SPEC® CFP2006 Result

Dell Inc.

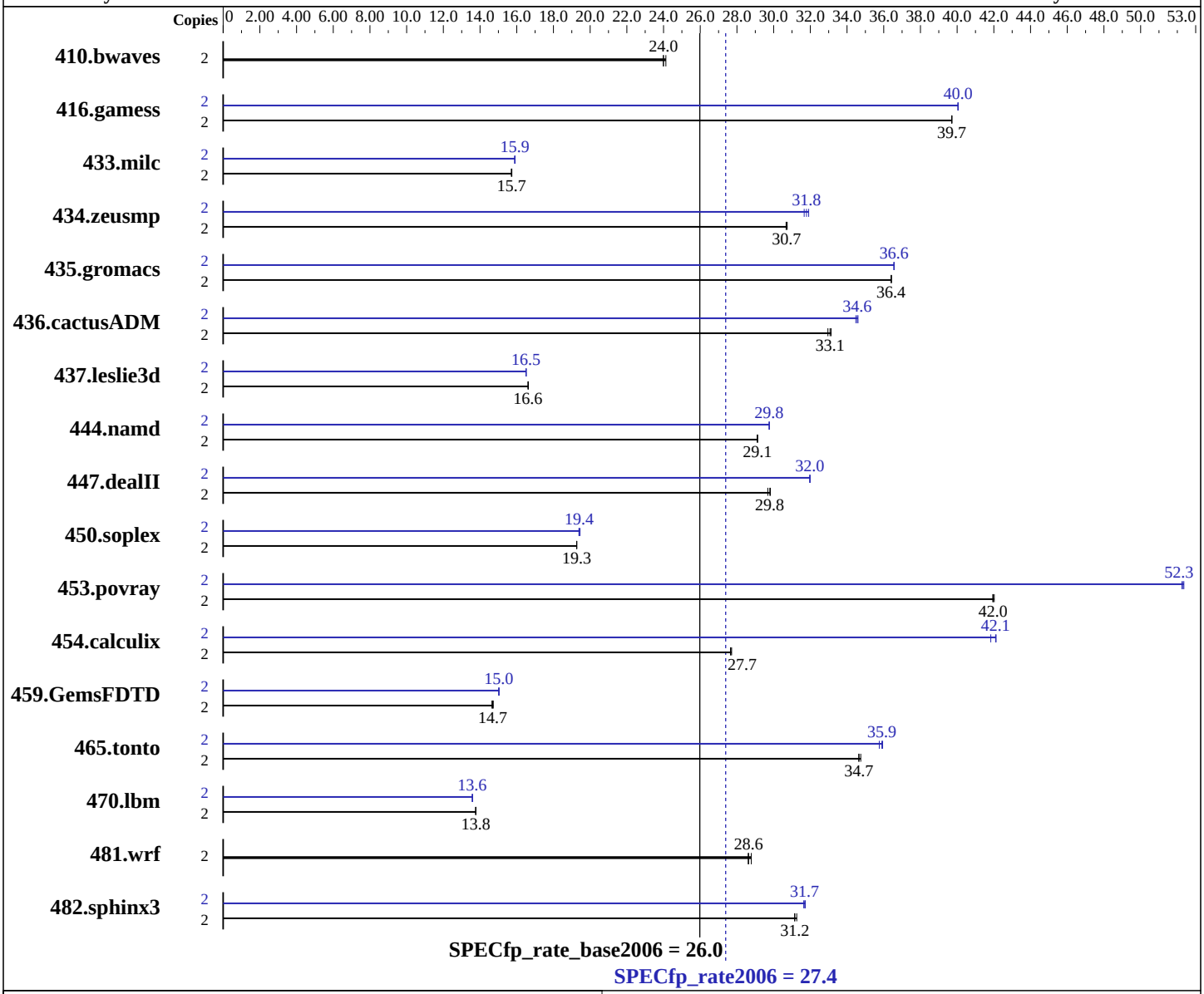
Dell Precision M6300 (Intel X9000, 2.80 GHz)

SPECfp®_rate2006 = 27.4

SPECfp_rate_base2006 = 26.0

CPU2006 license: 55
Test sponsor: Dell Inc.
Tested by: Dell Inc.

Test date: Apr-2008
Hardware Availability: Mar-2008
Software Availability: Mar-2008



Hardware

CPU Name:	Intel Core 2 Extreme X9000
CPU Characteristics:	800 MHz Bus Speed
CPU MHz:	2800
FPU:	Integrated
CPU(s) enabled:	2 cores, 1 chip, 2 cores/chip
CPU(s) orderable:	1 chip
Primary Cache:	32 KB I + 32 KB D on chip per core
Secondary Cache:	6 MB I+D on chip per chip

Continued on next page

Software

Operating System:	Windows Vista Ultimate (64-bit)
Compiler:	Intel C++ Compiler for Intel 64, Version 10.1 Build 20080312 Package ID: w_cc_p_10.1.021 Intel Visual Fortran Compiler for Intel 64, Version 10.0 Build 20080312 Package ID: w_fc_p_10.1.021 Microsoft Visual Studio 2005 SP1
Auto Parallel:	No
File System:	NTFS

Continued on next page

SPEC CFP2006 Result

Dell Inc.

Dell Precision M6300 (Intel X9000, 2.80 GHz)

SPECfp_rate2006 = 27.4

SPECfp_rate_base2006 = 26.0

CPU2006 license: 55
Test sponsor: Dell Inc.
Tested by: Dell Inc.

Test date: Apr-2008
Hardware Availability: Mar-2008
Software Availability: Mar-2008

L3 Cache: None
Other Cache: None
Memory: 4 GB (2x2 GB 667 MHz CL5 DDR2)
Disk Subsystem: 1 x 120GB SATA 7200 RPM
Other Hardware: None

System State: Default
Base Pointers: 32/64-bit
Peak Pointers: 32/64-bit
Other Software: MicroQuill SmartHeap Library 8.1 for x64

Results Table

Benchmark	Base							Peak						
	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
410.bwaves	2	**1133**	**24.0**	1127	24.1	1133	24.0	2	**1133**	**24.0**	1127	24.1	1133	24.0
416.gamess	2	986	39.7	986	39.7	**986**	**39.7**	2	978	40.0	**978**	**40.0**	978	40.1
433.milc	2	1169	15.7	**1168**	**15.7**	1167	15.7	2	1156	15.9	1154	15.9	**1154**	**15.9**
434.zeusmp	2	593	30.7	**593**	**30.7**	592	30.7	2	575	31.7	570	31.9	**572**	**31.8**
435.gromacs	2	392	36.4	392	36.4	**392**	**36.4**	2	**391**	**36.6**	391	36.6	390	36.6
436.cactusADM	2	721	33.1	**723**	**33.1**	725	33.0	2	**692**	**34.6**	693	34.5	691	34.6
437.leslie3d	2	1130	16.6	1132	16.6	**1131**	**16.6**	2	**1138**	**16.5**	1138	16.5	1139	16.5
444.namd	2	551	29.1	551	29.1	**551**	**29.1**	2	539	29.8	539	29.8	**539**	**29.8**
447.dealII	2	**768**	**29.8**	771	29.7	767	29.8	2	**716**	**32.0**	716	32.0	715	32.0
450.soplex	2	866	19.3	866	19.3	**866**	**19.3**	2	**859**	**19.4**	858	19.5	860	19.4
453.povray	2	254	41.9	**254**	**42.0**	253	42.0	2	**203**	**52.3**	204	52.3	203	52.4
454.calculix	2	595	27.7	597	27.6	**596**	**27.7**	2	**392**	**42.1**	394	41.8	392	42.1
459.GemsFDTD	2	1442	14.7	**1445**	**14.7**	1449	14.6	2	**1412**	**15.0**	1411	15.0	1413	15.0
465.tonto	2	**568**	**34.7**	566	34.7	568	34.6	2	**548**	**35.9**	548	35.9	550	35.8
470.lbm	2	1997	13.8	**1996**	**13.8**	1996	13.8	2	2024	13.6	2021	13.6	**2024**	**13.6**
481.wrf	2	776	28.8	**780**	**28.6**	781	28.6	2	776	28.8	**780**	**28.6**	781	28.6
482.sphinx3	2	**1251**	**31.2**	1252	31.1	1247	31.3	2	**1230**	**31.7**	1229	31.7	1232	31.6

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

Base Compiler Invocation

C benchmarks:
```
icl -Qstd=c99
```

C++ benchmarks:
```
icl
```

Fortran benchmarks:
```
ifort
```

Benchmarks using both Fortran and C:
```
icl -Qstd=c99 ifort
```

Base Portability Flags

410.bwaves: -DSPEC_CPU_P64
416.gamess: -DSPEC_CPU_P64
433.milc: -DSPEC_CPU_P64
434.zeusmp: -DSPEC_CPU_P64
435.gromacs: -DSPEC_CPU_P64
436.cactusADM: -DSPEC_CPU_P64 -Qlowercase /assume:underscore
437.leslie3d: -DSPEC_CPU_P64
444.namd: -DSPEC_CPU_P64 /TP
447.dealII: -DSPEC_CPU_P64 -DDEAL_II_MEMBER_VAR_SPECIALIZATION_BUG
450.soplex: -DSPEC_CPU_P64
453.povray: -DSPEC_CPU_P64 -DSPEC_CPU_WINDOWS_ICL
454.calculix: -DSPEC_CPU_P64 -DSPEC_CPU_NOZMODIFIER -Qlowercase
459.GemsFDTD: -DSPEC_CPU_P64
465.tonto: -DSPEC_CPU_P64
470.lbm: -DSPEC_CPU_P64
481.wrf: -DSPEC_CPU_P64 -DSPEC_CPU_WINDOWS_ICL
482.sphinx3: -DSPEC_CPU_P64

Base Optimization Flags

C benchmarks:
-fast -Qauto-ilp32 /F1000000000 -link /FORCE:MULTIPLE

C++ benchmarks:
-fast -Qauto-ilp32 -Qcxx_features /F1000000000 shlw64m.lib
-link /FORCE:MULTIPLE

Fortran benchmarks:
-fast -Qauto-ilp32 /F1000000000 -link /FORCE:MULTIPLE

Benchmarks using both Fortran and C:
-fast -Qauto-ilp32 /F1000000000 -link /FORCE:MULTIPLE

Peak Compiler Invocation

C benchmarks:
icl -Qstd=c99

C++ benchmarks:
icl

Fortran benchmarks:
ifort

Benchmarks using both Fortran and C:
icl -Qstd=c99 ifort

Peak Portability Flags

Same as Base Portability Flags

Peak Optimization Flags

C benchmarks:

433.milc: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Qunroll2 -Oa /F1000000000 -link /FORCE:MULTIPLE

470.lbm: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Qunroll2 -Qscalar-rep- -Qprefetch /F1000000000 -link /FORCE:MULTIPLE

482.sphinx3: -fast -Qauto-ilp32 -Qunroll2 /F1000000000 -link /FORCE:MULTIPLE

C++ benchmarks:

444.namd: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Oa -Qcxx_features /F1000000000 shlw64m.lib -link /FORCE:MULTIPLE

447.dealII: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Qunroll2 -Qprefetch -Qcxx_features /F1000000000 shlw64m.lib -link /FORCE:MULTIPLE

450.soplex: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Qcxx_features /F1000000000 shlw64m.lib -link /FORCE:MULTIPLE

453.povray: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Qunroll4 -Qansi-alias -Qcxx_features /F1000000000 shlw64m.lib -link /FORCE:MULTIPLE

Fortran benchmarks:

410.bwaves: basepeak = yes

416.gamess: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Qunroll2 -Ob0 -Qansi-alias -Qscalar-rep- /F1000000000 -link /FORCE:MULTIPLE

434.zeusmp: -Qprof_gen(pass 1) -Qprof_use(pass 2) -QxT -O2 -Qprec-div- -Qunroll0 -Qscalar-rep- /F1000000000 -link /FORCE:MULTIPLE

437.leslie3d: -Qprof_gen(pass 1) -Qprof_use(pass 2) -fast -Qauto-ilp32 -Qprefetch /F1000000000 -link /FORCE:MULTIPLE

Continued on next page

Peak Optimization Flags (Continued)

459.GemsFDTD: `-Qprof_gen`(pass 1) `-Qprof_use`(pass 2) `-fast -Qauto-ilp32 -Qunroll2 -Ob0 -Qprefetch /F1000000000 -link /FORCE:MULTIPLE`

465.tonto: `-Qprof_gen`(pass 1) `-Qprof_use`(pass 2) `-fast -Qauto-ilp32 -Qunroll4 -Qauto /F1000000000 -link /FORCE:MULTIPLE`

Benchmarks using both Fortran and C:

435.gromacs: `-Qprof_gen`(pass 1) `-Qprof_use`(pass 2) `-fast -Qauto-ilp32 -Oa -Qprefetch /F1000000000 -link /FORCE:MULTIPLE`

436.cactusADM: `-Qprof_gen`(pass 1) `-Qprof_use`(pass 2) `-fast -Qauto-ilp32 -Qunroll2 -Qprefetch /F1000000000 -link /FORCE:MULTIPLE`

454.calculix: `-fast -Qauto-ilp32 -Qunroll-aggressive /F1000000000 -link /FORCE:MULTIPLE`

481.wrf: `basepeak = yes`

The flags file that was used to format this result can be browsed at
`http://www.spec.org/cpu2006/flags/dell.ic10.1.windows.flags.20090714.html`

You can also download the XML flags source by saving the following link:
`http://www.spec.org/cpu2006/flags/dell.ic10.1.windows.flags.20090714.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 17:11:27 2014 by SPEC CPU2006 PS/PDF formatter v6932.
Originally published on 13 May 2008.