Modeling the Communication Behavior of Distributed Memory Machines by Genetic Programming*

L. Heinrich-Litan, U. Fissgus, St. Sutter, P. Molitor, and Th. Rauber

Computer Science Institute, Department for Mathematics and Computer Science, Martin-Luther-Universität Halle-Wittenberg, D-06099 Halle (Saale), Germany
<name>@informatik.uni-halle.de

Abstract. Due to load imbalance and communication overhead the behavior of the runtime of distributed memory machines is very complex. The contribution of this paper is to show that runtime functions predicting the execution time of the communication operations can be generated by means of the genetic programming paradigm. The runtime functions generated dominate those presented in literature, till today.

1 Introduction

Distributed memory machines (DMMs) provide large computing power which can be exploited for solving large problems or computing solutions with high accuracy. Nevertheless, DMMs are still not broadly accepted. One of the main reasons is the costly development process for a specific parallel algorithm on a specific DMM. This is due to the fact that parallel algorithms on DMMs may show a complex runtime behavior caused by communication overhead and load imbalance. Thus, there is considerable research effort to model the performance of DMMs. This includes modeling the runtimes of communication operations with parametrized formulas [6, 5, 4, 8, 1]. The modeling of the execution time of communication operations can be used in compiler tools, in parallelizing compilers or simply as concise information of the performance behavior for the application programmer.

Genetic algorithms (GAs) are stochastic optimization techniques which simulate the natural evolutionary process of beings. They often outperform conventional optimization methods when applied to difficult problems. We refer to [2] for an overview on complex problems in the area of industrial engineering, where GAs have been successfully applied. The genetic programming approach (GP) has been introduced by Koza [7] and is a special form of a genetic algorithm.

The contribution of this paper is to show that runtime functions of high quality, which model the execution time of communication operations, can be modeled by the genetic programming approach. To illustrate the effectiveness of the approach we have chosen [8] for comparison. In [8], collective communication

* This work has been supported in part by DFG grant Ra 524/5 and Mo 645/5.
operations from the communication libraries PVM and MPI are investigated and compared. Their execution time is modeled by runtime functions found by curve fitting. In this paper we demonstrate the effectiveness of GPs to model the execution time of communication operations on DMMs, by example. Especially we demonstrate that GP generates runtime functions of higher quality than those published in literature till now.

The paper is structured as follows. Section 2 gives a brief overview on the investigations made by [8]. Section 3 presents how performance modeling can be attacked by GPs. Experimental results are shown and discussed in Section 4.

2 Runtime Functions Generated by Curve Functions

First, we give an overview on the communication operations which we use in our experiments. Then we present the model of the communication behavior on the IBM SP2 obtained by curve fitting. The specific execution platform that [8] used for their experiments is an IBM SP2 with 37 nodes and 128 MByte main memory per processor from GMD St.Augustin, Germany, with IBM MPI-F, Version 1.41. They investigate and compare different communication operations. Due to space limitation, we only report on two communication operations.

**Single-transfer operation:** In MPI, the standard point-to-point communication operations are the blocking MPI\_Send() and MPI\_Recv(). For blocking send, the control does not return to the executing process before the send buffer can be reused safely. Some implementations use an intermediate buffer.

**Scatter operation:** A single-scatter operation is executed by the global operation MPI\_Scatter() which must be called by all participating processes. As effect, the specified root process sends a part of its own local data to each other process.

In [8], the runtimes of MPI communication operations on the IBM SP2 are modeled by runtime formulas found by curve fitting that depend on various machine parameters including the number of processors, the bandwidth of the interconnecting networks, and the startup times of the corresponding operations. The investigations resulted in the parameterized runtime formula

\[ f_{\text{scatter}}(p, b) = 10.28 \cdot 10^{-6} + 91.59 \cdot 10^{-6} \cdot p + 0.030 \cdot 10^{-6} \cdot p \cdot b \ [\mu\text{sec}] \]

and

\[ f_{\text{single}}(b) = 211.8 \cdot 10^{-6} + 0.030 \cdot 10^{-6} \cdot b \ [\mu\text{sec}] \]

for the scatter operation, and the single-transfer operation. The parameters \( b \) and \( p \) are the message size in bytes and the number of processors, respectively.

3 The Genetic Programming Approach

The usual form of genetic algorithm was described by Goldberg [3]. Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and natural genetics. They start with an initial population, i.e., an initial set of random solutions which are represented by chromosomes. The chromosomes evolve through successive iterations, called generations. During each