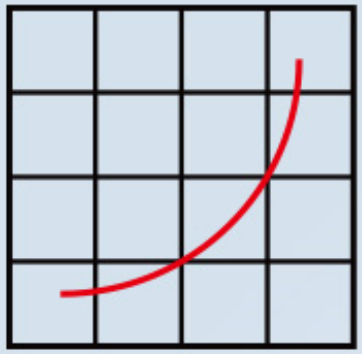


SPEC 2016亚洲峰会
SPEC 2016 ASIA SUMMIT

SPEC CPU2006: An Overview and Round Table Discussion

Jeff Reilly

SPEC CPU Committee Chair

Purpose and agenda

Purpose: Provide an overview of SPEC CPU2006 and have a round table forum/question and answer session.

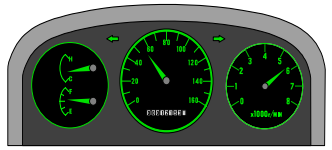
Agenda:

- **Review: What are benchmarks and what makes a good benchmark?**
- **A brief overview of SPEC CPU2006**
- **Question and answers/discussion**

Background: What are benchmarks?

A standard by which something can be measured or judged.

Specific to computers: A program (or suite of programs) used to assess the performance characteristics of one or more computer systems



Examples:

- Fuel efficiency/kilometers per gallon
- Grades in school
- SPEC CPU2006

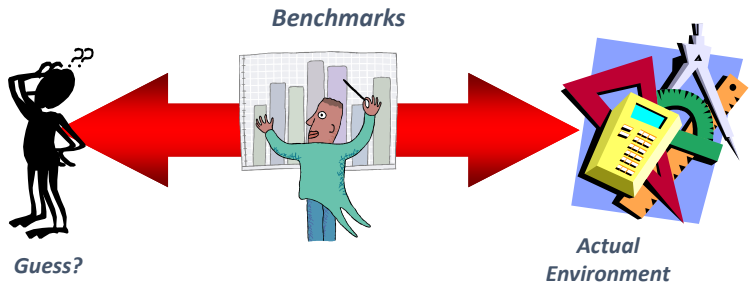
Benchmarks allow for comparisons between two or more items.

Background: Why use benchmarks?

Benchmarks provide information somewhere between “no information/guess” and “actual environment”.

In a perfect world, you would measure exactly want you want to evaluate but the following are issues...

- Time
- Money
- Available data
- Economy Of Scale



“Benchmarks provide successive approximations to reality”
This requires understanding both of the benchmark AND your needs!

A good benchmark is…

- Relevant
- Reproducible
- Fair
- Usable
- Well-defined/verifiable
- Recognized
- Simple
- Portable/Scalable

Not all benchmarks (including some popular ones) have all of these characteristics! Always considered this list…

What were SPEC's goals with SPEC CPU 2006?

"SPEC designed this suite to provide a comparative measure of compute-intensive performance across the widest practical range of hardware using workloads developed from real user applications."

Through the SPEC website, provide a resource to the performance community with reviewed results and other content about the benchmark.

What is SPEC CPU2006?

A benchmark suite, composed of real applications, for comparing the compute capabilities of a given system.

Emphasizes: Processor(s), memory, compiler

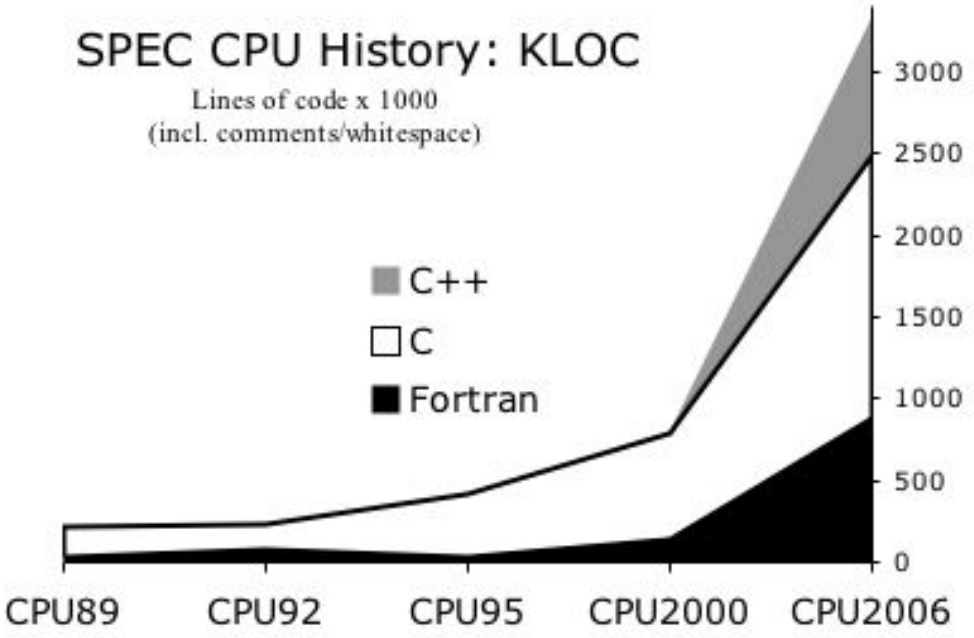
8 Metrics	Integer/ Floating Point	Speed For each benchmark, time to complete one invocation Rate Throughput for running user selected number of concurrent copies	Baseline All programs of a given language compiled the same way; no FDO Peak Each program may be compiled differently
SPECint2006	Integer	Speed	Peak
SPECint_base2006	Integer	Speed	Baseline
SPECint_rate2006	Integer	Rate	Peak
SPECint_rate_base2006	Integer	Rate	Baseline
SPECfp2006	Floating point	Speed	Peak
SPECfp_base2006	Floating point	Speed	Baseline
SPECfp_rate2006	Floating point	Rate	Peak
SPECfp_rate_base2006	Floating point	Rate	Baseline

What is SPEC CPU2006?

CINT2006: 12 benchmarks; 9 in C and 3 in C++.

CFP2006: 17 benchmarks; 3 in C, 4 in C++, 6 in FORTRAN and 4 in a mix of C and FORTRAN.

Based on real applications. Full details can be seen at www.spec.org.



Integer Workloads

CINT2006 (Integer Component of SPEC CPU2006):			
Benchmark	**Language**	**Application Area**	**Brief Description**
400.perlbench	C	Programming Language	Derived from Perl V5.8.7. The workload includes SpamAssassin, MHonArc (an email indexer), and specdiff (SPEC's tool that checks benchmark outputs).
401.bzip2	C	Compression	Julian Seward's bzip2 version 1.0.3, modified to do most work in memory, rather than doing I/O.
403.gcc	C	C Compiler	Based on gcc Version 3.2, generates code for Opteron.
429.mcf	C	Combinatorial Optimization	Vehicle scheduling. Uses a network simplex algorithm (which is also used in commercial products) to schedule public transport.
445.gobmk	C	Artificial Intelligence: Go	Plays the game of Go, a simply described but deeply complex game.
456.hmmer	C	Search Gene Sequence	Protein sequence analysis using profile hidden Markov models (profile HMMs)
458.sjeng	C	Artificial Intelligence: chess	A highly-ranked chess program that also plays several chess variants.
462.libquantum	C	Physics / Quantum Computing	Simulates a quantum computer, running Shor's polynomial-time factorization algorithm.
464.h264ref	C	Video Compression	A reference implementation of H.264/AVC, encodes a videostream using 2 parameter sets. The H.264/AVC standard is expected to replace MPEG2
471.omnetpp	C++	Discrete Event Simulation	Uses the OMNet++ discrete event simulator to model a large Ethernet campus network.
473.astar	C++	Path-finding Algorithms	Pathfinding library for 2D maps, including the well known A* algorithm.
483.xalancbmk	C++	XML Processing	A modified version of Xalan-C++, which transforms XML documents to other document types.

Floating Point Workloads

CFP2006 (Floating Point Component of SPEC CPU2006):			
Benchmark	**Language**	**Application Area**	**Brief Description**
410.bwaves	Fortran	Fluid Dynamics	Computes 3D transonic transient laminar viscous flow.
416.gamess	Fortran	Quantum Chemistry.	Gamess implements a wide range of quantum chemical computations. For the SPEC workload, self-consistent field calculations are performed using the Restricted Hartree Fock method, Restricted open-shell Hartree-Fock, and Multi-Configuration Self-Consistent Field
433.milc	C	Physics / Quantum Chromodynamics	A gauge field generating program for lattice gauge theory programs with dynamical quarks.
434.zeusmp	Fortran	Physics / CFD	ZEUS-MP is a computational fluid dynamics code developed at the Laboratory for Computational Astrophysics (NCSA, University of Illinois at Urbana-Champaign) for the simulation of astrophysical phenomena.
435.gromacs	C, Fortran	Biochemistry / Molecular Dynamics	Molecular dynamics, i.e. simulate Newtonian equations of motion for hundreds to millions of particles. The test case simulates protein Lysozyme in a solution.
436.cactusADM	C, Fortran	Physics / General Relativity	Solves the Einstein evolution equations using a staggered-leapfrog numerical method
437.leslie3d	Fortran	Fluid Dynamics	Computational Fluid Dynamics (CFD) using Large-Eddy Simulations with Linear-Eddy Model in 3D. Uses the MacCormack Predictor-Corrector time integration scheme.
444.namd	C++	Biology / Molecular Dynamics	Simulates large biomolecular systems. The test case has 92,224 atoms of apolipoprotein A-I.
447.dealII	C++	Finite Element Analysis	deal.II is a C++ program library targeted at adaptive finite elements and error estimation. The testcase solves a Helmholtz-type equation with non-constant coefficients.
450.soplex	C++	Linear Programming, Optimization	Solves a linear program using a simplex algorithm and sparse linear algebra. Test cases include railroad planning and military airlift models.
453.povray	C++	Image Ray-tracing	Image rendering. The testcase is a 1280x1024 anti-aliased image of a landscape with some abstract objects with textures using a Perlin noise function.
454.calculix	C, Fortran	Structural Mechanics	Finite element code for linear and nonlinear 3D structural applications. Uses the SPOOLES solver library.
459.GemsFDTD	Fortran	Computational Electromagnetics	Solves the Maxwell equations in 3D using the finite-difference time-domain (FDTD) method.
465.tonto	Fortran	Quantum Chemistry	An open source quantum chemistry package, using an object-oriented design in Fortran 95. The test case places a constraint on a molecular Hartree-Fock wavefunction calculation to better match experimental X-ray diffraction data.
470.lbm	C	Fluid Dynamics	Implements the "Lattice-Boltzmann Method" to simulate incompressible fluids in 3D
481.wrf	C, Fortran	Weather	Weather modeling from scales of meters to thousands of kilometers. The test case is from a 30km area over 2 days.
482.sphinx3	C	Speech recognition	A widely-known speech recognition system from Carnegie Mellon University

How can you use SPEC CPU2006?

Run it on your own machine.

- Order a license from SPEC.
- Compile and run on your own equipment.
- If you wish, submit to SPEC for review and publication. (https://www.spec.org/osg/submitting_results.html)

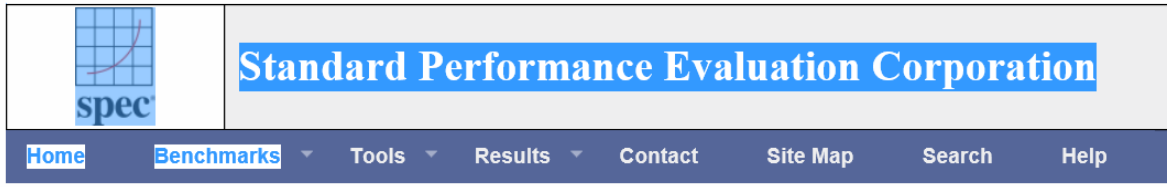
Ask your platform/compiler supplier to provide results.

Use/compare the results on the SPEC website.

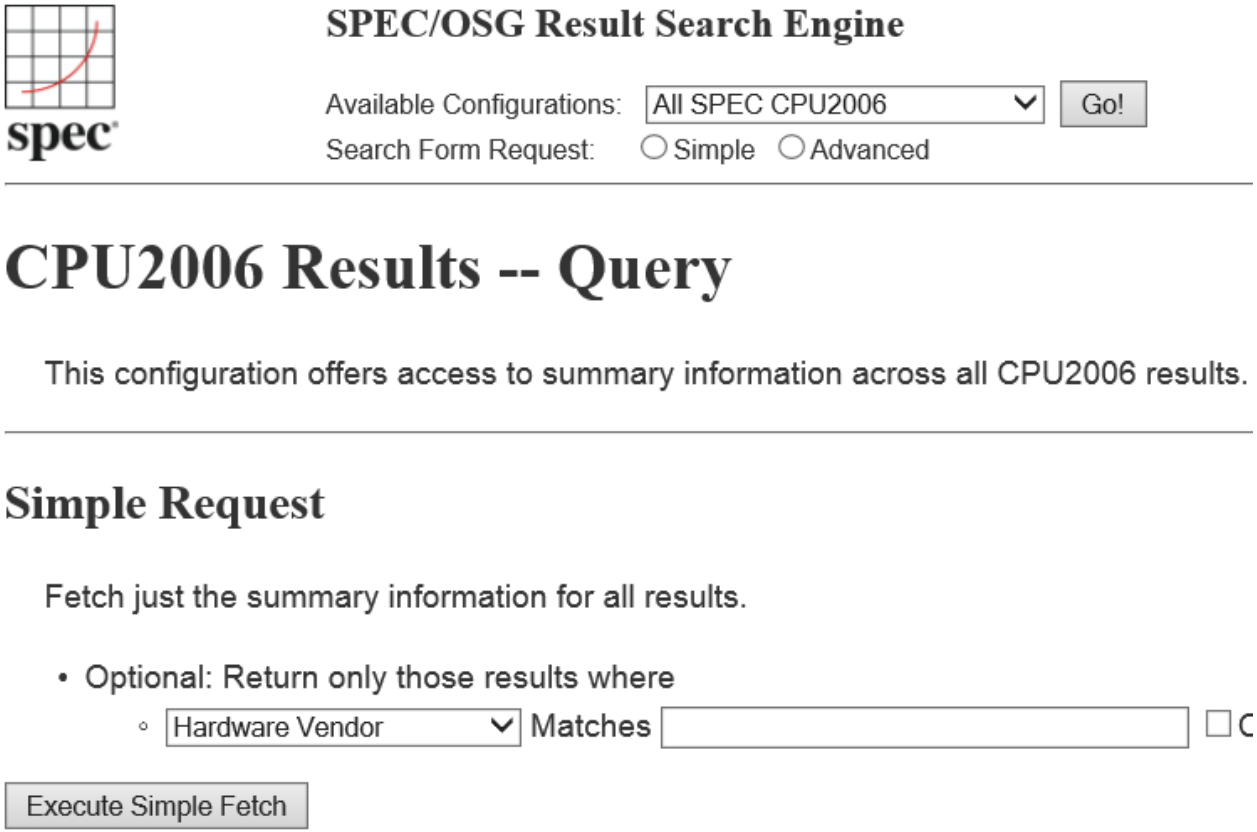
- The website has a database of results (over 30,000) (https://www.spec.org/cgi-bin/osgresults?conf=cpu2006)

How to search for CPU2006 results

Go to www.spec.org and in the top menu, click on "results" and select CPU2006



From there you can look at results by time or click on the link for "CPU2006 Search Form"



Example of CPU2006 result

SPEC® CINT2006 Result

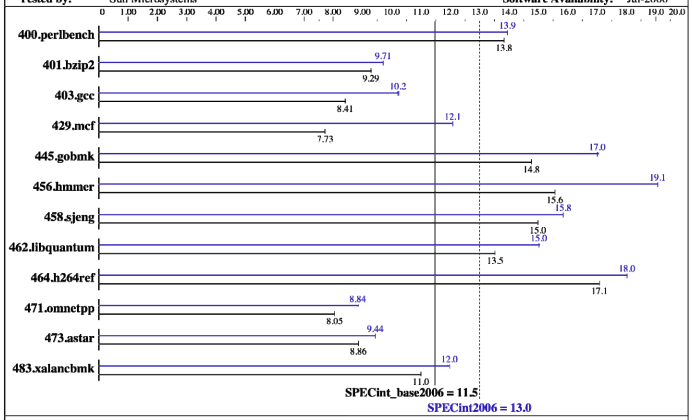
Copyright 2006-2009 Standard Performance Evaluation Corporation

Sun Microsystems
Sun Fire X2100

SPECint®2006 = 13.0
SPECint_base2006 = 11.5

CPU2006 license: 6
Test sponsor: Sun Microsystems
Tested by: Sun Microsystems

Test date: Apr-2006
Hardware Availability: May-2006
Software Availability: Jul-2006



Hardware

CPU Name:	AMD Opteron 156
CPU Characteristics:	
CPU MHz:	3000
FPU:	Integrated
CPU(s) enabled:	1 core, 1 chip, 1 core/chip
CPU(s) orderable:	1 chip
Primary Cache:	64 KB I + 64 KB D on chip per chip
Secondary Cache:	1 MB I+D on chip per chip
L3 Cache:	None
Other Cache:	None
Memory:	4 GB (4x1 GB, PC3200 CL3 ECC)
Disk Subsystem:	SATA,80GB,10K RPM
Other Hardware:	None

Software

Operating System:	Solaris 10 1/06
Compiler:	Sun Studio 11 with patch 120759-06
Auto Parallel:	No
File System:	ufs
System State:	Default
Base Pointers:	64-bit
Peak Pointers:	32/64-bit
Other Software:	None

SPEC CINT2006 Result

Copyright 2006-2009 Standard Performance Evaluation Corporation

Sun Microsystems
Sun Fire X2100

SPECint2006 = 13.0
SPECint_base2006 = 11.5

CPU2006 license: 6
Test sponsor: Sun Microsystems
Tested by: Sun Microsystems

Test date: Apr-2006
Hardware Availability: May-2006
Software Availability: Jul-2006

Results Table

Benchmark	Base Seconds	Base Ratio	Base Seconds	Base Ratio	Base Seconds	Base Ratio	Peak Seconds	Peak Ratio	Peak Seconds	Peak Ratio	Peak Seconds	Peak Ratio
400.perlbench	706	13.8	707	13.8	**707**	**13.8**	701	13.9	701	13.9	**701**	**13.9**
401.bzip2	1037	9.31	1039	9.29	**1039**	**9.29**	994	9.71	994	9.71	**994**	**9.71**
403.gcc	957	8.41	955	8.43	**957**	**8.41**	786	10.2	790	10.2	**789**	**10.2**
429.mcf	1180	7.73	1182	7.72	**1180**	**7.73**	755	12.1	755	12.1	**755**	**12.1**
445.gobmk	**711**	**14.8**	710	14.8	711	14.8	615	17.1	617	17.0	**617**	**17.0**
456.hmmer	600	15.6	**600**	**15.6**	600	15.6	489	19.1	**489**	**19.1**	489	19.1
458.sjeng	**808**	**15.0**	808	15.0	808	15.0	**764**	**15.8**	764	15.8	764	15.8
462.libquantum	**1532**	**13.5**	1532	13.5	1532	13.5	1377	15.0	1381	15.0	**1380**	**15.0**
464.h264ref	1296	17.1	1295	17.1	**1295**	**17.1**	1228	18.0	**1228**	**18.0**	1228	18.0
471.omnetpp	776	8.05	776	8.05	**776**	**8.05**	707	8.84	**707**	**8.84**	705	8.87
473.astar	792	8.87	792	8.86	**792**	**8.86**	744	9.44	**744**	**9.44**	744	9.44
483.xalancbmk	627	11.0	**628**	**11.0**	628	11.0	576	12.0	579	11.9	**577**	**12.0**

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

Operating System Notes

```
ulimit -s 131072  (shell): increases stack


/etc/system parameters
   tune_t_fsflushr=1
      Controls how many seconds elapse between runs of the
      page flush daemon, fsflush.
   autoup=900
      Causes pages older than the listed number of seconds to
      be written by fsflush.
```

Platform Notes

```
Default BIOS settings were used.
```

Base Compiler Invocation

C benchmarks:
cc

Continued on next page

SPEC CINT2006 Result

Copyright 2006-2009 Standard Performance Evaluation Corporation

Sun Microsystems
Sun Fire X2100

SPECint2006 = 13.0
SPECint_base2006 = 11.5

CPU2006 license: 6
Test sponsor: Sun Microsystems
Tested by: Sun Microsystems

Test date: Apr-2006
Hardware Availability: May-2006
Software Availability: Jul-2006

Base Compiler Invocation (Continued)

C++ benchmarks:
cc

Base Portability Flags

400.perlbench: -DSPEC_CPU_LP64 -DSPEC_CPU_SOLARIS_X64
401.bzip2: -DSPEC_CPU_LP64
403.gcc: -DSPEC_CPU_LP64 -DSPEC_CPU_SOLARIS
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_SOLARIS
464.h264ref: -DSPEC_CPU_LP64
471.omnetpp: -DSPEC_CPU_LP64
473.astar: -DSPEC_CPU_LP64
483.xalancbmk: -DSPEC_CPU_LP64 -DSPEC_CPU_SOLARIS

Base Optimization Flags

C benchmarks:
-fast -xipo=2 -xarch=amd64a

C++ benchmarks:
-fast -xipo=2 -xarch=amd64a -library=stlport4

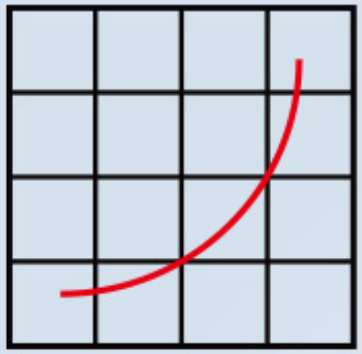
Base Other Flags

C benchmarks:
-V

C++ benchmarks:
-verbose=version

Peak Compiler Invocation

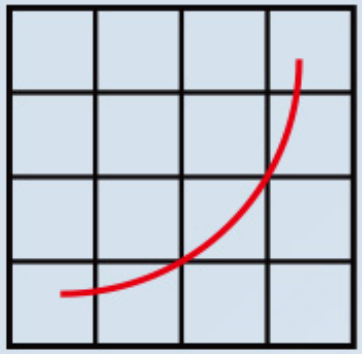
Same as Base Compiler Invocation



spec®

SPEC 2016亚洲峰会
SPEC 2016 ASIA SUMMIT

Q&A



SPEC 2016亚洲峰会
SPEC 2016 ASIA SUMMIT

Thank you!

info@spec.org
www.spec.org