SPEC® CINT2006 Result

Copyright 2006-2014 Standard Performance Evaluation Corporation

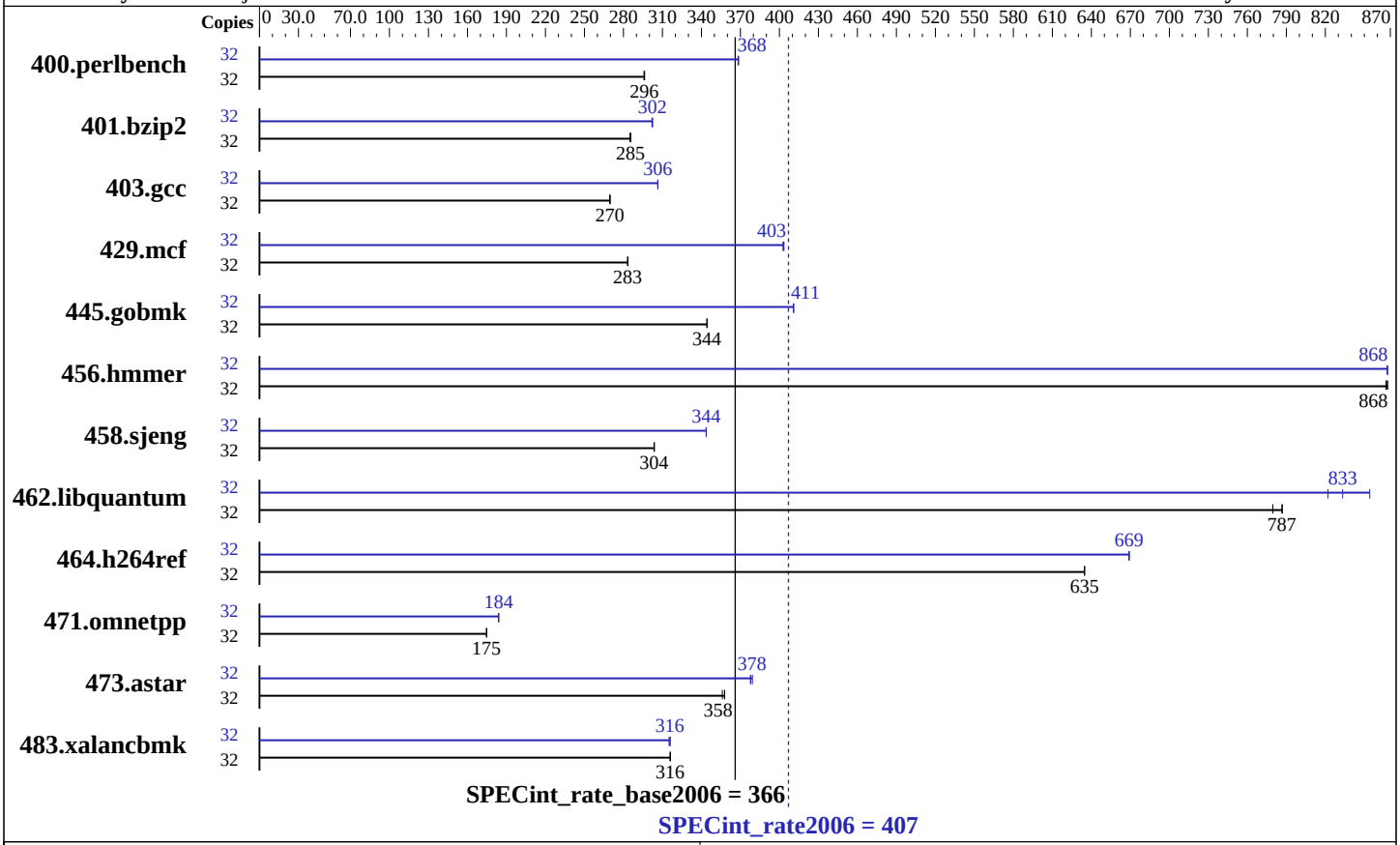
Fujitsu Limited
PRIMEQUEST 540A

SPECint®_rate2006 = 407

SPECint_rate_base2006 = 366

CPU2006 license: 19
Test sponsor: Fujitsu Limited
Tested by: Fujitsu Limited

Test date: Mar-2008
Hardware Availability: May-2008
Software Availability: Feb-2008



Hardware

CPU Name:	Dual-Core Intel Itanium 9150M
CPU Characteristics:	1.66GHz/24MB, 667MHz FSB
CPU MHz:	1667
FPU:	Integrated
CPU(s) enabled:	32 cores, 16 chips, 2 cores/chip
CPU(s) orderable:	2-16 chips
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:	256 GB (128 x 2GB DDR2-667 DIMMs)
Disk Subsystem:	2 x 147GB (SCSI Ultra 320, 10000rpm) No RAID configuration
Other Hardware:	None

Software

Operating System:	Red Hat Enterprise Linux 5.1, Kernel 2.6.18-53.el5 on an ia64
Compiler:	Intel C++ Compiler for Linux 10.1 (Build 20080112)
Auto Parallel:	No
File System:	ext2
System State:	Runlevel 1 (single user mode)
Base Pointers:	64-bit
Peak Pointers:	32/64-bit
Other Software:	MicroQuill Smartheap 8.0

SPEC CINT2006 Result

Copyright 2006-2014 Standard Performance Evaluation Corporation

Fujitsu Limited
PRIMEQUEST 540A

SPECint_rate2006 = 407

SPECint_rate_base2006 = 366

CPU2006 license: 19
Test sponsor: Fujitsu Limited
Tested by: Fujitsu Limited

Test date: Mar-2008
Hardware Availability: May-2008
Software Availability: Feb-2008

Results Table

Benchmark	Base							Peak						
	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
400.perlbench	32	1055	296	1057	296	**1056**	**296**	32	849	368	848	369	**849**	**368**
401.bzip2	32	**1082**	**285**	1081	286	1083	285	32	**1021**	**302**	1023	302	1021	302
403.gcc	32	956	270	955	270	**955**	**270**	32	840	307	841	306	**841**	**306**
429.mcf	32	**1030**	**283**	1030	283	1029	283	32	**724**	**403**	723	403	725	403
445.gobmk	32	**975**	**344**	975	344	975	344	32	817	411	817	411	**817**	**411**
456.hmmer	32	344	868	**344**	**868**	345	867	32	**344**	**868**	344	868	344	868
458.sjeng	32	1275	304	1275	304	**1275**	**304**	32	1126	344	1127	344	**1127**	**344**
462.libquantum	32	**843**	**787**	850	780	843	787	32	807	822	776	854	**796**	**833**
464.h264ref	32	1116	635	**1116**	**635**	1116	635	32	1058	669	1059	669	**1058**	**669**
471.omnetpp	32	1145	175	**1145**	**175**	1145	175	32	**1087**	**184**	1087	184	1087	184
473.astar	32	**628**	**358**	631	356	628	358	32	595	378	592	379	**594**	**378**
483.xalancbmk	32	699	316	698	316	**699**	**316**	32	701	315	**699**	**316**	699	316

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

General Notes

```
Processes are bound to CPUs using taskset.

limit stacksize unlimited

Memory system is in "Non Mirror Mode".

The following 2 environment variables were set
    MALLOC_MMAP_MAX_=0
    MALLOC_TRIM_THRESHOLD_=-1
This will cause use of sbrk() calls instead of
mmap() calls to get memory from the system.
```

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

Base Optimization Flags

C benchmarks:
-fast -IPF-fp-relaxed -ansi-alias -no-opt-prefetch-initial-values
-opt-prefetch-next-iteration -opt-prefetch-issue-excl-hint
-unroll-aggressive

C++ benchmarks:
-fast -IPF-fp-relaxed -ansi-alias -no-opt-prefetch-initial-values
-opt-prefetch-next-iteration -opt-prefetch-issue-excl-hint
-unroll-aggressive -Wl,-z,muldefs
/opt/SmartHeap_8/lib/libsmartheapC64.a
/opt/SmartHeap_8/lib/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 -ansi_alias -IPF_fp_relaxed -opt-mod-versioning -unroll-aggressive -inline-factor=150`

401.bzip2: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -IPF-fp-relaxed -ansi-alias -fno-alias -auto-ilp32 -opt-prefetch-next-iteration`

403.gcc: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -ansi_alias -auto-ilp32 -IPF_fp_relaxed -no-opt-prefetch-initial-values -opt-prefetch-next-iteration -unroll-aggressive`

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

445.gobmk: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -IPF_fp_relaxed -auto-ilp32 -no-opt-prefetch-initial-values -opt-prefetch-next-iteration -ansi-alias`

456.hmmer: `-fast -IPF_fp_relaxed -auto-ilp32 -no-opt-prefetch-initial-values`

458.sjeng: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -IPF_fp_relaxed -unroll-aggressive -no-prefetch -opt-prefetch-next-iteration`

462.libquantum: `-fast -IPF-fp-relaxed -auto-ilp32 -ansi-alias -opt-mod-versioning -no-opt-prefetch-initial-values -opt-prefetch-issue-excl-hint`

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

C++ benchmarks:

471.omnetpp: `-prof-gen`(pass 1) `-prof-use`(pass 2) `-fast -IPF-fp-relaxed -ansi-alias -fno-alias -inline-max-per-routine=50 -inline-factor=150 -Wl,-z,muldefs /opt/SmartHeap_8/lib/libsmartheapC64.a /opt/SmartHeap_8/lib/libsmartheap64.a`

473.astar: `-fast -IPF-fp-relaxed -no-prefetch -ansi-alias -fno-alias -inline-max-size=5000 -Wl,-z,muldefs /opt/SmartHeap_8/lib/libsmartheapC64.a /opt/SmartHeap_8/lib/libsmartheap64.a`

Continued on next page

Peak Optimization Flags (Continued)

483.xalancbmk: `-fast -IPF-fp-relaxed -unroll-aggressive -ansi-alias -no-opt-prefetch-initial-values -Wl,-z,muldefs /opt/SmartHeap_8/lib/libsmartheapC64.a /opt/SmartHeap_8/lib/libsmartheap64.a`

The flags file that was used to format this result can be browsed at
`http://www.spec.org/cpu2006/flags/Fujitsu.PQ580A.ipf.linux.flags.html`

You can also download the XML flags source by saving the following link:
`http://www.spec.org/cpu2006/flags/Fujitsu.PQ580A.ipf.linux.flags.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 v1.0.1.
Report generated on Tue Jul 22 18:38:24 2014 by SPEC CPU2006 PS/PDF formatter v6932.
Originally published on 15 April 2008.