Dynamic Page Migration in Software DSM Systems

Thomas Repantis* Christos D. Antonopoulos† Vana Kalogeraki* Theodore S. Papatheodorou†

*Department of Computer Science & Engineering
University of California, Riverside
Riverside, CA 92521, USA
{trep,vana}@cs.ucr.edu

†Computer Engineering & Informatics Department
University of Patras
26500 Patras, Greece
{cda,tsp}@hpclab.ceid.upatras.gr

Abstract

Computer clusters are a relatively new platform for high performance computing, that is attractive due to its flexibility and low cost. Of special interest seems the usage of the architecture of distributed shared memory in computer clusters, since it makes the work of the application programmer easier, by providing the illusion of shared memory.

Home-based software Distributed Shared Memory (DSM) systems offer the advantage of allowing page access faults to be satisfied with one request. Their performance however depends greatly on the distribution of pages across nodes. Dynamic page migration, when employed in DSMs offers several advantages: (i) reduces the latency of memory accesses by increasing the locality, (ii) improves resource utilization by considering the computational and communicational needs of the applications and adapting to the changing resource availability, and (iii) achieves the above with lower overhead than traditional approaches that rely on thread migration.

We propose a simple and efficient page migration mechanism [1], that dynamically allocates shared memory pages to home nodes. Each page has a designated home node and nodes that heavily modify the pages can become their new homes. In our protocol, to avoid redundant page transfers, we perform migration only when the number of modifications of a page becomes larger than a threshold. The migration information is piggybacked on the existing synchronization messages to minimize the communication overhead. The migration decision is taken locally, at the home of each page, according to the sizes of the diffs that have been applied to each page by the nodes that modified it. Each node sends its migration decisions to the barrier manager. The barrier manager then propagates the collected migration decisions to all nodes. Thus the page tables of all nodes are updated before the new barrier interval.

We have implemented our mechanism in the JIAJIA software DSM [2]. Performance evaluation using real application benchmarks shows that our mechanism significantly reduces remote page modifications (and hence transfers of pages and diffs), improves memory access latencies, and achieves better performance than its competitors. We observe that the cost of executing the algorithm and of migrating the pages is amortized by the benefits gained.

Keywords: *Distributed Shared Memory, page migration, page forwarding.*

References

[1] T. Repantis. *Implementation of Page Forwarding on Clusters.* Diploma thesis, University of Patras, Greece. `http://www.cs.ucr.edu/~trep/tsrDiplThesis.html`.

[2] W. Shi. *Improving the Performance of Software DSM Systems.* PhD thesis, Institute of Computing Technology, Chinese Academy of Sciences, 1999.