SPEC® CINT2006 Result

Copyright 2006-2014 Standard Performance Evaluation Corporation

NEC Corporation

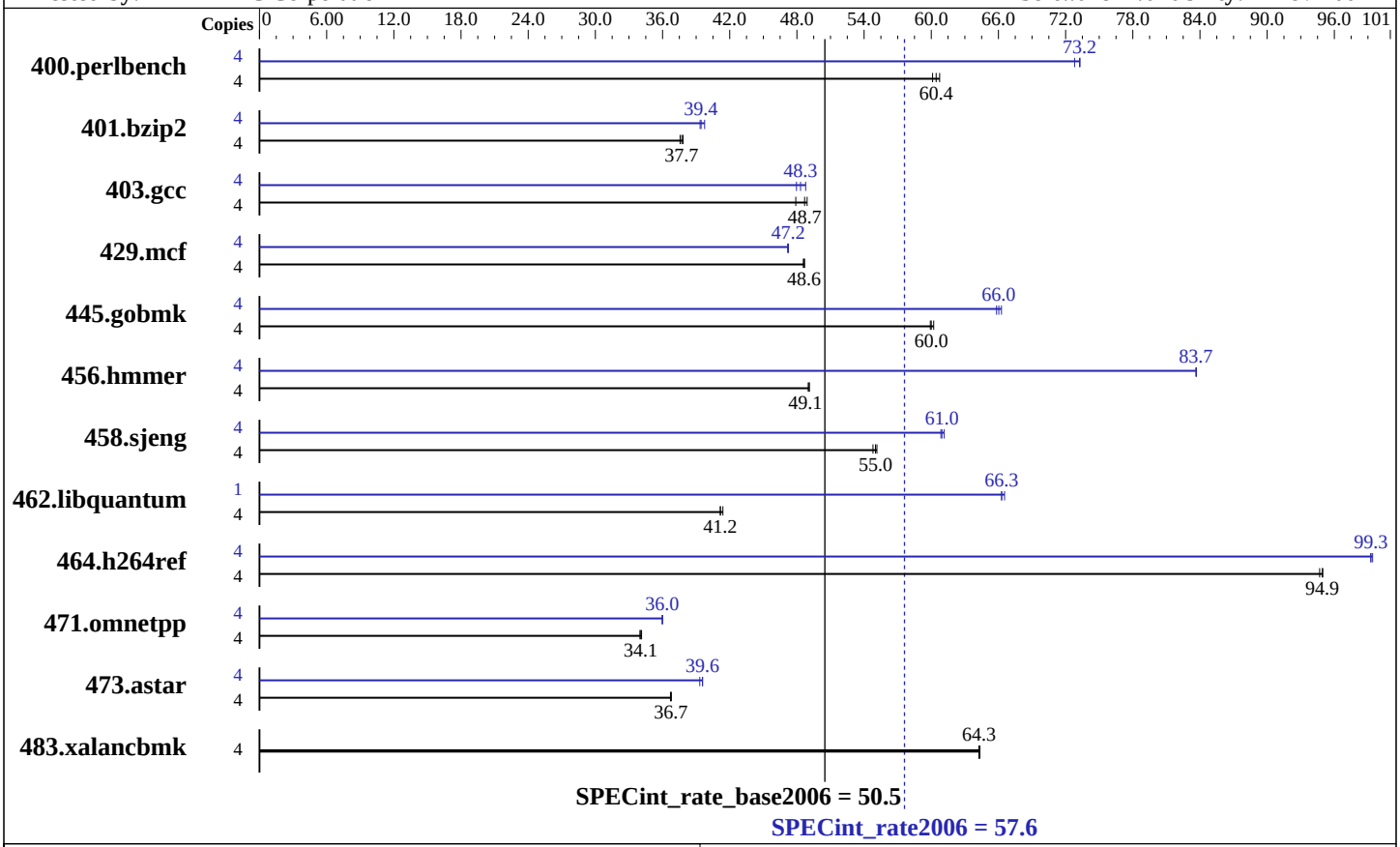
Express5800/120Bb-6
(Intel Xeon 5148 LV)

SPECint®_rate2006 = 57.6

SPECint_rate_base2006 = 50.5

CPU2006 license: 9006
Test sponsor: NEC Corporation
Tested by: NEC Corporation

Test date: Feb-2008
Hardware Availability: May-2007
Software Availability: Nov-2007



Hardware

CPU Name:	Intel Xeon 5148 LV
CPU Characteristics:	2.33 GHz, 4 MB L2, 1333 MHz bus
CPU MHz:	2333
FPU:	Integrated
CPU(s) enabled:	4 cores, 2 chips, 2 cores/chip
CPU(s) orderable:	1,2 chips
Primary Cache:	32 KB I + 32 KB D on chip per core
Secondary Cache:	4 MB I+D on chip per chip
L3 Cache:	None
Other Cache:	None
Memory:	8 GB (4x2 GB PC2-5300F, 2 rank, CL5-5-5, ECC)
Disk Subsystem:	1x73.2 GB SAS, 10000RPM
Other Hardware:	None

Software

Operating System:	SUSE Linux Enterprise Server 10 (x86_64) SP1, Kernel 2.6.16.46-0.12-smp
Compiler:	Intel C++ Compiler for Linux32 and Linux64 version 10.1 Build 20070913 Package ID: l_cc_p_10.1.008
Auto Parallel:	Yes
File System:	ext2
System State:	Run level 3 (multi-user)
Base Pointers:	32-bit
Peak Pointers:	32/64-bit
Other Software:	MicroQuill SmartHeap library 8.1 binutils-2.17.tar.gz, Version 2.17

SPEC CINT2006 Result

Copyright 2006-2014 Standard Performance Evaluation Corporation

NEC Corporation

Express5800/120Bb-6
(Intel Xeon 5148 LV)

SPECint_rate2006 = 57.6

SPECint_rate_base2006 = 50.5

CPU2006 license: 9006
Test sponsor: NEC Corporation
Tested by: NEC Corporation

Test date: Feb-2008
Hardware Availability: May-2007
Software Availability: Nov-2007

Results Table

Benchmark	Base							Peak						
	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
400.perlbench	4	650	60.1	**647**	**60.4**	643	60.7	4	537	72.8	**534**	**73.2**	533	73.3
401.bzip2	4	1028	37.6	**1023**	**37.7**	1020	37.8	4	981	39.4	**979**	**39.4**	971	39.8
403.gcc	4	**661**	**48.7**	672	47.9	658	48.9	4	671	48.0	660	48.8	**666**	**48.3**
429.mcf	4	751	48.6	**750**	**48.6**	749	48.7	4	772	47.2	**773**	**47.2**	773	47.2
445.gobmk	4	**699**	**60.0**	697	60.2	700	59.9	4	**635**	**66.0**	637	65.8	633	66.3
456.hmmer	4	762	49.0	**760**	**49.1**	760	49.1	4	446	83.7	**446**	**83.7**	446	83.6
458.sjeng	4	**880**	**55.0**	878	55.1	883	54.8	4	795	60.9	791	61.2	**794**	**61.0**
462.libquantum	4	2002	41.4	**2013**	**41.2**	2014	41.2	1	313	66.3	**313**	**66.3**	311	66.5
464.h264ref	4	932	95.0	935	94.7	**933**	**94.9**	4	**892**	**99.3**	892	99.3	890	99.4
471.omnetpp	4	733	34.1	**734**	**34.1**	736	34.0	4	695	36.0	**695**	**36.0**	695	36.0
473.astar	4	**764**	**36.7**	763	36.8	765	36.7	4	714	39.3	**710**	**39.6**	710	39.6
483.xalancbmk	4	430	64.3	429	64.3	**429**	**64.3**	4	430	64.3	429	64.3	**429**	**64.3**

Results appear in the order in which they were run. Bold underlined text indicates a median measurement.

Operating System Notes

'ulimit -s unlimited' was used to set the stacksize to unlimited prior to run
OMP_NUM_THREADS set to number of cores

General Notes

All benchmarks compiled in 32-bit mode except 401.bzip2,
456.hmmer, for peak, are compiled in 64-bit mode

Base Compiler Invocation

C benchmarks:
icc

C++ benchmarks:
icpc

Base Portability Flags

400.perlbench: -DSPEC_CPU_LINUX_IA32
462.libquantum: -DSPEC_CPU_LINUX
483.xalancbmk: -DSPEC_CPU_LINUX

Base Optimization Flags

C benchmarks:
`-fast -inline-calloc -opt-malloc-options=3`

C++ benchmarks:
`-xT -ipo -O3 -no-prec-div -Wl,-z,muldefs`
`-L/opt/SmartHeap_8.1/lib -lsmartheap`

Base Other Flags

C benchmarks:

403.gcc: `-Dalloca=_alloca`

Peak Compiler Invocation

C benchmarks (except as noted below):
`icc`

401.bzip2: `/opt/intel/cce/10.1.008/bin/icc`
`-L/opt/intel/cce/10.1.008/lib`
`-I/opt/intel/cce/10.1.008/include`

456.hmmer: `/opt/intel/cce/10.1.008/bin/icc`
`-L/opt/intel/cce/10.1.008/lib`
`-I/opt/intel/cce/10.1.008/include`

C++ benchmarks:
`icpc`

Peak Portability Flags

400.perlbench: `-DSPEC_CPU_LINUX_IA32`
401.bzip2: `-DSPEC_CPU_LP64`
456.hmmer: `-DSPEC_CPU_LP64`
462.libquantum: `-DSPEC_CPU_LINUX`
483.xalancbmk: `-DSPEC_CPU_LINUX`

Peak Optimization Flags

C benchmarks:

Continued on next page

Peak Optimization Flags (Continued)

400.perlbench: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -ansi-alias -prefetch`

401.bzip2: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -prefetch`

403.gcc: `-fast -inline-calloc -opt-malloc-options=3`

429.mcf: `-fast -prefetch`

445.gobmk: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-xT -O2 -ipo -no-prec-div -ansi-alias`

456.hmmer: `-fast -unroll2 -ansi-alias -opt-multi-version-aggressive`

458.sjeng: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -unroll4`

462.libquantum: `-fast -unroll4 -Ob0 -prefetch -opt-streaming-stores always -vec-guard-write -opt-malloc-options=3 -parallel -par-runtime-control`

464.h264ref: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -unroll2 -ansi-alias`

C++ benchmarks:

471.omnetpp: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-xT -O3 -ipo -no-prec-div -ansi-alias -opt-ra-region-strategy=block -Wl,-z,muldefs -L/opt/SmartHeap_8.1/lib -lsmartheap`

473.astar: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-xT -O3 -ipo -no-prec-div -ansi-alias -opt-ra-region-strategy=routine -Wl,-z,muldefs -L/opt/SmartHeap_8.1/lib -lsmartheap`

483.xalancbmk: `basepeak = yes`

Peak Other Flags

C benchmarks:

403.gcc: `-Dalloca=_alloca`

The flags file that was used to format this result can be browsed at
`http://www.spec.org/cpu2006/flags/NEC-Intel-ic10.1-INT-ia32-linux-flags.20090714.00.html`

You can also download the XML flags source by saving the following link:
`http://www.spec.org/cpu2006/flags/NEC-Intel-ic10.1-INT-ia32-linux-flags.20090714.00.xml`

SPEC and SPECint are registered trademarks of the Standard Performance Evaluation Corporation. All other brand and product names appearing in this result are trademarks or registered trademarks of their respective holders.

For questions about this result, please contact the tester.
For other inquiries, please contact webmaster@spec.org.

Tested with SPEC CPU2006 v1.0.
Report generated on Tue Jul 22 15:39:36 2014 by SPEC CPU2006 PS/PDF formatter v6932.
Originally published on 7 March 2008.