SPEC® CINT2006 Result

Copyright 2006-2014 Standard Performance Evaluation Corporation

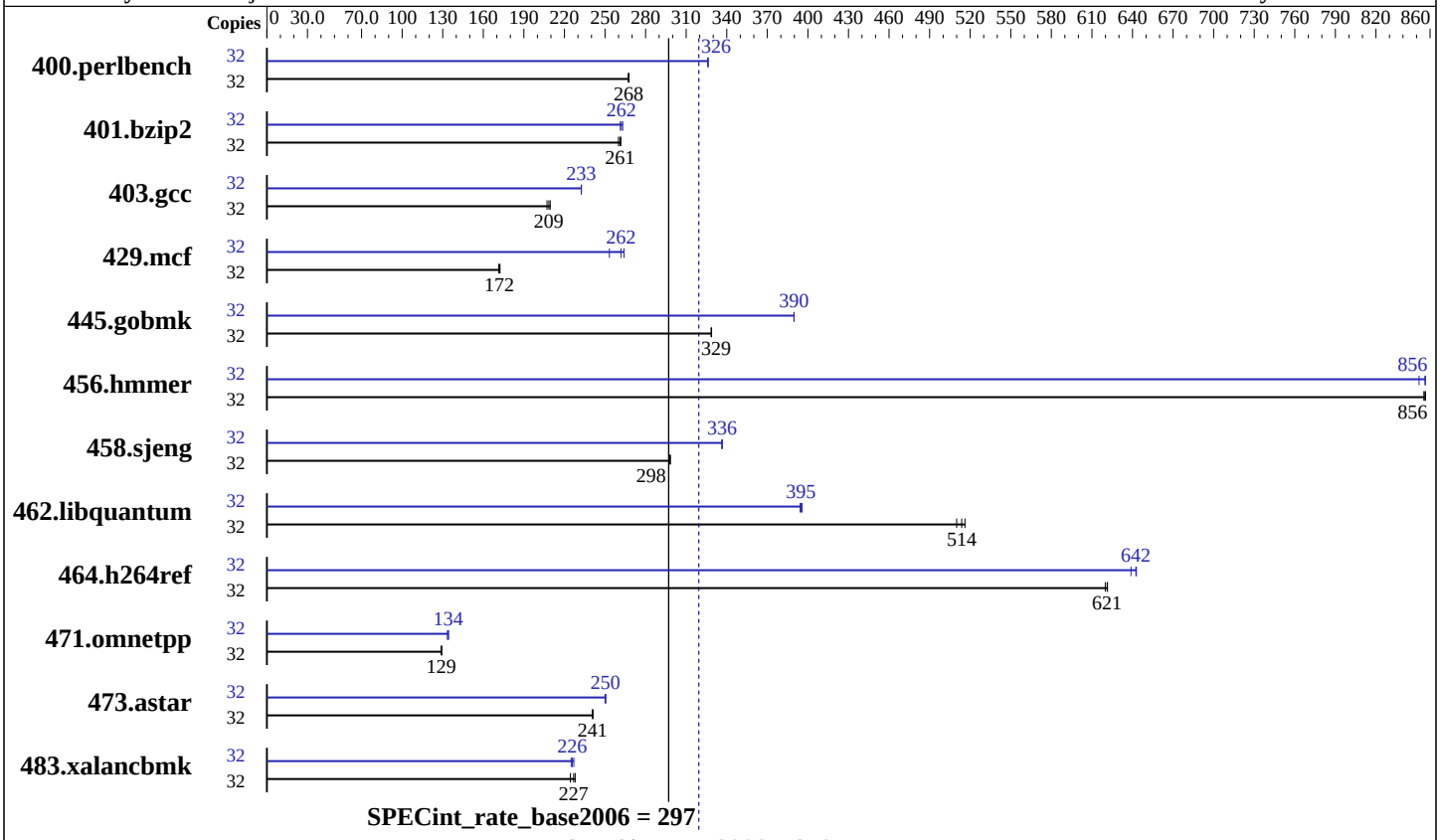
Fujitsu Limited
PRIMEQUEST 540A

SPECint®_rate2006 = 319

SPECint_rate_base2006 = 297

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 9130M
CPU Characteristics:	1.66GHz/8MB, 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:	4 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 = 319

SPECint_rate_base2006 = 297

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	**1169**	**268**	1168	268	1170	267	32	958	326	**959**	**326**	959	326
401.bzip2	32	1180	262	1188	260	**1183**	**261**	32	1174	263	1182	261	**1178**	**262**
403.gcc	32	**1235**	**209**	1230	209	1244	207	32	1108	232	**1108**	**233**	1108	233
429.mcf	32	1703	171	1693	172	**1697**	**172**	32	1152	253	**1115**	**262**	1105	264
445.gobmk	32	1021	329	1022	328	**1021**	**329**	32	861	390	**861**	**390**	861	390
456.hmmer	32	349	856	**349**	**856**	349	857	32	349	857	350	852	**349**	**856**
458.sjeng	32	1298	298	**1299**	**298**	1300	298	32	**1151**	**336**	1151	336	1150	337
462.libquantum	32	**1291**	**514**	1285	516	1300	510	32	**1679**	**395**	1675	396	1682	394
464.h264ref	32	1142	620	**1140**	**621**	1139	622	32	**1102**	**642**	1102	643	1108	639
471.omnetpp	32	1551	129	1548	129	**1550**	**129**	32	1499	133	**1491**	**134**	1490	134
473.astar	32	932	241	**933**	**241**	934	241	32	**897**	**250**	896	251	899	250
483.xalancbmk	32	**973**	**227**	984	224	969	228	32	**978**	**226**	982	225	973	227

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 18:29:44 2014 by SPEC CPU2006 PS/PDF formatter v6932.
Originally published on 15 April 2008.