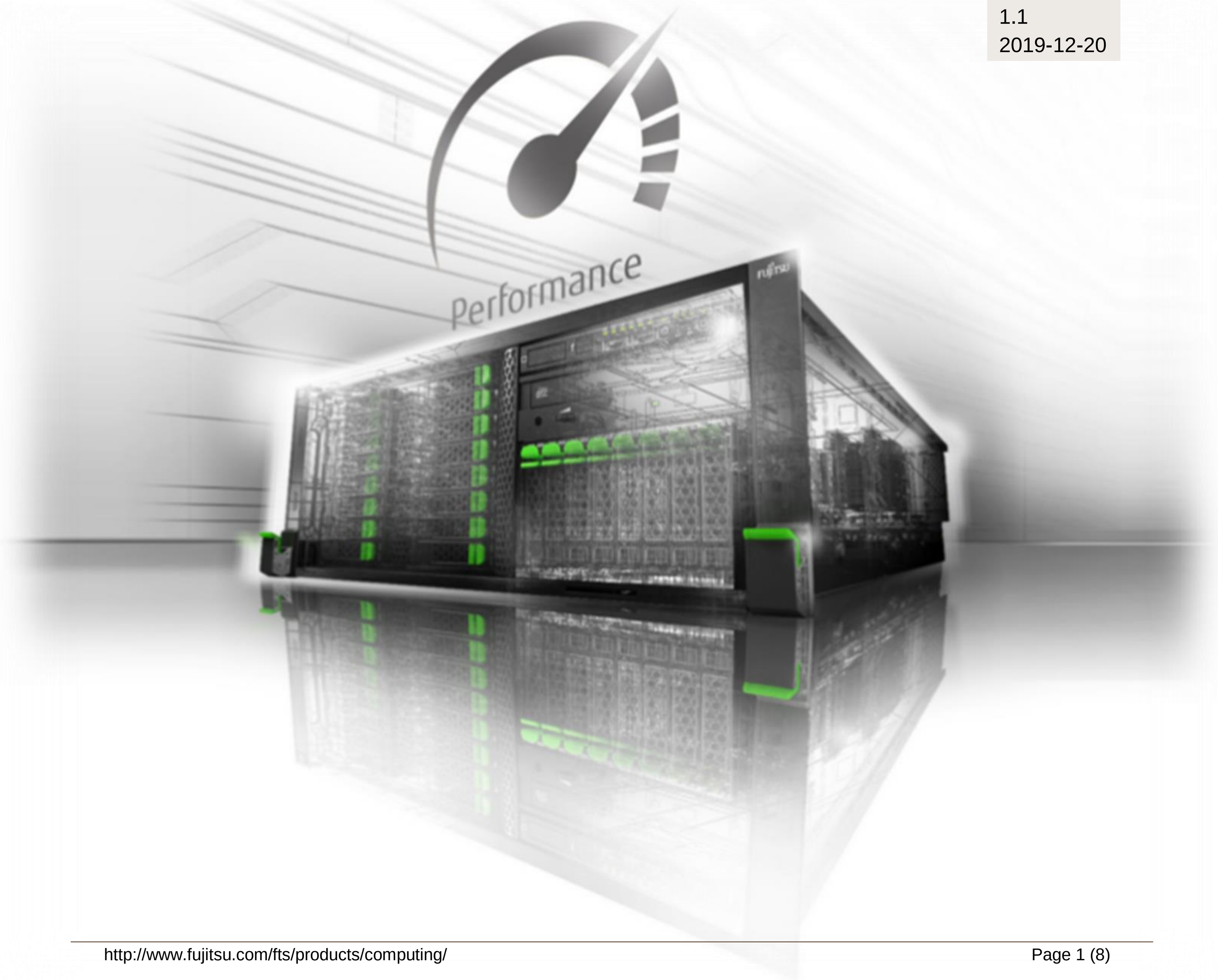
White Paper
FUJITSU x86 Servers & Workstations
Benchmark overview SPECcpu2017

This document presents the benchmark SPECcpu2017 which was developed and published by the Standard Performance Evaluation Corporation (SPEC).

This benchmark, in which processor and memory performance are measured, is a benchmark that is generally accepted by the industry. As a result of the benchmark concept realized in SPECcpu2017, it is possible to make across-the-board manufacturer comparisons.

Version
1.1
2019-12-20



Contents

Document history

Version 1.0 (2018-11-15)

Initial version

Version 1.1 (2019-12-20)

Added Power Efficiency item

SPECcpu2017 – An overview

The “Standard Performance Evaluation Corporation,” or briefly SPEC, is an organization that specializes in the development and issue of standardized benchmarks to evaluate the performance of computer systems. The members of SPEC are leading companies of the computer industry, such as Fujitsu. One of the benchmarks issued by SPEC, and the successor to CPU2006, is CPU2017. The purpose of this document is to provide an overview of the importance of this benchmark and the significance of its results.

Objective of the benchmark

When comparing systems, in principle, the individual customer application would be the measure of all things. However, due to shortage of time or for complexity or other reasons, you often do not have the opportunity to test a variety of possible systems with a customer application in order to find out which is the most suitable one. Benchmarks have been developed to nevertheless provide customers with objective criteria when choosing their system. These are standardized tools that measure the speed of individual system components or the speed of the entire system.

CPU2017 is a benchmark developed to enable a performance comparison between systems in the case of compute-bound activities. Strictly speaking, CPU2017 is not a single benchmark, but consists of two benchmark groups:

- SPECrate 2017 Integer and SPECspeed 2017 Integer, for measurement of the system performance in the case of integer operations
- SPECrate 2017 Floating Point and SPECspeed 2017 Floating Point, for measurement of the system performance in the case of floating-point operations

As a compute-bound benchmark, besides concentrating on the processor, it also concentrates on the memory architecture and the compiler.

Important: The latter components should not be ignored. Performance is not only a matter of the processor’s clock pulse frequency.

Important: Other system components, i.e. I/O, graphics or network, are not subject to a load test. If the performance bottleneck of an application is due to one of these components, the CPU2017 benchmark does not enable this to be recognized. Therefore, it depends on the load profile of an application whether or not CPU2017 is the right benchmark in the individual case.

Components of the benchmark

The benchmark group SPECrate 2017 Integer and SPECspeed 2017 Integer consists of 10 applications written in C, C++, and FORTRAN.

Benchmark (Integer)		**Application Area**
SPECrate 2017	**SPECcpeed 2017**	
500.perlbench_r	600.perlbench_s	Perl interpreter Derived from Perl V5.22.1. The workload includes SpamAssassin, MHonArc (an email indexer), and specdiff (a part of the CPU2017 tool suite).
502.gcc_r	602_gcc_s	GNU C compiler Based on gcc Version 4.5.0, generates code for the IA32 processor.
505.mcf_r	605_mcf_s	Route planning Derived from MCF, a single-depot vehicle scheduling in public mass transportation. The benchmark almost exclusively uses integer arithmetic.
520.omnetpp_r	620.omnetp_s	Discrete Event simulation Uses the OMNet++ discrete event simulator of a large 10 gigabit Ethernet network.
523.xalancbmk_r	623.xalancbmk_s	XML to HTML conversion via XSLT A modified version of Xalan-C++, which transforms XML documents into HTML, text, or other XML document types.
525.x264_r	625.x264_s	Video compression Encodes a video stream into H.264/MPEG-4 AVC format.
531.deepsjeng_r	631.deepsjeng_s	Artificial Intelligence: alpha-beta tree search (Chess) Based on Deep Sjeng WC2008, the 2008 World Computer Speed-Chess Champion. Achieves the highest possible playing strength.
541.leela_r	641.leela_s	Artificial Intelligence: Monte Carlo tree search (Go) Go playing engine featuring Monte Carlo-based position estimation.
548.exchange2_r	648.exchange2_s	Artificial Intelligence: recursive solution generator (Sudoku) Develops non-trivial 9x9 sudoku puzzles and tests many Fortran 95 array handling features for use with integer arrays.
557.xz_r	658.xz_s	General data compression Based on XZ Utils 5.0.5, does all compression and decompression entirely in memory. Performs no file I/O other than reading the input.

The benchmark group SPECrate 2017 Floating Point and SPECspeed 2017 Floating Point consists of 14 applications written in C, C++ and FORTRAN:

Benchmark (Floating Point)		Application Area
SPECrate 2017	SPECspeed 2017	
503.bwaves_r	603.bwaves_S	Explosion modeling Simulates blast waves in three-dimensional transonic transient laminar viscous flow.
507.cactuBSSN_r	607.cactuBSSN_s	Physics: relativity Based on the Cactus Computational Framework, solves the Einstein equation in vacuum using the Einstein Toolkit.
508.namd_r	--	Molecular Dynamics Derived from NAMD. Simulates large biomolecular systems.
510.parest_r	--	Biomedical imaging: optical tomography with finite elements Resolves the problem of reconstruction of interior properties of a 3D body from multiple observations at its two-dimensional surface.
511.provray_r	--	Ray-tracing Based on POV-Ray version 3.7, renders a 2560 x 2048 pixel image of a chess board, with the pieces placed on the board in the starting position.
519.lbm_r	619.lbm_s	Fluid Dynamics Implements the “Lattice-Boltzmann Method” to simulate incompressible fluids in 3D.
521.wrf_r	621.wrf_s	Weather forecasting Based on Version 3.6.1 of the Weather Research and Forecasting Model (WRF), generates the model at a horizontal resolution of 30 km every 3 simulated hours.
526.blender_r	--	3D rendering and animation Creates 3D images and calculates the Structual SIMilarity (SSIM) index to compare expected images.
527.cam4_r	627.cam4_s	Atmosphere modeling Based on Community Atmosphere Model (CAM)-5.0 which is a component of Community Earth System Model (CESM) 1.0.2, generates an atmospheric general circulation model.
--	628.pop2_s	Wide-scale ocean modeling (climate level) Based on Community Earth System Model (CESM) 1.0, simulates the earth’s climate system.
538.imagick_r	638.imagick_s	Image manipulation Converts bitmap images through various operations like resize, color space, and so on. The outputs are compared by calculating the Structural SIMilarity (SSIM) index.
544.nab_r	644.nab_s	Molecular Dynamics Based on Nucleic Acid Builder (NAB), which performs the kinds of intensive floating point calculations that commonly occur in the life sciences, simulates the state of molecular dynamics.
549.fotonik3d_r	659.fotonik3d_s	Computational Electromagnetics Based on the finite-difference time-domain (FDTD) method, solves the Maxwell equations and calculates the transmission coefficient of a photonic waveguide.
554.roms_r	654.roms_s	Regional ocean modeling Based on ROMS/TOMS version 3.2, calculates various energies and volumes.

All individual benchmarks share the fact that file-I/O only occurs when input files are read. Computation uses only cache and RAM. However, this requires the system to have a minimum memory capacity of 16 GB in the SPECspeed test, and (1 * number of processor cores) GB on 32-bit systems and (2 * number of processor cores) GB on 64-bit systems in the SPECrate test, not including the space for the operating system and other non-SPECcpu2017-specific tasks.

Measurement results and their interpretation

CPU2017 can carry out eight different measurements, which are distinguished in terms of means and methodology:

Benchmark	Number of single benchmarks	Arithmetic	Compiler optimization	Measurement result	
SPECspeed2017_int_peak	10	Integer	Aggressive (peak)	Speed	Performance
SPECspeed2017_int_energy_peak					Power Efficiency
SPECspeed2017_int_base			Conservative (base)		Performance
SPECspeed2017_int_energy_base					Power Efficiency
SPECrate2017_int_peak	10		Aggressive (peak)	Throughput	Performance
SPECrate2017_int_energy_peak					Power Efficiency
SPECrate2017_int_base			Conservative (base)		Performance
SPECrate2017_int_energy_base					Power Efficiency
SPECspeed2017_fp_peak	10	Floating point	Aggressive (peak)	Speed	Performance
SPECspeed2017_fp_energy_peak					Power Efficiency
SPECspeed2017_fp_base			Conservative (base)		Performance
SPECspeed2017_fp_energy_base					Power Efficiency
SPECrate2017_fp_peak	13		Aggressive (peak)	Throughput	Performance
SPECrate2017_fp_energy_peak					Power Efficiency
SPECrate2017_fp_rate			Conservative (base)		Performance
SPECrate2017_fp_energy_rate					Power Efficiency

The benchmarks are provided by SPEC as source code to allow execution of the benchmarks on various hardware models. Therefore, they have to be compiled first. However, the compilation of programs is not a standardized procedure. There are a couple of optional optimization possibilities that may be used by one programmer and ignored by another. Considering this problem, SPEC allows both: measuring with conservative (base) optimization where strict guidelines exist – essentially the use of the same optimization flags in an identical sequence for all individual benchmarks – as well as measuring with aggressive (peak) optimization where the guidelines are more generous, so each single benchmark can be optimized individually. The first measurement is mandatory, and the second one is only optional.

SPEC provides two different methods of performance measuring. One method (speed) consists of measuring how much time is required to execute a single task. The second method (rate) measures the throughput, which means how much time is required to execute a pre-defined number of multiple tasks in parallel.

The latest version of SPECcpu2017, version 1.1, introduced a new metric to measure the power efficiency. In this version, the performance per watt can also be measured by monitoring the maximum system power (W), average system power (W), and total energy (kJ) using a power meter and an external controller system during performance measurements. The power efficiency measurements are optional.

All benchmark results represent the geometric mean of normalized ratios determined for the individual benchmarks. “Normalized” means measuring how fast the test system runs in comparison to a reference system. SPEC chose the Sun “Fire V490” system with a 2100 MHz UltraSPARC IV processor as the reference system. Each benchmark was executed on this system and the reference times were determined. For every base benchmark, a value of “1” has been defined to be the reference system. The result for a test system means that it is able to do the same work as an equivalent number of reference systems in the same time as the reference system needs. For example, if a test system has a SPECspeed2017_int_base value of 2, this means that it has executed this benchmark about twice as fast as the reference system. Or, if a test system has a SPECrate2017_fp_base value of 7.8, this means that it has executed this benchmark at about 7.8 / [number of threads] as fast as the reference system, and that the test system has executed this benchmark in [number of threads] parallel threads. In general, if the number of threads for two systems is the same, the system with the higher result is faster. If the number of threads is different, the system with the higher result does more. The reason for this may be that it is faster or that it can do more in parallel.

The geometric mean is used to calculate the overall benchmark result from the single benchmark results. The effect of this method is a weighting in favor of the lower single results compared with the arithmetical-mean method. The intention behind this is to prevent any attempts to optimize compilers for an individual benchmark and thus to give this benchmark a disproportionately high weight. Moreover, this causes a shift towards the component that has remained unimproved. Better processors do not have such a strong impact on those single benchmarks that require many RAM accesses relative to cache accesses. Their worse results compared to other single benchmarks are emphasized by the use of the geometric mean.

Since minor fluctuations in performance are completely natural between measuring runs, all SPEC measurement results are deliberately rounded to three significant digits. For example, both a measured SPECrate2017_int_peak value of 1234.999 and a measured SPECrate_int_peak value of 1225.000 result in a rounded SPECint_rate value of 1230. The result files in html, pdf, ps and txt format contain such rounded values. Even requests with SPEC for results (e.g. greater than 1230) always refer to rounded values.

The interpretation of benchmark results depends on whether a complete system or a single component (processor, memory, or compiler) is to be compared with a competitive product. If the focus is on a single component, the other components have to be considered to be identical, as otherwise, values cannot be clearly assigned to single components. Therefore, the system configuration is documented with great precision in all publications.

Usually, we submit our SPECcpu2017 measurements for publication at SPEC. However, not all of our results appear on SPEC's web sites. As we archive the log data for all measurements, we are able to prove the correct implementation of the measurements – even ones that have not been submitted – at any time.

Literature

PRIMERGY Servers

http://www.fujitsu.com/fts/products/computing/servers/primergy/

PRIMEQUEST Servers

http://www.fujitsu.com/fts/products/computing/servers/mission-critical/

CELSIUS Workstations

http://www.fujitsu.com/fts/products/computing/pc/workstations/

PRIMERGY & PRIMEQUEST Performance

http://www.fujitsu.com/fts/products/computing/servers/primergy/benchmarks/

Benchmark descriptions

http://www.fujitsu.com/fts/products/computing/servers/primergy/benchmarks/benchmark-descriptions.html

This White Paper:

- http://docs.ts.fujitsu.com/dl.aspx?id=20f1f4e2-5b3c-454a-947f-c169fca51eb1
- http://docs.ts.fujitsu.com/dl.aspx?id=0f641c7e-bb5e-45e4-854f-cdd31faf5343

SPECcpu2017

http://www.spec.org/osg/cpu2017

Contact

FUJITSU

Website: http://www.fujitsu.com/

PRIMERGY & PRIMEQUEST Product Marketing

mailto:Primergy-PM@ts.fujitsu.com

PRIMERGY Performance and Benchmarks

mailto:primergy.benchmark@ts.fujitsu.com