

# Implementing Global Memory in Workstation Cluster

# Objective

- XFS uses aggregate memory of all cluster nodes for file cache
- This paper takes that idea further, to all memory services
- Local Mm access  $O(10^3)$  cheaper than disk. Remote Mm access on fast interconnect cluster still 5-10x cheaper than disk, so
  - Keep as much of needed state in global memory
  - How do you do this in a lightweight manner, don't want to make heavy replacement policy decisions

# Basic Algorithm

- Local memory –  $M_m$  accessed by a process on processor  $P$
- Global memory –  $M_m$  stored on processor  $P$  but accessed by another process
- Keep as much recent stuff in global memory
- Maintain dynamic balance between global and local based on workloads

# Basic Algorithm

P faults

- Case 1
  - Page in GIMm on Q; swap with a page from P's GIMm; P's LcMm increases, Q's GI/LcMm unchanged
- Case 2
  - Page in GIMm on Q; but P has no global pgs; take P's LRU local page, and swap with page on Q; balance unchanged
- Case 3
  - Page on disk; bring it to P's local; take LRU page in cluster (gl + lc), say on Q; move that to disk and move one of P's global pages there
- Case 4
  - Page is shared and in LcMm on Q; copy page on P; choose oldest page in cluster, say on R, for disk write; move a global page from P to R

# Basic Algorithm

- Page Replacement
  - Epoch, duration  $T$ , max pages to be replaced  $M$
  - On epoch start – Initiator determines  $\text{MinAge}$ , and weights each node  $w_i$  based on #old pages (out of  $M$  to be replaced)
  - If page should be evicted:
    - If older than  $\text{MinAge}$  then discard
    - Otherwise send to node  $i$  with probability  $w_i$
  - if nodes active  $\rightarrow$  few old pages  $\rightarrow M$  small  $\rightarrow \text{MinAge} = 0$ , so always write to disk only, don't use GSM

# Implementation

- Page-Ownership Directory – replicated on all
- Global Cache Directory – partitioned across all
- Page Frame Directory – disk location, plus LRU stats, other state...
- Page ID – address of node backing the page + disk address + caching-related-state
- Age stats collected through period TLB flushes!
- Basic Alg split
  - Page fault causes getpage
  - Pageout daemon causes putpage op
- Single master node for reconfig/mgt

# Performance