TECHNISCHE
UNIVERSITÄT
DRESDEN

Faculty of Computer Science Institute of Systems Architecture, Operating Systems Group

Understanding the propagation of hard errors to software and implications for resilient system design

M. Li, P. Ramachandran, S. Sahoo, S. Adve, V. Adve, Y. Zhou

presented by Bjoern Doebel

The old litany

- Shrinking feature sizes increase
  - Susceptibility to radiation
  - Manufacturing errors
  - Wear-off
  - Heat-induced errors
- Also: DVFS influences error rates
- Need hardware/software measures
  - Spend as few (additional) resources as possible
  - Require understanding of how hardware errors manifest

Design Goals

- Symptom-based vs. fault-based detection
- Don't handle masked faults.
- Optimize for the common case.
- Keep things customizable
- Leverage existing features instead of adding new ones.

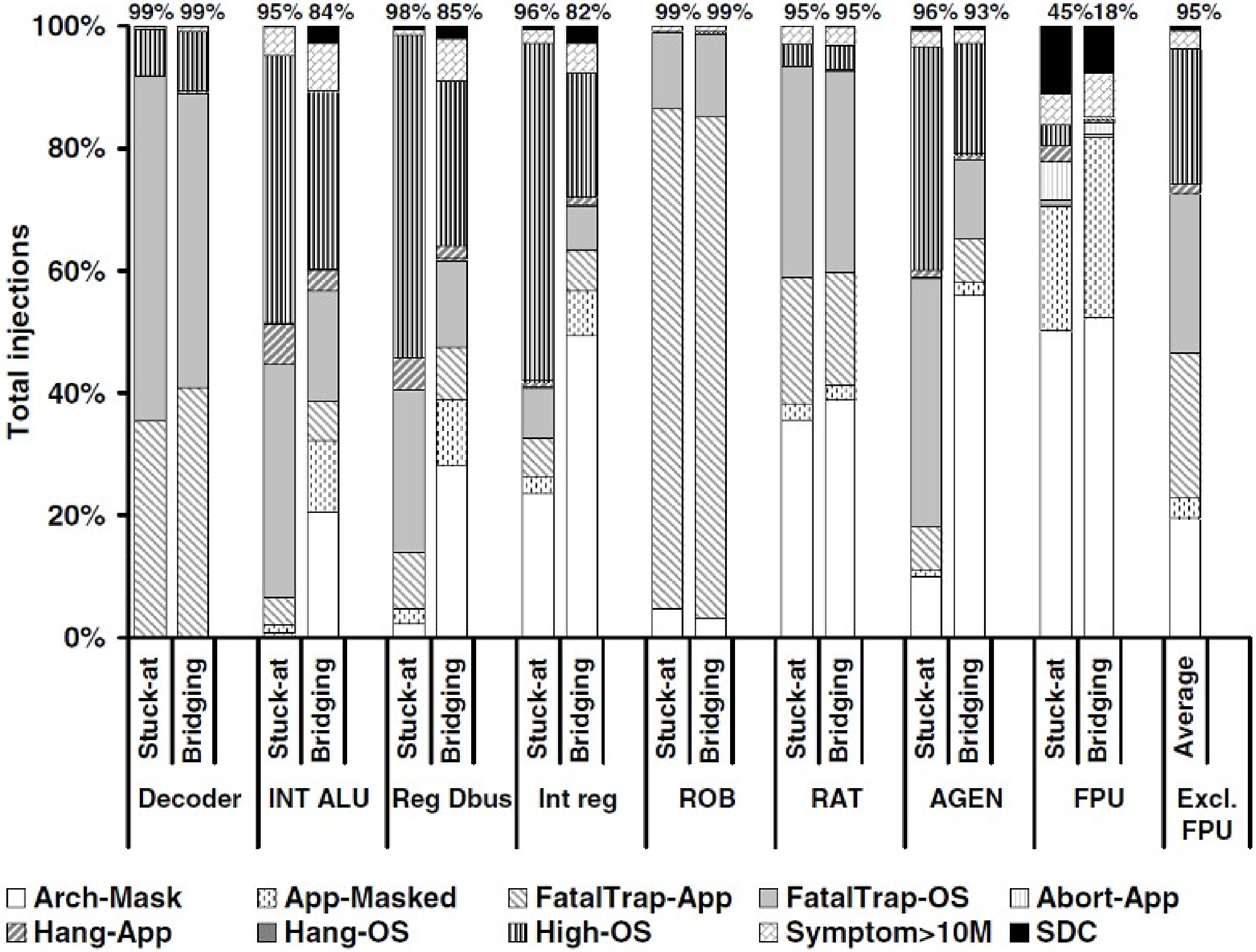
Fault injection experiments

- Target arch: SPARC v9, Solaris, SPEC benchmarks
- Environment: Simics + GEMS
  - Run in parallel for 10,000,000 cycles
  - Simics-only afterwards
- Inject hard (stuck, bridging) errors
- Fault injection in:
  - Instruction decoder
  - ALU
  - Register bus
  - Physical register file
  - Reorder buffer
  - Register Alias Table
  - Address generation unit
  - FPU

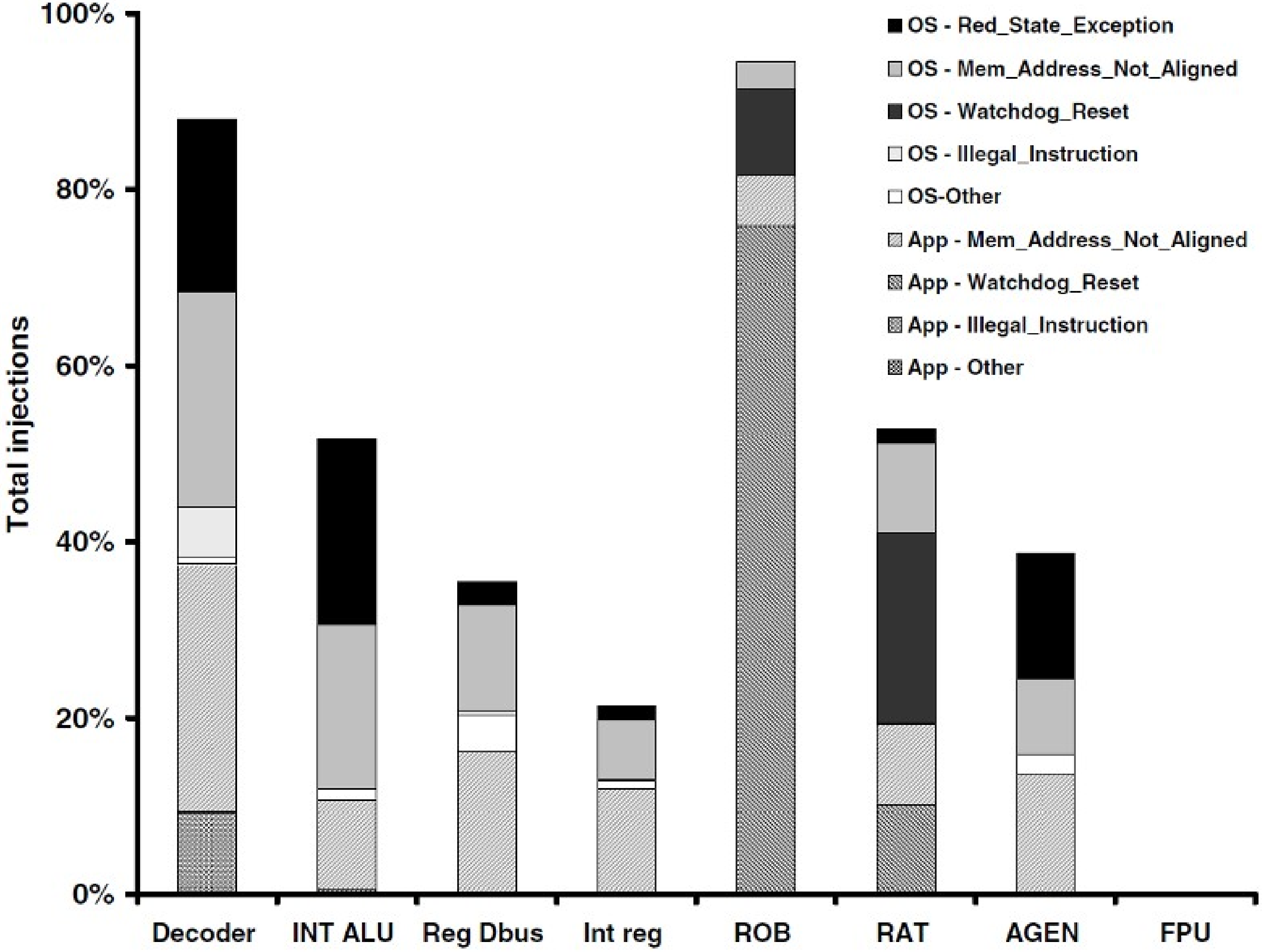
Symptom-based fault detection

- Fatal hardware traps
- Abnormal application exit / OS crash
- Application/OS hangs
  - Branch counting
- Excessive OS activity
  - Observation: normal OS activity <10,000 cycles

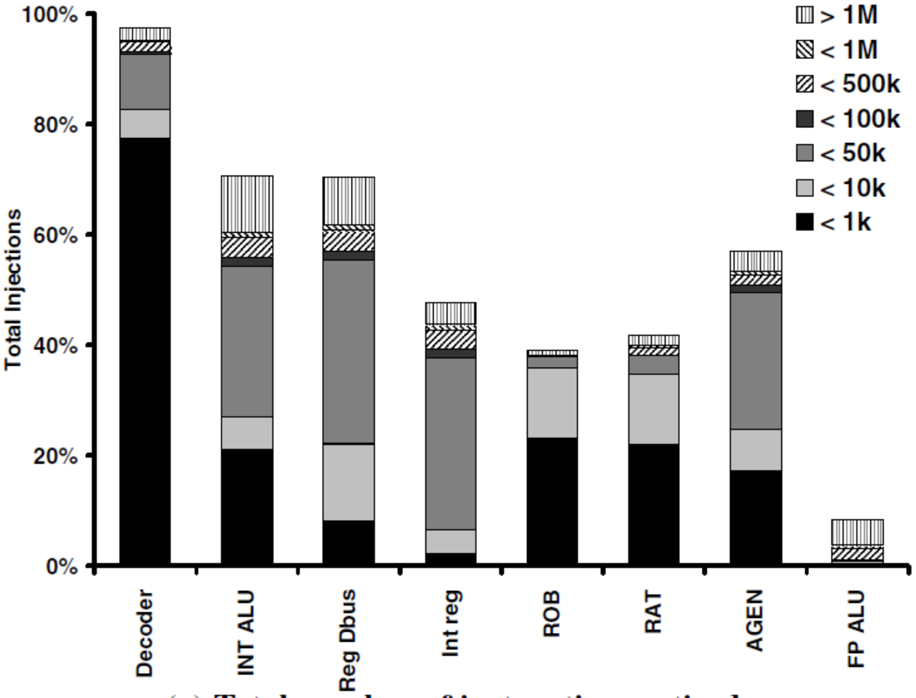
Initial results



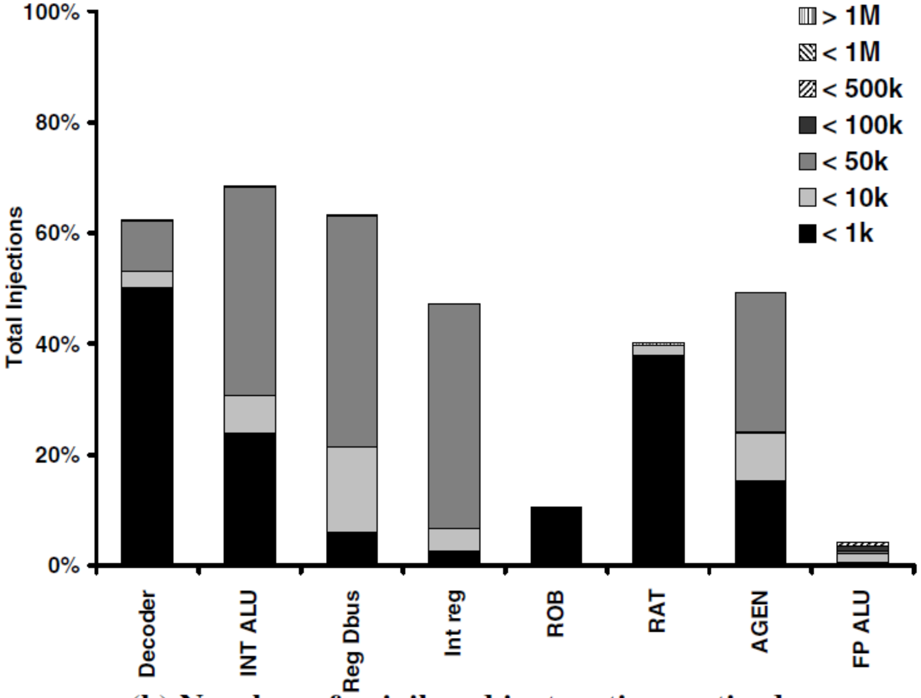
Fatal traps


100%
80%
60%
40%
20%
0%
Total injections
■ OS - Red_State_Exception
□ OS - Mem_Address_Not_Aligned
■ OS - Watchdog_Reset
□ OS - Illegal_Instruction
□ OS-Other
▨ App - Mem_Address_Not_Aligned
▨ App - Watchdog_Reset
▨ App - Illegal_Instruction
▨ App - Other
Decoder
INT ALU
Reg Dbus
Int reg
ROB
RAT
AGEN
FPU

Fault detection latency



(a) Total number of instructions retired from application state corruption to detection



(b) Number of privileged instructions retired from OS state corruption to detection

But what about soft errors?

- [Saggese2005]: “An experimental study of soft errors in microprocessors”
  - 53% of injected faults have no effect
  - 23% crash application
  - 13% silent data corruption
  - 12% incomplete execution

Discussion

- SPARC vs x86
  - Does the max. 10,000 cycles in kernel hold for Linux/x86? Is there an upper bound?
  - Fewer illegal instructions
  - No misaligned memory accesses
- “I already have all those expensive checkpoint/rollback features in my system, so no need to build something new.”