Fingerprinting Across Memory Interconnects

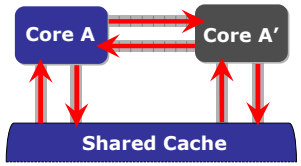
Srinivas Chellappa, Frédéric de Mesmay, Jared Smolens, Babak Falsafi, James Hoe, Ken Mai

Introduction

- Soft error rate becoming a problem in high performance microprocessors
- *Fingerprints* detect errors in dual modular redundant (DMR) systems

Problem:

- Past implementations assumed fixed-latency, dedicated, pipelined interconnects for fingerprint comparison
- Requires additional hardware
- Redundant core pairs fixed at design time



Observation:

- Modern CMPs (eg: Sun Niagara) include high-bandwidth memory interconnects to connect cores on-chip
- Plenty of spare bandwidth available

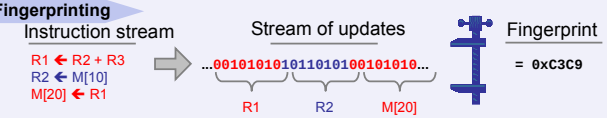
Can we use the existing on-chip interconnect for fingerprints?

Background

On-chip Interconnects

- Cross-bar type, message based
- New chips have high-bandwidth interconnects (Sun Niagara2 ~270 GB/s)
- Interconnect latency is variable due to queueing
- Spare bandwidth available

Fingerprinting

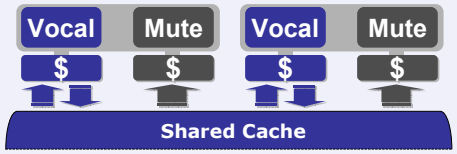


Architectural Fingerprints:

- Small (2-byte) hash of architectural state generated off the critical path of the processor
- Compared periodically between cores to detect errors
- Equivalent to comparing results of a group of instructions

Reunion Model

Dual Modular Redundancy (DMR)



- Redundant execution across pairs of cores
- Fingerprints periodically compared across cores
- Logical processor pair appears to be a single core

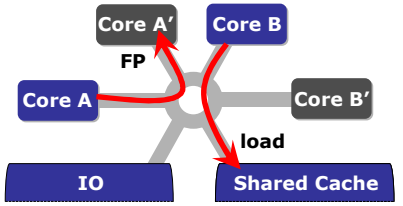
Vocal core: execution results visible to system
- Maintains baseline coherence protocol
- Stores globally to memory
- OoO execution, cache unchanged

Mute core: no effect on system state
- Cache not tracked by coherence protocol
- Mute stores visible only in private cache
- Cannot break vocal's coherence protocol

Design

Idea:

- Send fingerprints as messages on existing on-chip memory interconnect



Design Parameter:

- Fingerprint interval is the number of instructions between two fingerprint comparisons

Design Issues:

- Contention between fingerprints and memory messages could potentially slow down system

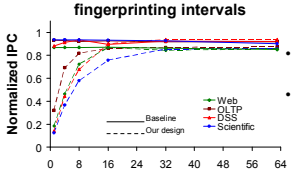
Results

Workloads

We evaluate the performance of our design with parallel commercial and scientific workloads on the Flexus timing simulator

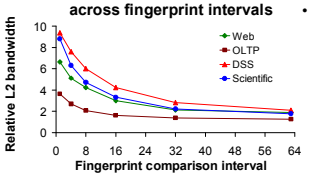
OLTP Web	Apache Zeus	Low IPC, low memory level parallelism
OLTP	DB2 Oracle	
DSS	DB2 Q1 DB2 Q2 DB2 Q17	Moderate to high IPC, high memory level parallelism
Scientific	em3d moldyn ocean sparse	IPC and memory access patterns vary

IPC trend across fingerprinting intervals



- Our baseline system is a Reunion system
- For >=32 instruction intervals, performance impact is negligible

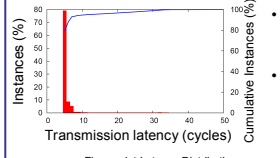
Interconnect bandwidth utilization across fingerprint intervals



- Interconnect bandwidth demands:

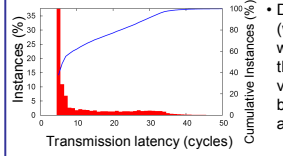
Low bandwidth demands for >=32 instruction intervals corresponds with the effects on IPC

Fingerprint latency distribution. Commercial workload (DSS query 1, 32-cycle FP interval)



- 90% of the transmissions are completed in 9 cycles or less
- Compare to typical (fixed) 10-cycle latencies when using dedicated datapaths

Scientific workload (ocean, 32-cycle FP interval)



- Difference in distribution (wrt commercial workloads) is because of the higher L2 request volume, interconnect bandwidth utilization, and higher IPC

Performance Factors

- **Fingerprint interval**: Affects performance the most. Our design is viable for intervals >=32 instruction fingerprint intervals
- **Workload characteristics**: Workload's L2 bandwidth usage impacts fingerprint transmission via the interconnect

Conclusion

- **On-chip memory interconnects are viable for communicating fingerprints**
- **Re-uses existing hardware infrastructure**

References

- J. C. Smolens, B. T. Gold, B. Falsafi, and J. C. Hoe, Reunion: Complexity-effective multicore redundancy,. in *Proceedings of ACM/IEEE International Symposium on Microarchitecture (MICRO-39)*, Dec 2006.
- J. C. Smolens, B. T. Gold, J. Kim, B. Falsafi, J. C. Hoe, and A. G. Nowatzyk, .Fingerprinting: Bounding soft-error detection latency and bandwidth,. in *Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems*, Oct 2004, pp. 224.234.
- *OpenSPARC T1 Microarchitecture Specification*, Sun Microsystems, Aug 2006.

Electrical & Computer ENGINEERING

www.ece.cmu.edu/~truss

Carnegie Mellon