Virtues & Obstacles of Hardware-assisted Multi-processor Execution Replay

Cristiano Pereira, Gilles Pokam, Klaus Danne, Ramesh Devarajan, Ali-Reza Adl-Tabatabai
Intel Corporation

- Parallel program execution changes from run to run due to non-determinism
- Non-determinism is source of complexity and makes parallel program behavior hard to predict and understand
- Shared-memory order is main cause of non-determinism
- SW-only Record & Replay cannot track shared-memory ordering efficiently
- Adoption of HW-assistance requires strong value proposition and simple design

Virtues

Debugging and Testing
• “Always on” allows in-house and customer reproducibility of concurrency bugs
• Integration with dynamic analysis tools (e.g. Intel® Parallel Studio)
• Time-travel debugging
• Order events across cores (e.g. Branch Traces)

Post-Silicon Validation
• Transfers failures from silicon to RTL or tester environment
• Traditional probe-based techniques are too complex
• Reduces cost of “triage”: false alarms due to SW bugs
• Cycle accuracy helps HW debug

High-Availability (HA)
• Replaces lock-step for replication & fail-over
• Exchange replay logs between primary and replica

Obstacles

Hardware Complexities
• Minimal hardware changes and simplicity are key
• No coherence piggybacking
• Writing logs out to disk without perturbation
• No cache modifications

Relaxed Memory Models & Instruction Atomicity
• Cannot simply use instruction counts to capture order
• Requires core modifications
• Instruction side-effects are exposed before completion (e.g. Intel® x86 Macro/Micro)

Replay-Speed
• Debugging can afford slow replay (e.g. serial replay)
• HA requires fast replay; hardware support likely needed

HotPar 2010, June 14-15, Berkeley, CA