Evaluation of Design Alternatives for a MP multiprocessor
- -

Classic line: " The distinguishing characteristic of a shared-memory multiprocessor architectures is the level of the memor hierarchy at which the CPUs are interconnected." Nearer is better for fine-grain parallelism, further is better for coarse grain parallelism - where is the mish-mash, the middle-line, the sweet spot??

CPU supported in all three models that they study:

2-way issue processor, dynamic scheduling, speculative execution and non-blocking caches.
Fully pipelines functional units
"to eliminate structural hazards" there are twosies of everything except the memory data port
16KB, 2-way SA iand d caches
32 entry centralized window instruction issue
32 entry ROB
1024 entry BTB
non-blocking L1 that supports upto 4 outstanding misses.
MIPS2 ISA

Shared L1-$ MP
- - - - - - - - - - - -
So, the entire address space is mapped to all the 4 l1s? whats the deal?

+ Low-latency interprocessor communication using shared-memory address space
+ Implies high performance on fine-grained applications
+ Prefetches of shared data also enhances parallel application performance
+ Eliminates complex cache coherence logic usually associated with cache-coherent mps
+ Implicitly provides SC - easy to program, easy to design h/w

- Access time to the L1 is increased by the time required to get through the crossbar
- 3 cycles..
- processors working on different data can now conflict in cache.
- L2 access = 10 cycles

Shared L2 $ MP
- - - - - - - - - - - -

Share the L2, place processors + L1s on one chip and the l2 on alother and connect 'em via a MCM.

Write through L1.

+ processor+cache is independent of all other processor+cache
+ L1 latency is small

- L2 latency increases from 10-14 cycles.
- 64 bit bus reduces occupancy from 2-4 cycles for 32 byte transfer, but they assume critical word first
  so, performance is not *greatly* impacted.

Shared Memory MP
- - - - - - - - - - - - - - -

+ L2 (access = 10 cycles, occupancy = 2 cycleS)
+ l1 is 1-cycle away

-Communication is via main memory through the system bus (SLOW)
- Limits interprocessor communication
- Cache to cache transfers will crawl (50 cycles) because all three of the others on the bus must check their tags for a match, agree which processor should source the data amd then recover the date from the corresponding cache.
- Must support snooping protocols

A common solution is to have non-uniform memory access which implies that we are pushing the communication away from the processor which limits impact on single-processor performance.

Evaluation:
- - - - - - - - -

hand-parallelized
- eqntott spec92
- MP3D,
- Ocean
- Volpack

compiler-parallelized
- ear
- fft

MP/OS workloads
- No user-level data is shared

Operating across different address-spaces