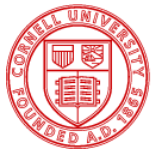
Can Hardware Performance Counters Be Trusted?

Vincent M. Weaver and Sally A. McKee
Cornell University

16 September 2008

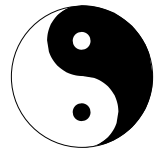
Motivation

- Gather Basic Block Vectors for SimPoint
- Attempt to validate
- Found variation



Cornell University
Computer Systems Laboratory

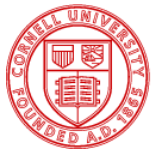
Fusion

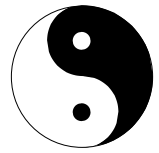


Hardware Performance Counters

- Available on all modern processors
- Used for validation
- Used for performance and workload characterization

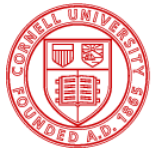
Can they be trusted?

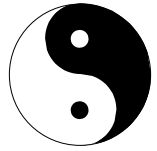




Related Work

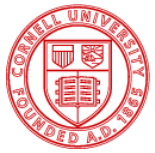
- Black, Huang, Lipasti, Shen. ICCD 1996
  PowerPC, Shorter benchmarks
- Korn, Teller, Castillo. IPCCC 2001
  MIPS, up to 25% error compared to sim
- Maxwell, Teller, Salayandia, Moore. LACSIS 2002
  Pentium III, $<$1% error, microbenchmarks
- Mytkowicz, Diwan, Hauswirth, Sweeney. NSF-NGS 2008
  VM effects can cause up to 5% run-time variation

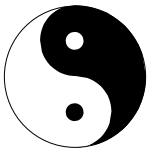




Retired Instruction Count

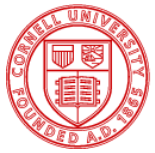
- Universally available
- Should be same on all implementations of ISA
- Used extensively with sampled execution (SimPoint, etc.)
- Part of IPC/CPI metrics

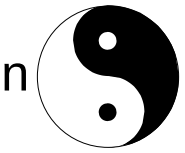




Experimental Setup

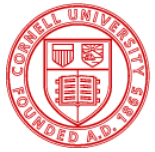
- Linux 2.6.25.4 with perfmon2 patches
- Statically linked 32-bit SPEC CPU 2000 and 2006, full reference inputs
- **Nine** systems, **seven** runs per benchmark per system, **48** SPEC 2000, **55** SPEC 2006
- Only **userspace** instructions counted
- Pin, Qemu and Valgrind DBI tools also investigated

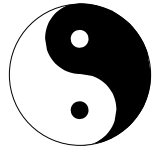




Experimental Systems

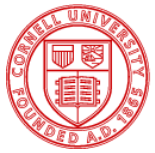
Processor	Speed	Memory	L1 I/D	L2 Cache
Pentium Pro	200MHz	256MB	8KB/8KB	512KB
Pentium II	400MHz	256MB	16KB/16KB	512KB
Pentium III	550MHz	512MB	16KB/16KB	512KB
Pentium 4	2.8GHz	2GB	12Kμ/16KB	512KB
Pentium D	3.46GHz	4GB	12Kμ/16KB	2MB
Athlon XP	1.73GHz	768MB	64KB/64KB	512KB
Phenom	2.2GHz	2GB	64KB/64KB	512KB
Core Duo	1.6GHz	1GB	32KB/32KB	1MB
Core2 Q6600	2.4GHz	2GB	32KB/32KB	4MB

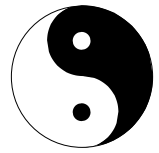




Sources of Variation

- We found sources of variation:
  - Inconsistent Instructions
  - Virtual Memory Layout
  - Hardware Effects
  - System Issues
  - DBI/Simulator Differences
- Can these be mitigated?



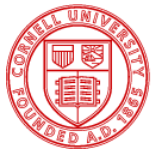


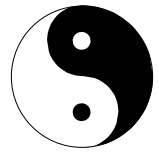
`fldcw` Instruction

- Pentium 4 `instr_retired:nobugsntag` counts `fldcw` as two instructions
- Can lead to a large overcount; `177.mesa` has additional 7 billion dynamic instructions, an overcount of 2.4%
- 12 of the benchmarks have overcount of at least 100M

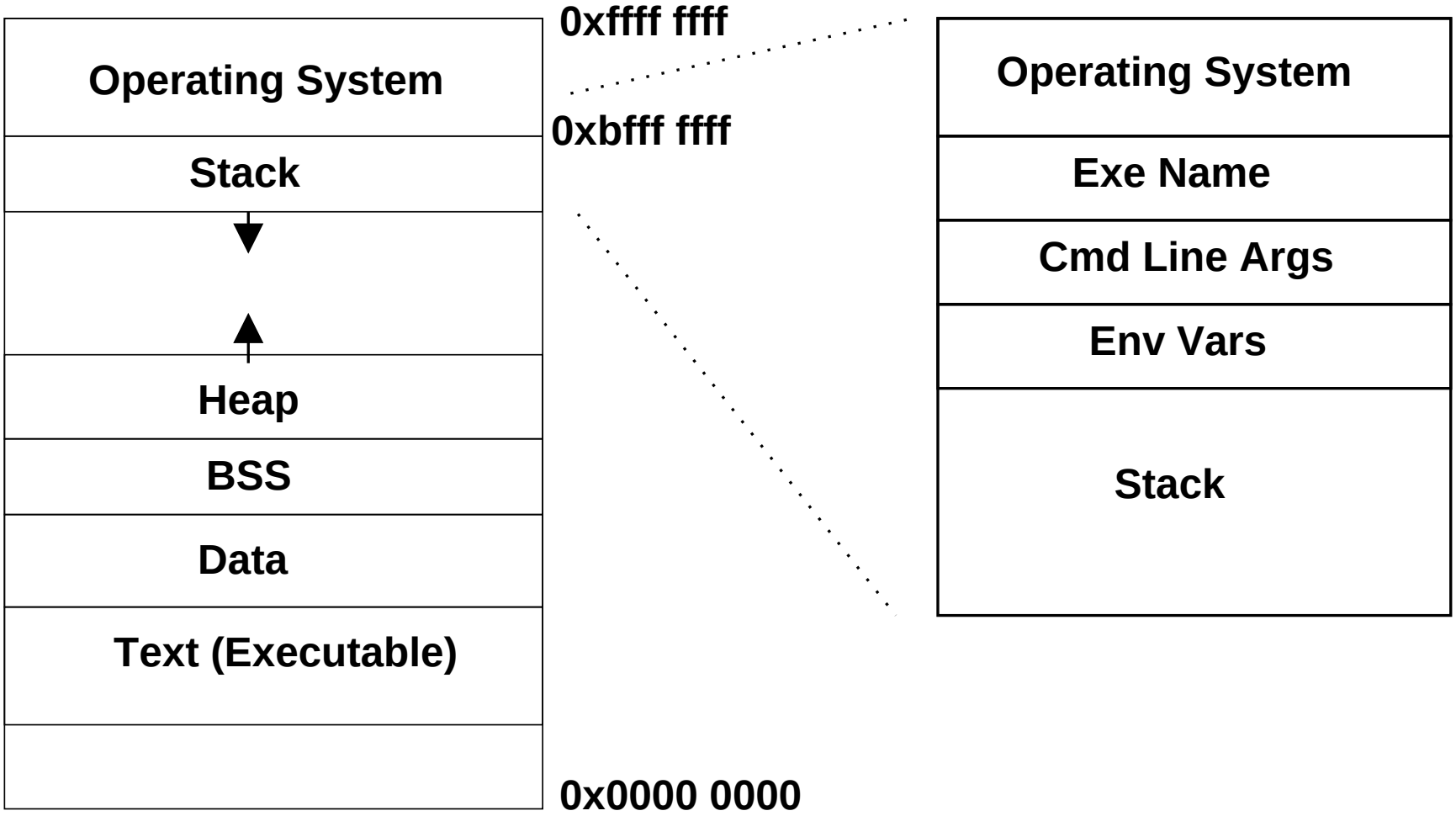
Mitigation:

- Use `instr_completed:nbogus` (Pentium D)
- Adjust using DBI-collected `fldcw` count

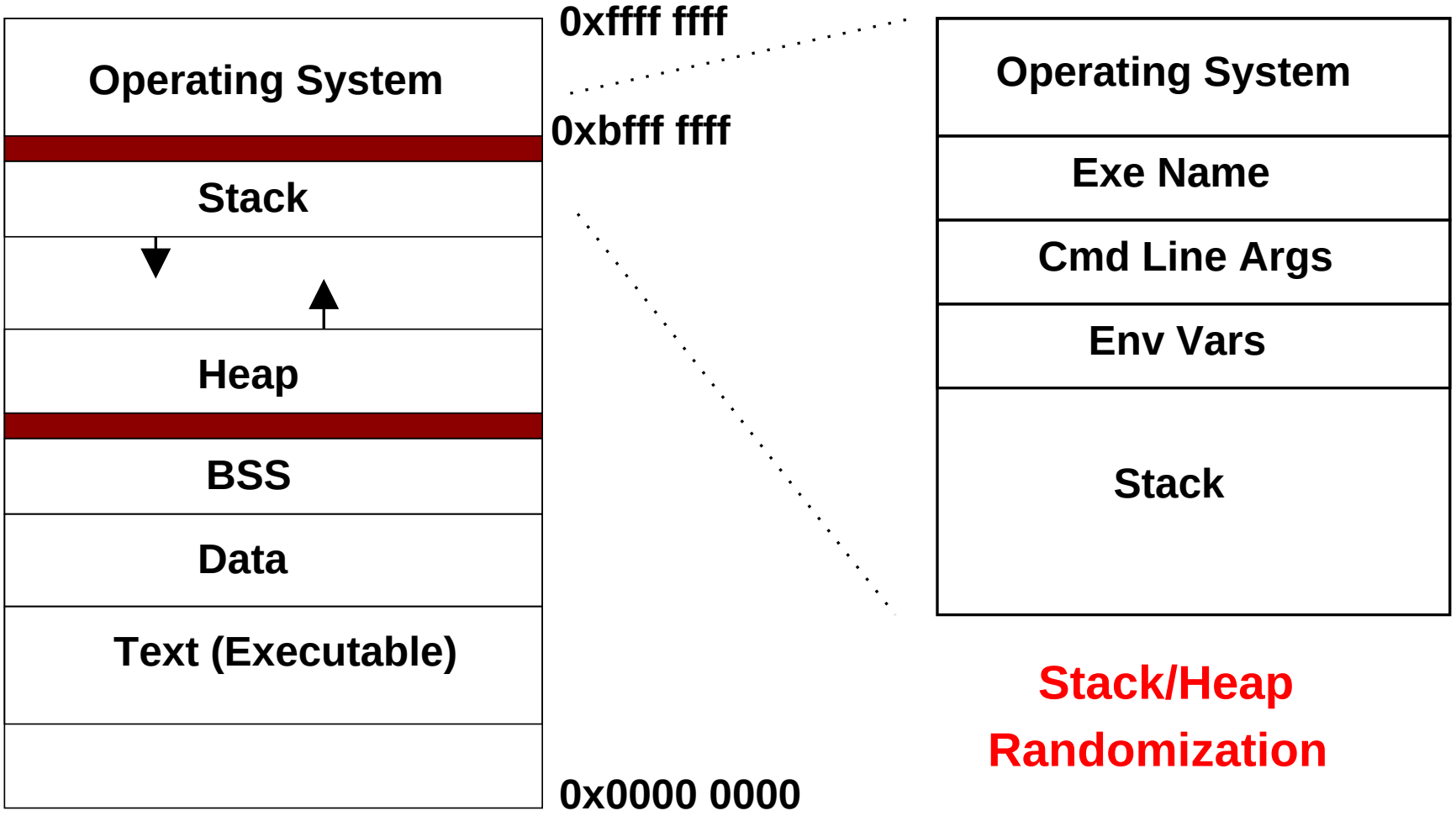




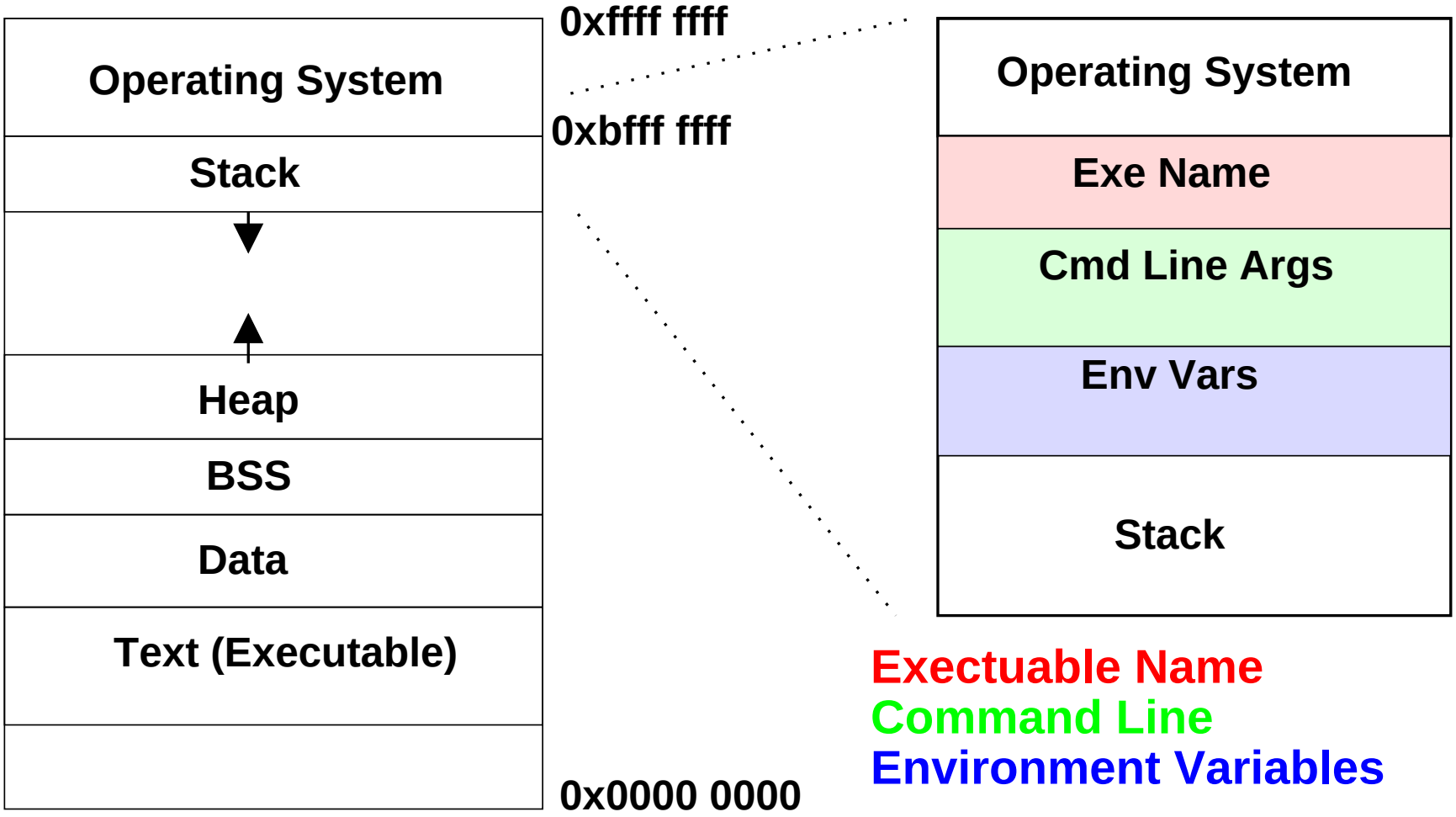
x86 32-bit Virtual Memory Layout



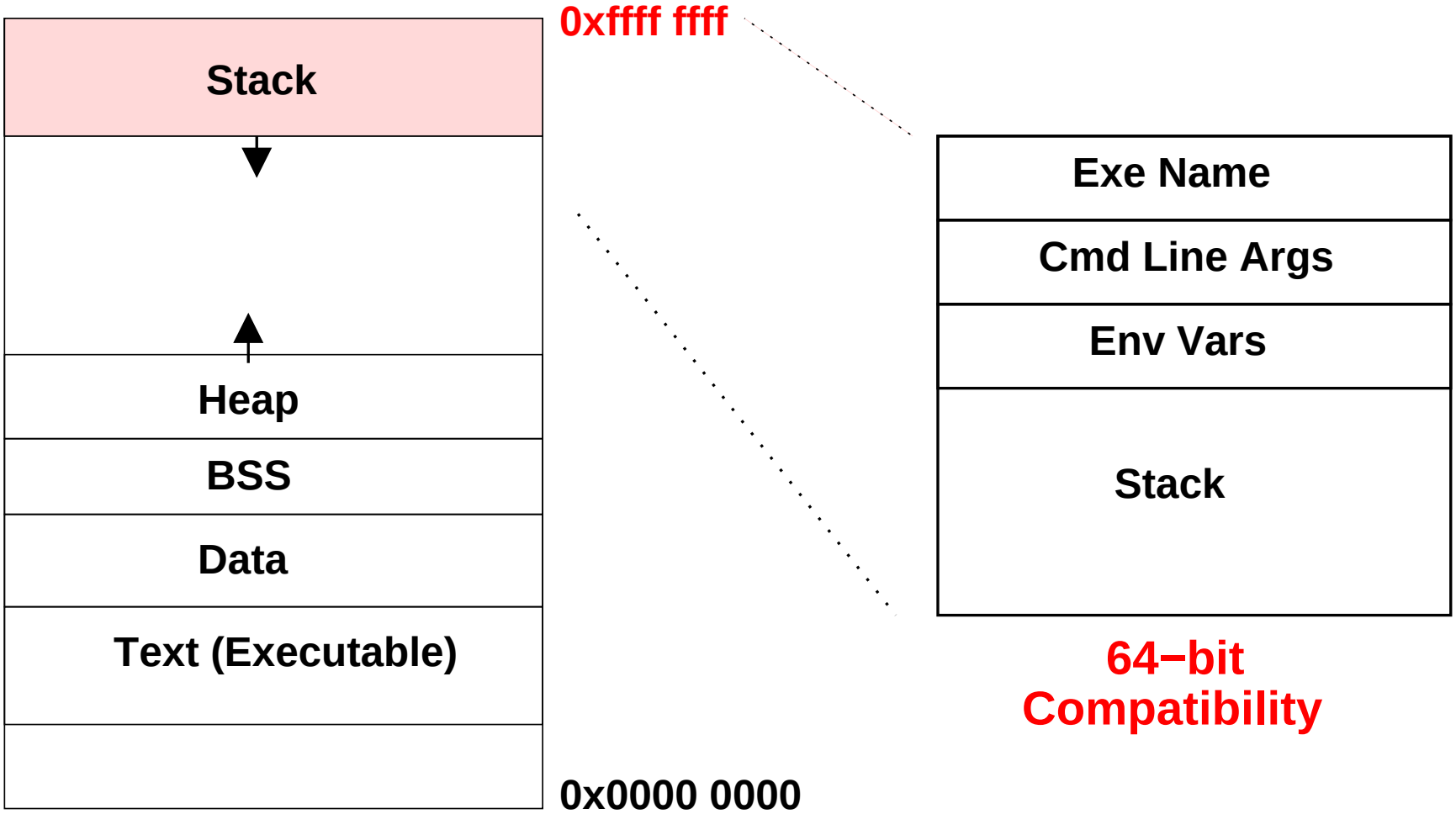
Virtual Memory Randomization



Stack Offset Changes



64-bit Compatibility

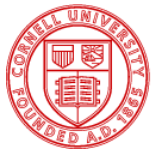


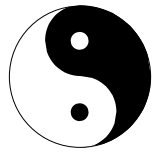
Virtual Memory Layout Impact

- Pointers as hash table keys:
  - Heap — `parser`
  - Stack — `perlbench`
- Optimized memory copies

Mitigation:

- `linux32 -3 -R` — enforce VM, disable randomization
- `/proc/sys/kernel/randomize_va_space`
- Enforce environment variable size



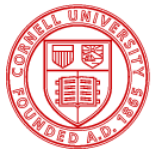


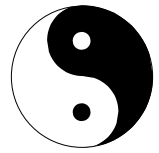
Hardware Effects

- Processor Errata
- Hardware Interrupts — cause extra counts

Mitigation:

- Be aware of errata
- Count or estimate interrupts



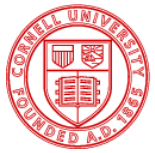


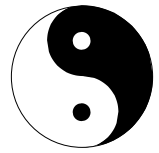
Operating System Effects

- Non-deterministic system calls: time, PID, thread synchronization, random numbers, network activity, IO
- Page faults

Mitigation:

- Modify benchmarks, use methods to reduce non-determinism
- Count pagefaults



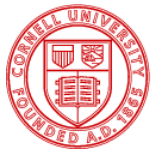


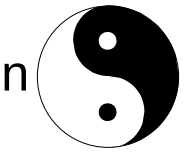
DBI Tool/Simulator Issues

- Instruction complexity: `rep` prefix
- Floating point rounding issues (`art`, `dealII`)
- Virtual Memory Layout

Mitigation:

- Fix simulator/DBI tool
- VM — same as with real hardware

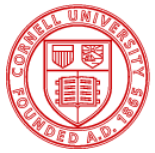


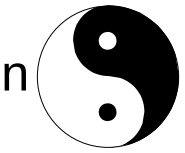


Results

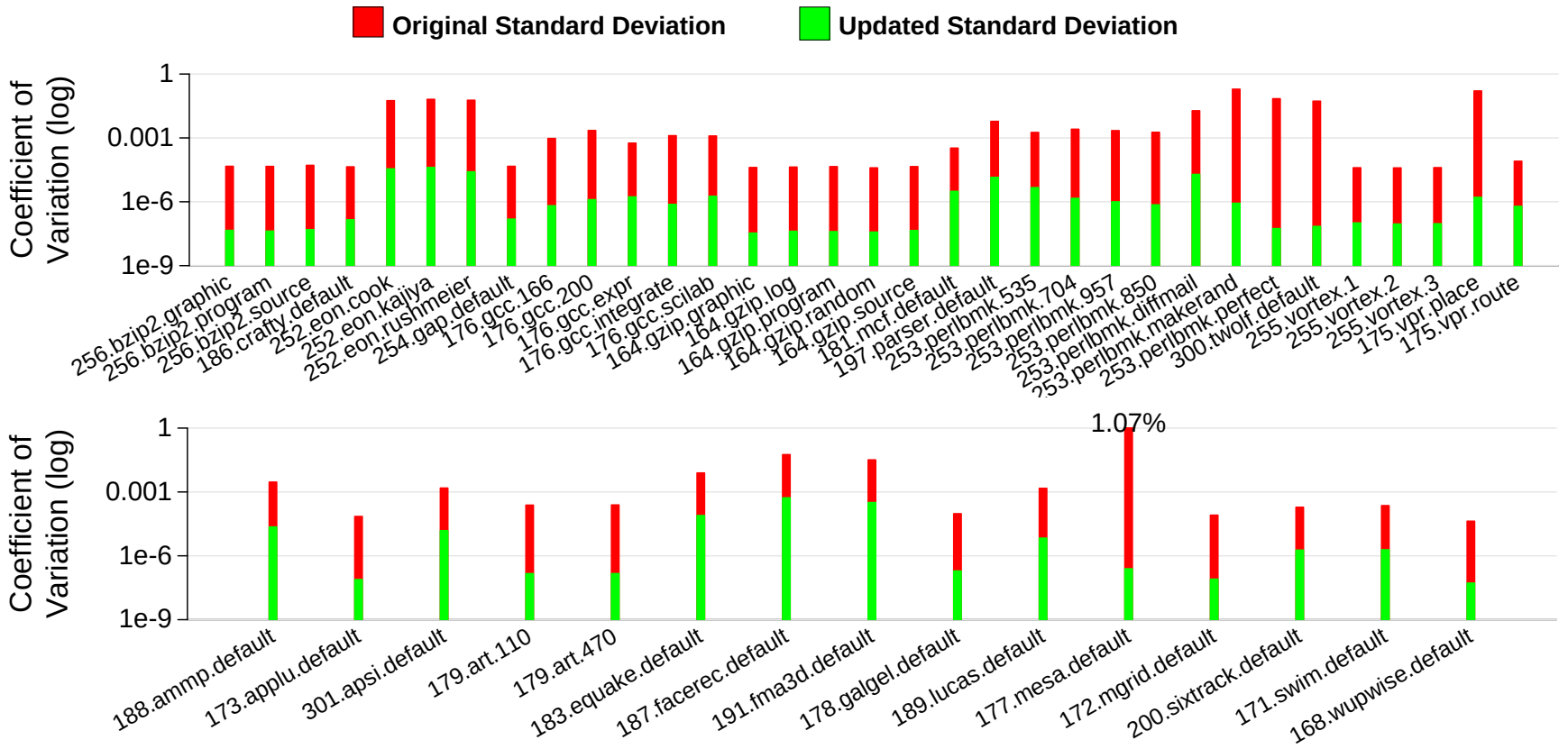
Two kinds of variation:

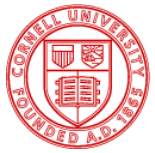
- Inter-machine (differences between systems)
- Intra-machine (differences on the same machine)





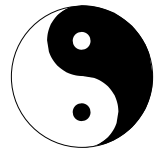
Coefficient of Variation – SPEC CPU 2000



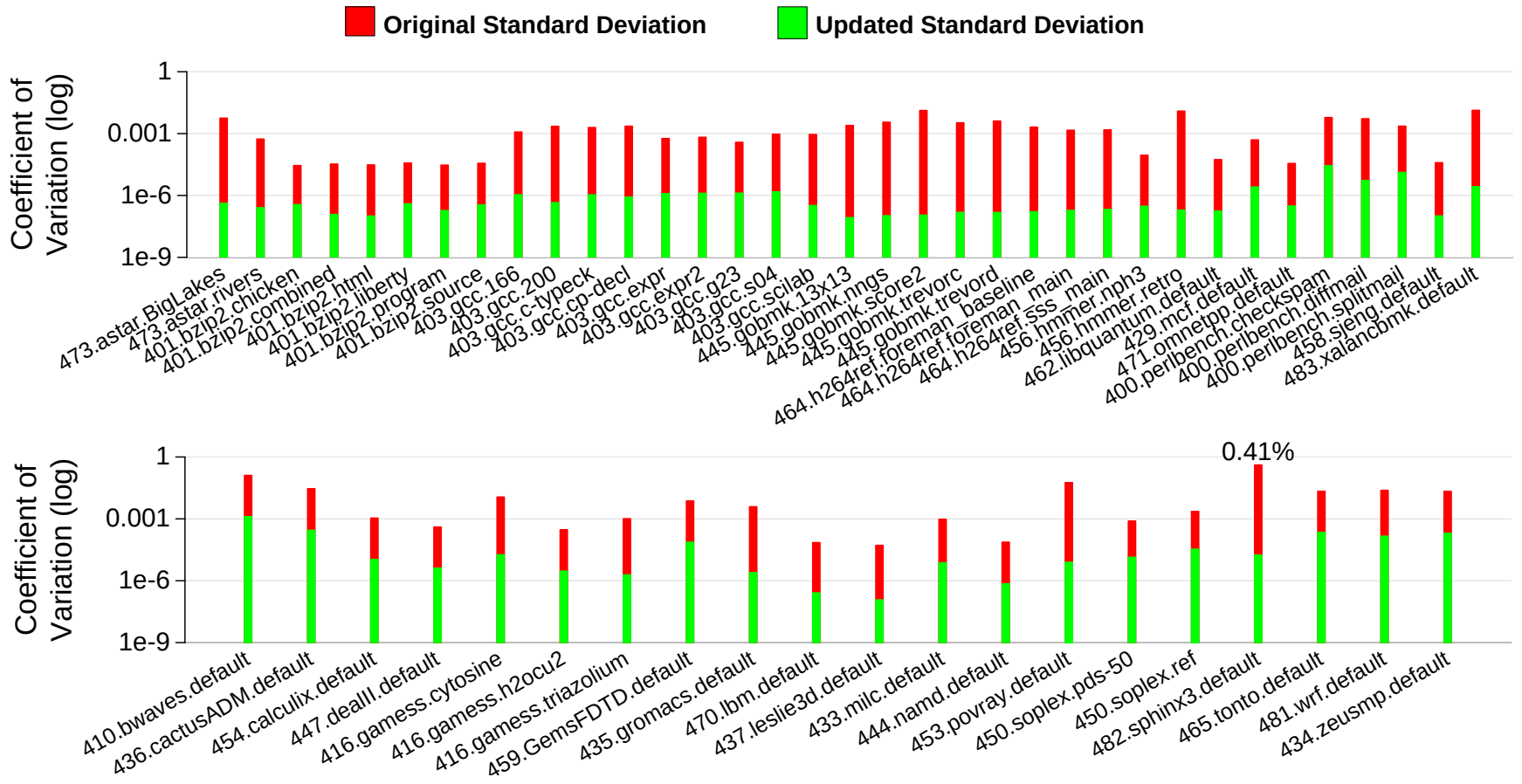


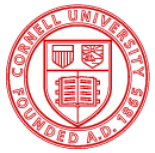
Cornell University
Computer Systems Laboratory

Fusion



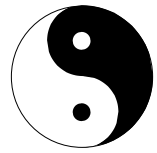
Coefficient of Variation – SPEC CPU 2006



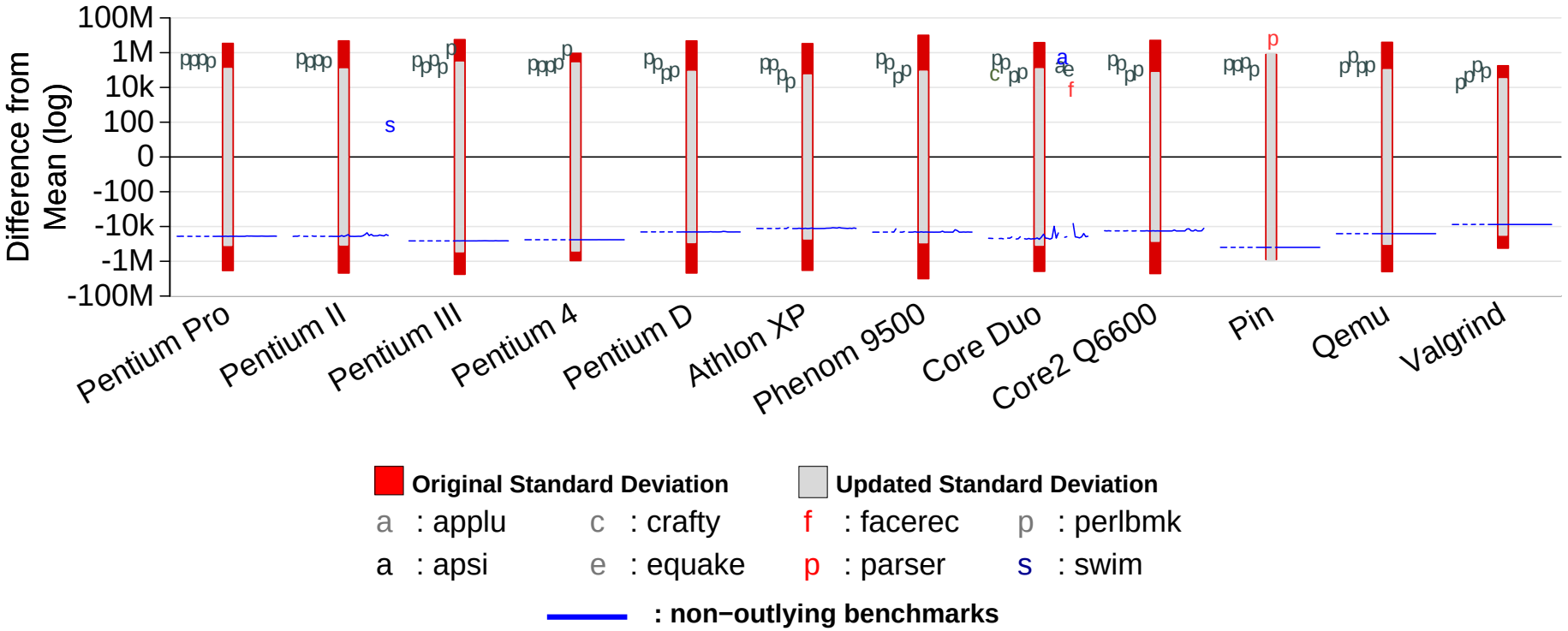


Cornell University
Computer Systems Laboratory

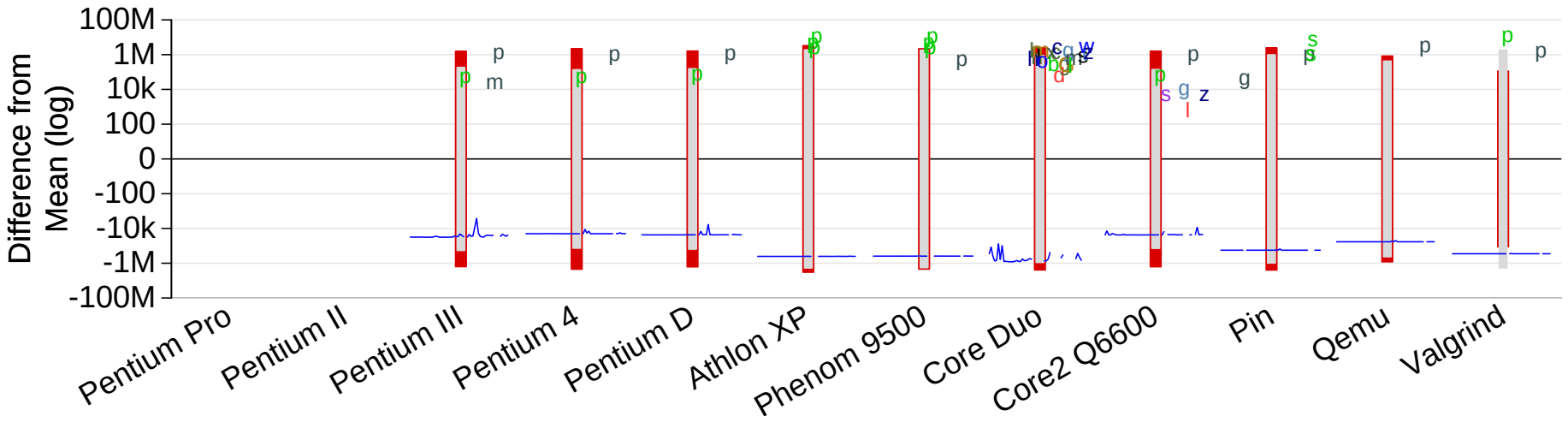
Fusion



Same Machine Results – SPEC CPU 2000



Same Machine Results – SPEC CPU 2006



Original Standard Deviation Updated Standard Deviation

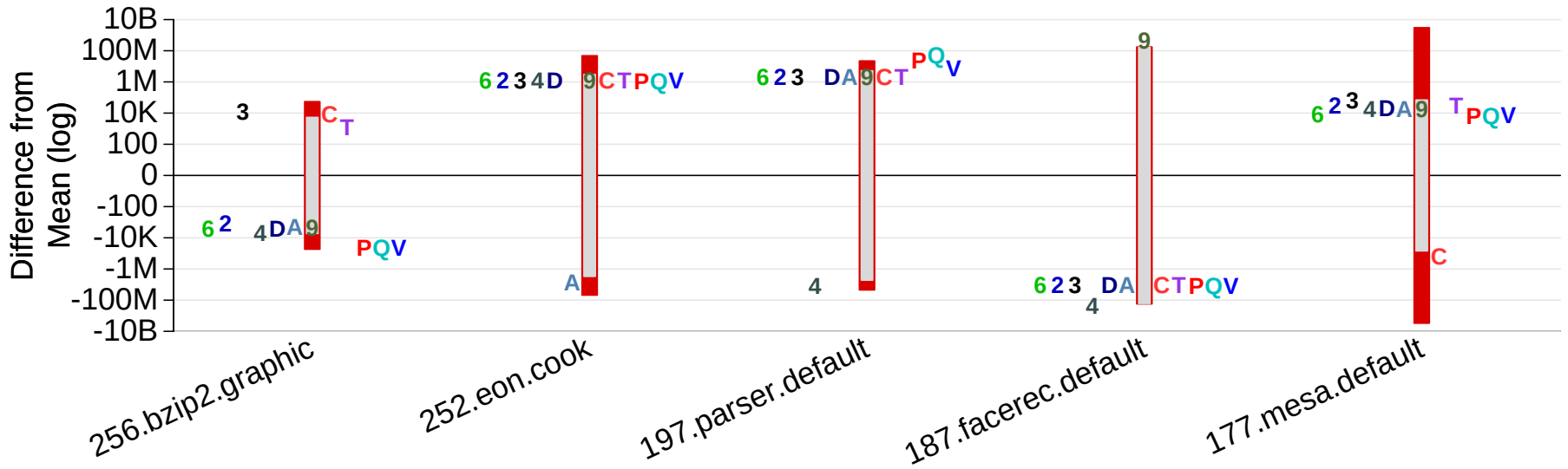
g : gcc.expr2 m : milc s : sjeng

g : gromacs p : perlbench s : soplex

l : leslie3d p : povray z : zeusmp

: non−outlying benchmarks

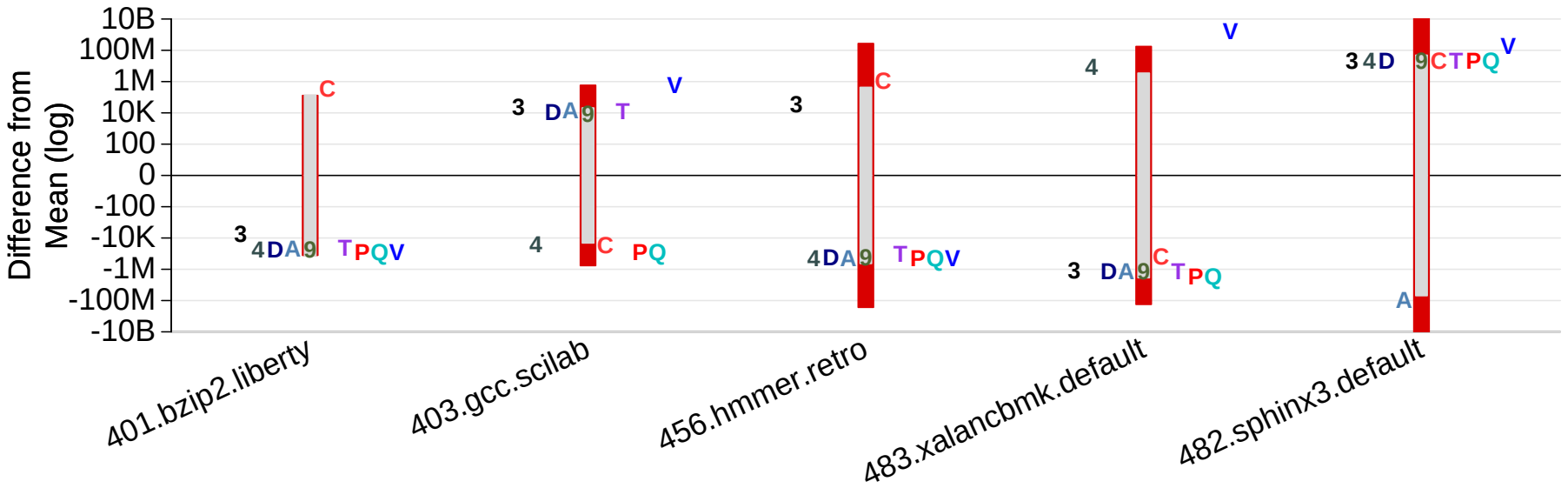
Cross Machine Results – SPEC CPU 2000



Original Standard Deviation Updated Standard Deviation

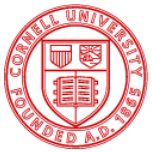
6	: Pentium Pro	4	: Pentium 4	9	: Phenom 9500	P	: Pin
2	: Pentium II	D	: Pentium D	C	: Core Duo	Q	: Qemu
3	: Pentium III	A	: Athlon XP	T	: Core2 Q6600	V	: Valgrind

Cross Machine Results – SPEC CPU 2006



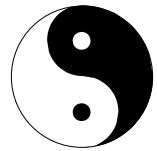
Original Standard Deviation Updated Standard Deviation

6	: Pentium Pro	4	: Pentium 4	9	: Phenom 9500	P	: Pin
2	: Pentium II	D	: Pentium D	C	: Core Duo	Q	: Qemu
3	: Pentium III	A	: Athlon XP	T	: Core2 Q6600	V	: Valgrind



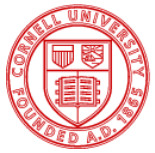
Cornell University
Computer Systems Laboratory

Fusion



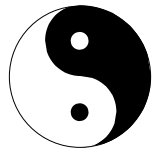
Conclusion

- The retired instruction counter can be trusted to have low variation both inter- and intra-machine
- These results hold across processor generations
- For best results, precautions must be taken



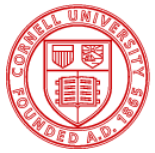
Cornell University
Computer Systems Laboratory

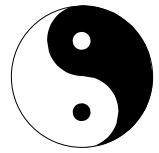
Fusion



Future Work

- Track down remaining sources of variation
- Non-x86 platforms
- Investigate other counter types
- Parallel workloads

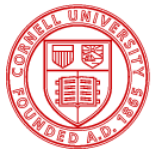


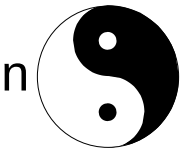


Tools

All code is available from our tools page:

http://fusion.csl.cornell.edu/tools/

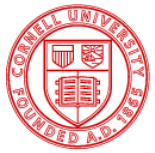




Questions?

All code is available from our tools page:

http://fusion.csl.cornell.edu/tools/



Cornell University
Computer Systems Laboratory

Fusion