Implementing Cooperative Prefetching and Caching in a Globally-Managed Memory System

Geoffrey M. Voelker, Eric J. Anderson, Tracy Kimbrel, Michael J. Feeley, Jeffrey S. Chase, Anna R. Karlin, and Henry M. Levy

1998
Overview

- Background
- Prefetching Global Memory System
- Motivation
- Usage
- Evaluation
Background

- Disks are much slower than processors
  - Also improving at a much slower rate
  - Disk stall time is a bottleneck
- Prefetching data from disk can reduce disk stalls and help performance
  - Programmer-annotated or compiler-annotated hints
  - Global memory on idle network nodes
  - Striping data across multiple disks using multiple nodes
Prefetching Global Memory System

- Three-level Memory Hierarchy
  - Local Memory
  - Global Memory
  - Disk

- Prefetch pages that will be needed sooner than some already in memory
  - Can use hints to help decide what to prefetch
  - Replaces pages deemed less valuable
    - Local memory to global memory
    - Global memory to disk
Cache Replacement in PGMS

- Local cache replacement
  - Local memory to global memory
- Global cache replacement
  - Global memory to disk
- Local prefetching
  - Disk-to-local, Global-to-local
- Global prefetching
  - Disk-to-global
Cache Replacement in PGMS

- Conservative local prefetching
  - Removing a page early can produce a stall
- Aggressive global prefetching
  - Little downside to removing a page early
Motivation

- $F_G = \text{Network Page Transfer Cost}$
- $F_D = \text{Local Disk Page Transfer Cost}$
- $F_G < F_D$ in high speed network

Figure 1: Prefetching in conventional and global-memory systems
Motivation

- **Global Memory**
  - Use network paging when possible since it is cheaper than local disk paging

- **Prefetching**
  - Bring pages to nodes before they are needed to avoid stalling
Usage

Figure 2: Communications for prefetch into global memory
PGMS Speedup

- Speedup versus no global memory and only default readahead prefetching for files accessed sequential
- GMS is the global memory system of previous research
  - similar to PGMS, but without prefetching

Figure 3: Application speedup on GMS and PGMS
Breakdown of PGMS Enhancements on Speedup

Figure 4: Speedup of Render application on PGMS
Breakdown of PGMS Enhancements on Prefetch and Stall Amounts

Figure 5: Prefetch request and stall breakdown for Render
Performance Benefits of Increasing Idle Nodes in Network

Figure 6: Execution time detail for Render
Cache Benefits of Increasing Idle Nodes in Network

Figure 7: Breakdown of cold and capacity misses for Render
Performance Benefits of Different Memory Dispersals in Network

Figure 8: Fixed total global memory size vs. fixed per-node global memory size
Performance Detriment Due to Competing Threads

Figure 9: Elapsed times for two Render processes executing simultaneously.
Any questions?
Thanks