A Multithreaded Message Passing Environment for ATM LAN/WAN [1]

SCCS # 690

Rajesh Yadav, Rajashekar Reddy, Salim Hariri, and Geoffrey Fox

Northeast Parallel Architectures Center (NPAC)
Syracuse University
Syracuse, NY 13244
{ryadav, kraja, hariri}@npac.syr.edu

Abstract

Large scale High Performance Computing and Communication (HPCC) applications (e.g. Video-on-Demand, and HPDC) would require storage and processing capabilities which are beyond existing single computer systems. The current advances in networking technology (e.g. ATM) have made high performance network computing an attractive computing environment for such applications. However, using only high speed network is not sufficient to achieve high performance distributed computing environment unless some hardware and software problems have been resolved. These problems include the limited communication bandwidth available to the application, high overhead associated with context switching, redundant data copying during protocol processing and lack of support to overlap computation and communication at application level. In this paper, we propose a Multithreaded Message Passing system for Parallel/Distributed processing, that we refer to as NYNET Communication System (NCS). NCS, being developed for NYNET (ATM wide area network testbed), is built on top of an ATM Application Programmer Interface (API). The Multithreaded environment allows applications to overlap computations and communications and provide a modular approach to support efficiently HPDC applications with different quality of service (QOS) requirements.

[1] This research is funded by Rome Laboratory (Contract # F30602-94-C-0165), Rome, NY

1 Introduction

Large scale High Performance Computing and Communication (HPCC) applications (e.g. Video-on-Demand and HPDC) would require storage and processing capabilities which are beyond existing single high performance computer systems. Current advances in processor technology, and advent of high speed networking technology such as ATM have made high performance network computing an attractive computing environment for such applications. Aggregate power of workstation clusters (Alpha cluster for example) is comparable to that of a supercomputer [10]. Compared to massively parallel computers, network computing is typically less expensive and more flexible.

Another reason for recent rapid growth in network computing area is the availability of Parallel/Distributed tools that simplify process management, inter-processor communication and program debugging in distributed computing environment. However, these advances cannot be fully exploited unless some hardware and software problems are resolved. These problems can be attributed to the high cost of operating system calls, context switching and the use of inefficient communication protocols (e.g TCP/IP). The complexity and the inefficiency of these communication protocols can be justified in 1970s where the networks were slow (operating in Kbps range) and not very reliable while the processing power was relatively three orders of magnitude higher than the network speed.

With advent of high speed networks operating, at Mbps and Gbps, new methods are needed to process protocols efficiently. Reducing the communication latency has been active research area in parallel processing field. Most of the techniques proposed are based on using active messages [2], reducing operating system overhead [1], and overlap computation with communication using Multithreading [7]. In distributed computing, most of the research focused on developing new communication protocols (XTP, FLIP) [11], streamlining existing ones (by merging several layers into one), or building intelligent hardware interface to off-load host from protocol processing.

The main objective of the research presented in this paper is to implement NYNET Communication System (NCS) based on these techniques (e.g. multithreading, reduce data copying and operating system overhead and parallel data transfer) that have been proven to be successful in parallel as well as distributed computing. NCS uses multithreading capability to provide efficient techniques to overlap computation and communication. Furthermore, multithreaded message passing represents an interesting distributed programming paradigm