Performance Metrics: SPEC Gigaflops vs. Linpack Gigaflops

Tom Elken, *SGI*
Manager, Performance Engineering
& Math Libraries

May 24, 2001

Overview

- System Performance Metrics
  - CPU Mhz
  - Peak Flops
  - SPEC CPU2000 Speed and Rate
  - SPEC CPU2000 Gflops
  - Linpack Gflops
  - System efficiency measure
  - Summary

Simple Performance Metrics

- CPU Mhz
  - Most popular metric in the press
  - MHz: A high frequency chip that can't do very much in a given cycle is like a runner with short legs
- Peak Flops
  - MHz x (Max # of Flops per cycle) / (repeat rate for those flops)
  - Applications are like running on a rocky, wavy road; having a long stride won't necessarily ensure a fast pace.
- Bus bandwidth
  - Having a non-scalable system like a single bus is like sharing a pair of shoes during a race; only one person can run at a time.

SPEC CPU2000

- Speed vs. Rate benchmarks
  - Speed: Performance on a single CPU
  - Rate: Performance on all CPUs of a system with multiple copies of the CPU2000 benchmark applications
- SPEC CFP2000:
  - Geometric Mean of the performance of 14 floating point applications. C and Fortran.
  - Compiler generated code only!
  - Rate exercises the local memory system

SPEC CPU2000, cont.

- SPEC CINT2000
  - Geometric Mean of the performance of 10 integer applications. C and C++.
  - Compiler generated code only!
  - Does not exercise the local memory system

SPEC CPU2000 Rates

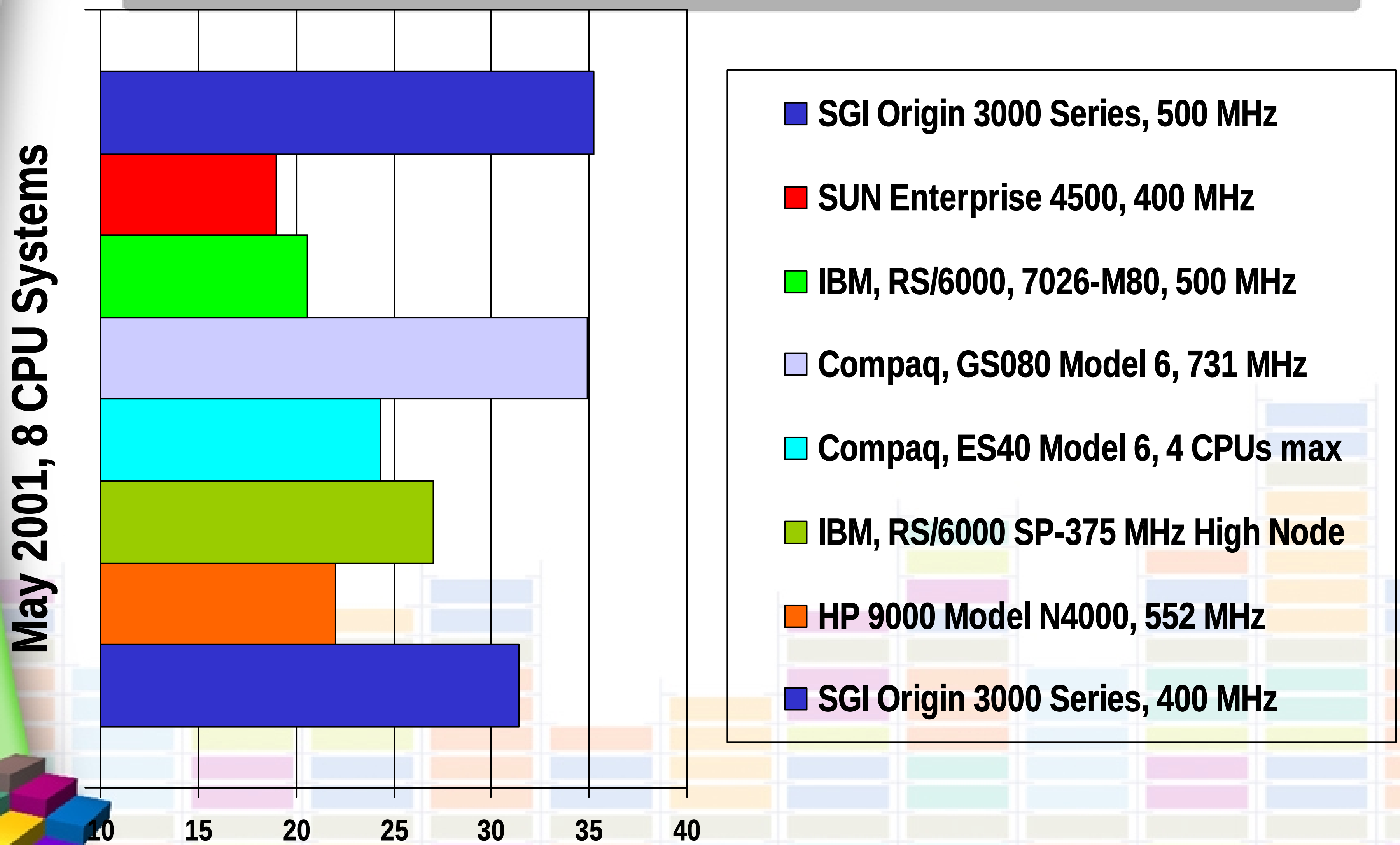
- SPEC CPU2000:
  - CPU intensive benchmarks developed from real user applications.
  - Performance depends on processor, memory system and compiler.
  - But: running a single processor benchmark on a multiprocessor system is not a good measure of full system performance.
  - CPU2000 rates: fill up the system with multiple copies of benchmarks to measure full system performance.
  - SGI Origin 3000 series does well when the full system is used.

SGI Origin 3000 Series: The Fastest Server Available

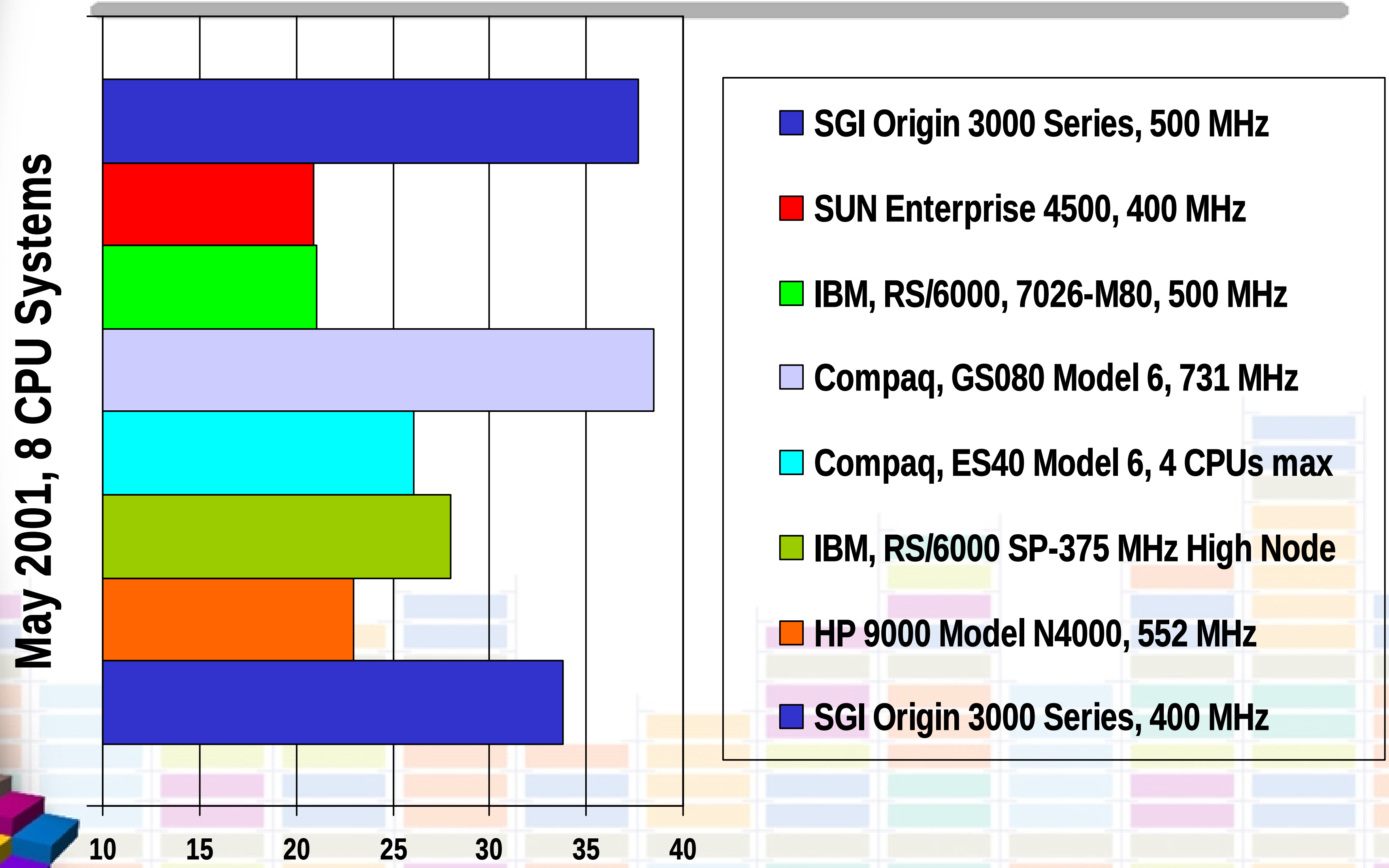
- On 8 or more CPUs:
  - The SGI Origin 3000 Series with MIPS R14000, 500 Mhz CPUs is the fastest server available*, as measured by the SPEC CPU2000 rate results (both FP and INT).
- Single CPU:
  - The SGI Origin 3000 Series with MIPS R14000, 500 Mhz CPUs is the fastest big server (8 or more CPUs) available on SPECfp2000 and SPECint2000.

* Exception: Peak SPECfp_rate2000, Compaq GS080 = 38.5; SGI Origin 3200 = 37.7

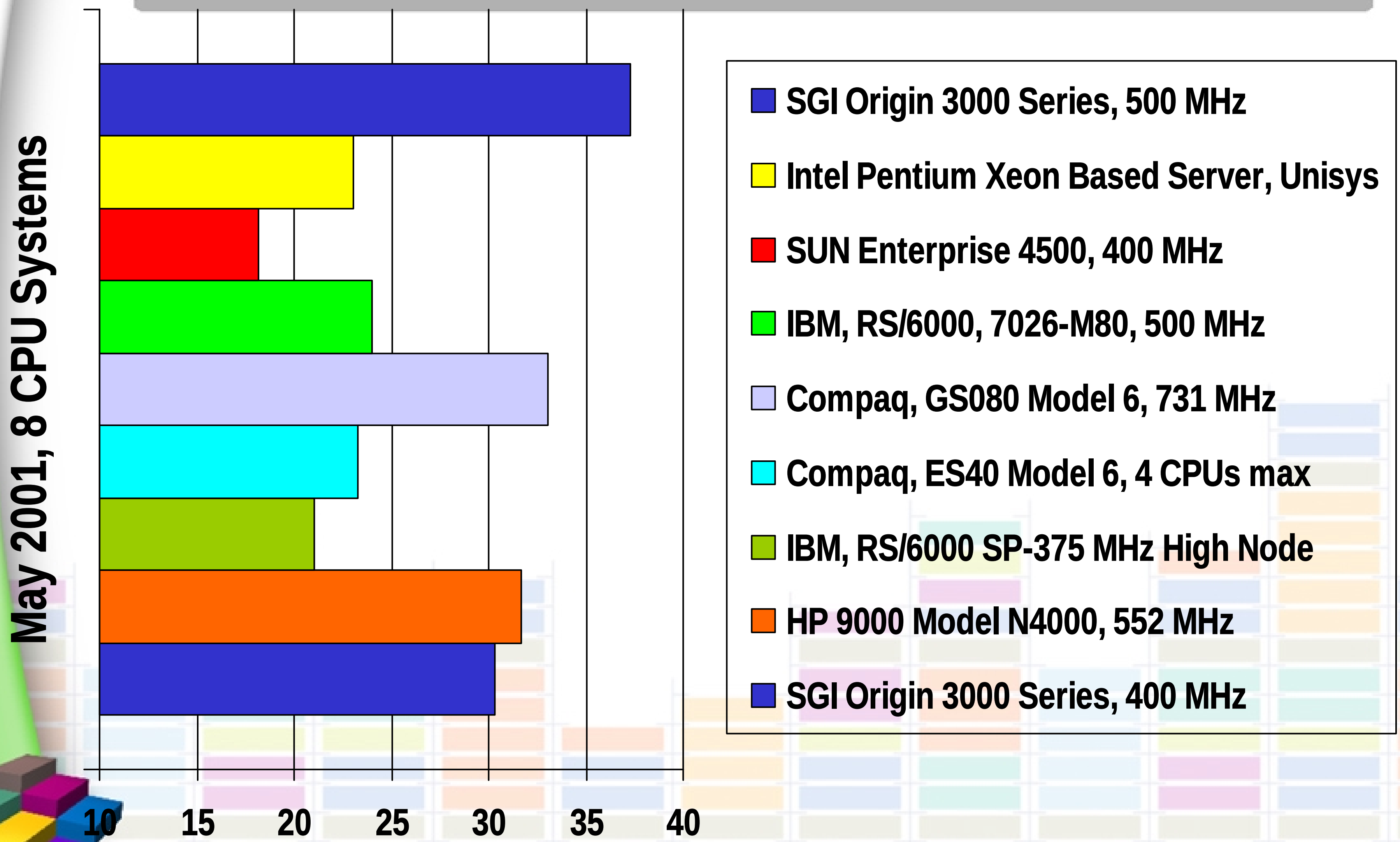
8 CPU Systems: SPEC CFP2000 Rates Base Results



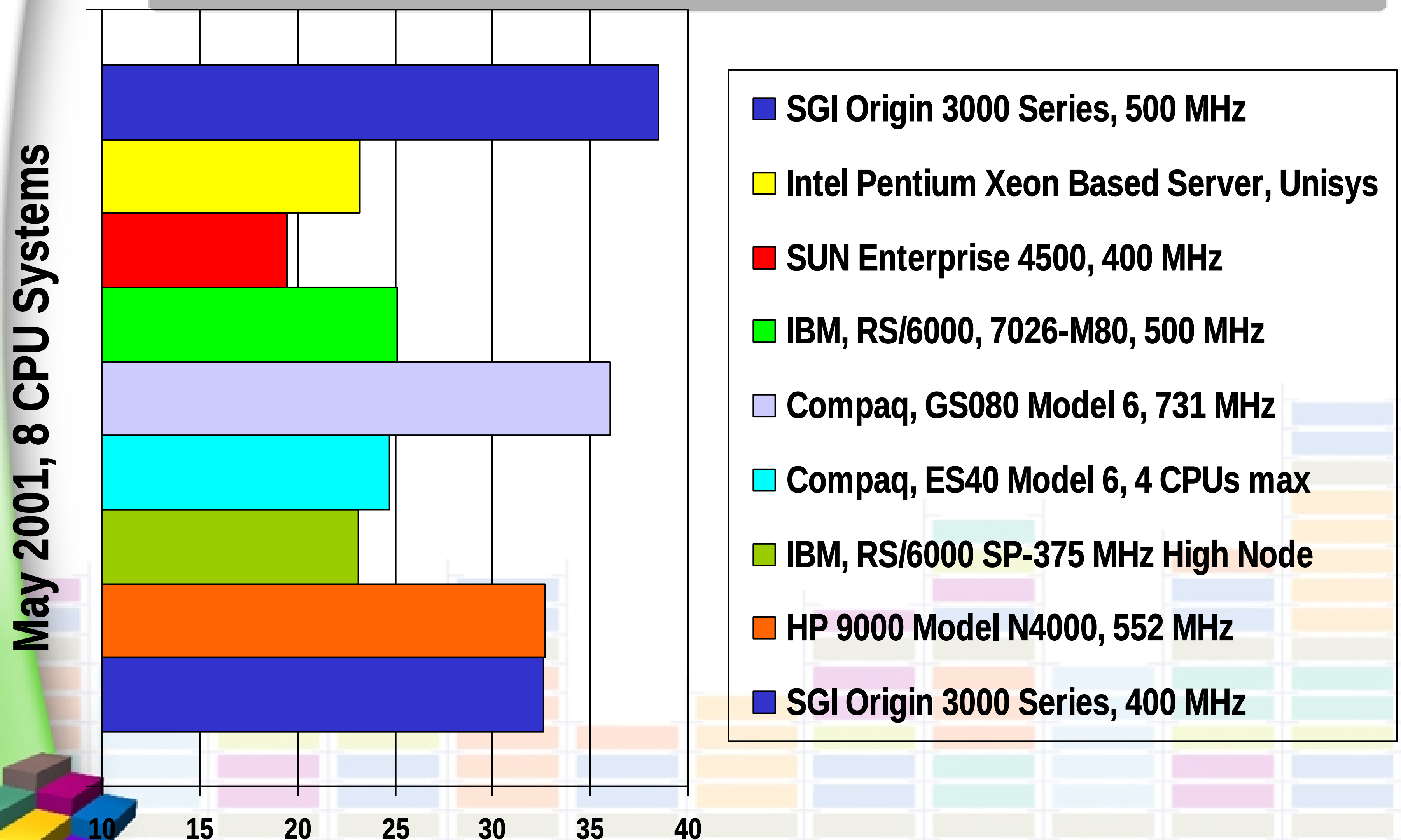
8 CPU Systems: SPEC CFP2000 Rates Peak Results



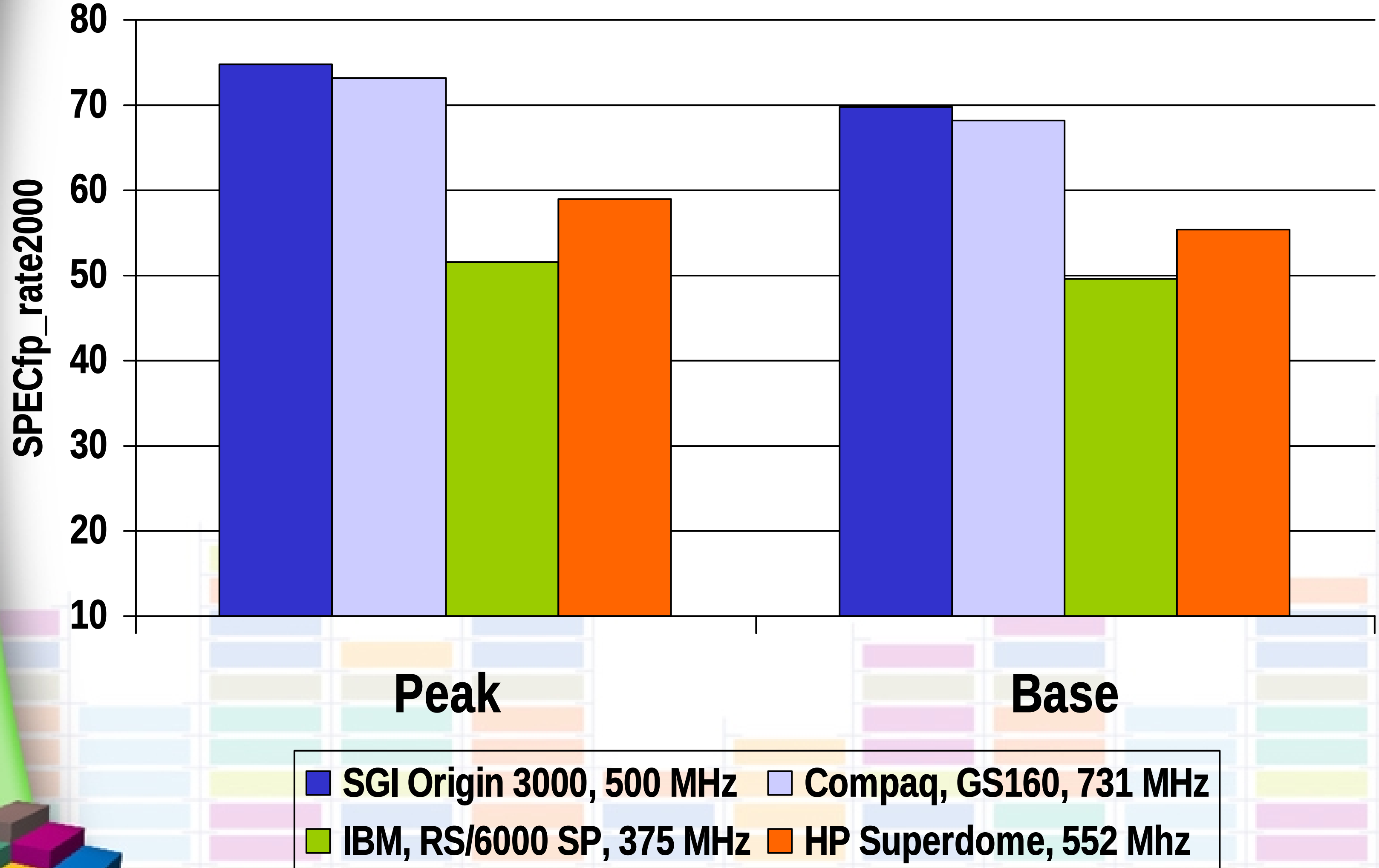
8 CPU Systems: SPEC CINT2000 Rates, Base Results



8 CPU Systems: SPEC CINT2000 Rates, Peak Results



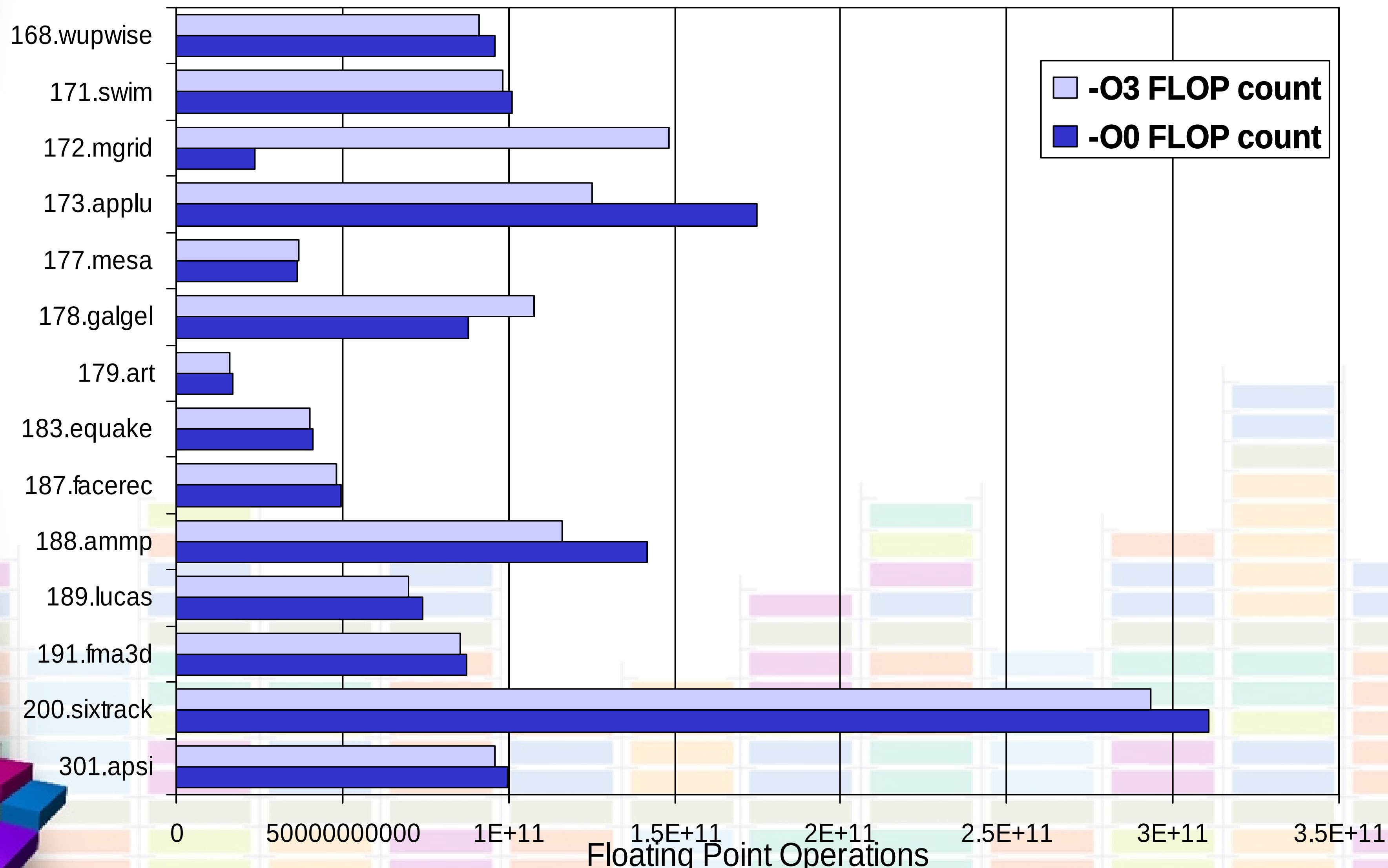
16 CPU Systems: SPECfp_rate2000



Counting FP operations in SPEC CFP2000

- Procedure:
  - Compile using
    - -O0 -TARG:madd=off (don't generate multiply-add instructions, so we can count FLOPS more accurately)
      Assumption was that "no optimization" would result in more FLOPs counted per code; and
    - -O3 -TARG:madd=off
  - Run CFP2000 codes using MIPS performance counter tool: perfex -e21 (event 21: graduated floating-point instructions).

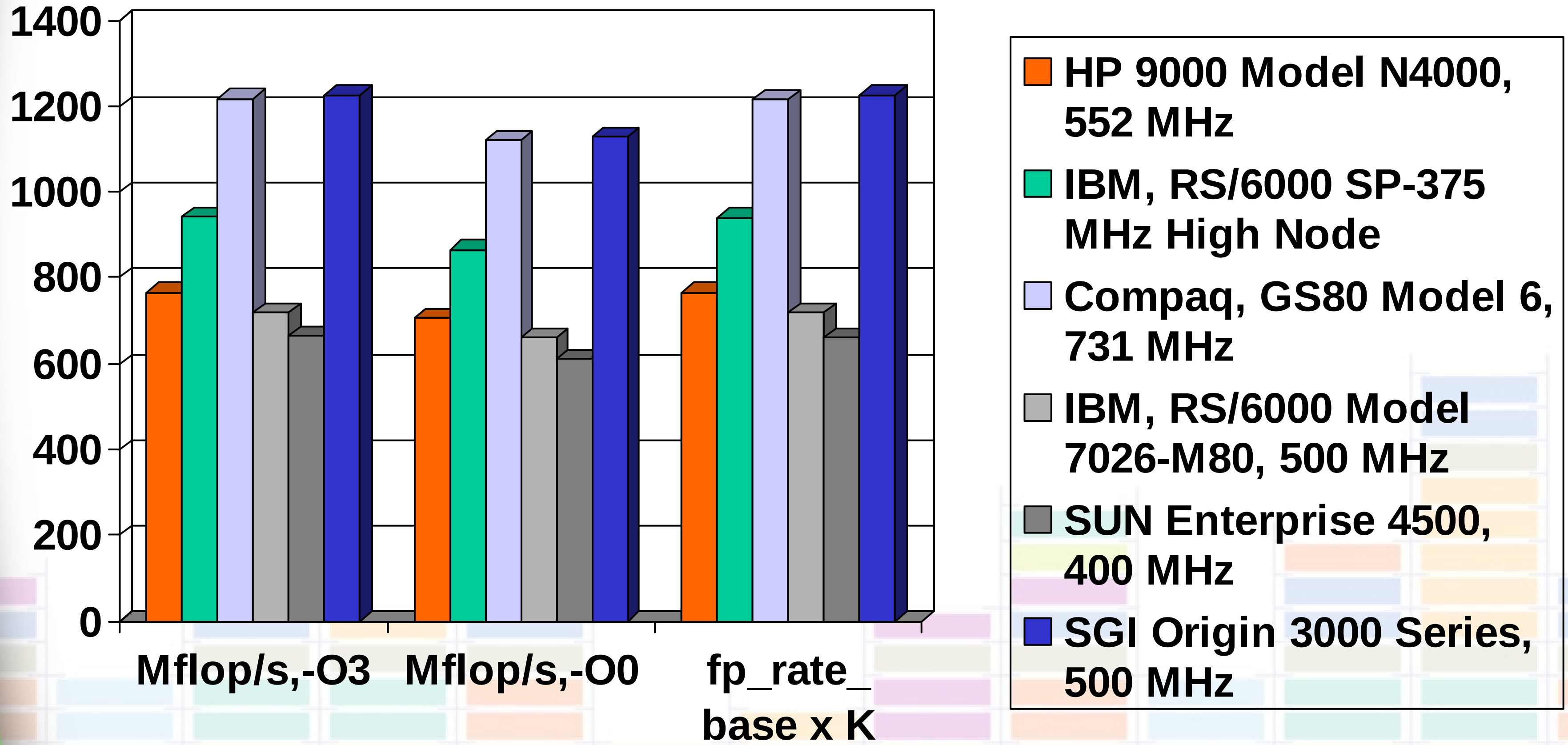
Measurement of CFP2000 flops -- MIPSpro compiler: -O3 vs. -O0



Comparing Flop counts

- Based on previous slide: SGI's perfex on -O3 compiled codes vs. -O0 compiled codes yields:
  - much larger Flop counts for "-O3" on:
    - 172.mgrid
  - smaller Flop counts for "-O3" on
    - 173.applu and 188.ammp
- Hypothesis: dividing these differing Flop counts by SPEC run times to get SPEC Mflops/sec., leads to a different-behaving metrics for systems
- Hypothesis is wrong. Mflops and SPEC marks are proportional using either counting method

Vendor comparison using three metrics (8P rate)



Both Mflop counting methods yield SPEC Mflop/s measures that are SPECfp_rate_base2000 times a constant (K); error is < 0.3%.

An 'ideal' FP operation counting method

- Run code and generate an "ideal" or "pixie" type of experiment file:

```
ssrun  -ideal a.out
```

- Run prof on the experiment file:

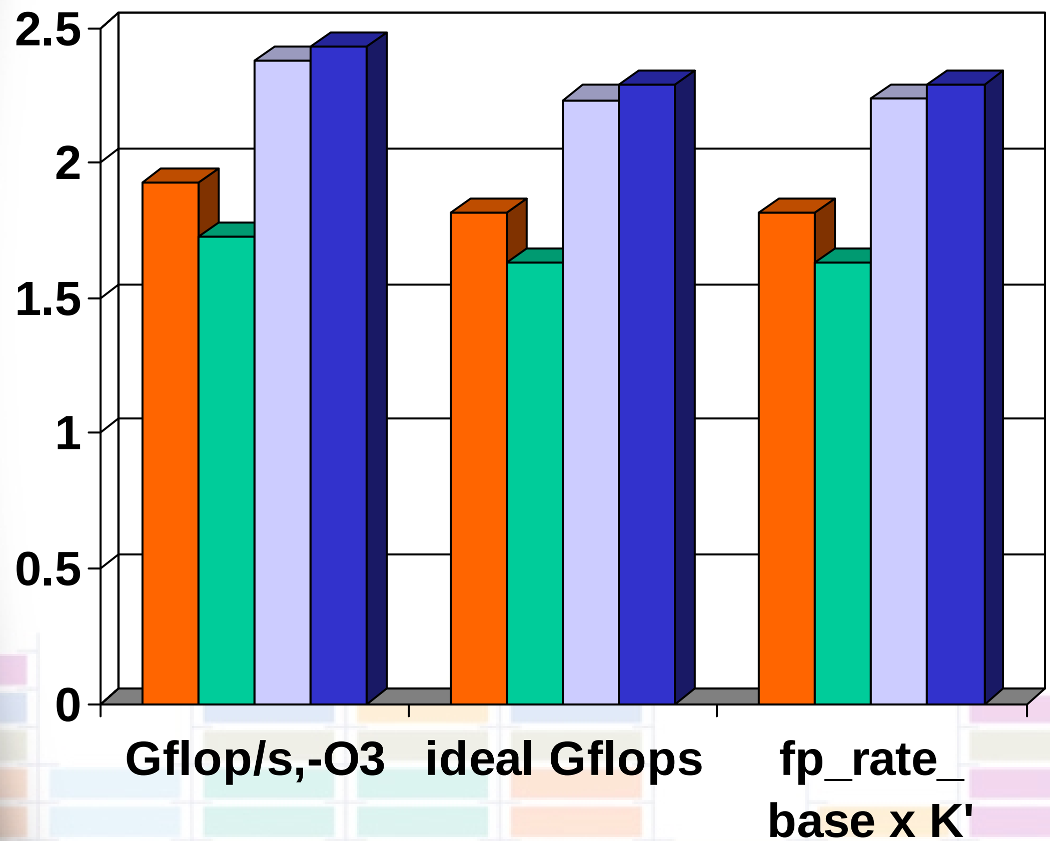
```
prof -archinfo a.out.ideal.m123456
```

- Search for "floating point operations" in the output:

```
224047269534: instructions executed
206713004712: floating point operations
 58992217886: integer operations
```

- This counting method used on the CFP2000 'base' executables also results in SPEC Mflops which are proportional to SPECfp2000 results...

Vendor comparison using three metrics (16P rate)



This is evidence that the proportionality holds at other CPU counts.

Both Mflop counting methods yield SPEC Mflop/s measures that are SPECfp_rate2000 times a constant (K'); error is < 0.3%.

SPEC Gflops: a non-controversial metric?

- SPEC Gflops is proportional to SPECfp_rate2000
- Three methods* of counting FP operations yield metrics that are directly proportional (+/- 0.3%)
- We use the "-O3" counts here since it yields larger Gflop ratings
- How would one get multi-vendor agreement on a FP counting method?   We can discuss...
- Although SPEC "base" results were shown here, the same proportionality holds with "peak" results.

* Table with all three counting method results is in Appendix

Why bother with SPEC Gigaflops?

- It might be more understandable than the SPEC CPU2000 numbers:
  - SPECfp2000 is defined as ratios:
    (100 x “Reference time”) / (run time) ,
    for 14 codes and then take the geometric mean of the ratios
  - SPEC Gflops is defined as
    (Gflop count) / (run time in secs.)
    for each code, then take the geometric mean
- SPEC code performance can be compared with peak flops ...

SPEC Gflops leads to Efficiency Metric

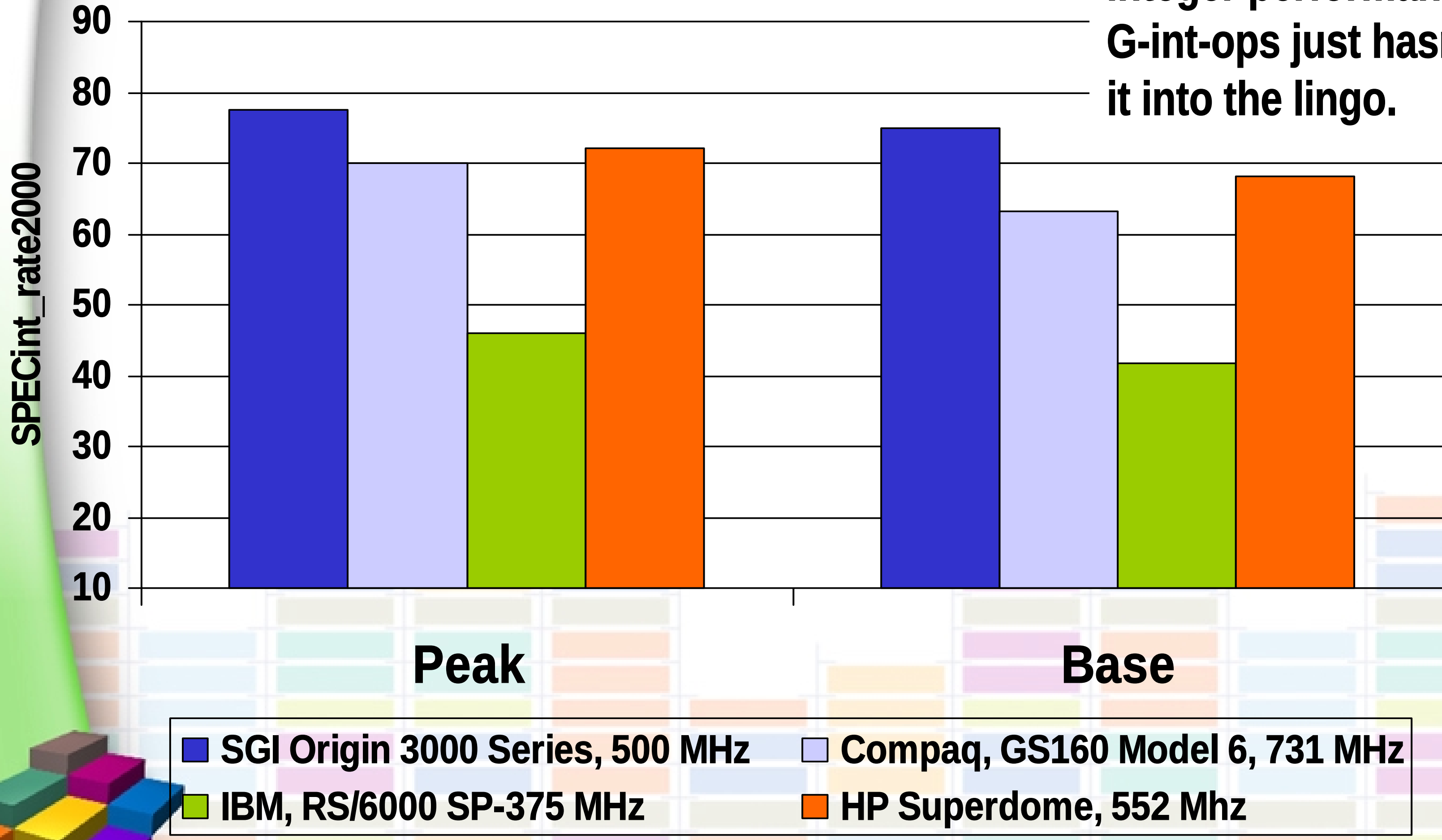
System:	SGI Origin 3000, 500 Mhz	Compaq GS160, 731 Mhz	HP Super-dome, 552 Mhz	IBM, RS6000-SP, 375 Mhz
Peak Gflops, 16 CPUs	16	23.4	35.3	24
SPECrate Base Gflops, 16 CPUs	2.43	2.38	1.93	1.73
Efficiency, 16 CPU systems	15%	10%	5%	7%

Other useful metrics?

- SPEC Gflops per $M
- SPEC Gflops per Watt
- SPEC Gflops per square foot
- etc.

16 CPU Systems: SPECint_rate2000

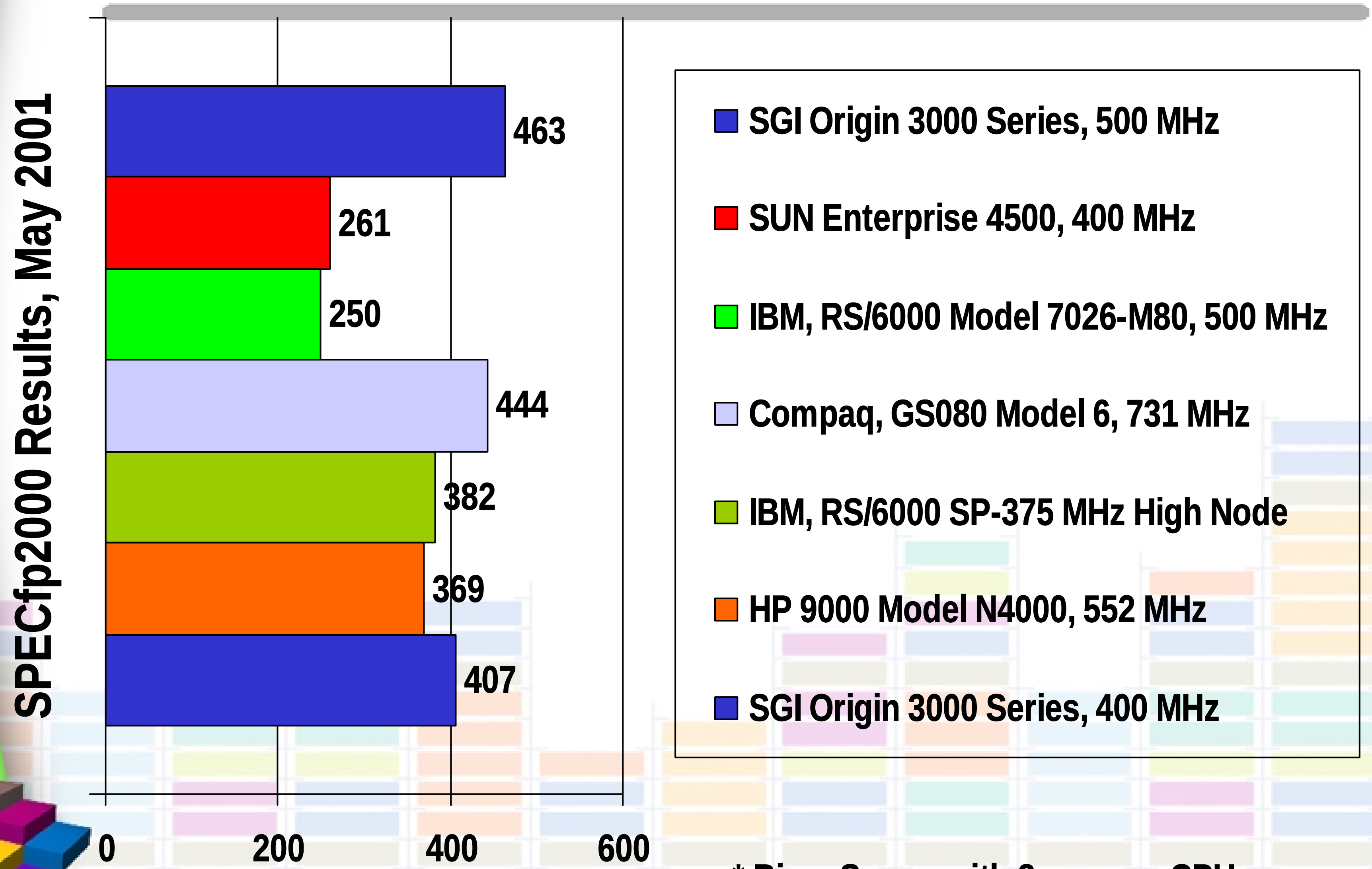
We don't want to ignore integer performance, but G-int-ops just hasn't made it into the lingo.



SPEC CPU2000 speed = single processor benchmarks

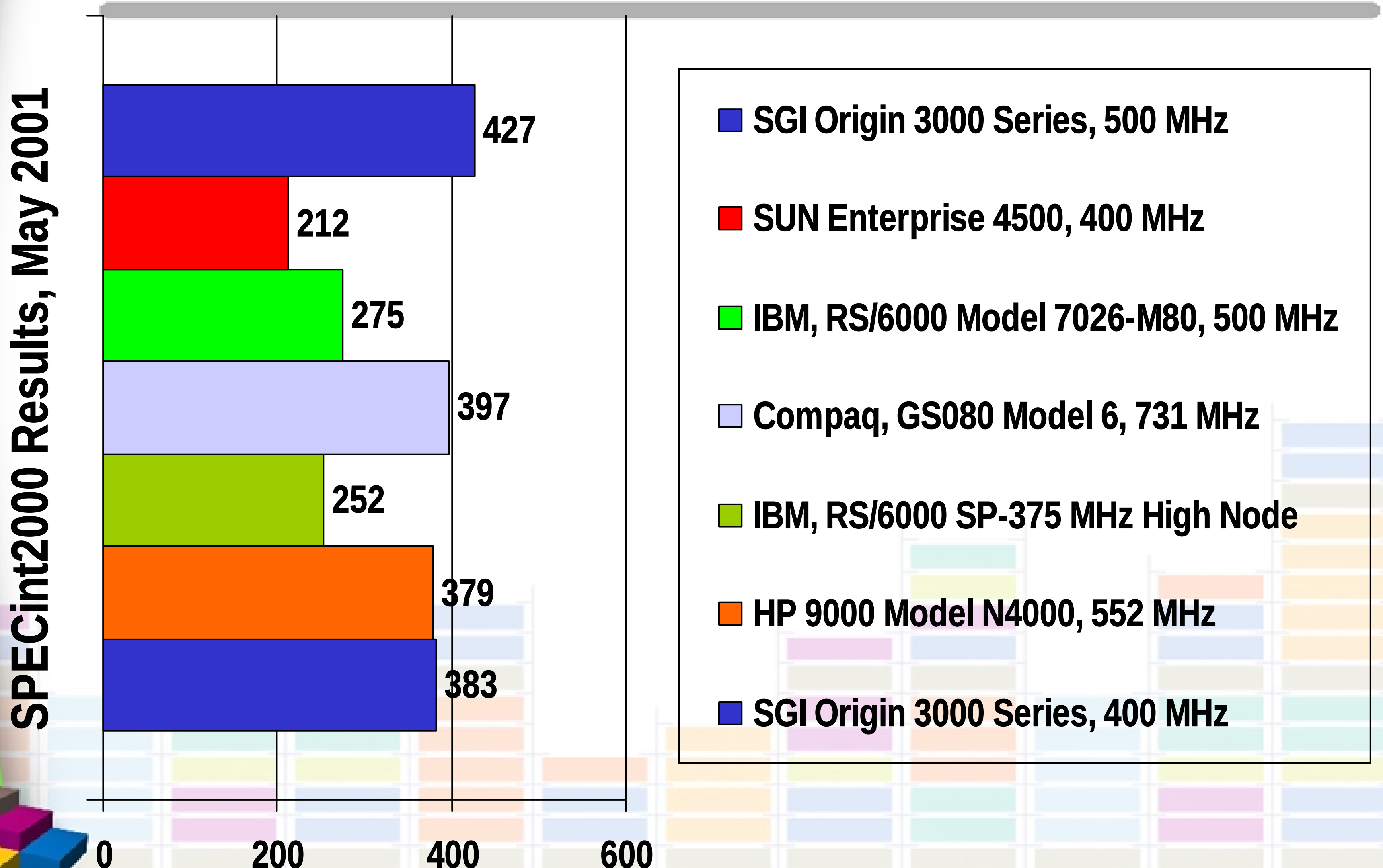
- SPEC CPU2000 speed benchmarks (SPECfp2000 and SPECint2000) do not stress the system as much as a multiple CPU benchmark
- Hardware vendors often do not support multiprocessor systems with as fast a CPU as their single processor systems
  - Intel Pentium 4 is not offered in a multiprocessor system
  - Compaq GS80 is slower than Compaq ES40
- Use SPEC CPU2000 rate benchmarks (SPECfp_rate2000 and SPECint_rate2000) over speed benchmarks as a measure of performance.
- All that said, SGI Origin 3000 Series compares favorably with other large servers on SPEC CPU2000 single processor benchmarks.

Single processor results on big* servers: SPECfp2000 Results



* Big = Server with 8 or more CPUs

Single processor results on big* servers: SPECint2000 Results



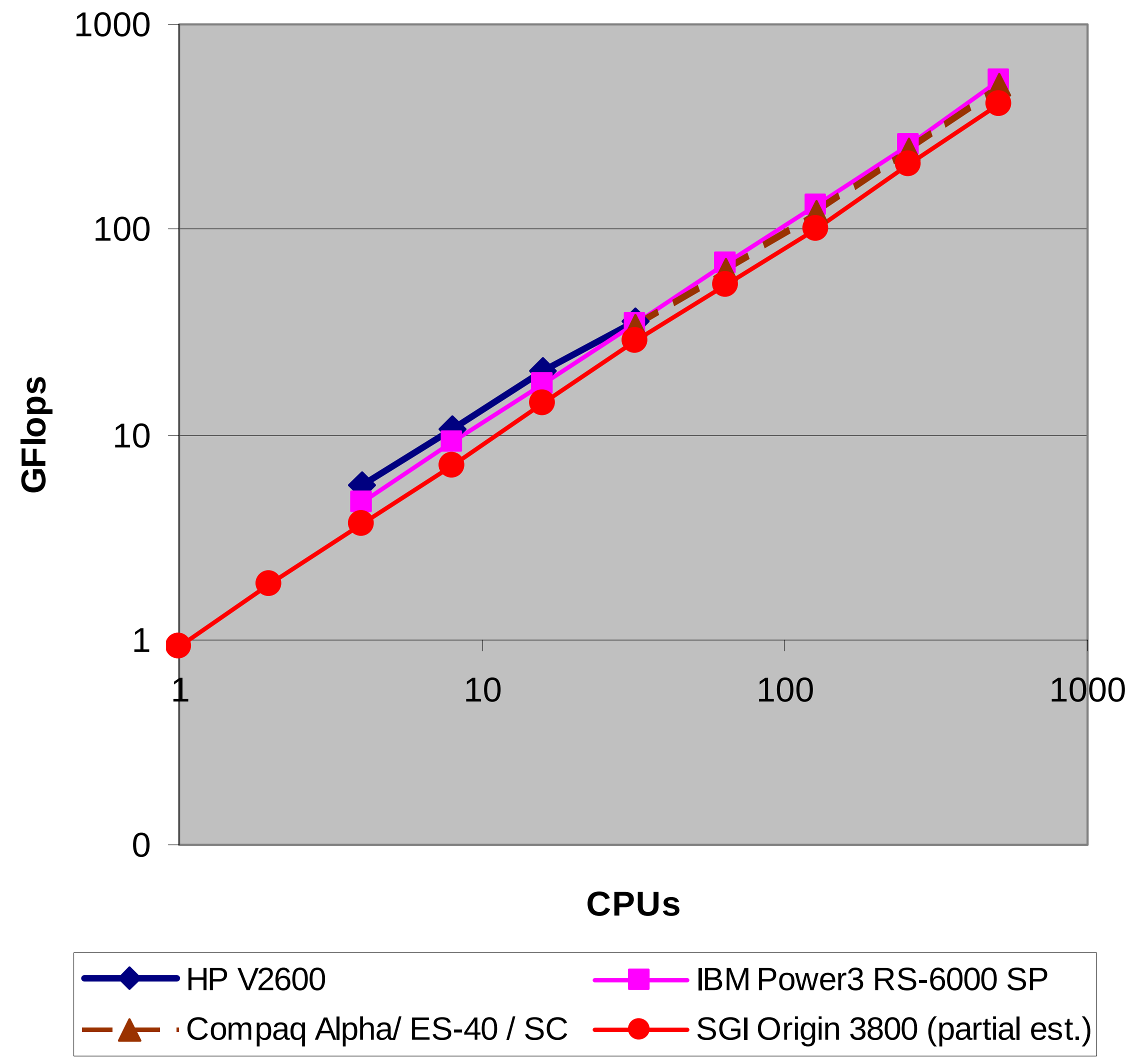
* Big = Server with 8 or more CPUs

Linpack as a metric

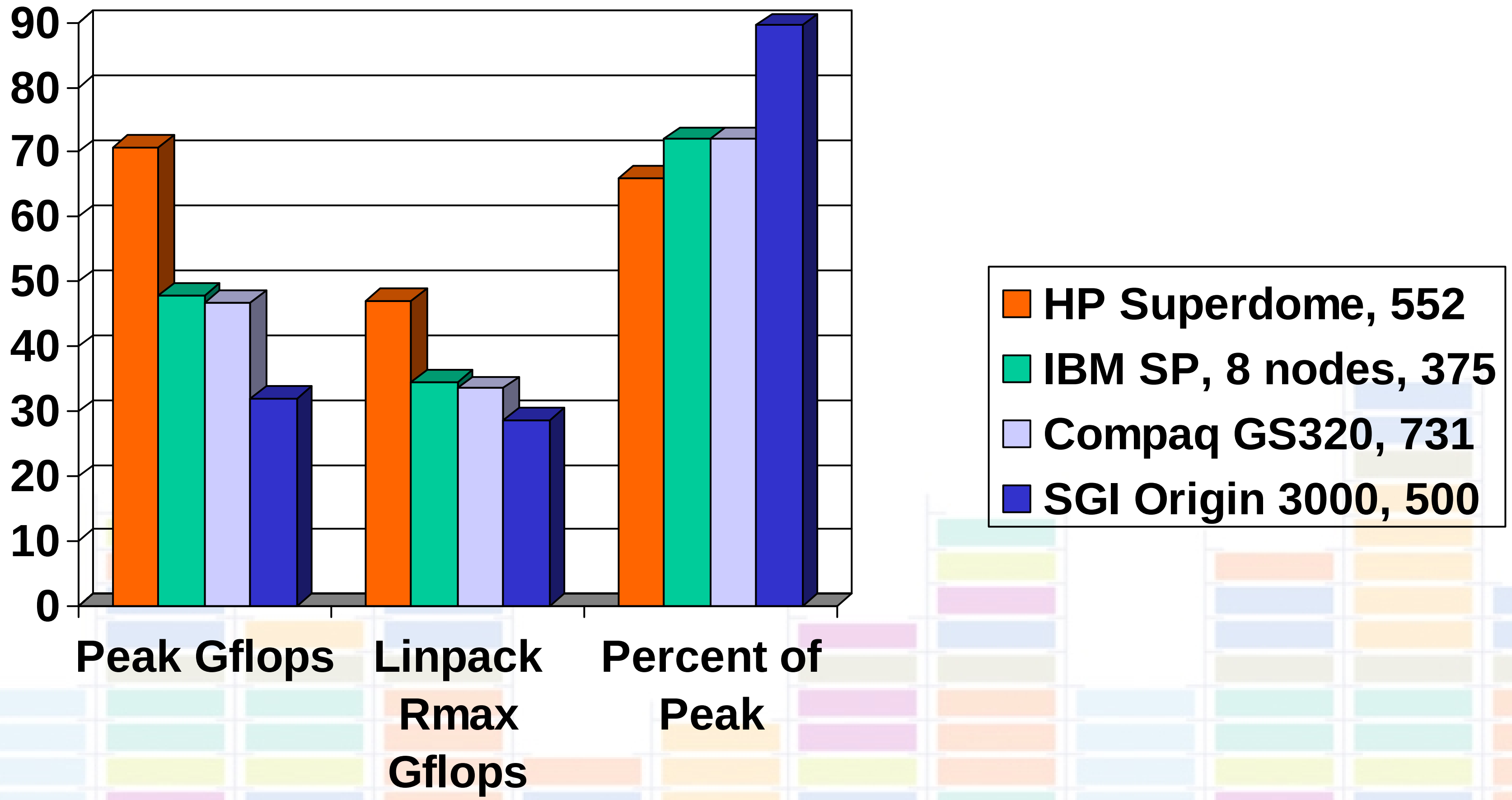
- Linpack:
  - performance of hand-written assembly code of the following operation contained in a three dimensional loop:
    - A(I,J) = A(I,J) + B(I,K)*C(J,K)
    - Are all of your applications based on the latter operation?
    - Will your programmers be rewriting your applications in assembly code?
    - Are your parallel applications embarassingly parallel ?
    - Do your applications place no demand on the memory system?
  - If "Yes" 4 times, Linpack is the metric for you!

Linpack HPC Comparison

- Linpack HPC important for Top 500 List and Gflops proponents
- Notes on chart:
  - HP: current per-CPU champ
  - Compaq and IBM post results up to 512P



Linpack HPC Comparison at 32 CPUs



Linpack Comments

- Everyone knows it's not representative of their applications
- But, it runs without too much difficulty on a variety of computer systems, including clusters
- People are looking for alternative metrics to assist in purchase decisions

IDC HPC Forum

- Recent draft proposal: “A New HPC Technical Computing Benchmark: The IDC Balanced Rating”
- Have proposed a draft metric which includes:
  - CPU: SPECfp_rate2000 and Linpack TPP
  - Memory: Main memory and largest cache memory size; Best and worst bandwidth (to near, far memory)
  - Scalability: Latency of a ping-pong; Bisection memory bandwidth
- Please contribute to definitions and weightings of components to IDC’s HPC Forum

Conclusions

- Few use any single benchmark to acquire large machines
- We recommend weighting, in this order
  - Your applications
  - Benchmarks which include real applications, e.g. SPEC CPU2000 and SPEC OMP2001
  - Pseudo-apps, like NAS Parallel
  - Microbenchmarks, like mpbench and the OpenMP microbenchmarks
  - Kernels, like STREAM & Linpack
  - Machine parameters, like Peak Flops

Appendix

SPEC CPU2000 Primer

- A replacement for SPEC CPU95
  - CPU2000: 26 benchmarks; 700 sec. avg run-time (Origin2000 300Mhz)
  - CPU95: 18 benchmarks; 100 sec. avg. run time (Origin2000 300Mhz)
- Approximately 7X more run-time than SPEC95; a much larger build time
- Often much more data/cache footprint
- Not an artificial benchmark

CFP2000 Descriptions

14 applications from a variety of fields

Benchmark	Language	Category
168.wupwise	Fortran 77	Physics / Quantum Chromodynamics
171.swim	Fortran 77	Shallow Water Modeling
172.mgrid	Fortran 77	Multi-grid Solver: 3D Potential Field
173.applu	Fortran 77	Parabolic / Elliptic Partial Differential Equations
177.mesa	C	3-D Graphics Library
178.galgel	Fortran 90	Computational Fluid Dynamics
179.art	C	Image Recognition / Neural Networks
183.equake	C	Seismic Wave Propagation Simulation
187.facerec	Fortran 90	Image Processing: Face Recognition
188.ammp	C	Computational Chemistry
189.lucas	Fortran 90	Number Theory / Primality Testing
191.fma3d	Fortran 90	Finite-element Crash Simulation
200.sixtrack	Fortran 77	High Energy Nuclear Physics Accelerator Design
301.apsi	Fortran 77	Meteorology: Pollutant Distribution

Actual Flop Counts for 3 counting methods

Code	-O0 / perfex	-O3 / perfex	ssrun -ideal / prof
168.wupwise	9.568E+10	9.094E+10	6.326E+10
171.swim	1.012E+11	9.837E+10	1.018E+11
172.mgrid	2.348E+10	1.484E+11	1.492E+11
173.applu	1.748E+11	1.252E+11	1.248E+11
177.mesa	3.622E+10	3.705E+10	3.555E+10
178.galgel	8.793E+10	1.078E+11	1.106E+11
179.art	1.7E+10	1.603E+10	1.418E+10
183.equake	4.11E+10	4.036E+10	3.976E+10
187.facerec	4.954E+10	4.81E+10	4.741E+10
188.ammp	1.415E+11	1.162E+11	1.252E+11
189.lucas	7.439E+10	6.981E+10	6.674E+10
191.fma3d	8.726E+10	8.543E+10	8.574E+10
200.sixtrack	3.109E+11	2.933E+11	1.854E+11
301.apsi	9.951E+10	9.594E+10	1.022E+11

CINT2000 Descriptions

12 diverse applications

Benchmark	Language	Category
164.gzip	C	Compression
175.vpr	C	FPGA Circuit Placement and Routing
176.gcc	C	C Programming Language Compiler
181.mcf	C	Combinatorial Optimization
186.crafty	C	Game Playing: Chess
197.parser	C	Word Processing
252.eon	C++	Computer Visualization
253.perlbmk	C	PERL Programming Language
254.gap	C	Group Theory, Interpreter
255.vortex	C	Object-oriented Database
256.bzip2	C	Compression
300.twolf	C	Place and Route Simulator