

Using Dynamic Binary Instrumentation to Generate Multi-Platform SimPoints

Vincent M. Weaver and Sally A. McKee
Cornell University

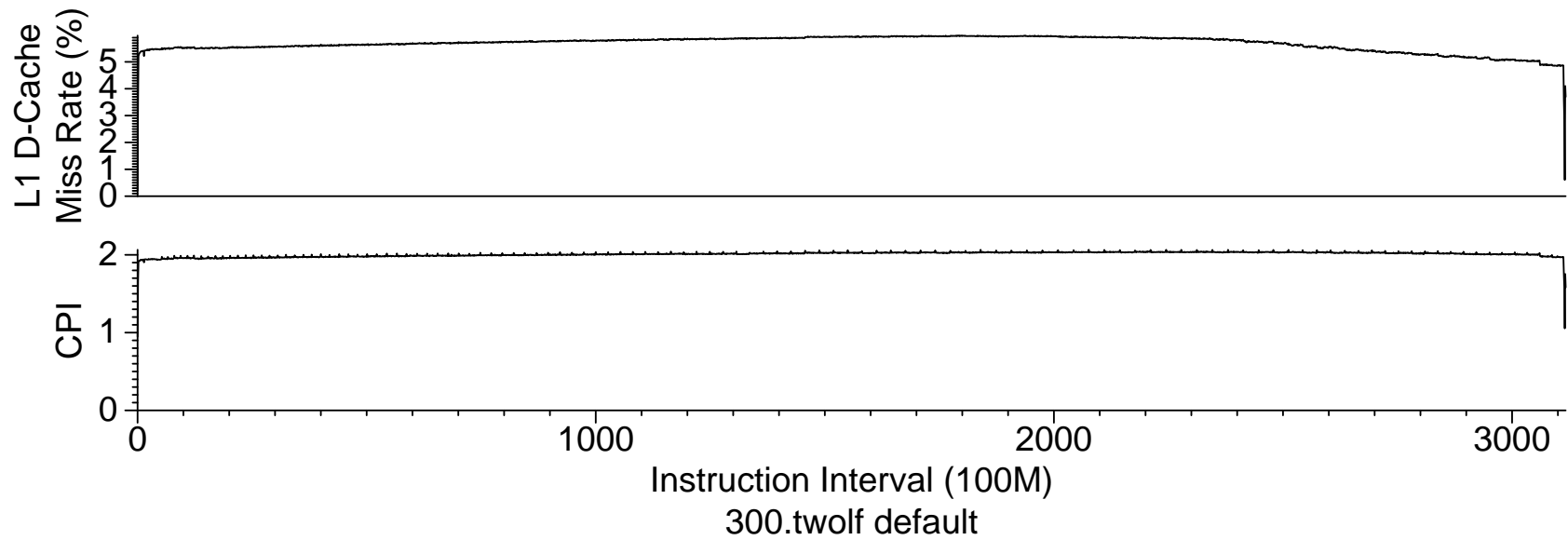
29 January 2008

Breakdown of Benchmark Methodology Used in Major Conferences

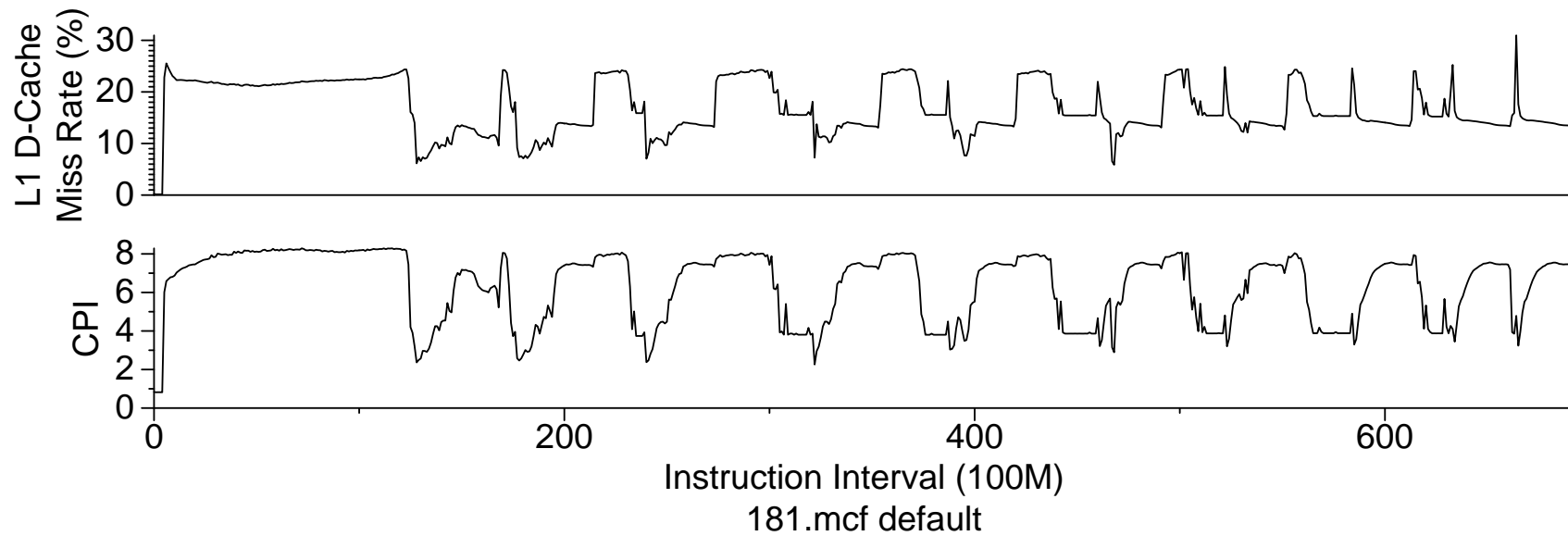
	HPCA, ISCA, MICRO (1995-2004) Yi et al HPCA 2005	ISCA (2006-2007)
Run Complete (simulator)	18%	19%
Run Multiple SimPoints	0%	4%
Run SMARTS	0%	11%
Run One SimPoint	0%	30%
Fast Forward X, Run Z	27%	30%
Run Reduced (train, MinneSPEC)	19%	22%
Run Z	23%	4%



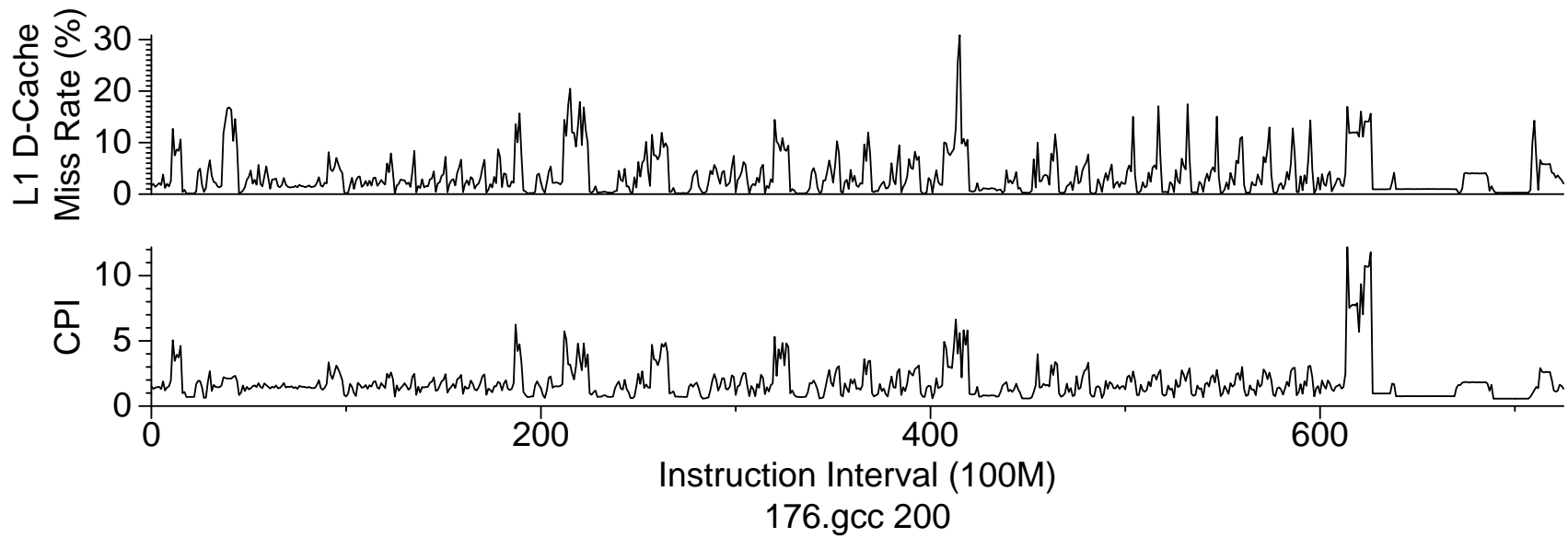
Phase Behavior — twolf



Phase Behavior — mcf



Phase Behavior — gcc.200



SimPoint Basic Block Vectors

- Instrument every Basic Block
- Increment unique counter on entry to each BB
- Save list of BBs and frequencies periodically

These BBVs are used by SimPoint utility to determine phase behavior



Dynamic Binary Instrumentation

- Instruments each BB at first execution
- Caches instrumented code
- Runs faster than simulation, slower than native



Previous Tools for Generating BBVs

- Atom — only Alpha with Tru64 UNIX
- Pin — only Intel architectures (IA32, Intel 64, IA64, XScale)
- Simulator mods — only simulator specific (SLOW)

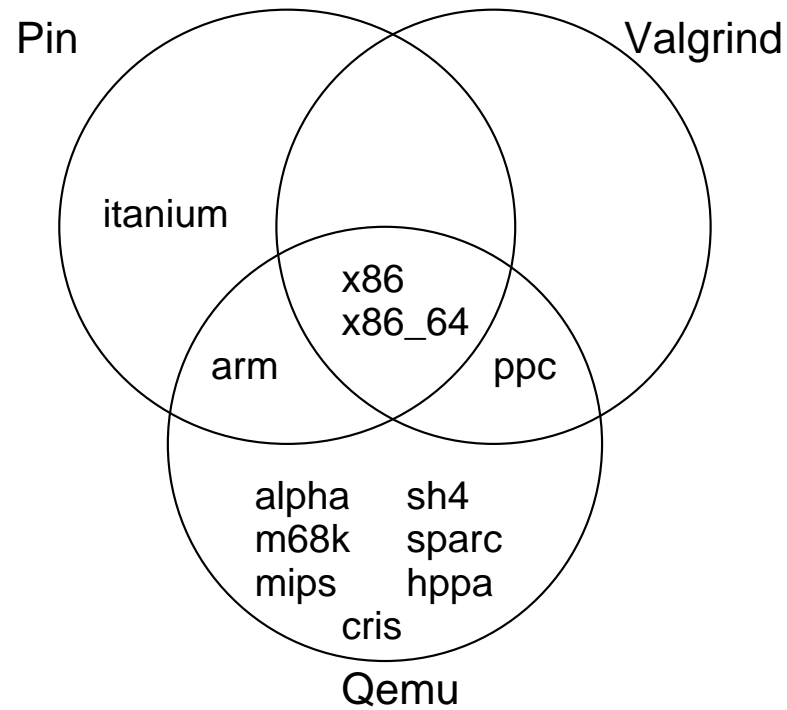


Goal – Expand Applicability of SimPoints

- Support more architectures
- Use existing DBI tools
- Validate generated BBVs
- Generate cross-platform BBV files



Solution: Valgrind and Qemu



Pin

- Intel proprietary DBI tool: IA32, IA64, Intel 64, Xscale (Windows, Linux, OSX)
- Plugins are written in C++
- Average slowdown of 15x on SPEC 2000



Qemu

- Open Source
- Full-system simulation or syscall-by-proxy
- Translates from arbitrary ISAs via DBI
- Runs ARM, IA32, Intel 64, MIPS, PPC, SPARC, HPPA, Etrax CRIS, Alpha, sh4, m68k
- Average slowdown of 28x on SPEC 2000



Valgrind

- Open Source, external plugin interface
- Originally designed to find memory access violations
- Translates to own IR, instruments, retranslates back to native ISA
- Runs on IA32, Intel 64 and PPC platforms (Linux and AIX)
- Average slowdown of 39x on SPEC 2000

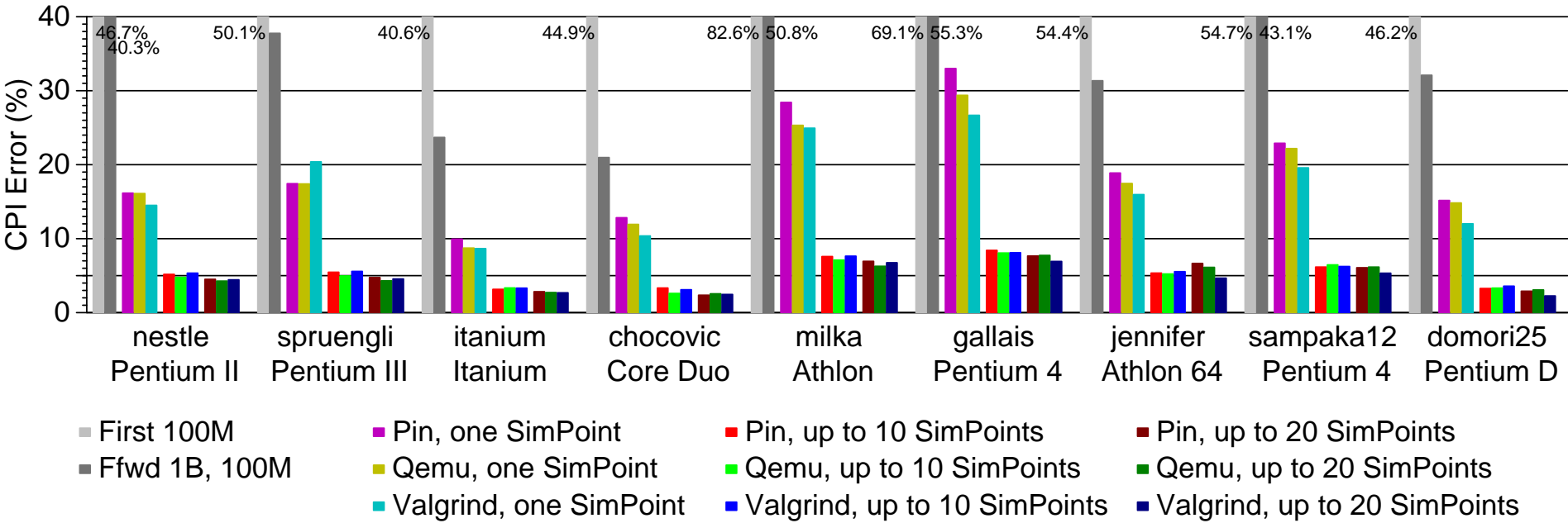


Validation Systems

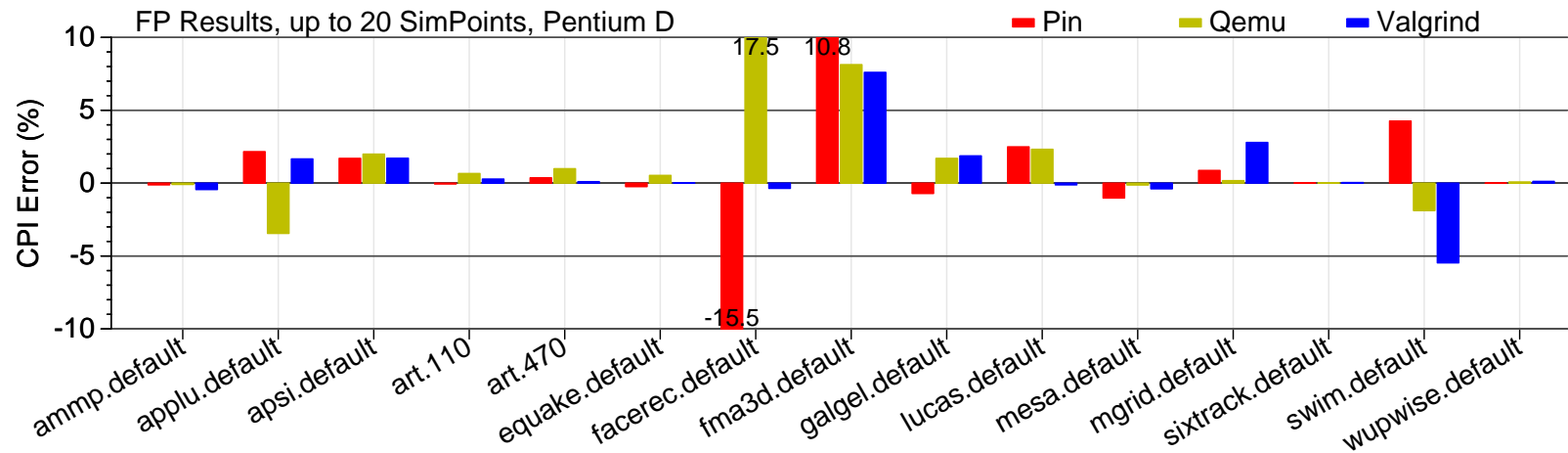
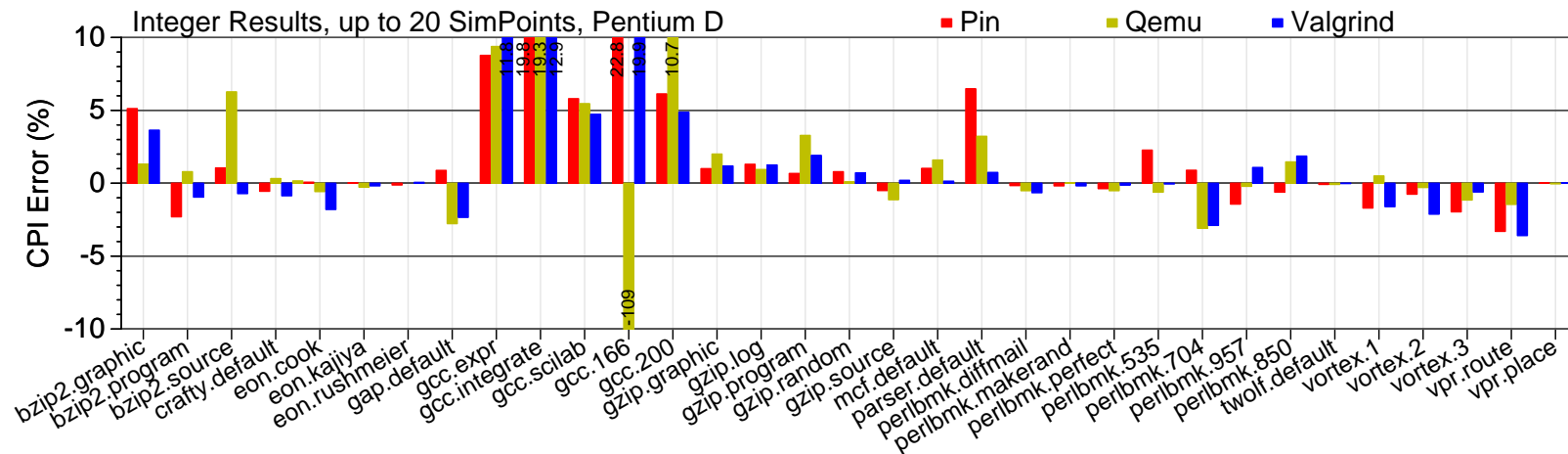
machine	processor	memory	L1 I/D	L2/L3 Cache
nestle	400MHz Pentium II	256MB	16KB/16KB	512KB
spruengli	550MHz Pentium III	512MB	16KB/16KB	512KB
itanium	800MHz Itanium (x86 mode)	1GB	16KB/16KB	96KB/3MB
chocovic	1.66GHz Core Duo	1GB	32KB/32KB	1MB
milka	1.733MHz Athlon MP	512MB	64KB/64KB	256KB
gallais	1.8GHz Pentium 4	256MB	12Ku/16KB	256KB
jennifer	2GHz Athlon64 X2	1GB	64KB/64KB	512KB
sampaka12	2.8GHz Pentium 4	2GB	12Ku/16KB	512KB
domori25	3.46GHz Pentium D	4GB	12Ku/16KB	2MB



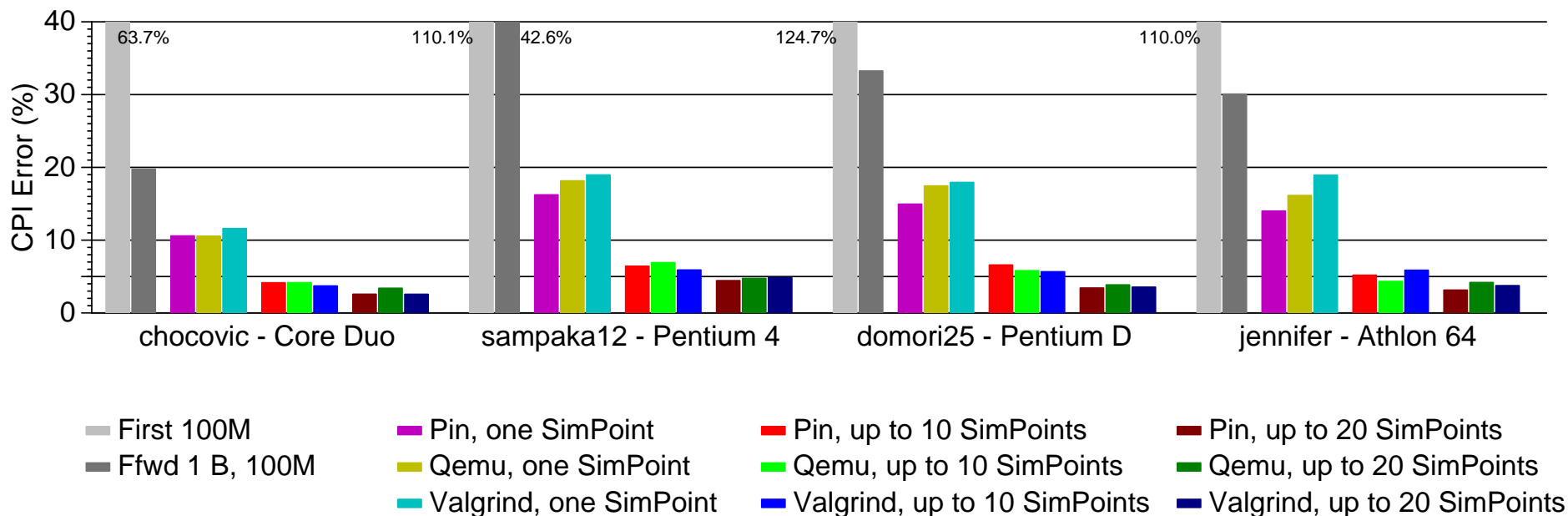
SPEC CPU 2000



SPEC CPU 2000 Breakdown – Pentium D



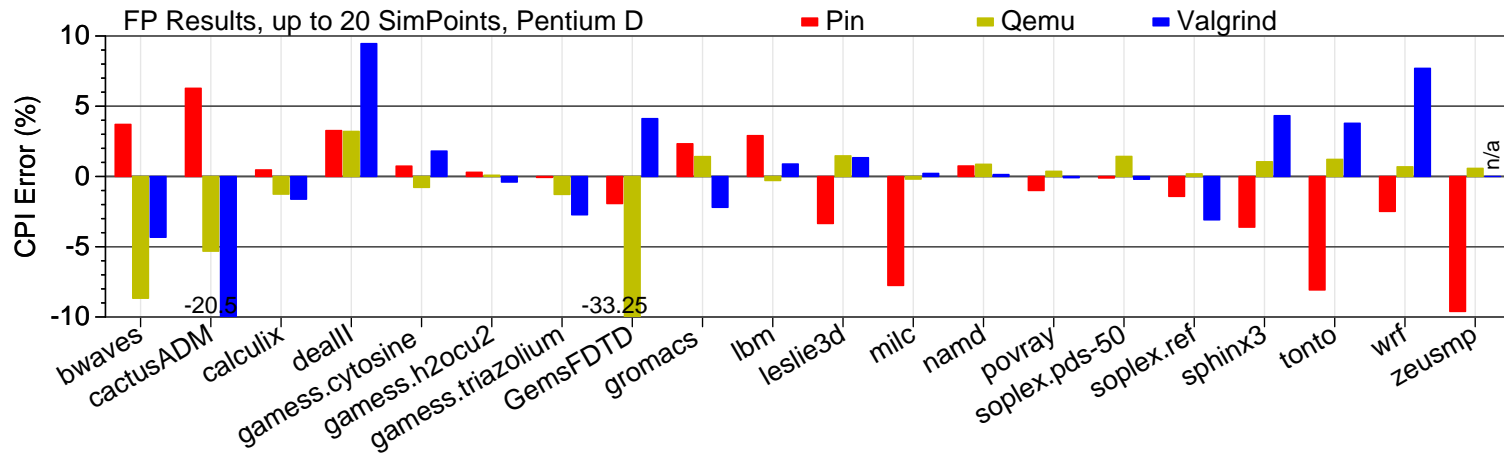
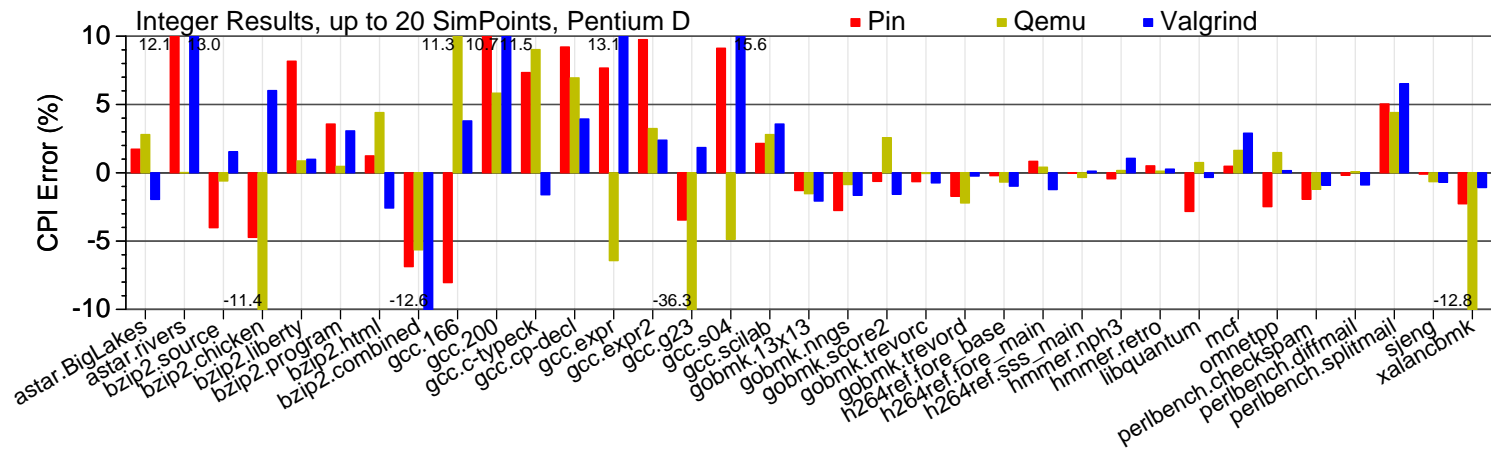
SPEC CPU 2006



machine	processor	memory	L1 I/D	L2 Cache
chocovic	1.66GHz Core Duo	1GB	32KB/32KB	1MB
jennifer	2GHz Athlon64 X2	1GB	64KB/64KB	512KB
sampaka12	2.8GHz Pentium 4	2GB	12Ku/16KB	512KB
domori25	3.46GHz Pentium D	4GB	12Ku/16KB	2MB



SPEC CPU 2006 Breakdown – Pentium D



Results – Average CPI error

- SPEC2000: 10 SimPoints ($<0.4\%$ of reference inputs)
 - Pin 5.32%
 - Qemu 5.04%
 - Valgrind 5.38%
- SPEC2006: 10 SimPoints ($<0.06\%$ of reference inputs)
 - Pin 5.58%
 - Qemu 5.30%
 - Valgrind 5.28%



Future Work

- Generating multi-platform results – MIPS BBV file from IA32 Qemu
- Running non-Linux binaries (Solaris, IRIX)
- Generating OS-aware SimPoints – Qemu runs full OS



Tools Available for Download

All code is available from:

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



Questions? Feedback?

All code is available from:

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

vince@csl.cornell.edu — <http://www.csl.cornell.edu/~vince>

sam@csl.cornell.edu — <http://www.csl.cornell.edu/~sam>

