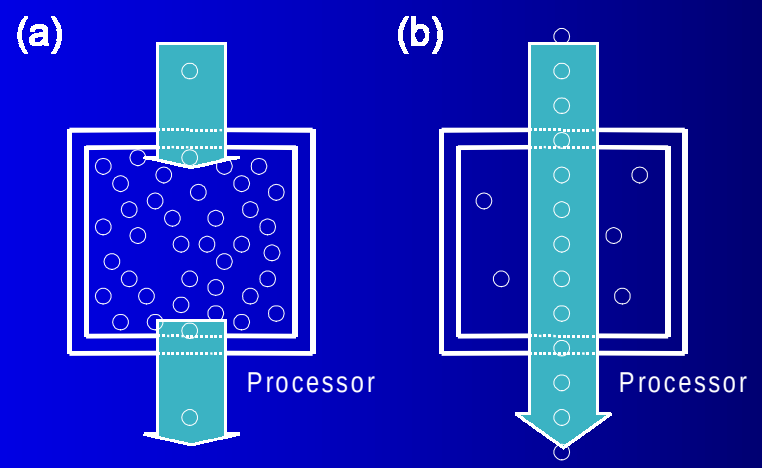
Multi-Level Memory Prefetching for Media and Stream Processing

Jason Fritts

Assistant Professor
Department of Computer Science,
Washington University

Introduction

- Multimedia is a dominant computer workload
- Traditional cache-based memory systems not well-suited to multimedia data
  - *cache memory designed for data with high temporal & spatial locality*
  - *Multimedia data has high spatial locality by low temporal locality*
- Growing processor-memory gap requires more efficient memory system for multimedia



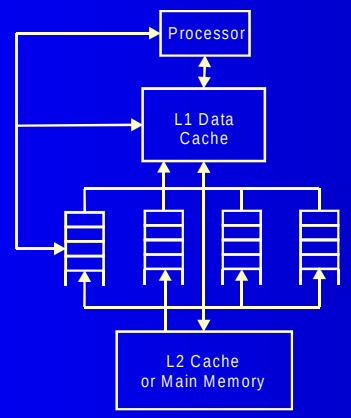
Prefetching Fundamentals

- **Four aspects of prefetching**
  - Detection *- how a memory access pattern is detected*
    - Pre-Pattern Detection
    - Post-Pattern Detection
  - Synchronization *- how & when prefetch requests are issued*
  - Storage *- where prefetched data is stored*
- **Software Prefetching**
  - Compiler statically determines predictable memory access patterns
  - Prefetch operation inserted into code for each individual prefetch request
  - Static Detection & Static Synchronization
- **Hardware Prefetching**
  - Hardware unit dynamically determines predictable memory access patterns
  - Hardware unit dynamically issues prefetch requests
  - Dynamic Detection & Dynamic Synchronization
- **Hybrid Hardware/Software Prefetching**
  - Compiler statically determines predictable memory access patterns
  - Prefetch operation inserted into code for each memory access pattern
  - Hardware table performs individual prefetch requests
  - Static Detection & Dynamic Synchronization

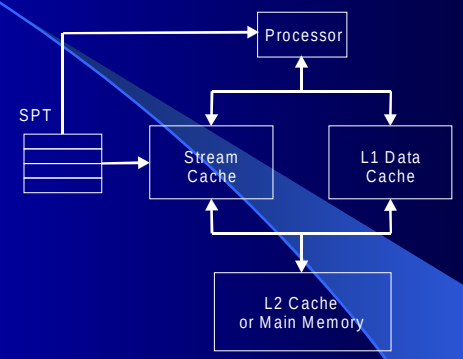
Prefetching Tradeoffs

- **Software Prefetching**
  - Advantage: *no hardware required*
  - Disadvantage: *increased code size*
    static pattern detection
- **Hardware Prefetching**
  - Advantage: *dynamic pattern detection*
  - Disadvantage: *cost of hardware (area, power, etc.)*
- **Hybrid Hardware/Software Prefetching**
  - Advantage: *little increase in code size*
  - Disadvantage: *some hardware needed*
    static pattern detection
- **Pimentel et al. (IPCCC 2000) found:**
  - Hybrid prefetching performed best on statically detectable patterns
  - Hardware prefetching performed best overall

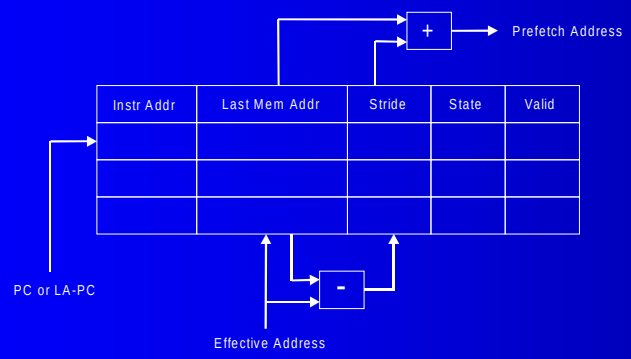
Streaming Prefetch Engines



Stream Buffers



Stream Cache



Stride Prediction Table (SPT)

Issues in Hardware Prefetching

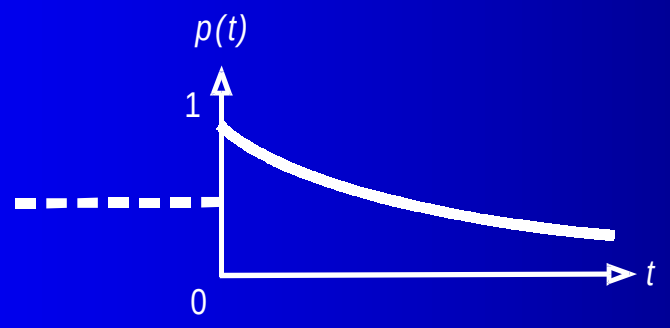
- Hardware prefetching usually occurs on-chip
  - In parallel or series w/ L1 cache
  - Prefetching performance usually best closest to L1 cache
- Prefetching increases bus bandwidth requirements by 30-100%
- Prefetch time varies in multi-level memory hierarchies
- Growing processor-memory gap requires aggressive prefetching
- Result: Need more aggressive & more efficient prefetching that:
  - Minimizes extra bus bandwidth
  - Supports longer memory latencies
  - Supports variable prefetch distance

Multi-Level Memory Prefetching

Adaptive Prefetching

More Aggressive Prefetching

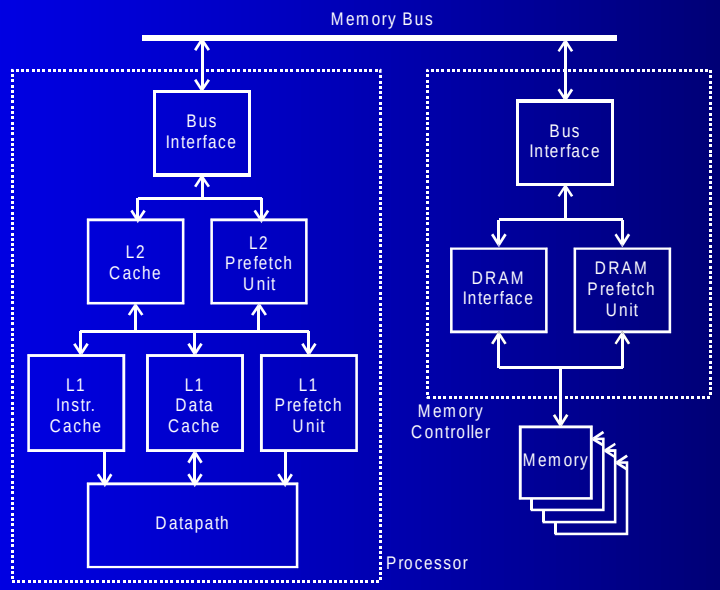
- Prefetch accuracy decreases with increasing prefetch distance
- More aggressive prefetching = prefetching further ahead
- To achieve same amount of useful data, prefetching further ahead in time requires prefetching more data



Prefetch Accuracy vs. Prefetch Distance

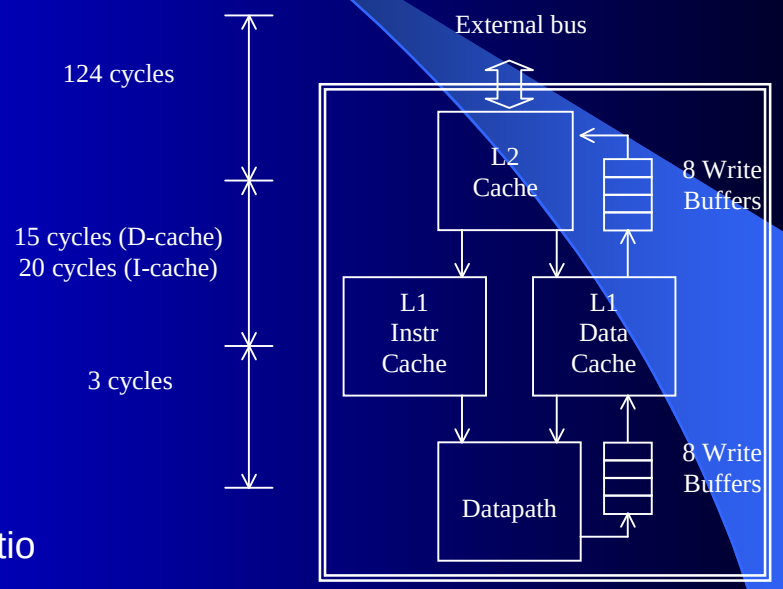
Multi-Level Memory Prefetch Hierarchy

- Conservative On-Chip Prefetching
  - Prefetch short distance ahead
- Aggressive Off-Chip Prefetching
  - Prefetch long distance ahead

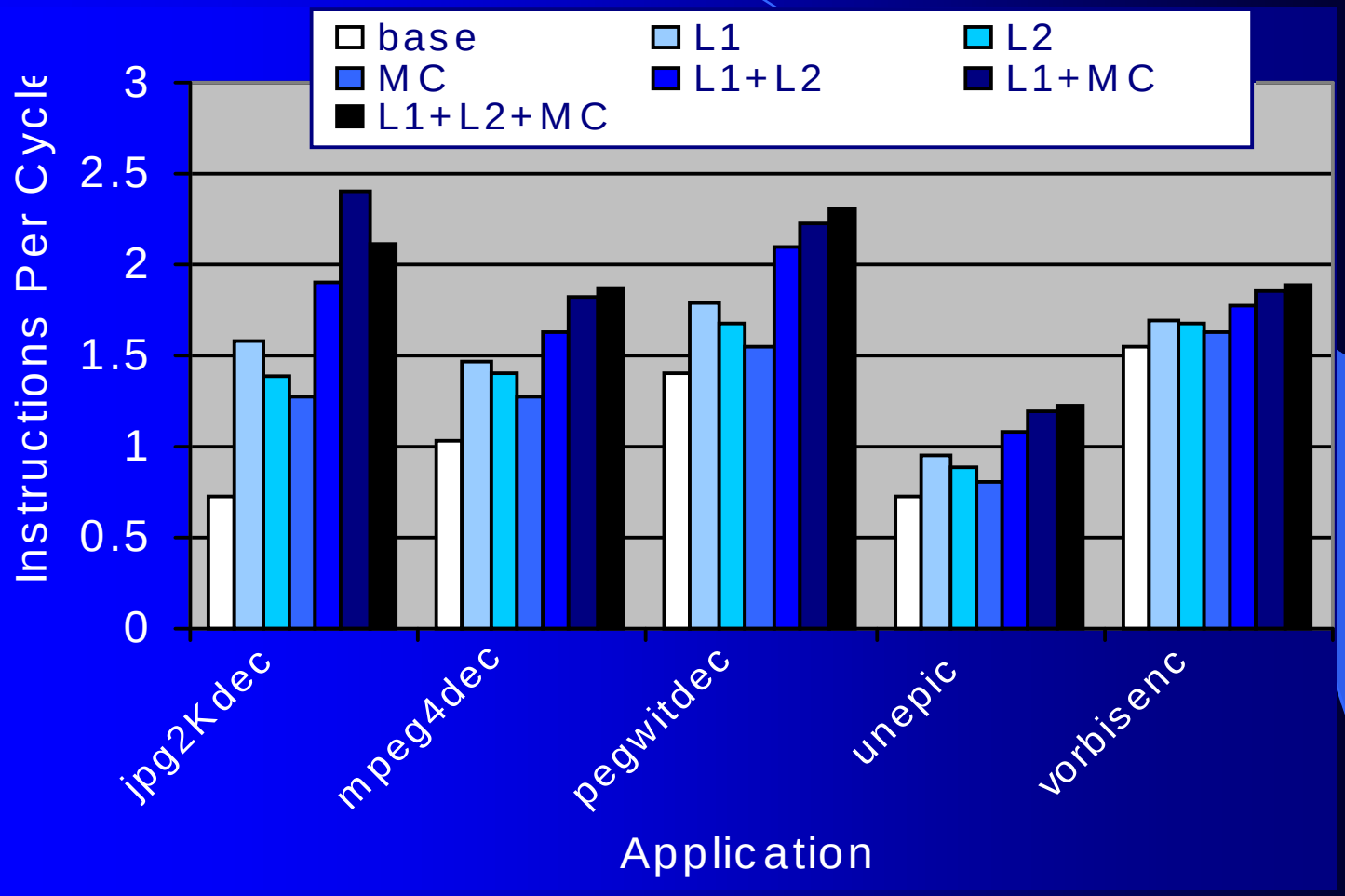


Base Architecture Model

- **Architecture model**
  - 4-issue media processor
  - 1 GHz processor frequency
- **L1 Cache**
  - 32 KB direct-mapped L1 data cache w/ 64 byte lines
  - 16 KB direct-mapped L1 instruction cache w/ 256 byte lines
- **On-Chip L2 Cache**
  - 256 KB 4-way set associate w/ 64 byte lines
- **External Memory**
  - 64-bit processor-memory bus
  - 8:1 processor-to-memory frequency ratio
- **Memory Prefetching**
  - 8-way stream buffers
    - 5 entries per stream at L1 level
    - 3 entries per stream at L2 and MC levels



Initial Results



Summary of Results

- Compared single-level prefetching vs. multi-level prefetching
  - 8-way stream buffer w/ 5 entries at L1 level
  - 8-way stream buffer w/ 3 entries at L2 and MC levels
- Average speedups for single-level prefetching
  - 45% for L1 prefetching
  - 34% for L2 prefetching
  - 24% for MC prefetching
- Average speedups for multi-level prefetching
  - 66% for L1+L2 prefetching
  - 84% for L1+MC prefetching
  - 89% for L1+L2+MC prefetching
- L1+L2+MC has 89% speedup, but excess bandwidth of ~100%
- L1+MC has 84% speedup, but excess bandwidth of ~50%

Summary and Conclusions

- Proposed multi-level memory prefetching for bandwidth-efficient aggressive prefetching of streaming data
  - Perform conservative prefetching on-chip and aggressive prefetching off-chip
  - Designed to:
    - provide aggressive prefetching (i.e. enable prefetching further ahead in time)
    - reduce extra bandwidth consumption from prefetching
- Multi-level memory prefetching provides valuable benefits
  - Nearly twice the speedup
    - off-chip prefetching aggressively tackles growing processor-memory gap
  - Half the extra bandwidth consumption
    - conservatively prefetching on-chip minimizes extra bandwidth consumption on system bus
  - Prefetching at L1 and MC levels identified as best overall prefetching method
- Definitive conclusions require more thorough evaluation
  - Test different prefetching methods
  - Explore variety of parameters (including varying latencies)
  - Examine implicit vs. explicit control/communication between prefetch levels
  - Study impact of multi-level memory prefetching on general-purpose workloads

Future Work

- Adaptive Prefetching
  - Three adaptive methods:
    - Adaptive separation of predictable vs. demand-based data
      - allows for separate, smaller and faster memory structures for different data types
    - Adapting post-pattern prefetch distance
    - Adapting pre-pattern prefetch distance
  - Examine effectiveness of three adaptive prefetching methods
  - Will prefetch distances achieve steady-state or constantly fluctuate?
  - How well does separation of predictable vs. demand-based data work?
    - Too much crossover will decrease performance…
    - How much smaller can each memory structure be made?

- Examine variety of performance metrics:
  - Speedup
  - Bandwidth
  - Power
  - Miss Ratio & Miss CPI
  - Tradeoffs in Size/Area of Hardware Prefetch Units