SPEC® CINT2006 Result

Copyright 2006-2014 Standard Performance Evaluation Corporation

Hewlett-Packard Company

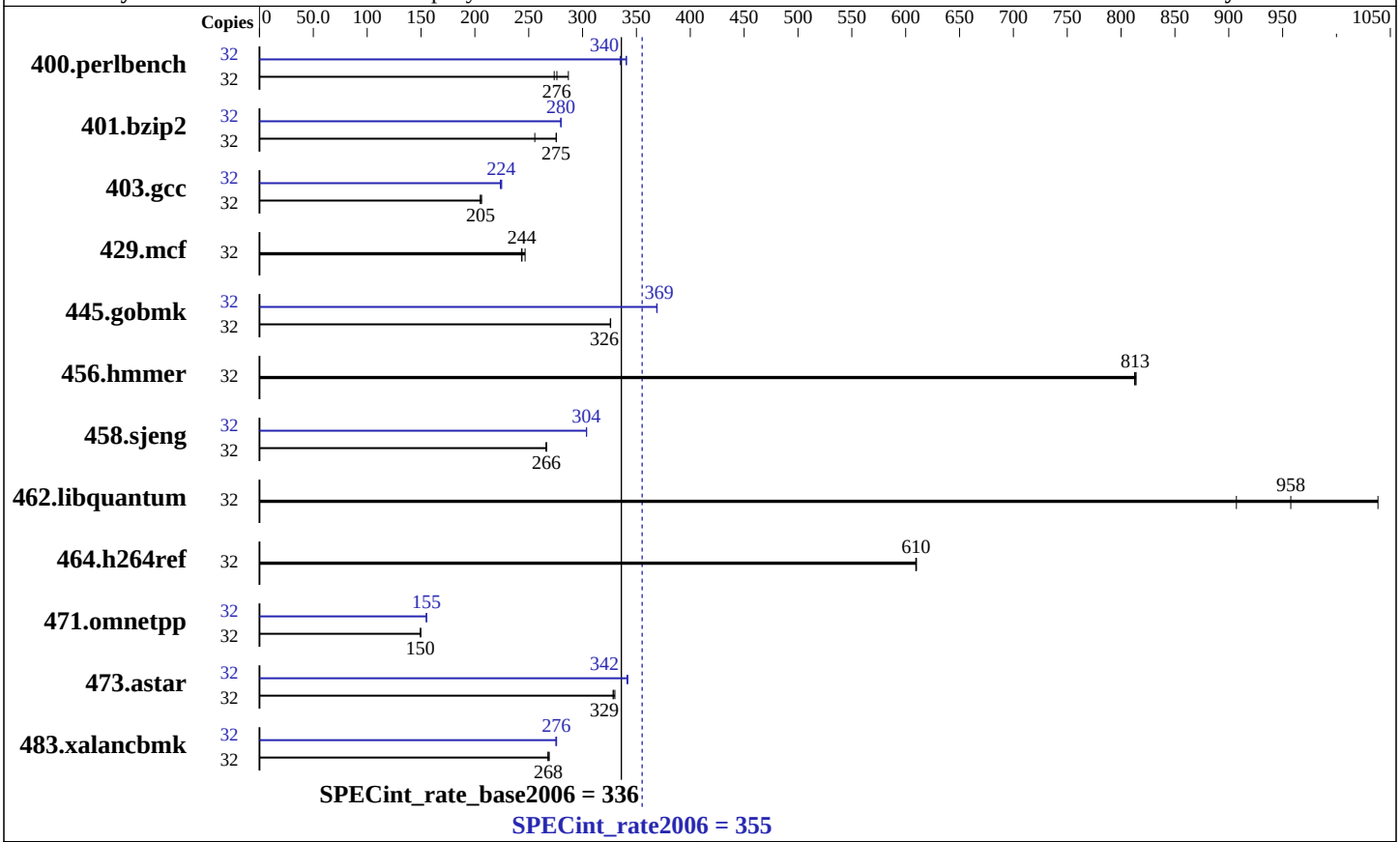
HP Integrity rx8640
(1.6GHz/24MB Dual-Core Intel Itanium 2)

SPECint®_rate2006 = 355

SPECint_rate_base2006 = 336

CPU2006 license: 03
Test sponsor: Hewlett-Packard Company
Tested by: Hewlett-Packard Company

Test date: Dec-2006
Hardware Availability: Sep-2006
Software Availability: Nov-2006



Hardware

CPU Name:	Dual-Core Intel Itanium 2 9050
CPU Characteristics:	1.6GHz/24MB, 533MHz FSB
CPU MHz:	1600
FPU:	Integrated
CPU(s) enabled:	32 cores, 16 chips, 2 cores/chip
CPU(s) orderable:	1-16 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:	128 GB (64x2GB DIMMs)
Disk Subsystem:	73GB 15K RPM SCSI
Other Hardware:	None

Software

Operating System:	Red Hat Enterprise Linux AS release 4 (Update 4)
Compiler:	Intel C++ Compiler 9.1 for Linux (Build 20061105)
Auto Parallel:	No
File System:	ext3
System State:	Multi-user
Base Pointers:	64-bit
Peak Pointers:	64-bit
Other Software:	MicroQuill Smartheap 8.0

SPEC CINT2006 Result

Copyright 2006-2014 Standard Performance Evaluation Corporation

Hewlett-Packard Company

HP Integrity rx8640
(1.6GHz/24MB Dual-Core Intel Itanium 2)

SPECint_rate2006 = 355

SPECint_rate_base2006 = 336

CPU2006 license: 03
Test sponsor: Hewlett-Packard Company
Tested by: Hewlett-Packard Company

Test date: Dec-2006
Hardware Availability: Sep-2006
Software Availability: Nov-2006

Results Table

Benchmark	Base							Peak						
	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
400.perlbench	32	1142	274	**1132**	**276**	1090	287	32	917	341	934	335	**918**	**340**
401.bzip2	32	1120	276	1208	256	**1121**	**275**	32	1103	280	**1103**	**280**	1104	280
403.gcc	32	1256	205	**1254**	**205**	1249	206	32	**1148**	**224**	1145	225	1152	224
429.mcf	32	1200	243	**1197**	**244**	1183	247	32	1200	243	**1197**	**244**	1183	247
445.gobmk	32	1029	326	**1030**	**326**	1031	326	32	909	369	910	369	**910**	**369**
456.hmmer	32	367	813	**367**	**813**	367	814	32	367	813	**367**	**813**	367	814
458.sjeng	32	1455	266	1453	267	**1454**	**266**	32	**1275**	**304**	1276	304	1275	304
462.libquantum	32	638	1040	**692**	**958**	731	907	32	638	1040	**692**	**958**	731	907
464.h264ref	32	1161	610	**1162**	**610**	1162	609	32	1161	610	**1162**	**610**	1162	609
471.omnetpp	32	1338	150	1337	150	**1337**	**150**	32	1289	155	1291	155	**1290**	**155**
473.astar	32	681	330	**683**	**329**	684	328	32	657	342	**657**	**342**	658	341
483.xalancbmk	32	821	269	**823**	**268**	825	268	32	802	275	800	276	**801**	**276**

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

Operating System Notes

```
stacksize set to unlimited prior to run
```

Platform Notes

```
System was configured as a single partition with
4 cells and 4 processors (8 cores) per cell.
Memory was configured as 100% cell local.

The following config file entry was used to bind
processes to cores using the Linux "numactl" utility:
submit = let "MYNUM=$SPECCOPYNUM" ; let "NODE=\$MYNUM/8" ; numactl --cpubind \$NODE --membind \$NODE $command
```

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`

C++ benchmarks:
`-fast -IPF_fp_relaxed -ansi-alias -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 -IPF_fp_relaxed -ansi-alias`

Continued on next page

Peak Optimization Flags (Continued)

401.bzip2: Same as 400.perlbench

403.gcc: Same as 400.perlbench

429.mcf: basepeak = yes

445.gobmk: Same as 400.perlbench

456.hmmer: basepeak = yes

458.sjeng: Same as 400.perlbench

462.libquantum: basepeak = yes

464.h264ref: basepeak = yes

C++ benchmarks:

471.omnetpp: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF_fp_relaxed
-ansi-alias -Wl,-z,muldefs
/opt/SmartHeap_8/lib/libsmartheapC64.a
/opt/SmartHeap_8/lib/libsmartheap64.a

473.astar: -prof_gen(pass 1) -prof_use(pass 2) -fast -IPF_fp_relaxed
-ansi-alias -inline-factor=150 -Wl,-z,muldefs
/opt/SmartHeap_8/lib/libsmartheapC64.a
/opt/SmartHeap_8/lib/libsmartheap64.a

483.xalancbmk: Same as 471.omnetpp

The flags file that was used to format this result can be browsed at
http://www.spec.org/cpu2006/flags/IPF_intel91_flags.20090715.html

You can also download the XML flags source by saving the following link:
http://www.spec.org/cpu2006/flags/IPF_intel91_flags.20090715.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 10:56:24 2014 by SPEC CPU2006 PS/PDF formatter v6932.
Originally published on 9 January 2007.