SPEC® CINT2006 Result

SGI

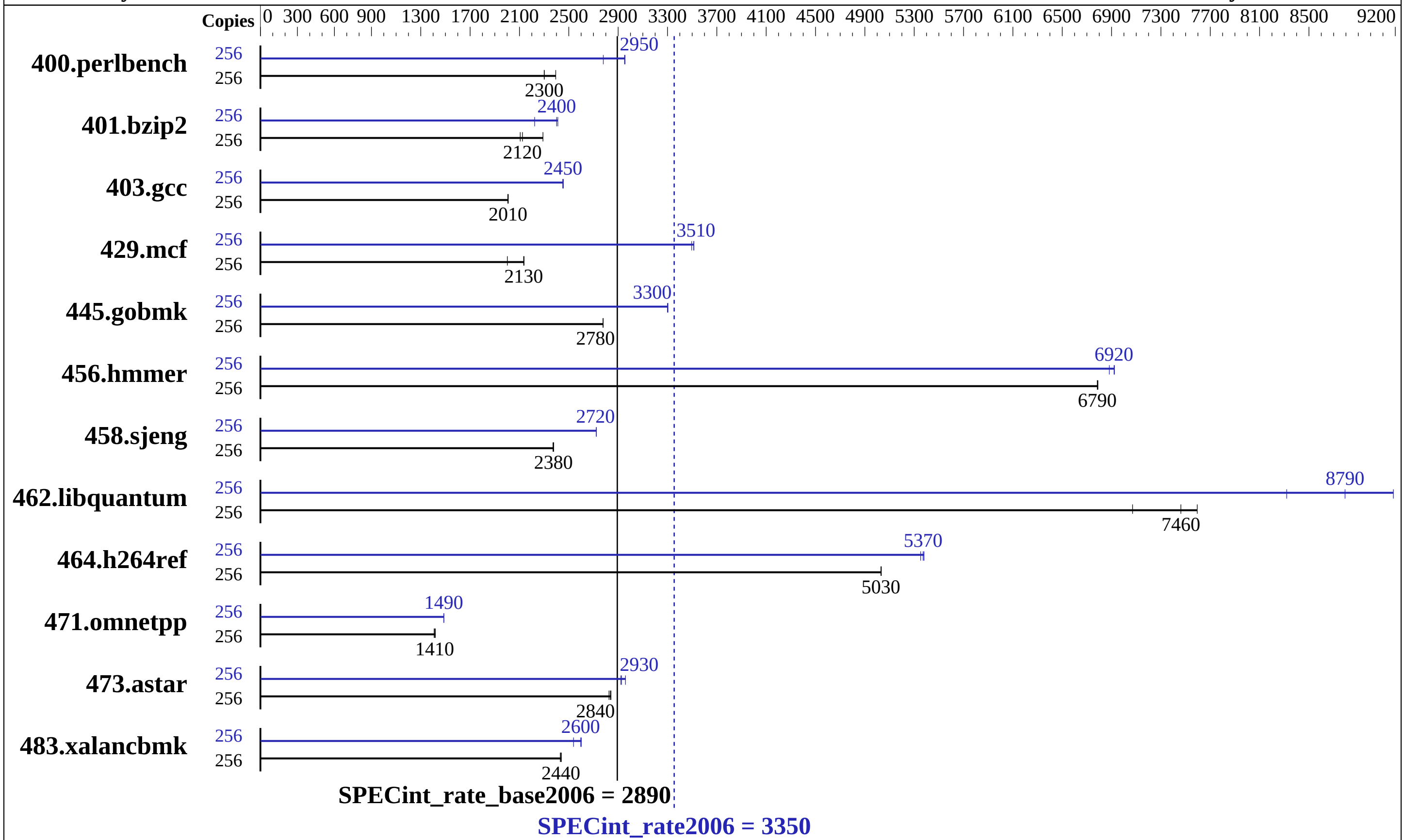
SGI Altix 4700 Density System (Itanium Processor 9150M 1.66GHz/24M)

SPECint®_rate2006 = 3350

SPECint_rate_base2006 = 2890

CPU2006 license: 4
Test sponsor: SGI
Tested by: SGI

Test date: Oct-2007
Hardware Availability: Nov-2007
Software Availability: Nov-2007



Hardware

CPU Name:	Dual-Core Intel Itanium 9150M
CPU Characteristics:	667MHz FSB
CPU MHz:	1669
FPU:	Integrated
CPU(s) enabled:	256 cores, 128 chips, 2 cores/chip
CPU(s) orderable:	4 to 256 blades with 2 chips per blade
Primary Cache:	16 KB I + 16 KB D on chip per core
Secondary Cache:	1 MB I + 256 KB D on chip per core
L3 Cache:	12 MB I+D on chip per core
Other Cache:	None
Memory:	512 GB (8*1GB DDR2-400 DIMMS per 4 core module)
Disk Subsystem:	2.4 TB RAID 4+1 32 x 73 GB SCSI (Seagate Cheetah 15k rpm)
Other Hardware:	None

Software

Operating System:	SUSE Linux Enterprise Server 10 (ia64) SP1, Kernel 2.6.16.53-0.16-default
Compiler:	Intel C++ Compiler for Linux 10.1 (Build 20071005) MicroQuill SmartHeap Library 8 (www.microquill.com)
Auto Parallel:	No
File System:	xfs
System State:	Multi-user
Base Pointers:	64-bit
Peak Pointers:	32/64-bit
Other Software:	SGI ProPack 5 Service Pack 3

SPEC CINT2006 Result

SGI

SGI Altix 4700 Density System (Itanium Processor 9150M 1.66GHz/24M)

SPECint_rate2006 = 3350

SPECint_rate_base2006 = 2890

CPU2006 license: 4
Test sponsor: SGI
Tested by: SGI

Test date: Oct-2007
Hardware Availability: Nov-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	256	1045	2390	1087	2300	**1087**	**2300**	256	900	2780	846	2960	**847**	**2950**
401.bzip2	256	1078	2290	**1163**	**2120**	1173	2110	256	1111	2220	1025	2410	**1029**	**2400**
403.gcc	256	1028	2010	**1027**	**2010**	1026	2010	256	**840**	**2450**	841	2450	839	2460
429.mcf	256	1166	2000	**1094**	**2130**	1092	2140	256	**665**	**3510**	668	3500	664	3510
445.gobmk	256	967	2780	**966**	**2780**	966	2780	256	814	3300	**813**	**3300**	813	3300
456.hmmer	256	352	6790	**352**	**6790**	352	6780	256	347	6880	**345**	**6920**	345	6920
458.sjeng	256	1306	2370	1303	2380	**1304**	**2380**	256	1137	2720	**1137**	**2720**	1137	2720
462.libquantum	256	750	7070	**711**	**7460**	698	7590	256	638	8320	578	9180	**603**	**8790**
464.h264ref	256	1126	5030	1126	5030	**1126**	**5030**	256	1058	5350	1053	5380	**1055**	**5370**
471.omnetpp	256	1136	1410	**1132**	**1410**	1128	1420	256	1076	1490	**1076**	**1490**	1076	1490
473.astar	256	**634**	**2840**	636	2830	632	2840	256	607	2960	615	2920	**614**	**2930**
483.xalancbmk	256	724	2440	727	2430	**725**	**2440**	256	696	2540	679	2600	**680**	**2600**

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

General Notes

```
Processes were bound to CPUs using dplace.
limit stacksize unlimited
```

Base Compiler Invocation

C benchmarks:
`icc`

C++ benchmarks:
`icpc`

Base Portability Flags

400.perlbench: `-DSPEC_CPU_LP64 -DSPEC_CPU_LINUX_IA64`
401.bzip2: `-DSPEC_CPU_LP64`
403.gcc: `-DSPEC_CPU_LP64`
429.mcf: `-DSPEC_CPU_LP64`
445.gobmk: `-DSPEC_CPU_LP64`
456.hmmer: `-DSPEC_CPU_LP64`
458.sjeng: `-DSPEC_CPU_LP64`
462.libquantum: `-DSPEC_CPU_LP64 -DSPEC_CPU_LINUX`
464.h264ref: `-DSPEC_CPU_LP64`
471.omnetpp: `-DSPEC_CPU_LP64`

Continued on next page

Base Portability Flags (Continued)

473.astar: -DSPEC_CPU_LP64
483.xalancbmk: -DSPEC_CPU_LP64 -DSPEC_CPU_LINUX

Base Optimization Flags

C benchmarks:
-fast -IPF-fp-relaxed -opt-prefetch-next-iteration
-no-opt-prefetch-initial-values -ansi-alias

C++ benchmarks:
-fast -IPF-fp-relaxed -opt-prefetch-next-iteration
-no-opt-prefetch-initial-values -ansi-alias -Wl,-z,muldefs
libsmartheapC64.a libsmartheap64.a

Peak Compiler Invocation

C benchmarks:
icc

C++ benchmarks:
icpc

Peak Portability Flags

Same as Base Portability Flags

Peak Optimization Flags

C benchmarks:

400.perlbench: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF-fp-relaxed
-opt-prefetch-next-iteration -inline-factor=150 -ansi-alias

401.bzip2: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF-fp-relaxed
-opt-prefetch-next-iteration -fno-alias -auto-ilp32
-ansi-alias

403.gcc: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF-fp-relaxed
-opt-prefetch-next-iteration -unroll-aggressive -auto-ilp32
-no-opt-prefetch-initial-values -ansi-alias

Continued on next page

Peak Optimization Flags (Continued)

429.mcf: -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -auto-ilp32 -ansi-alias

445.gobmk: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -auto-ilp32 -no-opt-prefetch-initial-values -ansi-alias

456.hmmer: -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -auto-ilp32 -no-opt-prefetch-initial-values -ansi-alias

458.sjeng: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -unroll-aggressive -ansi-alias

462.libquantum: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -unroll-aggressive -opt-mod-versioning -ansi-alias

464.h264ref: -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -auto-ilp32 -unroll-aggressive -opt-mod-versioning -no-prefetch -fno-alias -inline-factor=150 -ansi-alias

C++ benchmarks:

471.omnetpp: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -fno-alias -inline-factor=150 -ansi-alias -Wl,-z,muldefs libsmartheapC64.a libsmartheap64.a

473.astar: -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -no-prefetch -inline-factor=150 -ansi-alias -Wl,-z,muldefs libsmartheapC64.a libsmartheap64.a

483.xalancbmk: -fast -IPF-fp-relaxed -opt-prefetch-next-iteration -unroll-aggressive -no-prefetch -ansi-alias -Wl,-z,muldefs libsmartheapC64.a libsmartheap64.a

The flags file that was used to format this result can be browsed at
http://www.spec.org/cpu2006/flags/Intel-ic91-ipf.html

You can also download the XML flags source by saving the following link:
http://www.spec.org/cpu2006/flags/Intel-ic91-ipf.xml

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 14:10:53 2014 by SPEC CPU2006 PS/PDF formatter v6932.
Originally published on 14 November 2007.