Asynchronous Distributed Broadcasting in Cluster Environment

Sándor Juhász and Ferenc Kovács

Department of Automation and Applied Informatics
Budapest University of Technology and Economics
1111 Budapest, Goldmann György tér 3. IV. em., Hungary
{juhasz.sandor,kovacs.ferenc}@aut.bme.hu

Abstract. Improving communication performance is an important issue in cluster systems. This paper investigates the possibility of accelerating group communication at the level of message passing libraries. A new algorithm for implementing the broadcast communication primitive will be introduced. It enhances the performance of fully-switched cluster systems by using message decomposition and asynchronous communication. The new algorithm shows the dynamism and the portability of the software solutions, while it has a constant asymptotic time complexity achieved only with hardware support before. Test measurements show that the algorithm really has a constant time complexity, and in certain cases it can outperform the widely used binary tree approach by 100 percent. The presented algorithm can be used to increase the performance of broadcasting, and can also indirectly speed up various group communication primitives used in standard message passing libraries.

1 Introduction

Clusters play an increasingly important role in solving problems of high computational challenge, because they scale well, and provide high performance, and good fault tolerance at lower cost than traditional super computers. The speed of internode communication is in focus of many research efforts, because it often hinders efficient implementation of communication intensive algorithms. Thanks to hardware improvements, cluster systems of our days can take benefit of extensions of Ethernet standard (Fast and Gigabit Ethernet), as well as new standards providing high performance (more Gbps) and low latency ($< 10\ \mu s$) such as Myrinet, SCI, Quadrics, or InfiniBand. The prices of active network elements were also dropped, thus clusters systems usually use a fully-switched network topology, reducing competition for the physical bandwidth and thus providing a collision-free environment for communication.

The peak performance of communications is limited by the physical properties of the underlying network, but previous studies [1,2] concluded that the performance of real-life parallel applications is much more sensitive on different software overheads. Due to the inefficiencies and overheads at the levels of application, message passing

D. Kranzlmüller et al. (Eds.): EuroPVM/MPI 2004, LNCS 3241, pp. 164–172, 2004.

subsystems (such as PVM [3] or MPI [4]), and operating systems, the physical transfer time itself – especially for smaller messages – is only a fraction of the total application-level delay. This paper seeks to speed up the group communication of message passing libraries. While providing basic elements for sending and receiving messages, these libraries also offer group communication primitives to ease the creation of complex communication patterns at the application level. Among these primitives broadcasting plays an emphasized role, because it is widely used in itself, and also as a building block of other communication primitives (*allgather, alltoall, allreduce*). This paper presents a method for enhancing the performance of broadcasting by software means. Our new algorithm uses message decomposition and asynchronous communication to achieves an execution time complexity of O(1) without hardware support.

The rest of the paper is organized as follows: Section 2 introduces the commonly used broadcasting methods – both with and without hardware support – used in cluster environments. Section 3 describes our symmetrical algorithm providing a new approach of data distribution in fully-switched cluster systems. Section 4 compares the performance of the widely used tree and the new method, and verifies the correspondence of the measured curves and the performance predicted by the theory. The paper concludes with summarizing the results and showing their application possibilities.

2 Overview of Broadcasting Methods

Following the recommendations of the MPI standard [4] most communication subsystems implement all the group communication primitives based on the point-to-point transfer functions. This technique allows a fast and portable implementation of the group primitives (only the bottom, point-to-point layer must be rewritten for other platforms). The efficiency of the different implementations is strongly influenced by the topology of the underlying connection network. Because of its wide practical use, this paper focuses on the virtual crossbar (fully switched) topology. All execution time estimations use the widely accepted [5,6] linear model, where the communication time t_c equals to

$$t_c(n) = t_0 + nt_d \,, \tag{1}$$

where n is message size, t_0 is the initial latency, and t_d is the time needed to transfer one data unit (reciprocal of the effective bandwidth).

The simplest way of broadcasting is the linear method, where the data transfer is controlled from a single source. This is the most straightforward method, following exactly the philosophy of one to all data distribution: the source node sends the data to be distributed to each of the partner nodes one by one. This technique is simple and easy to implement, but not very efficient: it has a linear increase of execution time as the number p of destination nodes grows:

$$t_c(n, p) = p(t_0 + nt_d) \Rightarrow O(p) \,, \tag{2}$$

Because the source node plays a central role as a single sender, this algorithm has the advantage of reducing collisions on a shared medium, that is why it was preferred