



Piranha

A Scalable Architecture Based on
Single-Chip Multiprocessing

Luiz André Barroso
Kourosh Gharachorloo
Robert McNamara
Andreas Nowatzky
Shaz Qadeer
Barton Sano
Scott Smith
Robert Stets
Ben Verghese



COMPAQ

Increasing Complexity of Processor Designs

- Pushing limits of instruction-level parallelism
 - multiple instruction issue
 - speculative out-of-order (OOO) execution
- Driven by applications such as SPEC
- Increasing design time and team size

Processor (SGI MIPS)	Year Shipped	Transistor Count (millions)	Design Team Size	Design Time (months)	Verification Team Size (% of total)
R2000	1985	0.10	20	15	15%
R4000	1991	1.40	55	24	20%
R10000	1996	6.80	>100	36	>35%

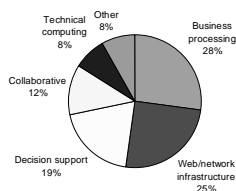
courtesy: John Hennessy, IEEE Computer, 32(8)

- Yielding diminishing returns in performance



Importance of Commercial Applications

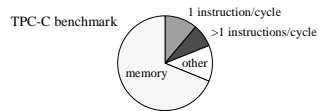
Approximate Breakdown of
Server Market Revenue by Workload in 1998 (IDC)



- Total server market size in 1998: ~\$50-60B
 - technical applications: less than \$5B
 - commercial applications: over \$35B



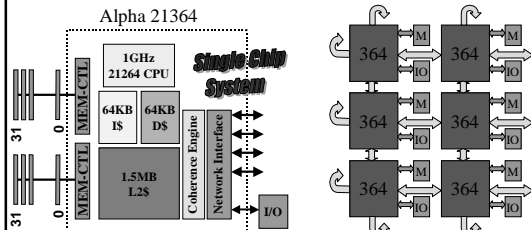
Challenges for Commercial Applications



- Memory system dominant factor in overall performance [Barroso et al., ISCA'98]
- Small gains from multiple instruction issue and OOO execution [Ranganathan et al., ASPLOS'98]
- No use for floating-point and multimedia functionality
- Further questions viability of more complex processors

Q

Exploiting Higher Levels of Integration

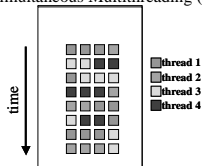


- lower latency, higher bandwidth benefits: [Barroso et al., HPCA'00]
- reuse of existing CPU core addresses complexity issues
- incrementally scalable glueless multiprocessing

Q

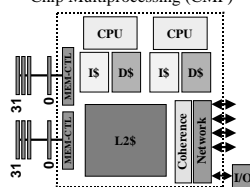
Exploiting Parallelism in Commercial Apps

Simultaneous Multithreading (SMT)



Example: Alpha 21464

Chip Multiprocessing (CMP)



Example: Hydra, IBM Power4

- SMT superior in single-thread performance database performance of SMT: [Lo et al., ISCA'98]
- CMP addresses complexity by using simpler cores

Q

Piranha Project

- Explore chip multiprocessing for scalable servers
- Focus on parallel commercial workloads
- Small team, modest investment, short design time
- Address complexity by using:
 - simple processor cores
 - standard ASIC methodology
- Piranha's CMP approach extremely compelling for server workloads with explicit thread-level parallelism



Outline

- Background
- Piranha Architecture
- Performance Evaluation
- Summary



Piranha Processing Node

CPU

Alpha core:
1-issue, in-order,
500MHz



Piranha Processing Node

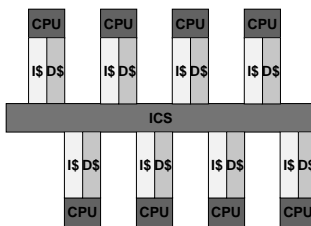


Alpha core:
1-issue, in-order,
500MHz
L1 caches:
I&D, 64KB, 2-way



a

Piranha Processing Node

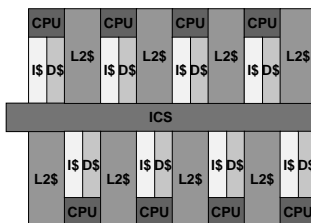


Alpha core:
1-issue, in-order,
500MHz
L1 caches:
I&D, 64KB, 2-way
Intra-chip switch (ICS)
32GB/sec, 1-cycle delay



a

Piranha Processing Node

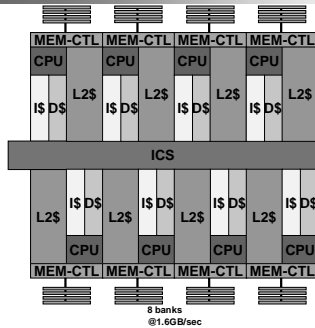


Alpha core:
1-issue, in-order,
500MHz
L1 caches:
I&D, 64KB, 2-way
Intra-chip switch (ICS)
32GB/sec, 1-cycle delay
L2 cache:
shared, 1MB, 8-way



a

Piranha Processing Node

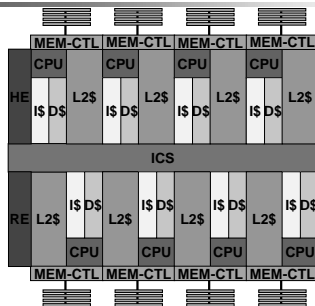


Alpha core:
1-issue, in-order,
500MHz
L1 caches:
I&D, 64KB, 2-way
Intra-chip switch (ICS)
32GB/sec, 1-cycle delay
L2 cache:
shared, 1MB, 8-way
Memory Controller (MC)
RDRAM, 12.8GB/sec

8 banks
@1.6GB/sec

a

Piranha Processing Node

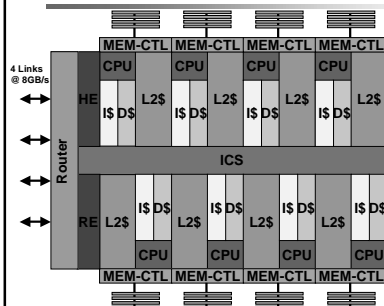


Alpha core:
1-issue, in-order,
500MHz
L1 caches:
I&D, 64KB, 2-way
Intra-chip switch (ICS)
32GB/sec, 1-cycle delay
L2 cache:
shared, 1MB, 8-way
Memory Controller (MC)
RDRAM, 12.8GB/sec
Protocol Engines (HE & RE):
µprog., 1K µinstr.,
even/odd interleaving

8 banks
@1.6GB/sec

a

Piranha Processing Node

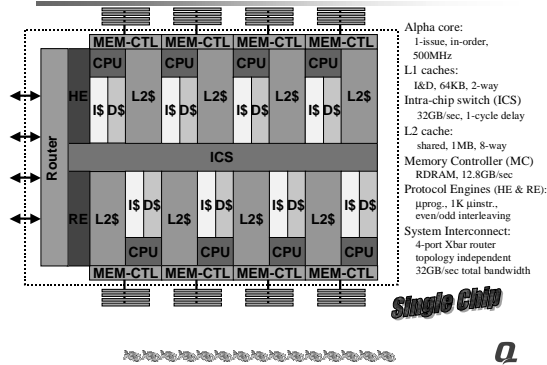


Alpha core:
1-issue, in-order,
500MHz
L1 caches:
I&D, 64KB, 2-way
Intra-chip switch (ICS)
32GB/sec, 1-cycle delay
L2 cache:
shared, 1MB, 8-way
Memory Controller (MC)
RDRAM, 12.8GB/sec
Protocol Engines (HE & RE):
µprog., 1K µinstr.,
even/odd interleaving
System Interconnect:
4-port Xbar router
topology independent
32GB/sec total bandwidth

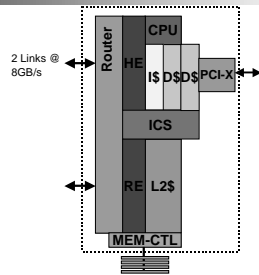
8 banks
@1.6GB/sec

a

Piranha Processing Node



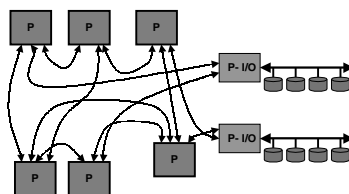
Piranha I/O Node



- I/O node is a full-fledge member of system interconnect
 - CPU indistinguishable from Processing node CPUs
 - participates in global coherence protocol

a

Example Configuration



- Arbitrary topologies
- Match ratio of Processing to I/O nodes to application requirements

a



L2 Cache and Intra-Node Coherence

- 8 banks based on cache line address interleaving
- No inclusion between L1s and L2 cache
 - total L1 capacity equals L2 capacity
 - L2 misses go directly to L1
 - L2 filled by L1 replacements
- L2 keeps track of all lines in the chip
 - sends Invalidates, Forwards
 - orchestrates L1-to-L2 write-backs to maximize chip-memory utilization
 - cooperates with Protocol Engines to enforce system-wide coherence



a

Protocol Characteristics

- ‘Stealing’ ECC bits for memory directory
 - $8 \times (64+8)$ $4 \times (128+9+7)$ $2 \times (256+10+22)$ $1 \times (512+11+53)$
 - 
 - 0 28 44 53
 - Data-bits
ECC
Directory-bits
- Directory (2b state + 40b sharing info)
 - 
 - 2b 20b 2b 20b
- Dual representation: limited pointer + coarse vector
- “Cruise Missile” Invalidations (CMI)
 - limit fan-out/fan-in serialization with CV
- Several new protocol optimizations



a

Outline

- Background
- Piranha Architecture
- Performance Evaluation
- Summary



a

Experimental Methodology

- Workloads
 - TPC-B: 600MB SGA, 500 transactions, 8 processes/CPU
 - TPC-D (Q6): 500 MB SGA, 4 processes/CPU
 - Oracle DBMS
- Simulation Environment: SimOS-Alpha
 - full system simulation (includes OS)
 - CPU models:
 - single-issue, in-order, blocking caches
 - out-of-order speculative OOO, non-blocking caches
 - Memory system:
 - shared L2 cache
 - NUMA model



a

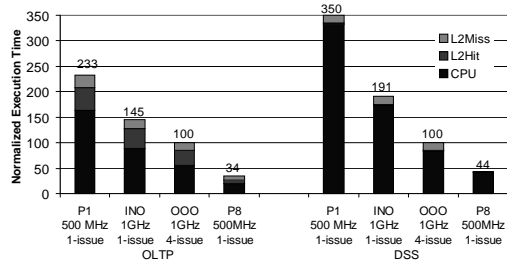
Simulated Architectures

Parameter	Piranha (P8)	Next-Generation Microprocessor (OOO)	Full-Custom Piranha (P8F)
Processor Speed	500 MHz	1 GHz	1.25 GHz
Type	in-order	out-of-order	in-order
Issue Width	1	4	1
Instruction Window Size	-	64	-
Cache Line Size	64 bytes	64 bytes	64 bytes
L1 Cache Size	64 KB	64 KB	64 KB
L1 Cache Associativity	2-way	2-way	2-way
L2 Cache Size	1 MB	1.5 MB	1.5 MB
L2 Cache Associativity	8-way	6-way	6-way
L2 Hit / L2 Fwd Latency	16 ns / 24 ns	12 ns / NA	12 ns / 16 ns
Local Memory Latency	80 ns	80 ns	80 ns
Remote Memory Latency	120 ns	120 ns	120 ns
Remote Dirty Latency	180 ns	180 ns	180 ns



a

Single-Chip Piranha Performance

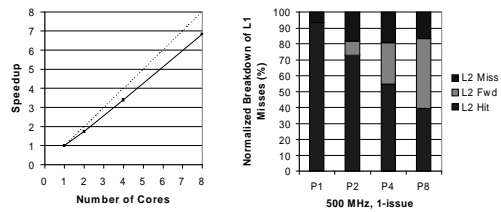


- Piranha's performance margin 3x for OLTP and 2.2x for DSS
- Piranha has more outstanding misses ➡ better utilizes memory system



a

Single-Chip Performance (Cont.)

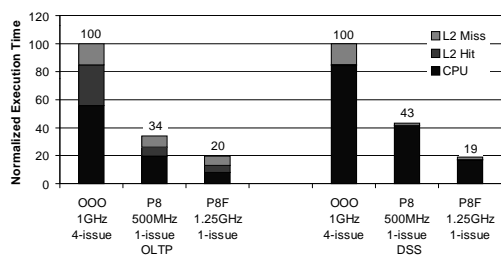


- Near-linear scalability
 - low memory latencies
 - effectiveness of highly associative L2 and non-inclusive caching



a

Potential of a Full-Custom Piranha

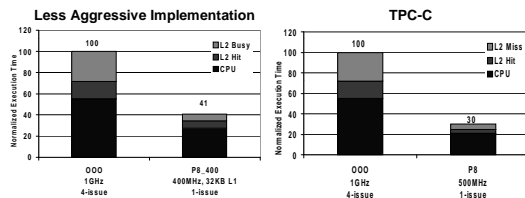


- 5x margin over OOO for OLTP and DSS
- Full-custom design benefits substantially from boost in core speed



a

Pirahna Performance (other)



- Pirahna design insensitive to variations in architecture parameters and workloads
- Pirahna's multi-chip scaling similar to OOO's (not shown here)



a

Implementation & Status

- RTL-level C++ simulator
 - allows mixed C++/Verilog execution
- ASIC methodology
 - using Verilog and industry-standard CAD tools
 - IBM ASIC process
- Engagement of Compaq's NonStop Hardware Division
- Current status
 - Alpha core (in Verilog) under debug and synthesis for timing
 - other modules at code completion stage and being translated to Verilog



Q

Related Work on CMP

- Hydra [Hammond ASPLOS'98] and other CMP TLDS work
 - thread-level data speculation not needed for commercial workloads
- MAJC [Tremblay Microprocessor Forum'99]
 - focuses on client appliances
- IBM Power4 [Diefendorff Microprocessor Report, Oct.'99]
 - most similar in focus to Piranha
 - opts for fewer, larger cores



Q

Summary

- Commercial workloads are rich in explicit thread-level parallelism and poor in ILP
- CMP is an excellent match to this application domain
- Piranha explores an extreme point in CMP design
 - use many simple cores
 - aggressively optimize memory and interconnect systems
- Piranha departs from increasing core complexity trends
- CMP is inevitable in future systems
 - key questions are:
 - number and complexity of CPU cores
 - best partitioning of on-chip memory hierarchy



Q

Additional Piranha Team Members

- Research
 - Joel McCormack
 - Mosur Ravishankar
- NonStop Hardware Division
 - Tom Heynemann
 - Dan Joyce
 - Harland Maxwell
 - Harold Miller
 - Brian Robinson
 - Sanjay Singh
 - Jeff Sprouse
- Former contributors
 - Basem Nayfeh
 - Joan Pendleton
 - Daniel Scales

a
