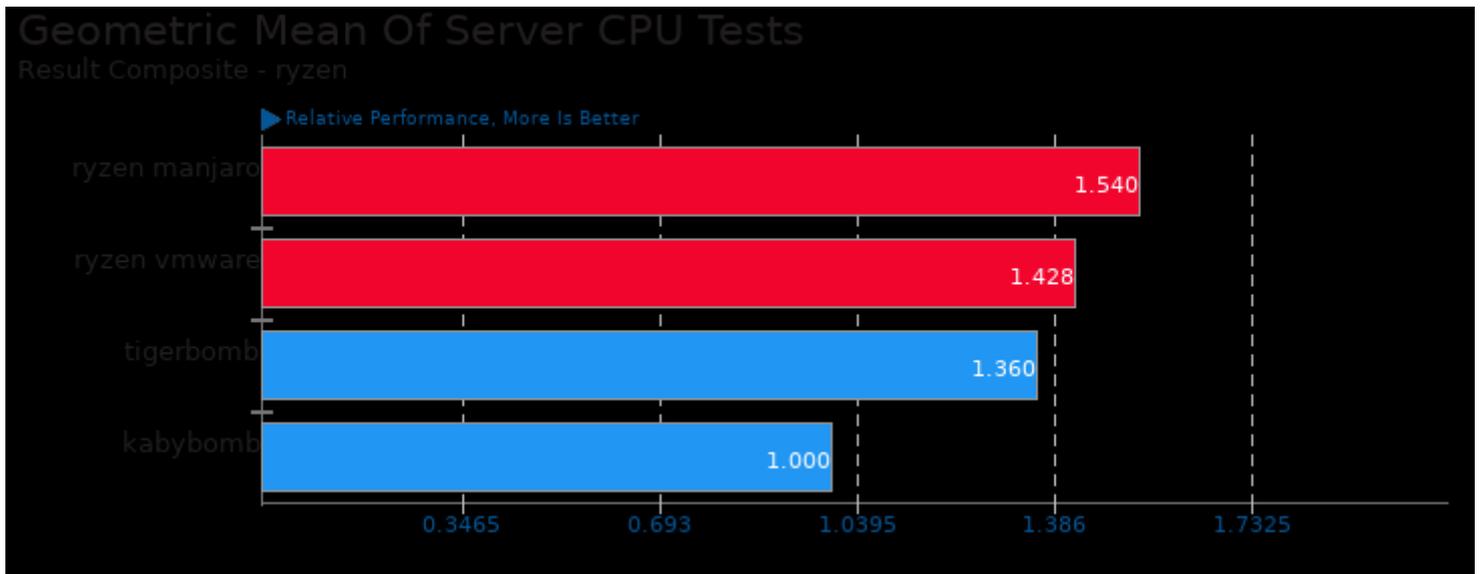
Geometric mean based upon tests: pts/coremark, pts/aircrack-ng, pts/x264, pts/x265, pts/ffmpeg, pts/dav1d, pts/swet, pts/compress-7zip and pts/compress-pbzip2



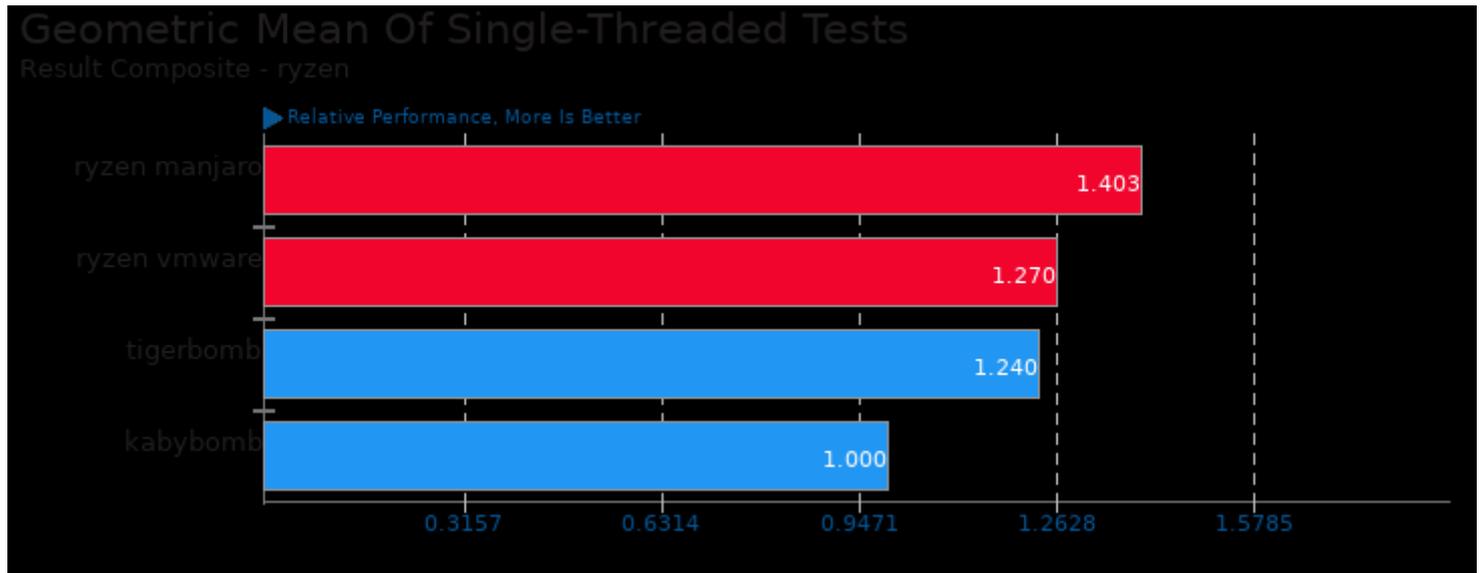
Geometric mean based upon tests: pts/cloverleaf and pts/hmmer



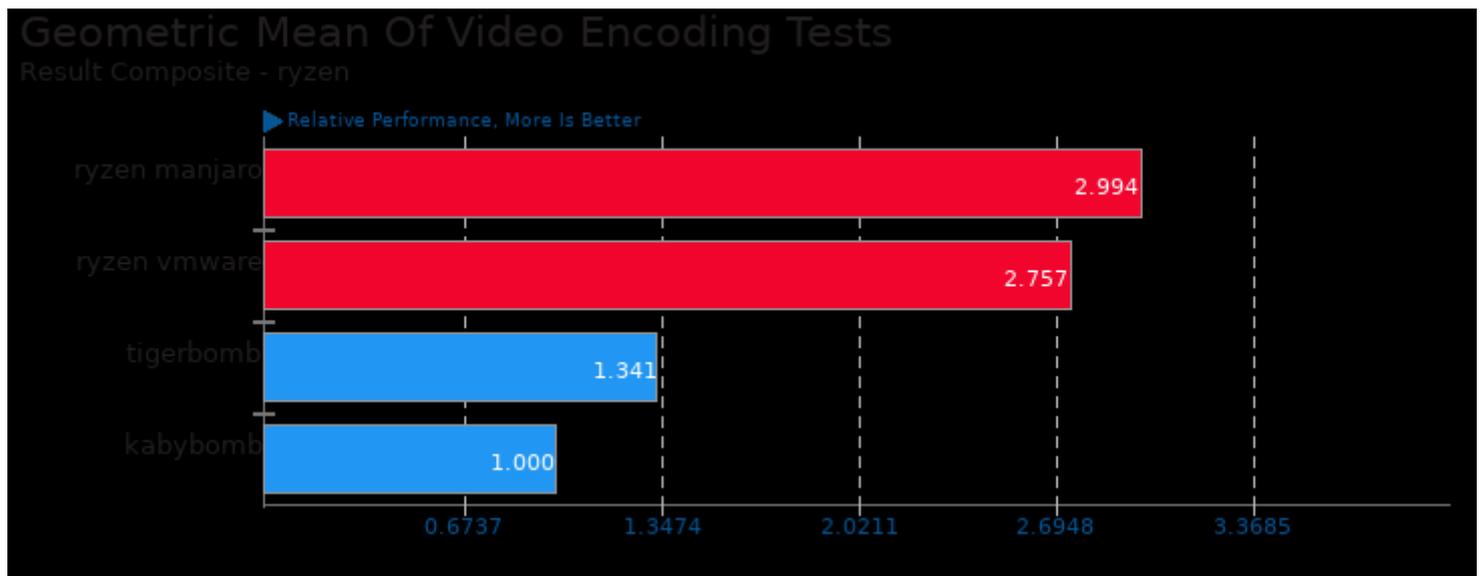
Geometric mean based upon tests: pts/ffte, pts/cloverleaf, pts/himeno, pts/hmmer and pts/mafft



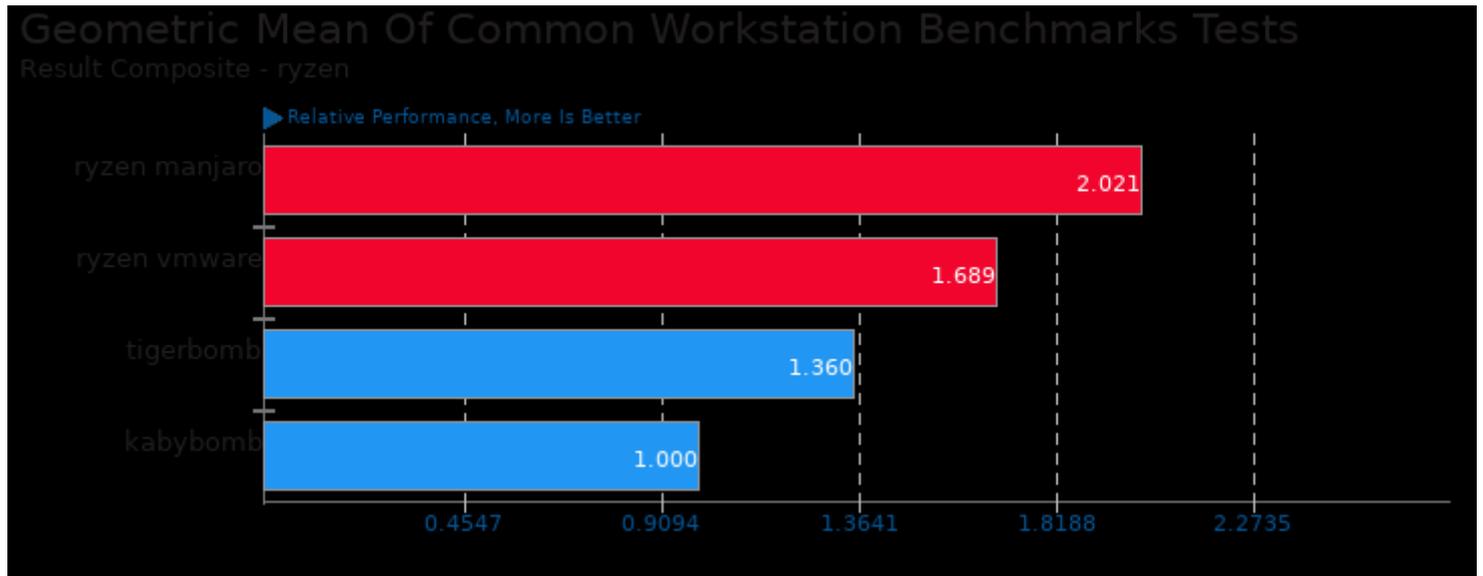
Geometric mean based upon tests: pts/x264, pts/x265, pts/dav1d, pts/himeno, pts/compress-7zip, pts/glibc-bench and pts/cython-bench



Geometric mean based upon tests: pts/polybench-c, pts/lzbench, pts/java-scimark2, pts/bork, pts/luajit, pts/scimark2, pts/botan, pts/swet, pts/espeak, pts/glibc-bench and pts/multichase



Geometric mean based upon tests: pts/x264, pts/x265, pts/ffmpeg and pts/dav1d



Geometric mean based upon tests: pts/himeno, pts/x265 and pts/swet

*This file was automatically generated via the Phoronix Test Suite benchmarking software on Monday, 18 March 2024 23:42.*