
Motivating Commodity Multi-Core Processor Design for System-Level Error Protection

Nidhi Aggarwal, Parthasarathy Ranganathan[†], Norman P. Jouppi[†],
James E. Smith, Kewal K. Saluja, George Krejci[†]

University of Wisconsin-Madison

[†]Hewlett Packard Labs

SELSE 2007

Outline

- Motivation
 - Tale of two trends
 - Availability features in commodity CMPs
 - Key Insights
 - Design alternatives
 - Proposed CMP design
 - Evaluation and results
 - Future work and conclusions
-

Problem: A tale of two trends

- More transistors \Rightarrow more cores & more integration
 - 2-8 core systems shipping; 32 cores by 2010 (Intel)
 - Integrated memory controller; more integration coming
 - Consequences:
 - Increased integration with lots of shared components
 - System-level availability concerns move up to the socket
- Smaller transistors \Rightarrow less reliability
 - Electrical noise, process variation, natural radiation
 - Soft error FIT/chip has 8% degradation/bit/generation
 - Consequence:
 - more failures, much lower MTTF

High availability with multiple cores that are less reliable?

Availability features in Commodity CMPs

- Analysis of representative systems
 - Five multi-core SMP architectures
 - Intel Xeon, AMD Opteron, Sun Niagara, IBM Power, Intel Montecito
 - Two gold standards
 - IBM zSeries, HP NonStop

Analyzing Commodity CMPs

- Necessary & sufficient conditions for high availability (White et al.)
 - Redundancy
 - Fault detection
 - Fault Isolation
 - Online repair/reconfiguration
- Components in a CMP
 - Core
 - Caches
 - Memory
 - I/O
 - System data and control buses

Details in the paper and technical report

Key Insights

- No fault isolation in shared components
 - Correlated errors can go undetected even in TMR
- Single component failure dramatic
 - What if a core fails?
 - What if the memory controller fails?
 - No reconfiguration support
- High availability solutions are proprietary
 - Custom designed hardware

Enabling High Availability

■ Goals

- Fault isolation
- Reconfiguration support
- Commodity hardware

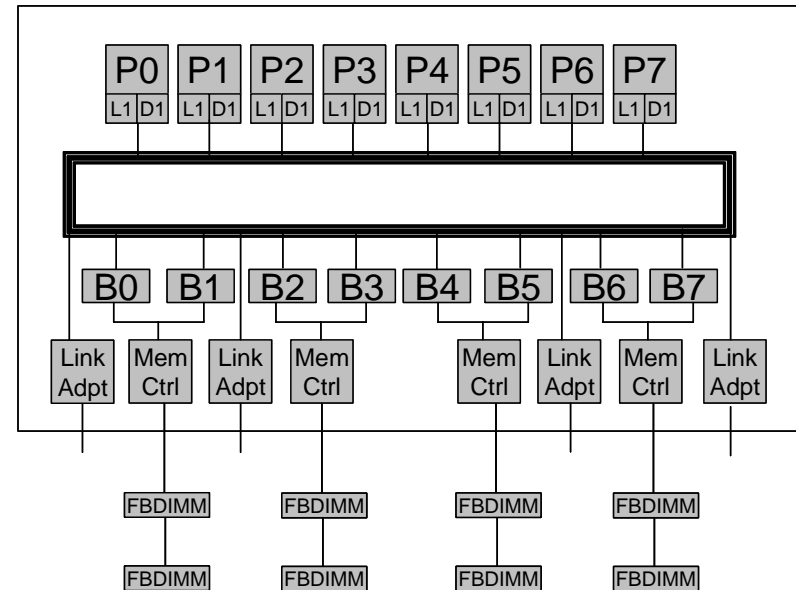
■ Assumption

- Commodity hardware is designed for “pareto points”

Design Alternatives - Current CMP Proposals

■ Shared Design

- ❑ 8 cores on a socket
- ❑ 8 private L1 caches
- ❑ 1 shared L2 cache
- ❑ 2-4 shared memory controllers
- ❑ Shared I/O interface



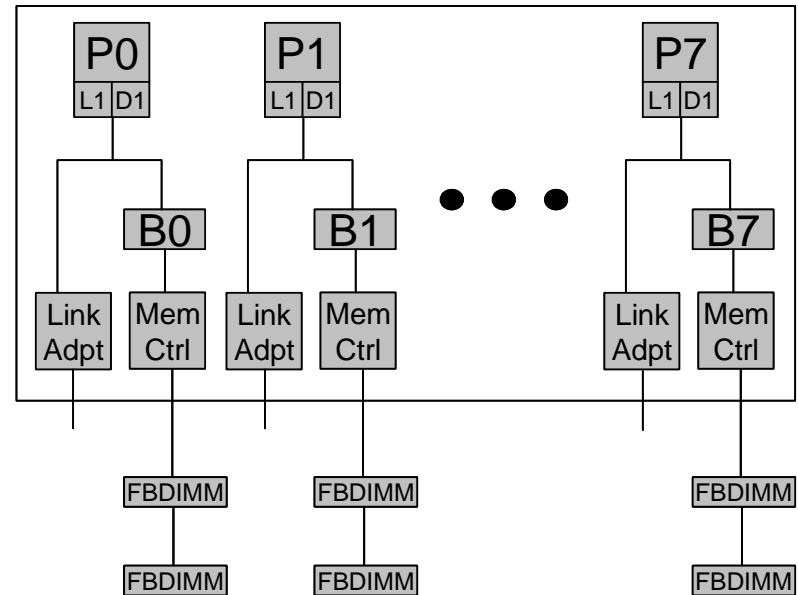
- ❌ Fault isolation
- ❌ Reconfiguration support
- ✅ Commodity hardware

Outline

- Motivation
- Configurable Isolation
 - Concept
 - Implementation
 - Using Configurable Isolation
- Evaluation and results
- Future work and conclusions

Design Alternatives – Custom Design

- Full Isolation Design
 - 8 cores
 - 8 private L1 caches
 - 8 private L2 caches
 - 8 private mem. controllers
 - 8 private interfaces to I/O



- Fault isolation
- Reconfiguration support
- Commodity hardware

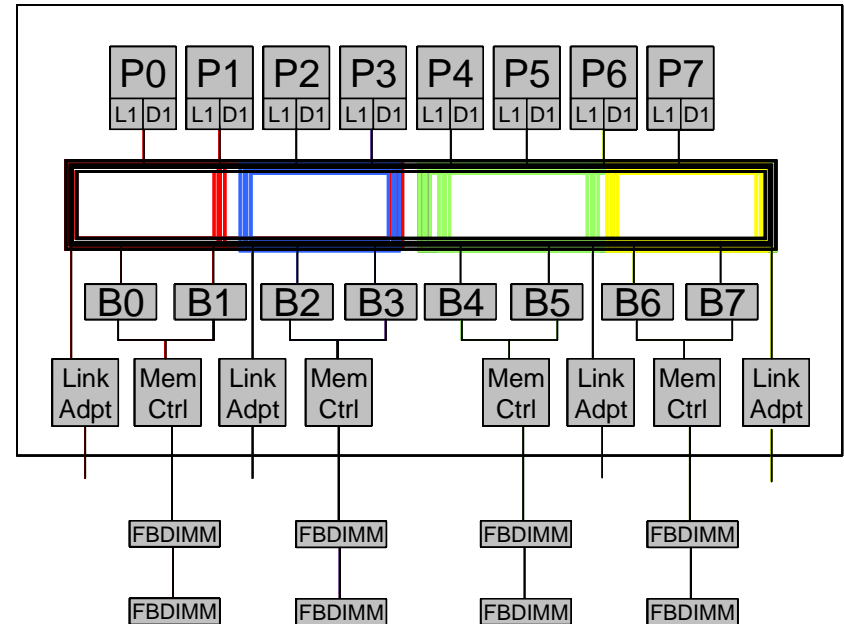
Proposal - Configurable Isolation

A set of techniques that provide optional logical fault isolation for shared components.

- High performance & high availability modes
- High performance mode
 - CMP resources shared for maximum utilization
- High availability mode
 - Degree of sharing configurable for selective isolation

High Availability Mode

- Split CMP resources into domains
- Map domains to different colors
- Color domains are units of fault containment
 - Failure in color-shared component affects same colored cores only
- Number of colors is configurable
 - More redundancy
 - Smaller fault zones



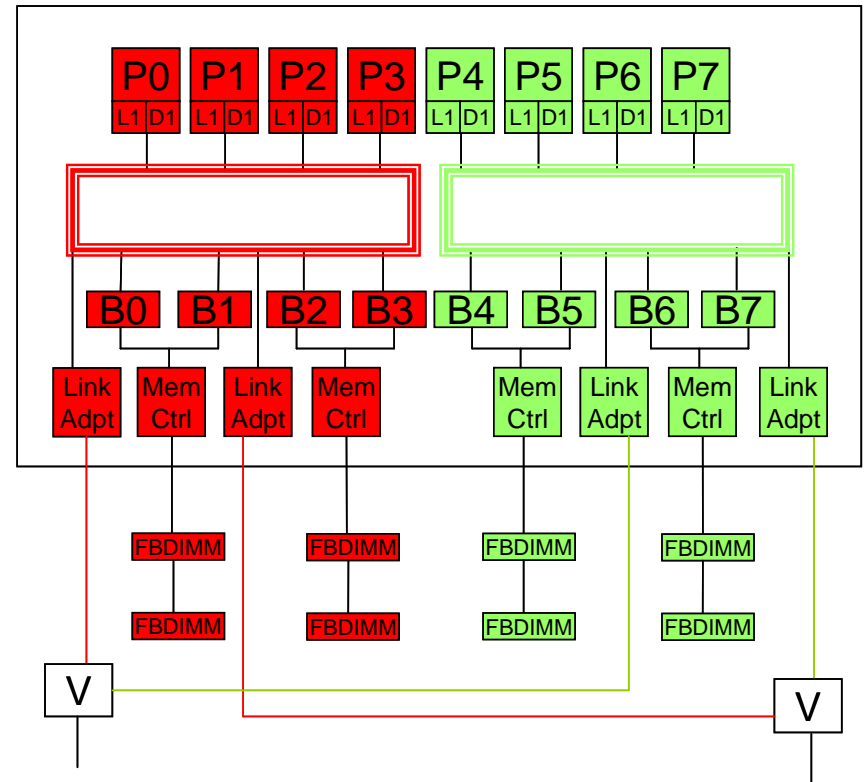
Changes in a Commodity CMP

- Changes to enable transient fault isolation
 - Cross links and input multiplexers in interconnect
 - Ring and bank addressing
- Changes to enable reconfiguration
 - 1st mode bit and tags to cache lines from banks connected to same memory controller
 - 2nd mode bit and tags to cache lines from banks connected to different memory controller
 - Extra tag bits required is $\log_2(\text{number of banks})$
- Estimated area overhead
 - ~1%

Transient Fault Detection Using Configurable Isolation

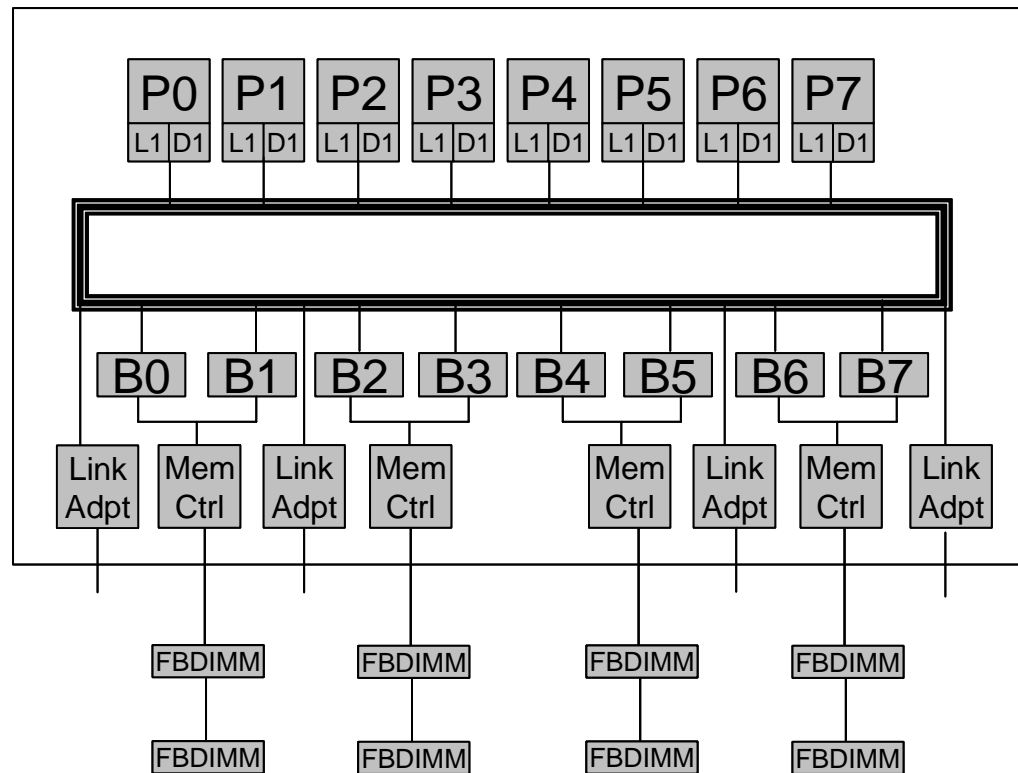
■ NonStop like design

- ❑ 8 cores on a socket
- ❑ 8 private L1 caches
- ❑ Color shared L2 banks
- ❑ Color shared memory controllers
- ❑ Color shared link controllers
- ❑ S/W isolated TLB entries
- ❑ OS assisted memory partitioning
- ❑ Voters in I/O hub or hypervisor



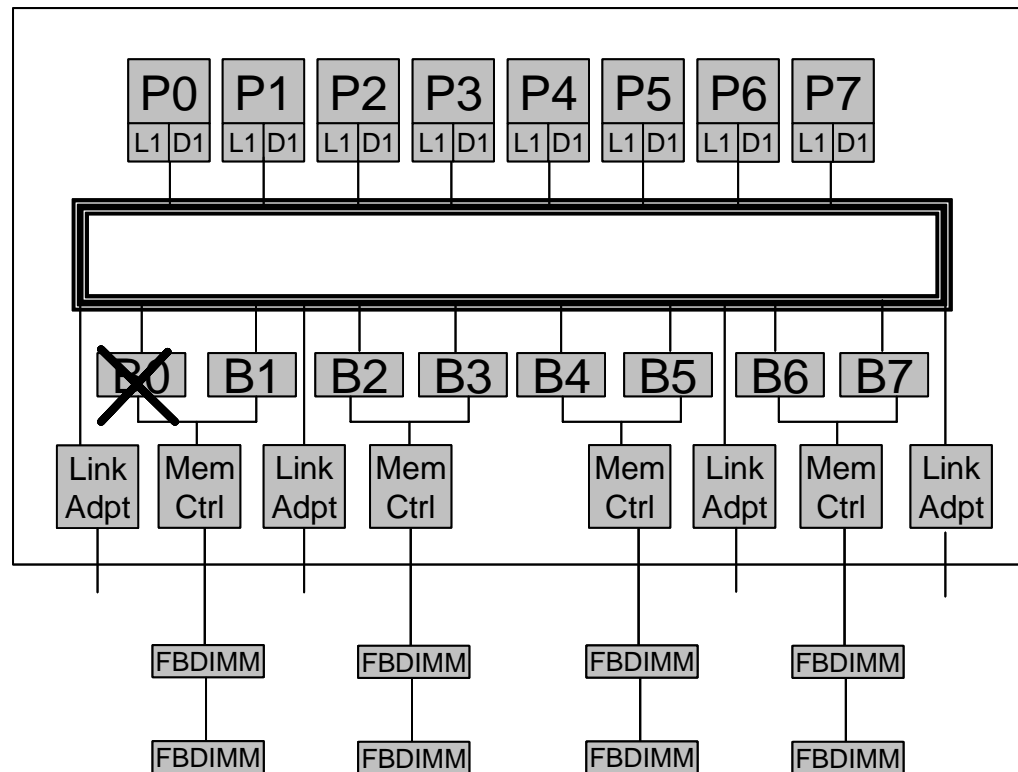
Using Configurable Isolation

- Map out faulty components



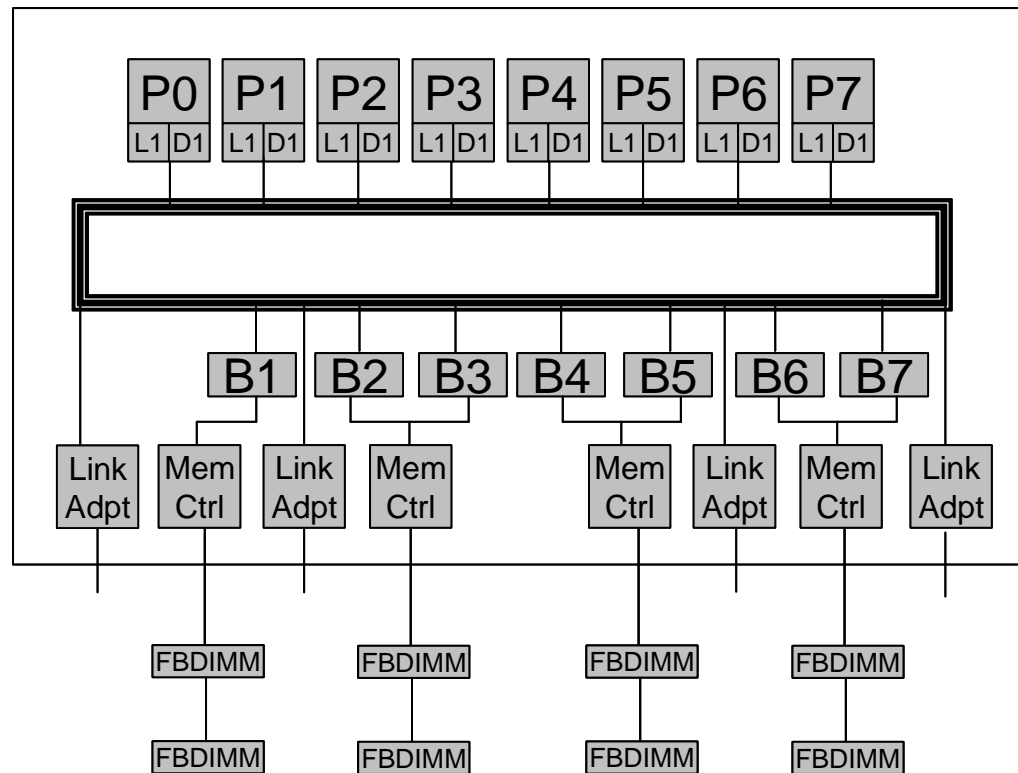
Using Configurable Isolation

- Map out faulty components



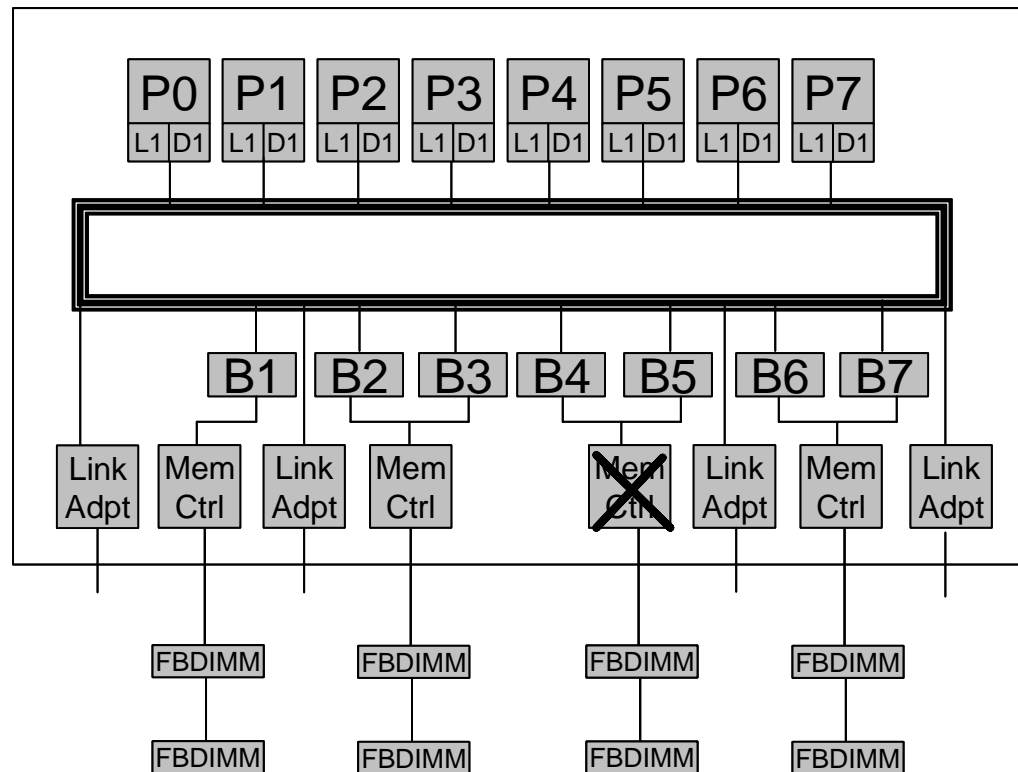
Using Configurable Isolation

- Map out faulty components



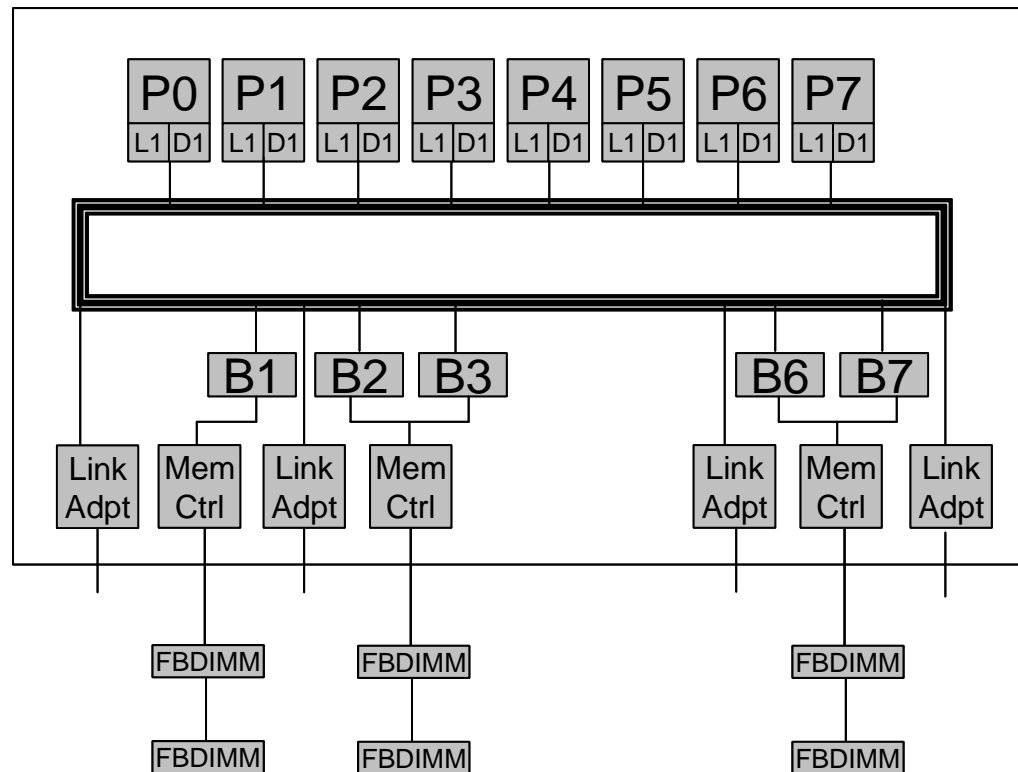
Using Configurable Isolation

- Map out faulty components



Using Configurable Isolation

- Map out faulty components



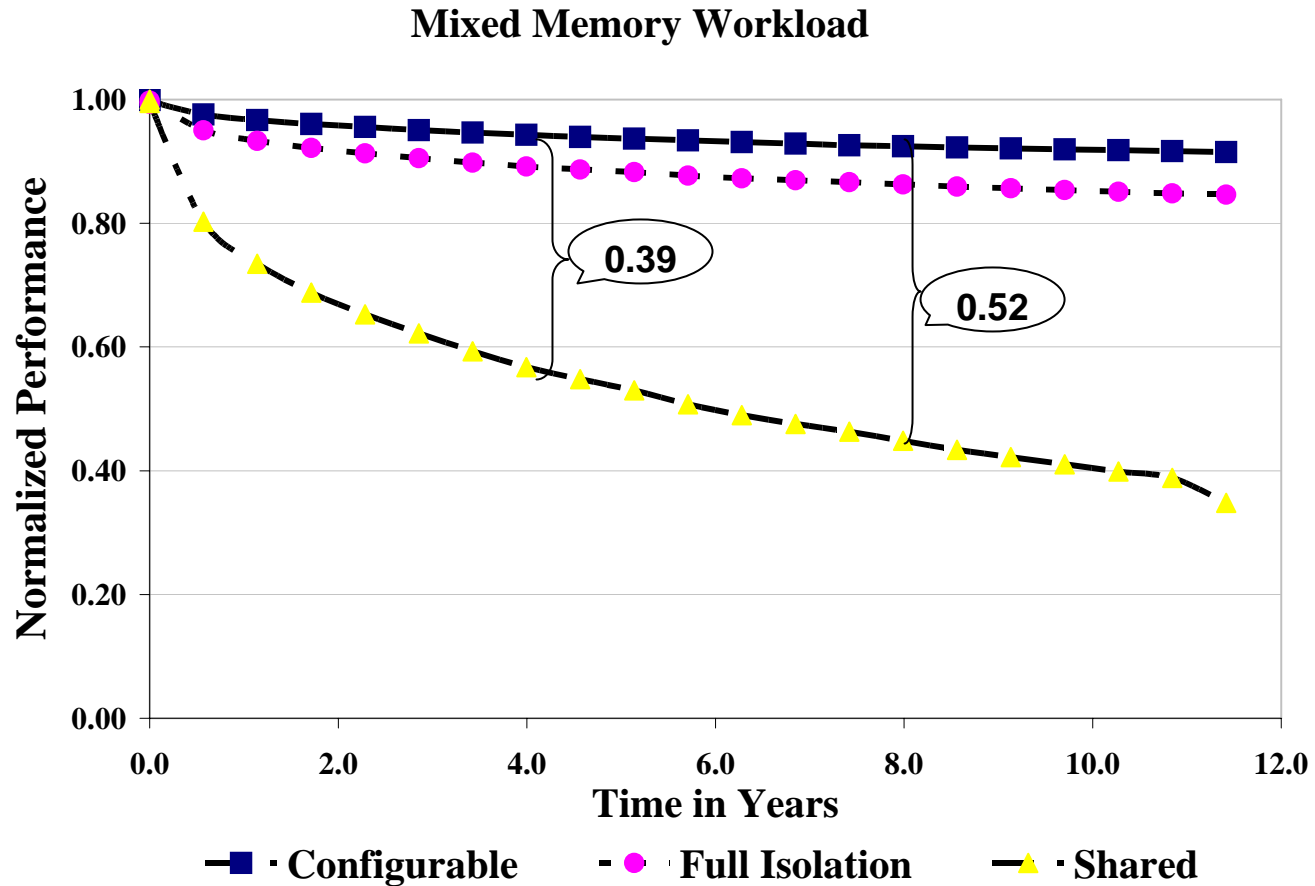
Configurable Isolation

- ☑ Fault isolation
- ☑ Reconfiguration support
- ☑ Commodity hardware

Evaluation Methodology

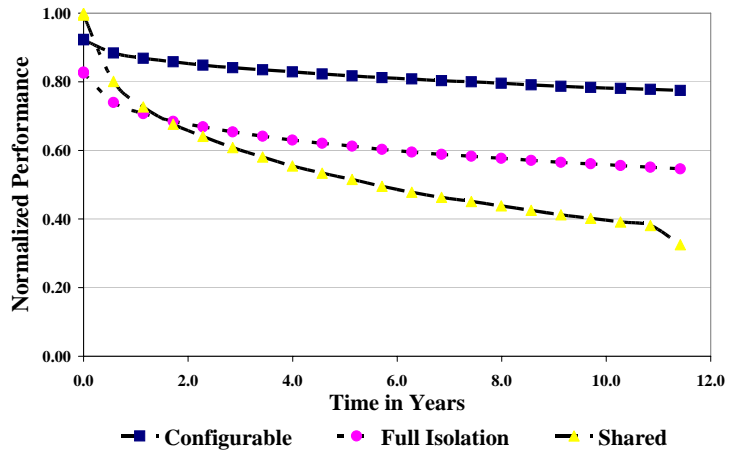
- **Impact of hard faults**
 - Measure system computing capacity over 100,000 hours
 - Three architectures– shared, full isolation & configurable isolation
- **Three workloads**
 - Large, mixed and small memory footprint (based on cache capacity)
- **State of the art industrial fault model**
 - FIT rates and distributions for various components
- **Two phase simulation methodology**
 - Exhaustive full system simulation of system configurations
 - Monte Carlo simulation for fault injection
 - 10,000 runs for 100,000 simulated hours each (approx. 11 years)

Example Results

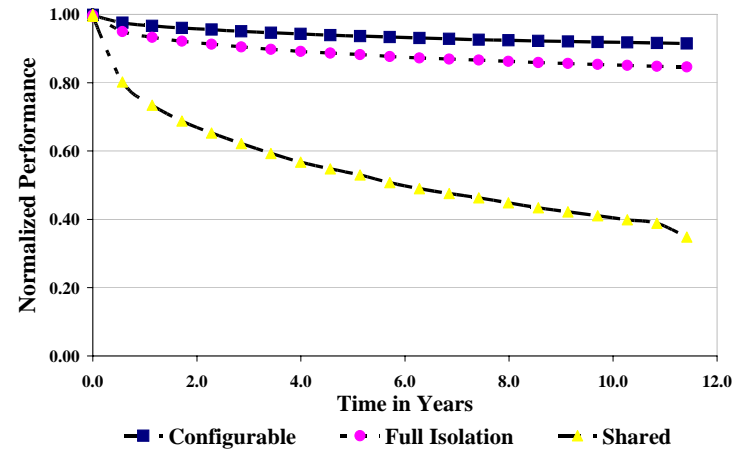


Results

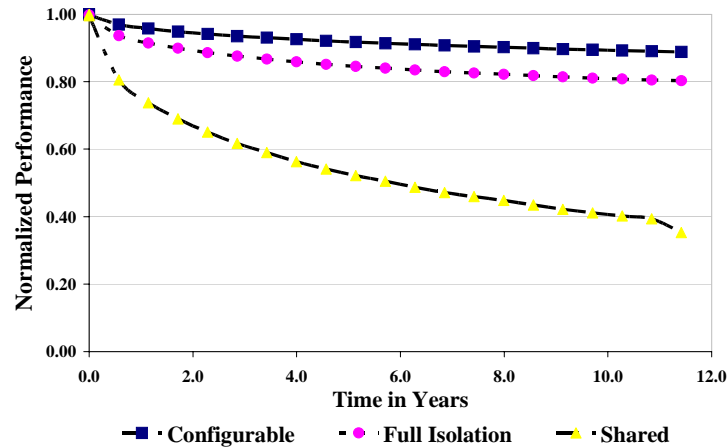
Large Memory Workload



Mixed Memory Workload



Small Memory Workload



Future Work

- More detailed evaluation
- Enhancements to configurable isolation
 - Multiple processes per core
 - Dynamic power reprovisioning from unused cores
 - Topology aware reconfiguration
 - Workload requirement aware reconfiguration

Summary

- Commodity CMP designs focus on high resource utilization
- Motivate CMP design for system level error protection
- Our approach:
 - Configurable isolation in commodity CMPs
 - Provides isolation from correlated faults
 - Ability to reconfigure after hard faults
 - Incremental overhead to enable high availability ~1%

Backup Slides

Summary of current CMP availability features

■ Cores

- ❑ Soft error detection limited to register files with ECC
- ❑ No fault isolation in Opteron, Xeon and Niagara
- ❑ Limited isolation in Montecito and Power5 in electrical and logical partitions
- ❑ All architectures susceptible to soft errors in logic
 - Montecito in lock-step configuration is an exception

Summary of current CMP availability features

■ Caches

- All architectures share at least one level of cache
- Resilient to errors in array
 - ECC or parity checks at all cache levels
- No tolerance to multi-bit errors in Opteron and Xeon
 - Niagara, Power5 & Montecito handle some classes of multi-bit errors
- No tolerance to errors in cache circuitry or interconnect
 - Entire socket susceptible to transient fault in cache controller state machine

Summary of current CMP availability features

■ Memory

- Most fault tolerant resource
- All conditions for high availability typically satisfied in arrays
- Sophisticated techniques present in all architectures
 - Chip kill, background scrubbing, DIMM sparing
- Memory access control circuitry unprotected
- Some architectures better than others
 - Shared Northbridge more vulnerable than on-chip controllers

HP NonStop

Component	Redundancy	Fault Isolation	Fault detection	Online repair
Core	Dual or Triple Modular Redundancy	Isolated to a processor.	Compare results of I/O.	Reintegration of new processing element, dedicated reintegration link
Cache	Dual or Triple Modular Redundancy	Isolated to cache	Compare results of I/O.	Replace processor slice.
Memory	Dual or Triple Modular Redundancy	Isolated to memory. No shared memory used. Only communication through SAN.	Compare results of I/O. Symmetric handling of interrupts to keep memory across redundant Processing elements coherent.	Replace processor slice
I/O	Dual redundant SAN	Independent fabrics	Self checked circuits	Online replacement of Logical Synchronization Units
System Data and Control Buses	Redundant buses	Independent buses	CRC checksums	No information available

zSeries

Component	Redundancy	Fault Isolation	Fault detection	Online repair
Core	8 spare processors spread across four books	Processor checkstops on failure. Both processors must stop because of the shared interface.	Mirrored pipeline. Register files have ECC or parity with retry. Checkpoint in separate ECC protected register file.	Dynamic Core sparing, checkpoint at each instruction boundary, Concurrent Book Add, checkpoint can be transplanted in to a dormant spare processor.
Cache	Active redundant L2 cache, redundant L2 rings	Special uncorrectable error codes	L1 - parity protected, L2 - ECC protected, ECC words don't cover words in close proximity, XOR checking of control signals from L2 chips, retry on error detection	New cache can be added at the L2 ring interface
Memory	2 spare DRAM chips for each card, two main storage controller, 4 physical copies of store protect keys	Erroneous DRAM chip and store key is not used	TMR for store keys, ECC and memory scrubbing for memory, extra ECC code space, special Uncorrectable Error Codes to indicate of error originated other than in memory	Concurrent Book Add, Chipkill ECC
I/O	Three Memory Bus Adapters per book, resources shard across all partitions	Single Memory Bus Adapter clockstep design.	Parity protected and command reject request in case of error, forced hang if requestor id broken, special UE for known uncorrectable data	Operation retries, concurrent book add
System Data and Control Buses	Redundant buses	Independent buses, immediate checkstop on UE on control bus, cleanly generate ECC when erroneous data travels across interfaces.	Parity, parity protected tag bit sent with uncorrectable data	Call for repair if errors exceed a threshold

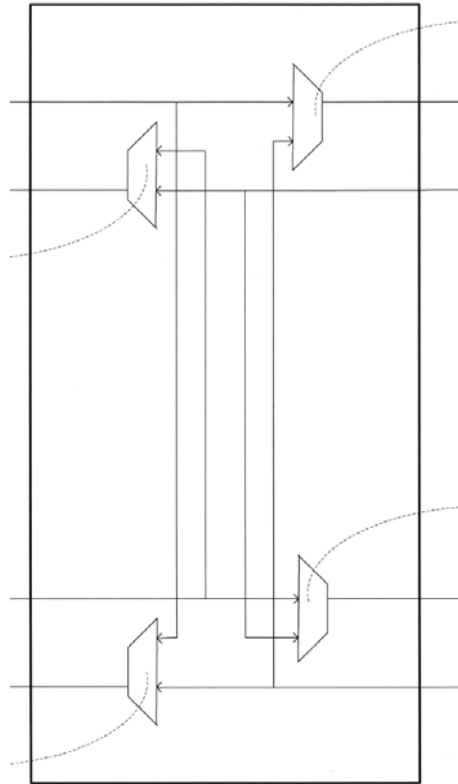
Cache

CMP type	Redundancy	Fault Isolation	Fault detection	Online repair
AMD Opteron	None beyond ECC	None beyond ECC	L1 data cache - ECC, L2 data and tags - ECC, L1 instruction cache - parity, machine check exception	L1 data cache - ECC, L2 data and tags- ECC, L1 instruction cache – parity
SUN Niagara	None beyond ECC	L2 banks are independent	I cache - parity, Dcache - parity, L2 cache - ECC, L2 cache scrubbers	Rollback of instructions following erroneous load, retry
Intel Xeon	None beyond ECC	None beyond ECC	ECC	ECC
IBM Power 5	Spare bits, spare lines	Faulty portions deconfigured, bit steering, cache line delete	Parity at L1 and ECC at L2	ECC, bit steering, invalidate and retry
Intel Montecito	ECC	Intel Cache Safe technology (Pellston) – disable L3 cache lines with errors	ECC	ECC

Core

CMP type	Redundancy	Fault Isolation	Fault detection	Online repair
AMD Opteron	None - but all cores are similar	None	None except parity in TLB	None
SUN Niagara	None - but all cores are similar. Only one floating point unit.	None	Each thread has own error reporting registers, multiple types of error traps depending on the origin of error, TLBs have parity, register files - ECC	Software scrubbing for DTLB, replay instruction from Write stage for correctable errors in register files
Intel Xeon	None - but all cores are similar	None	None, lockstep support for some vendors	None
IBM Power 5	One unlicensed CPU	Faulty CPUs are de-configured, in LPAR only the affected partition is terminated	Extensive error checking including parity for all intermediate results, error thrown to firmware and OS	Dynamic processor sparing
Intel Montecito	None - but all cores are similar	Hardware partitioning, electrically isolated partitions	Lockstep support, internal logic soft error checking	None

Ring Configuration Unit



Ring Configuration Unit (RCU)
Internal Design