Evaluation of the Communication Performance on a Parallel Processing System

Luís Paulo Santos Vítor Castro Alberto Proença
Dep. Informática, Universidade do Minho,
4719 Braga Codex, PORTUGAL
e-mail: psantos@di.uminho.pt

Abstract. This article presents an evaluation study of point-to-point and collective communication performance on a parallel processing system, a 16 node Parsytec PowerXplorer, using three different communication environments: PARIX, PVM and MPI.

1 Introduction

Most current massively parallel processing systems (MPP) are distributed memory message passing computers. Applications developed to run on these machines commonly use three types of communication: point-to-point, collective communication and collective computation [1, 2]. These communication operations incur costs which include software overheads (communication and synchronization protocols), hardware latencies and message delays (network and memory contention).

This article presents a systematic study of communication costs on a 16 node Parsytec PowerXplorer with three different communication environments: PARIX [3], PVM [4] and MPI [2, 5]. The tests were performed assuming the user view of the whole computer system through a high-level language, i.e., a particular software interface to a given computer architecture [6, 7].

This work used the methodology presented by Hwang and Xu [1] for collective operations, which is a generalization of Hockney’s model [8]. Most of the recommendations of the PARKBENCH Committee on Parallel Benchmarks [6, 7] are also considered.

Quantifying the costs of communication operations has several advantages. As the understanding of these operations increases, informed decision making during the design and/or execution of parallel applications becomes feasible, and it is easier to identify weaknesses on communication libraries and/or on the workload distribution strategies.

2 Scientific foundations

Roger Hockney proposed the following model for communication times in point-to-point operations [7, 8]:

\[ t(n) = t_0 + \frac{n}{r_\infty} \] (1)
where $n$ is the length in bytes of the user data field in the message, $t_0$ is the startup time (or latency) and $r_\infty$ is the asymptotic bandwidth which is the maximum achievable bandwidth when the message length approaches infinity.

The message length required to achieve a performance of $r_\infty/2$ is given by

$$n_{1/2} = t_0 \times r_\infty$$

and is known as the half performance length. $n_{1/2}$ can also be seen as the length of the message that could be sent during the startup time. In practice, $n_{1/2}$ is a good measure of the message length needed to approach $r_\infty$.

The pair of $(r_\infty, n_{1/2})$ parameters completely characterizes the performance of a given operation. For long messages ($n \gg n_{1/2}$) the startup time may be neglected and only $r_\infty$ is needed, while for short messages ($n \ll n_{1/2}$) only the startup time $t_0$ is necessary, although usually the specific performance ($r_\infty = t_0^{-1}$) is used instead.

Xu and Hwang [1] generalized this model for collective operations involving $p$ nodes. The time to complete a communication is given by

$$t(n, p) = t_0(p) + \frac{n}{r_\infty(p)}$$

where $t$ is still a linear function of message length $n$, but the startup time $t_0(p)$ and the asymptotic bandwidth $r_\infty(p)$ are both functions of the number of nodes $p$ involved in the communication. The half-performance length is given by

$$n_{1/2}(p) = t_0(p) \times r_\infty(p)$$

An additional metric, the aggregated asymptotic bandwidth $R_\infty$ was derived. $R_\infty$ is the ratio of the total number of data bytes transmitted by all nodes to the total time needed to execute the operation, as $n$ approaches infinity. For a point-to-point communication $R_\infty = r_\infty$; for broadcast, gather, scatter, reduction and scan $R_\infty = p \times r_\infty$; for the total exchange $R_\infty = p^2 \times r_\infty$.

3 The environment

All experiments were performed on a 16 node PowerXplorer from Parsytec. Each PowerXplorer node contains a Motorola PowerPC 601 80 MHz RISC microprocessor for computation and an Inmos T805 30 MHz Transputer for communication. Both processors are closely coupled via shared memory. The nodes are interconnected via a 2-dimensional grid using the Transputer links (a 4*4 grid in this particular system).

PARIX 1.3.1 (PARallel extensions to unIX) is the operating system actually being supplied with the PowerXplorer [3]. PARIX provides, among other services, message routing and multi-user partitioning of the 2D grid of processors. A partition is a set of processors that are exclusively allocated to one user. PARIX ensures that simultaneous users of different partitions will not conflict with each others.