PCODE: Efficient Parallel Computing Over Distributed Environments

Jehoshua Bruck Danny Dolev Ching-Tien Ho Rimon Orni Ray Strong
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
bruck@systems.caltech.edu
{dolev,rimono}@cs.huji.ac.il
{ho,strong}@almaden.ibm.com

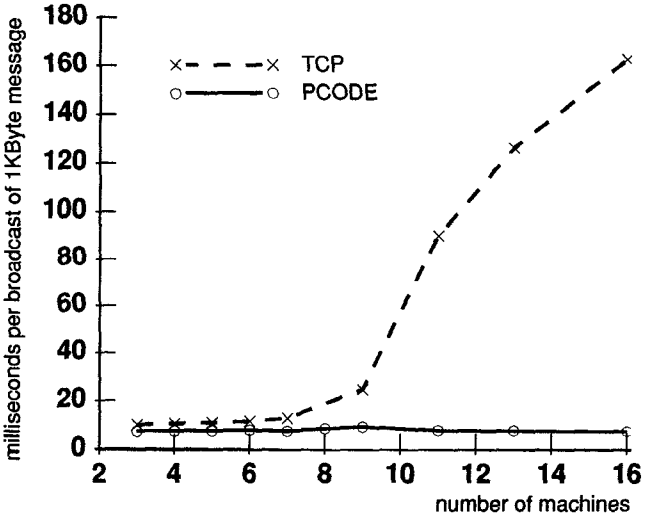
Parallel computing on clusters of workstations and personal computers has very high potential, since it leverages existing hardware and software. In fact, there are a number of existing commercial parallel programming environments that can run on top of clusters of workstations, for example, PVM, IBM PPE and EXPRESS by Parasoft. Parallel programming environments offer the user a convenient way for expressing parallel computation and communication. The communication part consists of the usual point-to-point communication as well as collective communication.

However, existing programming environments for clusters are built on top of a point-to-point communication layer (send and receive) over local area networks (LANs) and, as a result, suffer from poor communication performance. For example, a broadcast that is implemented using a TCP/IP protocol (which is a "reliable" point-to-point protocol) over a LAN is obviously inefficient as it is not utilizing the fact that the LAN is a broadcast medium.

Here we consider a system model that consists of a set of processors that communicate via asynchronous and unreliable broadcast messages. A processor has three logical layers of software. The lowest layer is a LAN-communication layer, typically a User Datagram Protocol (UDP), that interfaces the LAN. The second layer is the transport layer (this is where our new protocol fits). The upper layer is the user-communication layer, that in our case is a set of collective communication routines of a parallel programming environment. Our goal is to create a transport layer which utilizes the fact that a LAN is a broadcast domain and to make the collective communication part of a parallel programming environment more efficient. The challenge in achieving this goal is that the LAN-communication facility within a broadcast domain, typically UDP, is unreliable.

Reliable broadcast in distributed systems is a topic that was studied extensively for more than a decade. In fact, there are a number of existing projects and systems that provide a reliable transport layer as well as other services for distributed computing. Examples are the V system, ISIS, Psync, Amoeba, Trans, Transis and Totem. However, we have observed that the properties required from the user-communication layer while devising the known reliable broadcast protocols for *distributed systems* are different from the properties of the user-communication layer associated with *parallel systems*.

The main contributions of our work are: (i) We have studied the requirements associated with collective communication for parallel computing. We have observed that the main difference between a distributed computing paradigm and a message passing parallel computing paradigm is that, in a distributed environment the activity of every processor is independent while in a parallel environment the collection of the user-communication layers in the processors can be modeled as *a single global program*. Also, the typical fault model in parallel computing is that if a single processor fails then the execution stops and the recovery is handled by global techniques (like checkpointing). We have formalized the requirements by defining the notion of a *correct global program*. This notion provides a precise specification of the interface between the transport layer and the user-communication layer. (ii) We have developed a new communication protocol that is driven by a *global program*, and proved its correctness. (iii) We have implemented our protocol and run it over a collection of IBM RS/6000 workstations, using the AIX operating system and communicating via UDP broadcast. The experimental results we obtained indicate that the performance advantage of the PCODE protocol can be as high as an order of magnitude in the case of computing with 16 workstations. See the following figure for a comparison between the PCODE protocol and a TCP based implementation.



PODC 94 - 8/94 Los Angeles CA USA