Software Distributed Shared Memory – Scalability and New Applications

Mats Brorsson

Department of Information Technology, Lund University
P.O. Box 118, S-221 00 LUND, Sweden
email: Mats.Brorsson@it.lth.se

Revision 2, August 3, 1999

Summary

Software Distributed Shared Memory (S-DSM) can be used to build cost-effective high-performance computing platforms out of clusters of individual computers. In this project we propose to extend this to study the use of many small, inexpensive processor nodes in a hierarchical fashion for new applications. The primary research issues here are the design of software DSM for embedded processing nodes and scalability.

The driving vision application is a large-scale system for distributed surveillance with several hundreds of intelligent nodes where each node collects data from its environment, processes the data and cooperates with each other for intelligent decisions.

The project is a cooperation with Axis Communications AB in Lund which will contribute in a reference group and with equipment to build an experimental S-DSM system out of an embedded platform.

1 Introduction and motivation

Clusters of PCs or workstations have emerged as a cost-effective alternative to high-end compute servers. They are regularly used for scientific applications and are rapidly becoming mainstream, partly thanks to the Beowulf effort [1]. Traditionally, clusters have been programmed with a message passing programming model but with Software Distributed Shared Memory (S-DSM) systems and OpenMP, a shared address space model is now viable, also for clustered systems [3, 4, 9]. Research on S-DSM have so far concentrated on small systems and scientific and engineering applications. For these applications, S-DSM can be very cost-effctive.

The research proposed here aims to build on the existing knowledge to investigate scalability issues of S-DSM systems consisting of many simple processing nodes and and to study new applications beyond the scientific applications that traditionally have been used. The vision that will drive our research we outline here is an example future application for large-scale distributed systems with software DSM. This application is highly relevant for Axis Communications AB who is a partner in this application. One of their products consists of a video camera and a small processing card that can be connected to Internet. An important application of this product is for surveillance purposes.

We envision a distributed survillance system consisting of many simple processing nodes with multiple sensors such as video cameras, microphones, transponders, etc. These sensor nodes cooperate in order to perform intelligent decisions about wether there has been a secutiry breach or not. They also communicate with several servers that collect information, compiles it and presents it for surveillance personell. Locality is achieved in the system with as much local processing of sensor data as possible. This will require substantial processing power in each node. Similar systems are already under study, but not from the viewpoint of system architecture [8].

A distributed surveillance system such as the one outlined above creates several problems. The current processing board that is used to handle the communication from a surveillance camera such as the one manufactured by Axis does not have enough computing power to also analyse images and to make decisions. Therefore it is important to be able to add computing power easily. Another problem is that many nodes will need to access information produced by other nodes. This calls for efficicient handling of shared information.

The problem of adding computing power to the nodes can be achieved by using a more powerful computer system in each node. This is, however, not feasible in a product line such as the one offered by Axis. For them, it would be better if they instead could use existing embedded computer cards and just plug them together to form a small multiprocessor interconnected via a high-speed interconnection network. Several of these small multiprocessors would then be connected to an ordinary LAN-technology such as Ethernet.

In order to make it manageable to program large-scale distributed surveillance systems it is important that the system support a shared address space. The ability of distributed systems to be able to manage shared information has also been emphasised by several internationally recognised researchers, e.g. in [2]. We believe that a shared

address space model will provide a tremendous advantage in terms of programming support and these systems could therefore benefit from a S-DSM system. The system should be programmed as one system, and not as hundreds of separate systems. The main problem here is the scalability of S-DSM which is largely unexplored as of today.

2 Problem statements

The example application outlined in the introduction does not yet exist. Also, it is completely out of scope for this project to develop such as application. Instead, we will use it as an inspiration to study software DSM for simple processing nodes and scalability issues. As example programs we will use existing image processing software that we believe is representative.

The main research problems here are:

- How are current S_DSM systems adapted to the special environment of small embedded processor systems.
- How do current S-DSM systems scale to several hundreds of processors?
- What application support is needed to program large-scale clusters effectively?
- How should S-DSM systems, hardware and software, be designed to support scalability more effectively?
- What interconnection network support is suitable for large-scale embedded clusters with software DSM?

3 Approach

The group already has a considerable amount of experience in software DSM systems for scientific and engineering applications [5, 6, 7, 10, 11]. We now intend to expand this to new applications, especially database and web servers. We intend to carry out the project in two parts over a period of five years.

- The design and implementation of clusters of simple processing nodes with software DSM (2.5 years)
- Scalability studies of large software DSM embedded systems (2.5 years)

3.1 The design and implementation of clusters with simple processing nodes

This part of the project deals with the design of software DSM for systems using simple processing nodes. For this purpose development systems for the Etrax processor provided to us by Axis (see section 10) will be used. There are numerous new research issues here. The main challenge is to design an S-DSM system that does not rely on the elaborate memory management system of commercial microprocessors but which is missing in many embedded processors. In order to do this, we will probably use some sort of fine-grain memory coherence similar to what is used in Shasta [12]. The upside of using embedded processors is that many of them, including Etrax designed at Axis, have extensive I/O-support on the chip which could be used for high-performance low-latency communication in an S-DSM system.

The development system from Axis will, in addition to a processor, contain programmable logic devices and we will therefore be able to experiment with hardware features that can facilitate the implementation of software DSM. A prototype consisting of a small number of processors will be built and evaluated in this part of the project. The actual S-DSM system for this platform will as far as possible be built on a system developed in a separate project internally at the department of Information Technology at Lund University. We will in the first year start with a four-processor prototype to be extended to 16 processors in the second year.

3.2 Scalability studies of large-scale software DSM systems

The first embedded software DSM system will only consist of a few tens of processors. In order to carry out a meaningful scalability study, we intend to expand this system to 256 processing nodes. This will in particular stress the interconnection technology needed in order to deliver reasonable performance for the software DSM system. The structure and design of a scalable S-DSM itself is also a major focus of this part since this has not been studied yet.

In order to obtain meaningful applications to evaluate the system we will work closely with Axis in this part of the project. It would be impractical to use a full-blown distributed surveillance system as we envision because of the large number of input devices needed. Instead we will use data files that represent input devices and besides being much less costly, this will ensure the repeatibility of performance experiments.

4 Expected results and impact

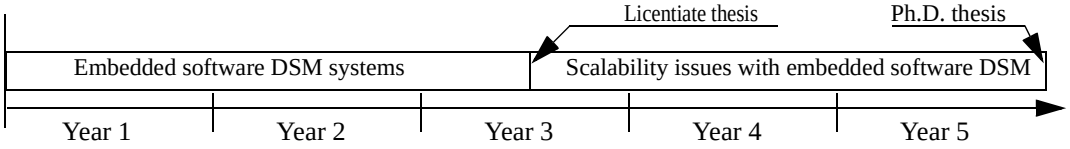
We expect that the project will produce a wide range of research results in the area of software DSM, parallel database servers and embedded parallel systems, in particular:

- Protocols and techniques for the design of software DSM for simple processing nodes.
- Performance evaluations of software DSM systems with simple processing nodes.

- Evaluation techniques for scalable software DSM systems.
- Interconnection technologies for scalable software DSM.
- Software technologies for scalable software DSM.

5 Project plan

The project will be carried out by one Ph.D. students during five years. The project will be carried out according to the following timeline in which we assume 20% departmental duties, such as teaching, for the student.



We propose that the project starts in January 2000 provided that there is a suitable Ph.D. candidate available.

The results from the project will be published in internationally renowned conferences and journals. From the first year of the project we expect to present one state-of-the-art report describing the design, implementation and performance evaluation of a small cluster of embedded processors with a software DSM system.

In addition to this report, we will produce deliverables and reports according to the CFP of PAMP. It is also expected that around three M.Sc. students each year will do their thesis project within the context of this project proposal.

6 Preliminary budget

All costs are in kSEK. Salary costs have been adjusted for 3% annual increase. The supervision cost is calculated as 10% of the salary of an Associate Professor for each student.

Cost item	2000	2001	2002	2003	2004	Total
Student 2	320	330	340	350	360	1700
Supervision (Mats Brorsson)	55	56	58	60	61	290
Computers and software	50	50	50	50	50	250
Travelling costs	25	50	50	50	50	225
Experimental platform	64	192	256	512		1024
Sum	514	678	754	1022	521	3489

The table above does not include university overheads. The cost for the experimental platform provided by Axis is estimated to 4 kSEK per processor and the net funding asked for from PAMP is 2465 kSEK excluding overheads.

7 Related research

Many references to related research are found in the introductory section of this proposal. Software DSM for scientific and engineering applications has been studied by many groups for some time now. A good survey of the current state-of-the-art has been made by Iftode and Singh in [4]. They also identify several new research directions, most notably the support for fine-grained memory coherence and scalability studies, both of which we intend to study within this project.

There is quite some activity in the area of embedded systems for high-performance processing. There is a yearly workshop devoted to this area in conjunction with International Parallel Processing Symposium (now merged with Symposium on Parallel and Distributed Processing to IPDPS – International Parallel and Distributed Processing Symposium): Workshop on Embedded HPC Systems and Applications. We will of course build our research on these results but as far as we know, there is very little done in the area of server applications on embedded clusters, especially with a shared address space.

8 Relation to PAMP profile

We believe that this project proposal fits very well in the profile of PAMP. The envisioned application of a distributed surveillance system is a real-time application for a large-scale system. A shared address space programming model has the potential to make the development of these system much easier than what is possible with current practice. Results from the project will contain methods to parallelise applications, methods for performance evaluation and methods to design I/O systems (interconnection networks); all of these are explicitly mentioned in the profile.

9 Industrial relevance

First of all, it is expected that the project will directly contribute to the technical know-how of Axis Communications in terms of distributed parallel applications and system design for a distributed shared address space in software. However, the results are of course also applicable in standard cluster of workstation systems and as such they will contribute to highly cost-effective parallel execution and programming of applications such as database and web servers. This will especially benefit small and medium sized enterprises since it will bring high-end computing power to a fraction of the cost of high-end parallel computer servers.

The vision about a large-scale surveillance system is shared by Axis Communications and this will ensure their activie participation in the project.

10 Context

The group conducting research on parallel computer systems at department of Information Technology at Lund University currently consists of:

- Mats Brorsson, Ph.D., Associate Professor and project leader
- Sven Karlsson, M.Sc., Ph.D. student
- Andreas Rodman, M.Sc., Ph.D. student

The research in the group has during the last few years focused on programming models for distributed shared memory architectures, especially the trade-offs between shared address space and message passing, and on protocol improvements for software DSM systems.

Mats Brorsson is representing Lund University as a member in EuroTools, a working group for the promotion of European tools and research on high-end computing. The group also has informal, but regular, contacts with Department of Computer Science at University of Copenhagen (Eric Jul and Povl Koch).

Mats Brorsson will together with Pallas (http://www.pallas.de) and other partners form a consortium for a project proposal to the new IST programme financed by the European Union. This project proposal will focus on programming tools and instrumentation for OpenMP and is complementary to the proposal as described here.

This project will be done in cooperation with Axis Communications AB. Axis has so far been a network client product company and now also want to expand in the technology sector. As part of this, they are interested in how new server applications can be used in embedded systems. The cooperation consists of two parts. The first part is to form a reference group consisting of group members from department of Information Technology and technical personell from Axis. The purpose of this reference group is to transfer information in both directions in a structural manner. The project will gain most of its input regarding distributed applications for highly-scalable software DSM for embedded clusters from this group. The second part is that Axis communication will contribute with equipment for the development of a large-scale embedded cluster with software DSM. The equipment will consist of hardware (processor development boards), software (operating system and communications software) and a limited amount of support. The processor development boards will contain an FPGA that can be used to study hardware features that may facilitate the design and implementation of the scalable S-DSM system.

Contact person at Axis is Niklas Morberg, Axis Communications AB, Scheelevägen 16, 223 70 Lund, tel. 046-270 18 00.

References

[1] D. J. Becker, T. Sterling, D. Savarese, J. E. Dorband, U. A. Ranawak, C. V. Packer, Beowulf: A Parallel Worstation for Scientific Computation, in *Proceedings of the International Conference on Parallel Processing*, 1995.

[2] J. Carter, New Directions and Challenges for Distributed Shared State Management, in NSF Workshop on New Challenges and Directions for Systems Research, St. Louis, Missouri, July 31-August 1, 1997. http://www.cs.utexas.edu/users/new_directions

[3] John Hård, *OdinMP - A proposal on how to implement the OpenMP standard on Networks of Workstations*, M.Sc. thesis, Department of Information Technology, Lund University, February 1999.

[4] L. Iftode and J. P. Singh, *Shared Virtual Memory: Progress and Challenges*, Proceedings of the IEEE, March 1999, Vol. 87, no. 3, pp. 498-506.

[5] HS. Karlsson and M. Brorsson, Producer-Push - a Protocol Enhancement to Page-based Software Distributed Shared Memory Systems, in *Proceedings of the 1999 International Conference on Parallel Processing* (ICPP'99), Aizu-Wakamatsu, Japan, September 1999. (to appear)

[6] S. Karlsson and M. Brorsson, *An Infrastructure for Portable and Efficient Software DSM*, Technical report, Department of Information Technology, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden, April 1999. To be presented at the 1st Workshop on Software Distributed Shared Memory (WSDSM'99), June 1999.

[7] S. Karlsson and M. Brorsson, A Comparative Characterization of Message Communication in Applications using MPI and Shared Memory on an IBM SP2, in *Proceedings of 1998 Workshop on Communication, Architecture, and Applications for Network-based Parallel Computing*, Las Vegas, January 31 - February 1, 1998, pp. 189-201.

[8] MIT AI Lab VSAM Home Page, A Forest of Sensors, Massachussetts Institute of Technology, Artificial Intelligenc Lab, http://www.ai.mit.edu/projects/vsam/

[9] OpenMP consortium, *OpenMP: A Proposed Standard API for Shared Memory Programming*, White paper, http://www.openmp.org.

[10] E. W. Parsons, M. Brorsson and K. C. Sevcik, *Predicting the Performance of Distributed Virtual Shared Memory Applications*, IBM Systems Journal, Volume 36, No. 4, 1997, pp. 527-549.

[11] A. Rodman and M. Brorsson, Programming Effort vs. Performance with a Hybrid Programming Model for Distributed Memory Parallel Architectures, in *Proceedings of EuroPar'99*, September 1999. (to appear)

[12] D. J. Scales, K. Gharachorloo, Towards Transparent and Efficient Software Distributed Sahred Memory, in *Proceedings of the 16th ACM Symposium on Operating System Principles*, St. Malo, France, October 1997.