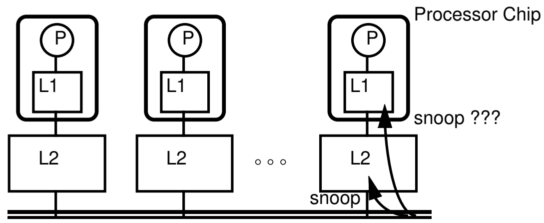
Design Trade-Offs In Small Scale SM MPs

- Recap
  - **Spectrum of SM designs**
  - **Cache Coherency**
  - **Protocols**
    - » **write-through**
    - » **3-state write-back invalidation**
    - » **4-state (Illinois) Invalidate Protocol**
    - » **4-state (Dragon) update protocol (today)**
- Multilevel Inclusion
- Evaluating Design Trade-offs

Multilevel Cache Hierarchies



- Independent snoop hardware for each level?
  - **processor pins for shared bus**
  - **contention for processor cache access**
- Snoop only at L2 and propagate relevant transactions
- Inclusion property

  (1) contents L1 is a subset of L

  (2) any block in modified state (D, S-D) in L1 is in modified state in L2

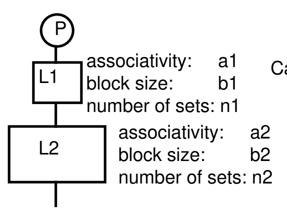
  1 => all transactions relavant to L1 are relavant to L2

  2 => on BusRd L2 can wave off memory access and inform L1

Forwarding Relevant Transactions to L1

- Which ones?
  - **Pass all transactions relevant to L2 on to L1**
  - **L2 maintains “inclusion bits” to provide better filter**
- What to do?
  - **if L1 is write-through, just invalidate**
  - **if L1 is write-back**
    - » **add extra state to L2 (dirty-but-stale)**
    - » **request flush from L1 on Bus Rd**
- Whenever L2 block is invalidated, must also invalidate in L1

Maintaining Inclusion

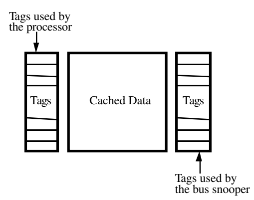


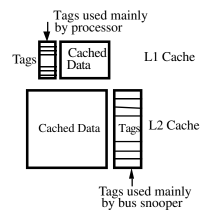
Capacity S1 = a1*b1*n1

- Can we establish cache parameters such that inclusion is maintained automatically?
  - **lower level caches get huge and complex (Baer & Wang, ISCA ‘88)**
- Common case that works
  - **a1 = 1, a2 ≥ 1, b1 = b2, n1 ≤ n2, and any replacement policy such that a new block is placed in both L1 and L2**
- Common cases that don’t
  - **a1 = a2 = 2, b1 = b2, n2 = K*n1, LRU (L1 hits change order w/i set)**
  - **separate L1 inst and data, a1 = 1, unified L2, a2 = 2**
  - **a1 = a2 = 1 , b1 = 1, b2 = 2**

Contention of Cache Tags

- Snooping from the bus can interfere with processor access to its cache!
  - **dual tags**



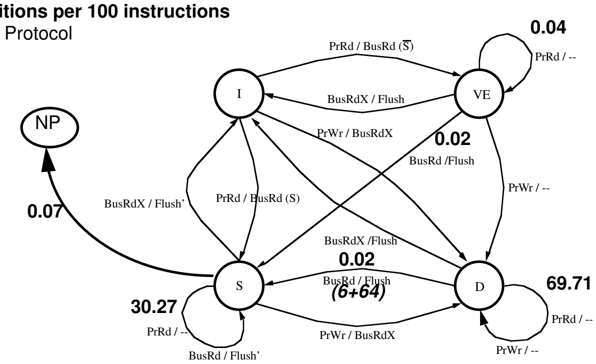


Design Tradeoffs in SMP

- Par. Arch = Computer Arch + Comm. Arch
- Comm. Arch in SMP = Extension of the memory system
- Memory system design
  - **huge parameter space**
    - » **memory interleaving, bus design**
    - » **block size, associativity, capacity, write policy**
  - **new parameters**
    - » **coherency protocol**
    - » **cache line size (false sharing)**
- => quantitative evaluation based on workload
  - **use the frequency of state transitions**

Transition Based Analysis

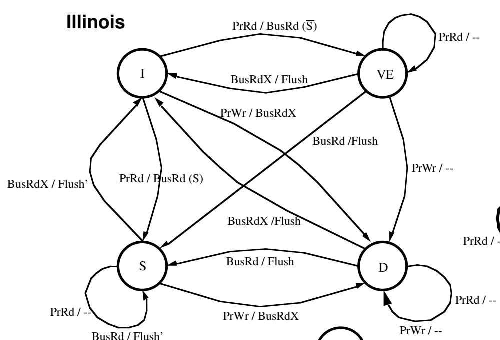
Transitions per 100 instructions
Illinois Protocol
LU



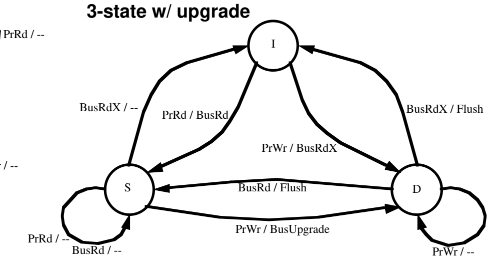
- Compute frequency (transitions per instructions)
- Multiply by bytes per transition => bytes per inst
- Multiply by instruction issue rate

Example Study

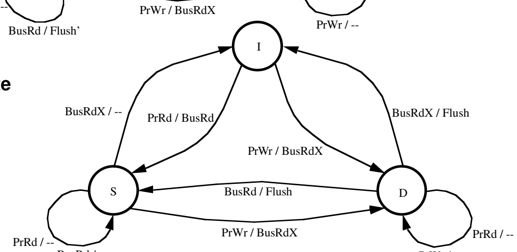
Illinois



3-state w/ upgrade



3-state

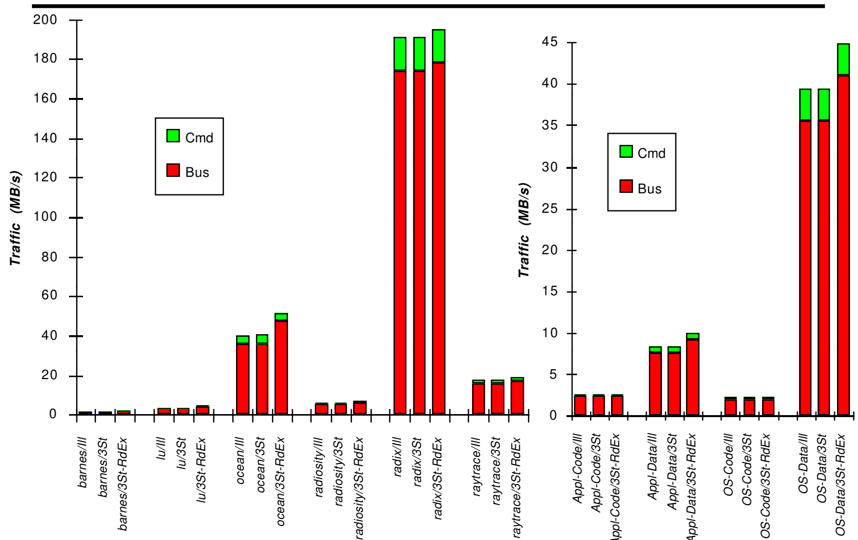


PrRd / --

Bandwidth per transition

Bus Transaction	Address / Cmd	Data
BusRd	6	64
BusRdX	6	64
BusWB	6	64
BusUpgd	6	--

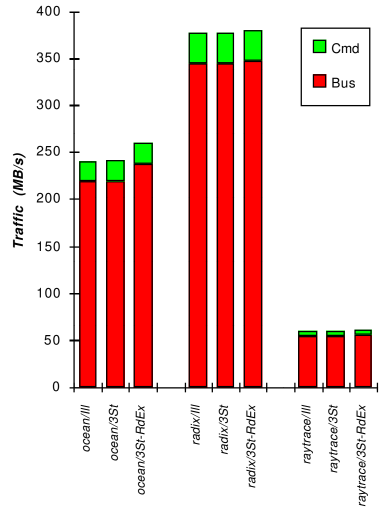
Bandwidth Trade-off



1 MB Cache, 200 MIPS / 200 MFLOPS Processor

VE -> D are infrequent BusUpgrade is cheap

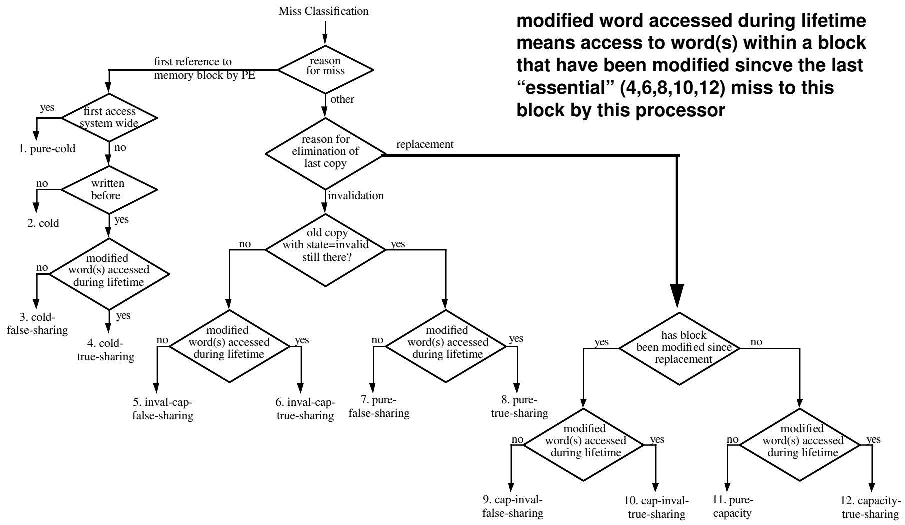
Smaller (64KB) Caches



Cache Block Size

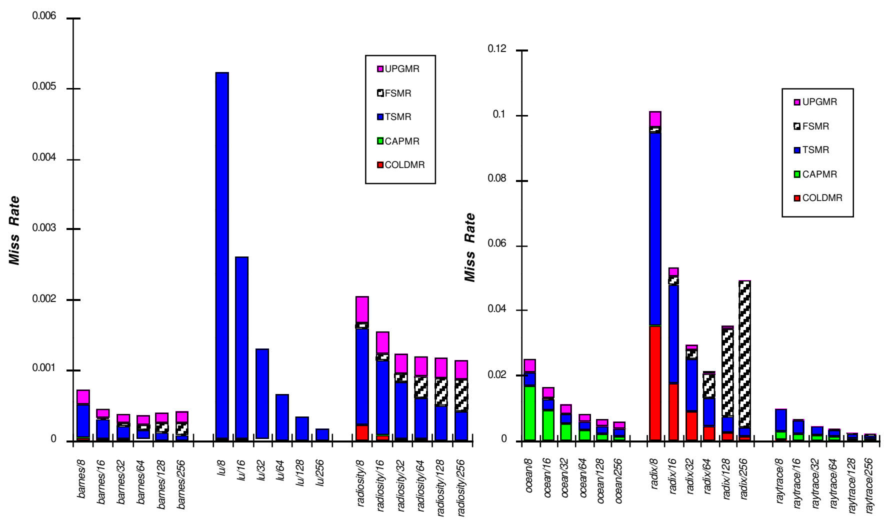
- Trade-offs in uniprocessors with increasing block size
  - **reduced cold misses (due to spatial locality)**
  - **increased transfer time**
  - **increased conflict misses (fewer sets)**
- Additional concerns in multiprocessors
  - **parallel programs have less spatial locality**
  - **parallel programs have sharing**
  - **false sharing**
  - **bus contention**
- Need to classify misses to understand impact
  - **cold misses**
  - **capacity / conflict misses**
  - **true sharing misses**
    - » **one proc writes words in a block, invalidating a block in another processor's cache, which is later read by that process**
  - **false sharing misses**

Miss Classification

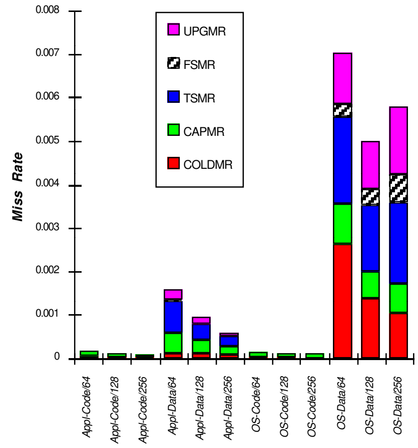


modified word accessed during lifetime means access to word(s) within a block that have been modified sincve the last "essential" (4,6,8,10,12) miss to this block by this processor

Breakdown of Miss Rates with Block Size

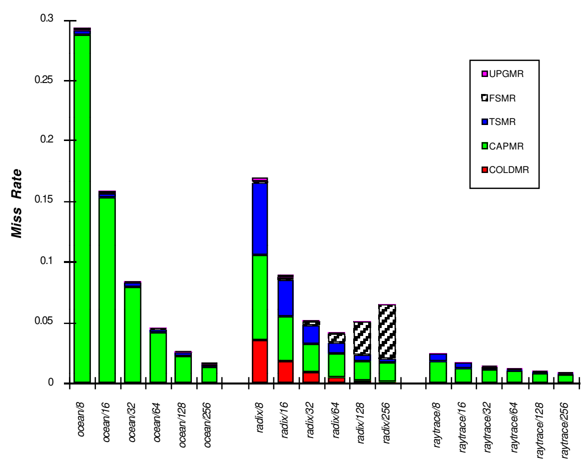


Breakdown (cont)

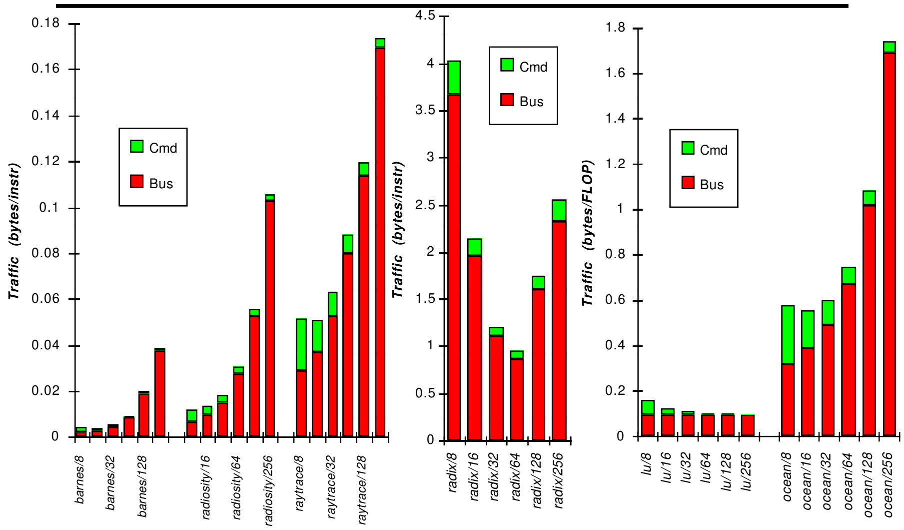


1 MB Cache

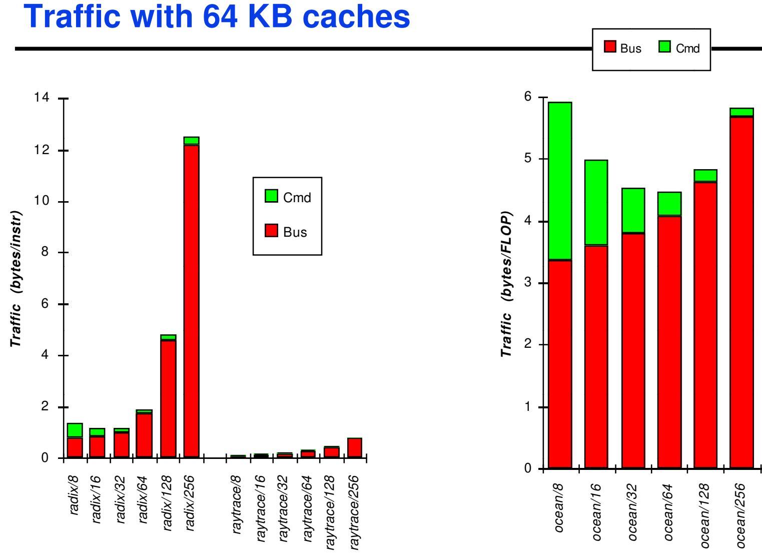
Breakdown with 64KB Caches



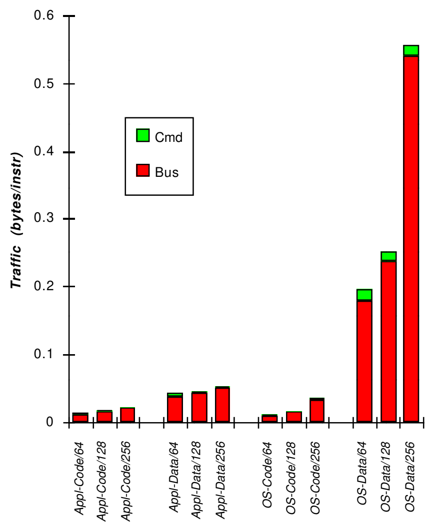
Traffic



Traffic with 64 KB caches



Traffic SimOS 1 MB



Coping with False Sharing

- Software
  - **Coarse-grain tasks**
  - **Per-processor Heaps**
  - **Copying Data**
  - **Higher Dimensional Arrays**
  - **Padding Arrays**
  - **Merging Multiple Arrays**
  - **Splitting an array**
  - **Aligning Arrays**
- Hardware
  - **delayed consistency models (delay invalidate till synchron point)**
  - **coherence granularity smaller than fetch granularity**
  - **update protocols**