RAMP-White / FAST-MP

Hari Angepat and Derek Chiou

Electrical and Computer Engineering

University of Texas at Austin

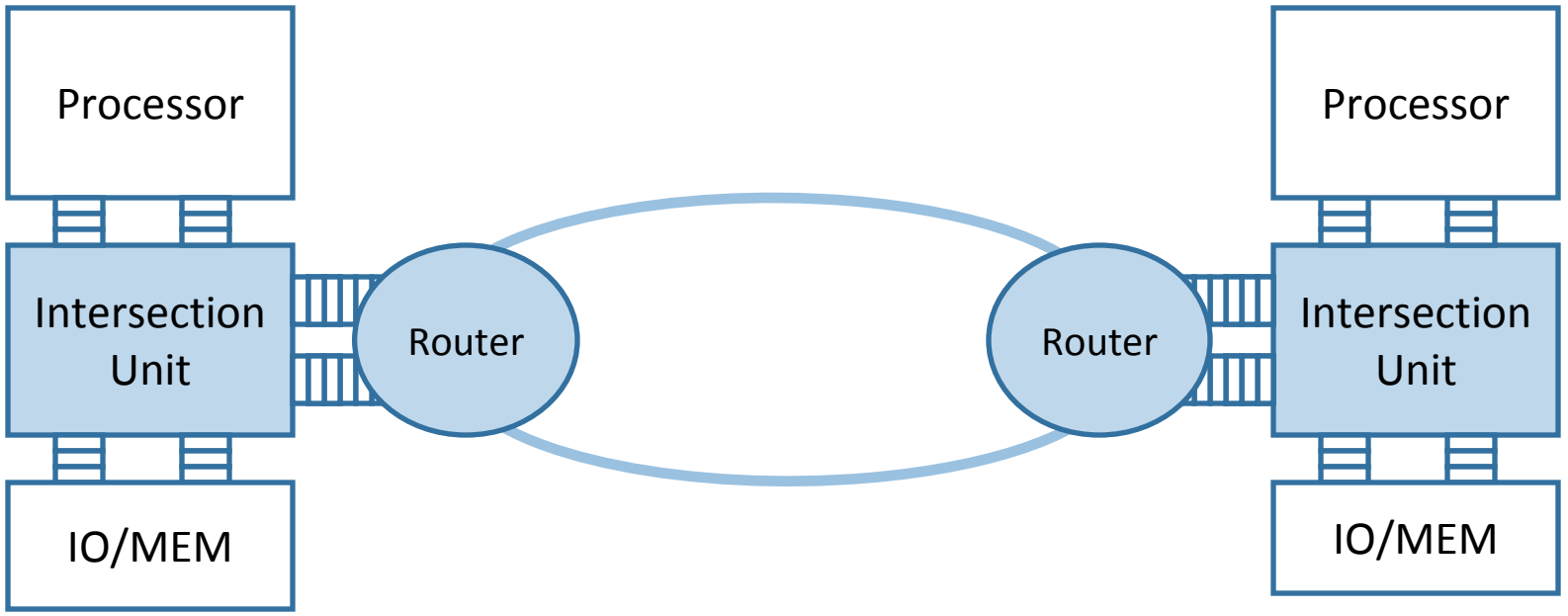
Supported in part by
DOE, NSF, SRC,Bluespec, Intel, Xilinx, IBM, and Freescale

RAMP-White Overview

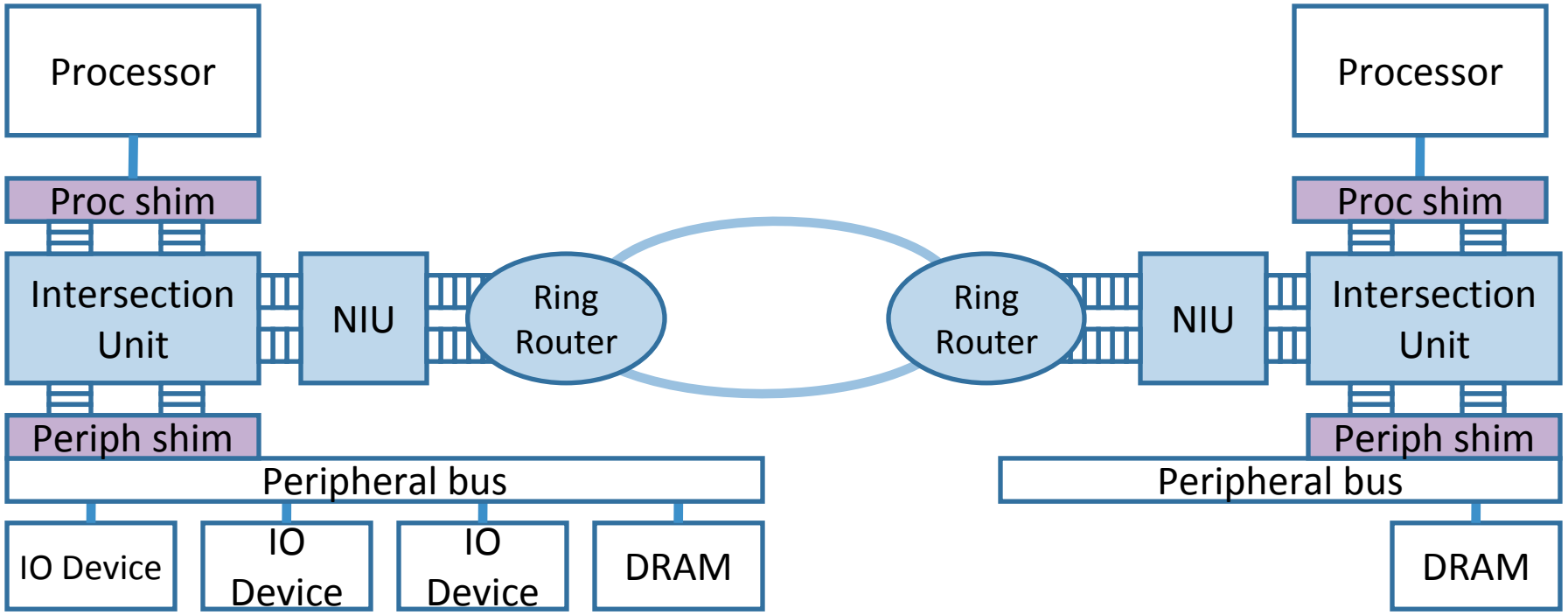
- Use existing FPGA processor implementations to build scalable, flexible, coherent shared memory platforms that run standard operating systems
- Standard ISA/OS enables more complex applications such as software emulators (QEMU) when desired.

RAMP White Architecture

- Classic shared memory machine design

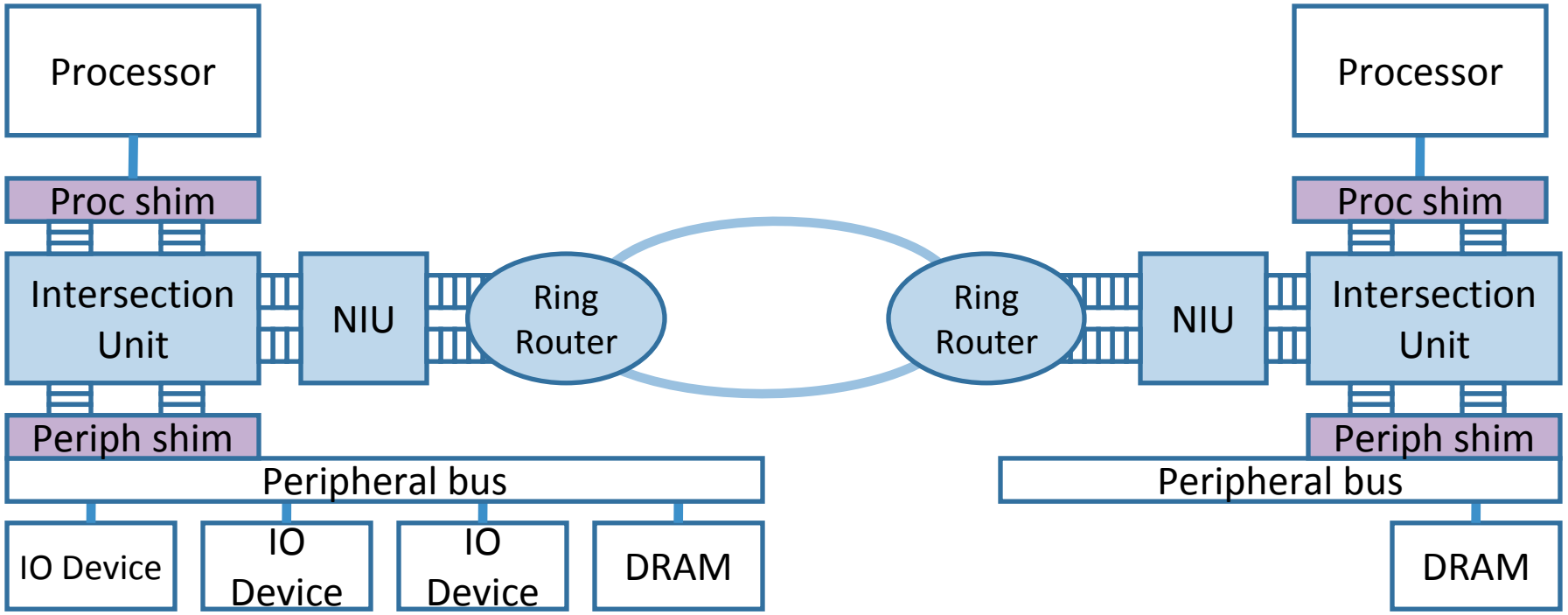


RAMP White Architecture



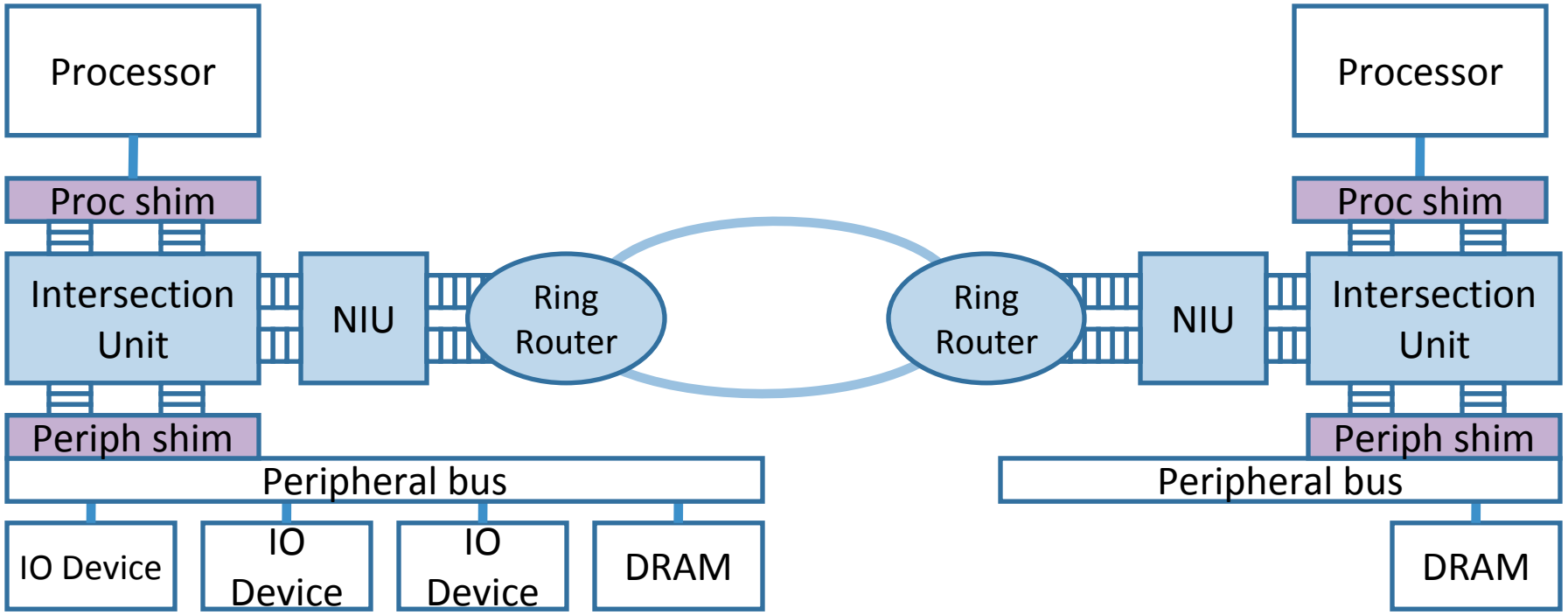
RAMP White Architecture

- Model CMP/SMP targets
  - Coherent shared memory platform
  - Single image OS
- RAMP scalability (1K cores) via spatial and temporal replication



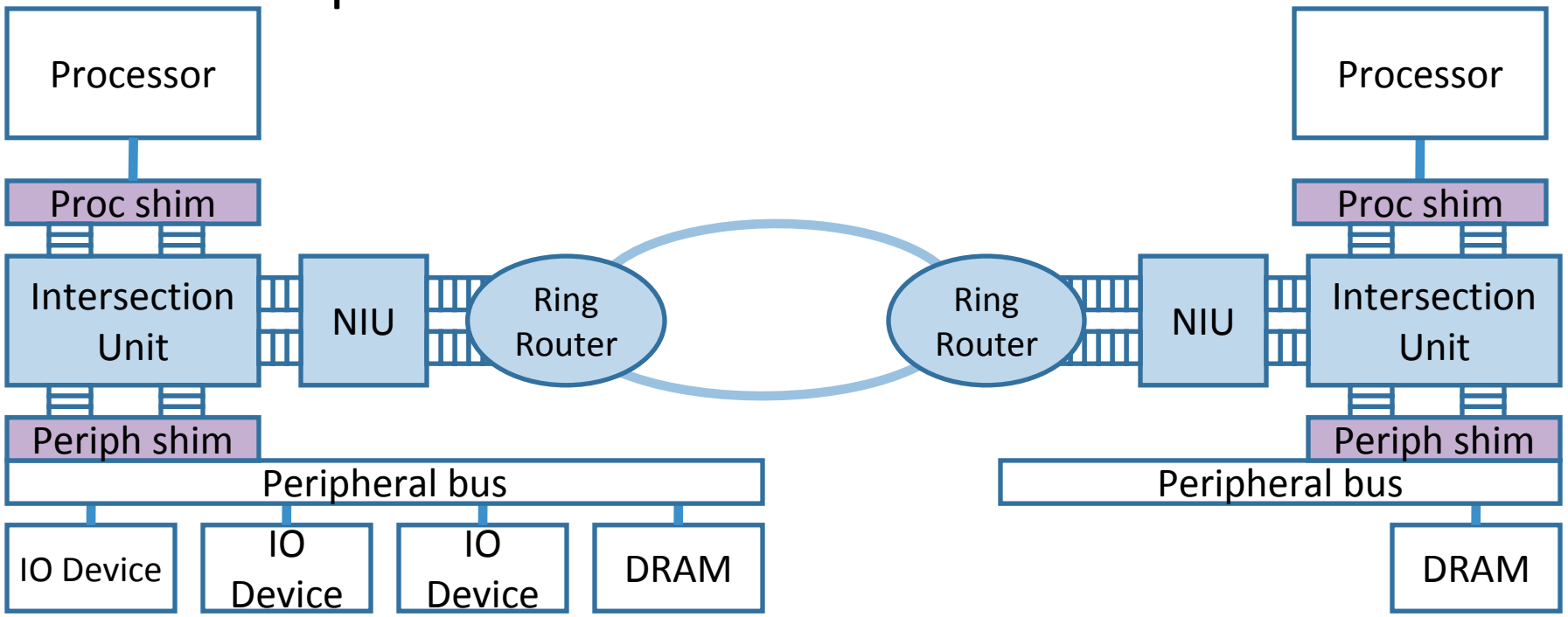
RAMP White Architecture

- Ability to use commodity cores:
  - SparcV8: Leon3 soft-core
  - PowerPC: PPC405 hard-core
  - Configurable coherence protocol, enginesc



RAMP White Architecture

- Configurable modules:
  - NIC, network, coherence engine, intersection unit
- Modules connected by ***Connectors***:
  - Point-to-point FIFOs that can model target time if required
- Shim adapters



RAMP-White Status

- Working:
  - Multi processor Leon
  - Soft-fp kernel and userspace as initramfs
  - Standard pthread Splash benchmarks
- Still debugging:
  - Multichip crossing with scalable interrupt components
  - Integration with parametrizable FAST cache model
- See me during retreat if interested in Alpha release

Prototype (See at Demo)

- Hardware
  - Sparc V8 32bit soft-core processor (Leon3)
  - 50 Mhz core clock, soft-FP, 16KB Icache, Dcache bypassed
  - GRLIB Components {serial, ethernet, ddr, jtag}
- Software
  - Linux SMP 2.6.21 for Leon3
  - Pthread-based Splash2 benchmarks
  - RAM disk rootfs with simple userspace apps
- Platform
  - BEE2 control FPGA with JTAG based programming
  - Ethernet for kernel loading/debugging

FAST-MP

FAST-MP: High Level Goal

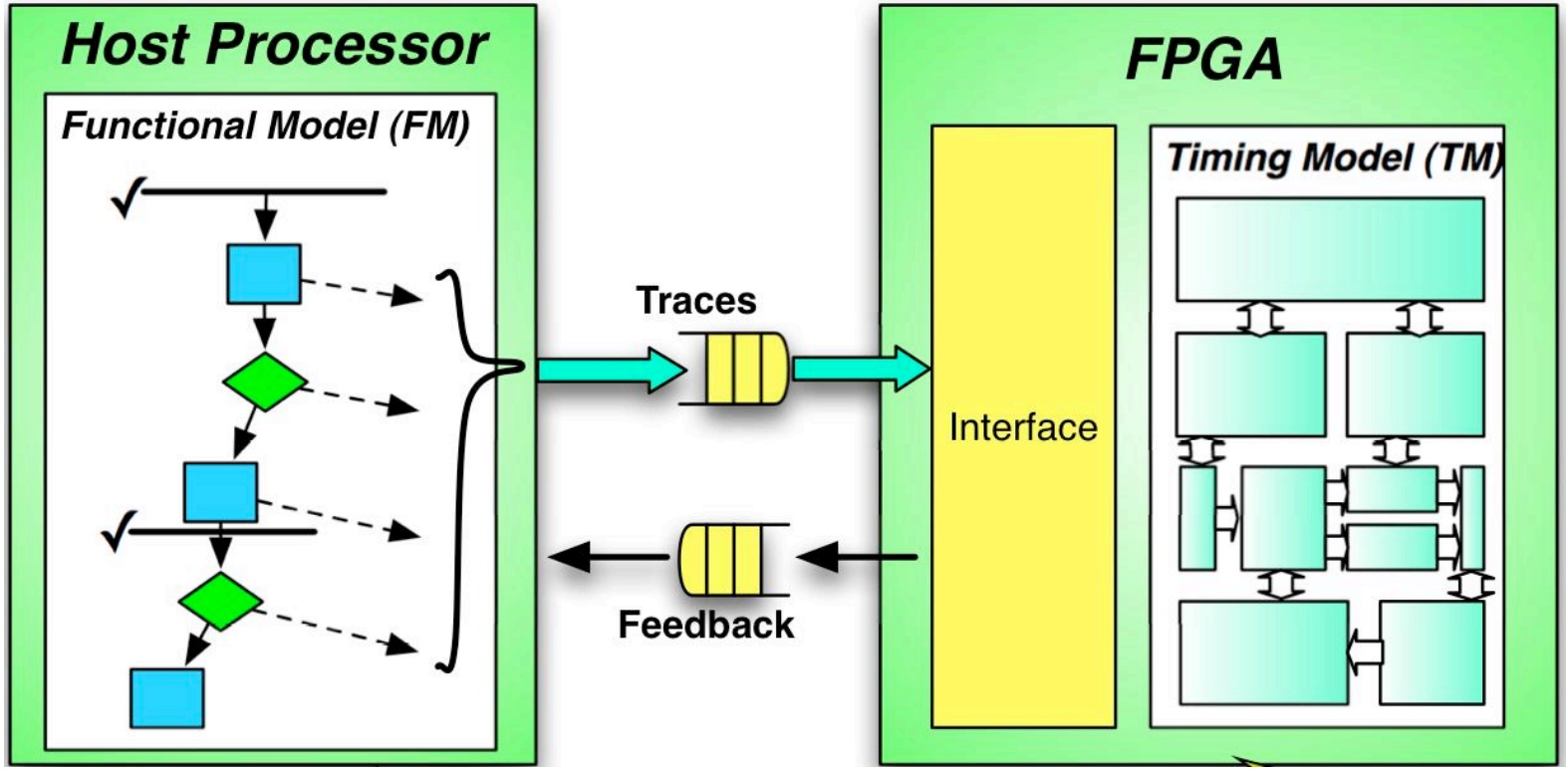
- Multi-resolution coherent shared memory target emulation
  - Predict performance/power for wide range of micro-architectures at accuracies ranging from cycle accurate to functional-only
  - Capable of running real ISAs aided by binary translation (x86, Sparc, PowerPC, etc), operating systems (unmodified Windows, Linux), compilers, applications (SQLServer, Apache, etc)
  - Extensible/flexible (new instructions, different micro architectures)

Performance Modeling on RAMP-White

- RAMP-White host predicts RAMP-White target performance perfectly
  - Predicting performance of arbitrary micro-architectures requires additional support
- FAST (FPGA Accelerated Simulation Techniques) uses a ***timing model*** to predict performance of arbitrary micro-architecture
  - Special purpose structure designed to predict time
    - Very small (complex model in a fraction of an FPGA)
  - Uses same functional model for any micro-architecture
- White as a scalable ***functional model*** for FAST-MP

FAST (FPGA Accelerated Simulation)

- Speculative FM with checkpoint/rollback of FM when FM/TM paths diverge
  - Ex) branch mispredict/resolve



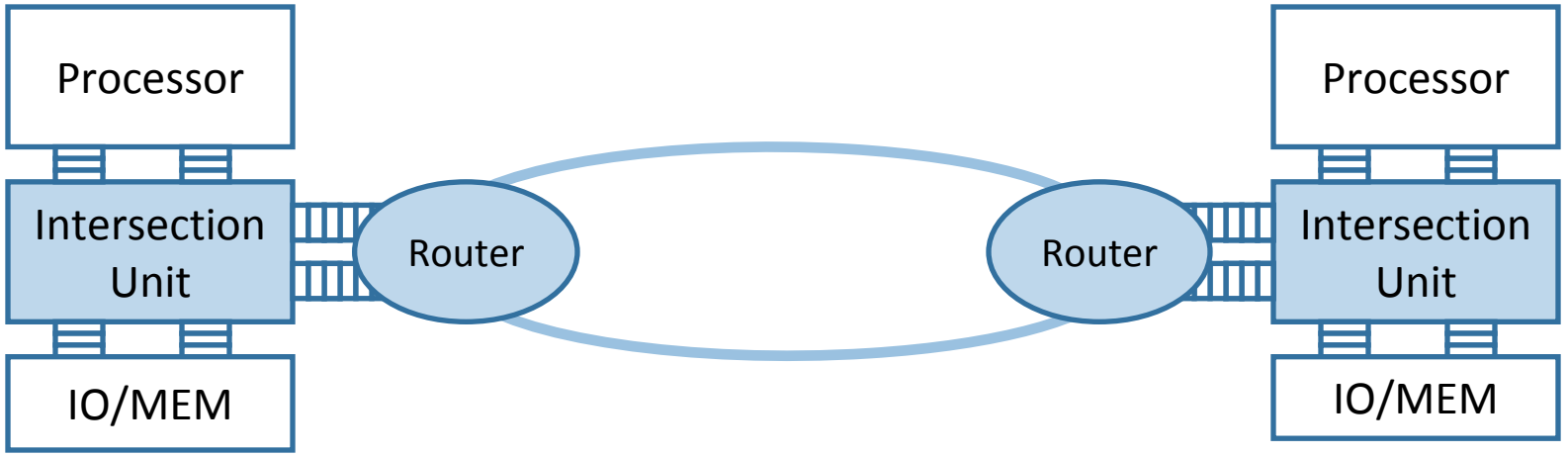
FAST-MP Approach

- Multicore functional model executes as it wishes
  - Functional instruction stream generated (per core) and sent to timing model
  - Rollback when functional model execution differs from timing model
    - Branch mispredictions, address speculation, etc.
- ***Possible for functional model to access memory in different order than target***

FAST-MP Memory Reordering

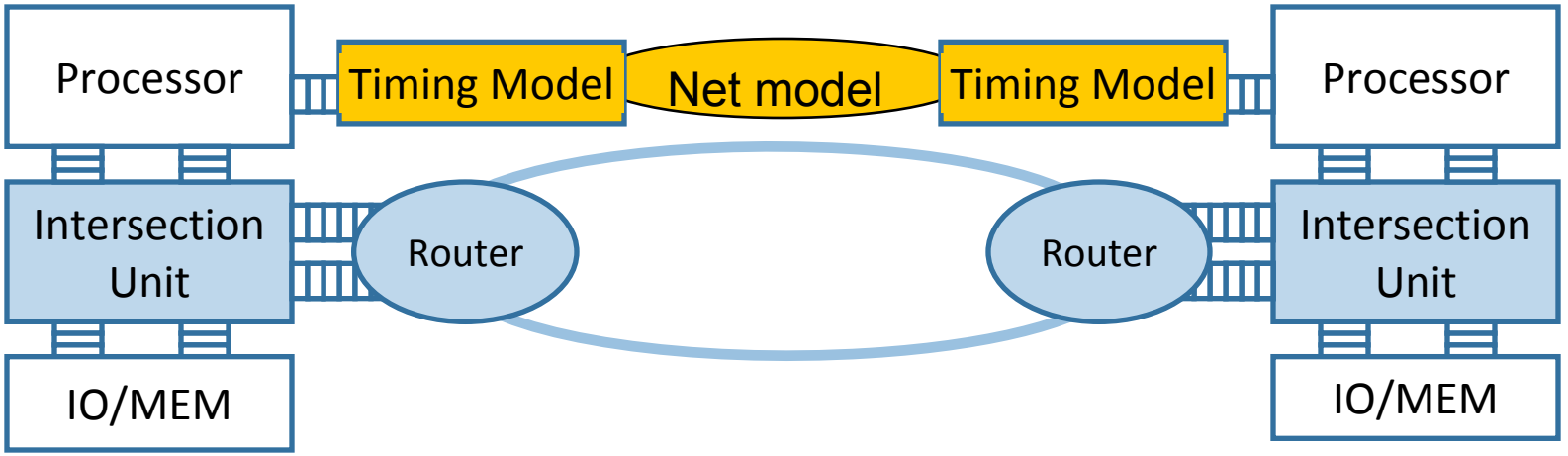
- All memory references tagged with a version number
- FM passes a version number in trace to TM
  - essentially a precondition on the validity of the given trace
- If TM version != FM version
  - Freeze timing models (to avoid corrupting TM)
  - Rollback functional models to restore correct memory/architectural state
  - Use TM directed order to re-execute

White


Processor
Intersection Unit
IO/MEM
Router
Router
Processor
Intersection Unit
IO/MEM

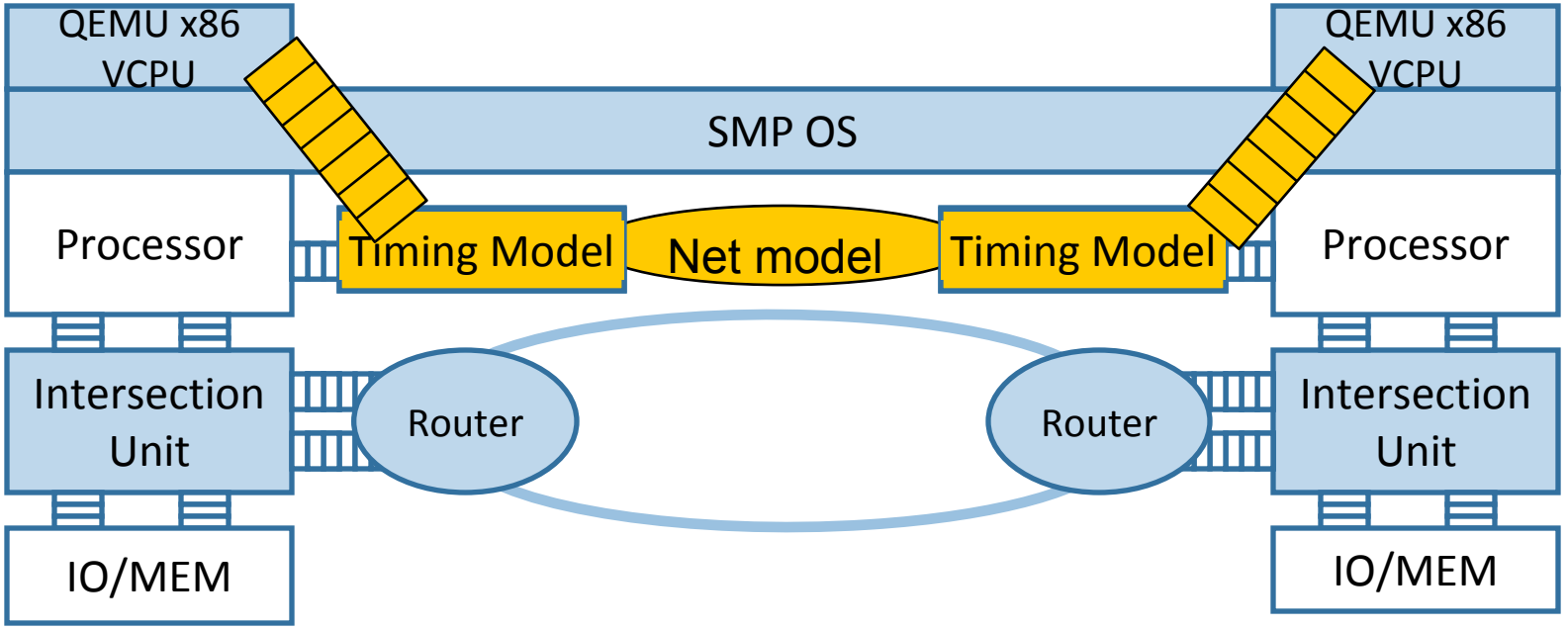
White + Timing Model

- PowerPC/Sparc ISA with arbitrary timing model



White + VM + Timing Model

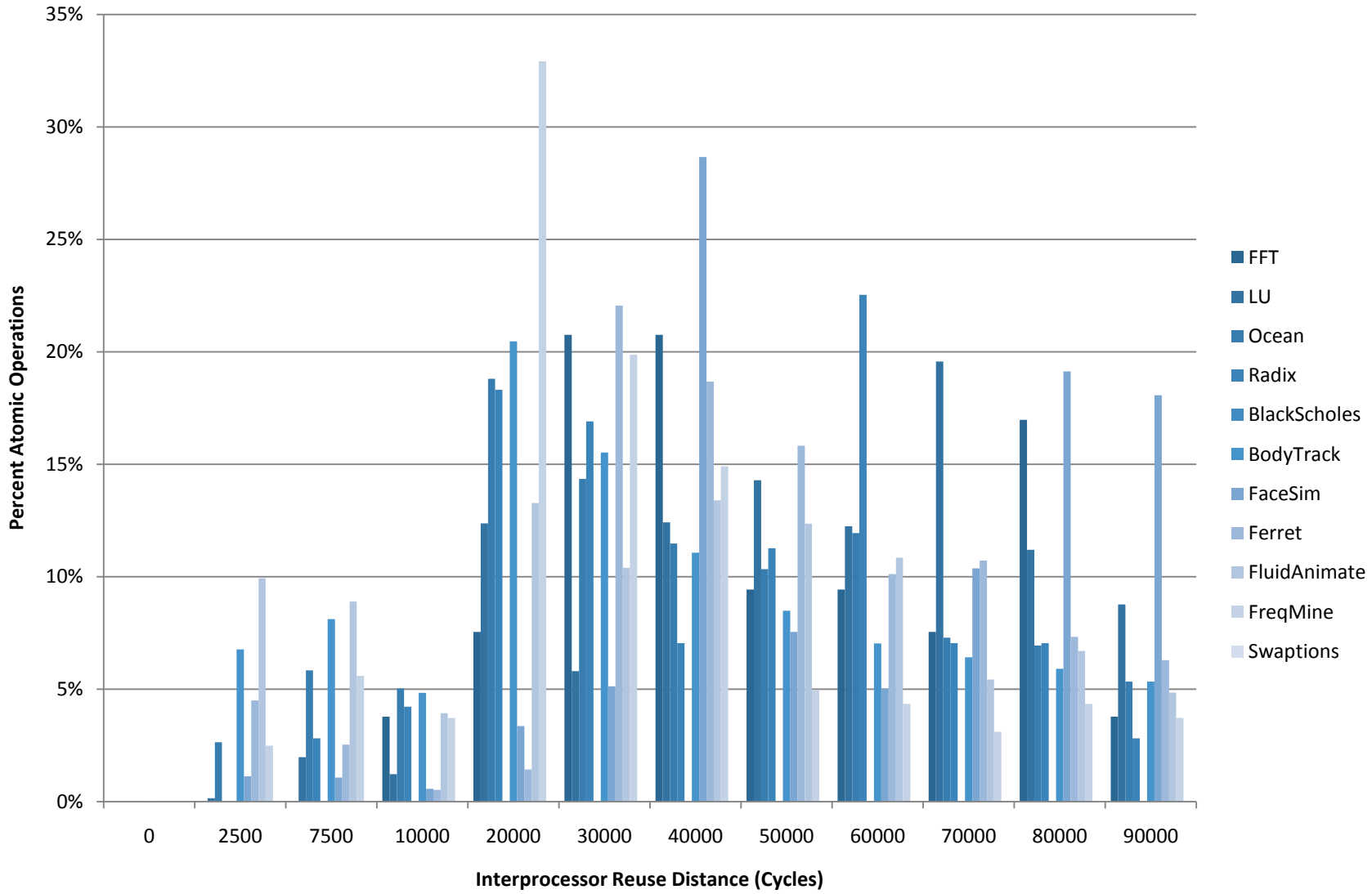
- Sparc ISA with QEMU to emulate any ISA
- Requires trace/rollback:
  - Hardware
  - Software (QEMU) - can also be hardware accelerated



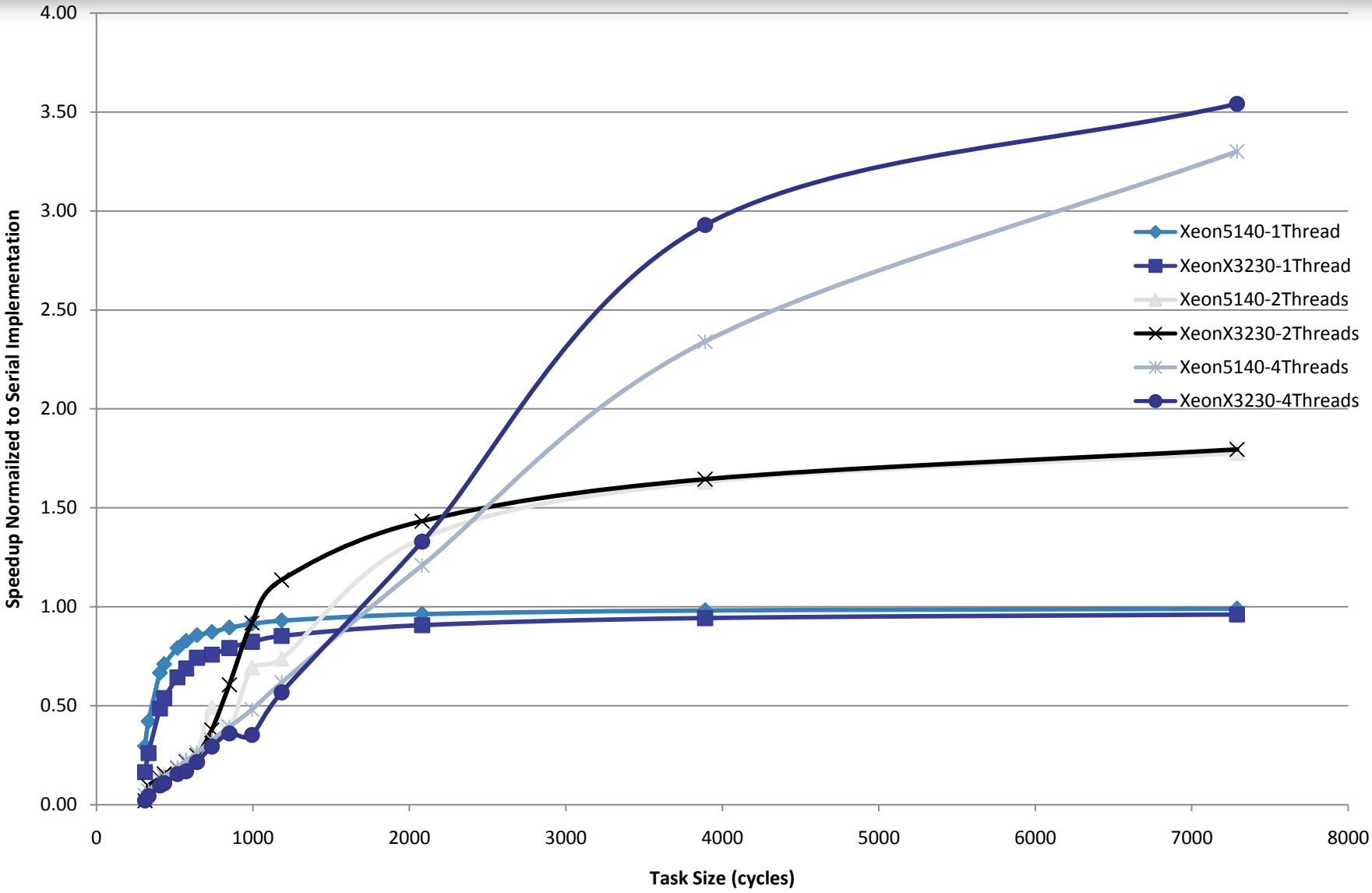
Probability of Reordered Memory Ops

- Functionally-driven speculation in a MP costly if timing ordered memory references conflict
  - Preliminary study with on X86 applications studying atomic operations
  - Use Pin dynamic instrumentation tool to monitor every atomic operation running a multi-threaded app
  - Analyze inter-atomic distance for existing shared memory workloads (Splash2, Parsec)

Interprocessor Atomic Reuse Distance



Task Size Scaling on Intel CMPs



FAST-MP Can Be Less Than Accurate!

- Nearly accurate
  - Functional model backpressured by timing model
    - Don't want to overflow buffers
  - Each functional core roughly at correct instruction relative to other cores
  - Do not rollback to reorder memory operations
    - Still correct, just locks taken in different order
  - Eliminate rollback overheads, probably quite accurate
  - Model RAMP-White on FAST-MP to check accuracy
- Functional + cache
  - Run with just cache simulators
- Etc.

QEMU on White-Leon3

- QEMU 0.9.1 with patches
  - Some issues remaining with Dyngen for V8 ISA with Leon3 cross compiler
- For initial Linux Boot:
  - X86 instructions: 1
  - QEMU uOPs: ~3.1
  - Sparc instructions: ~22.5
    - → High overheads involved in address computation, segmentation checks, software tlb, etc
- Can modify/replace Leon3 to improve efficiency
  - MicroOP-based processor

Conclusions

- Initial RAMP-White Alpha design functional
- FAST-MP
  - Provide various ISAs
  - Cycle-accuracy to purely functional
  - Developing power models
- FAST-MP will run on top of RAMP-White as well as standard multicore system

Questions…