Expert Systems with Applications 37 (2010) 6293–6301

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Predicting the performance measures of an optical distributed shared memory multiprocessor by using support vector regression

M. Fatih Akay *, Ipek Abasıkeleş

Dept., of Computer Engineering, Cukurova University, Adana, Turkey

ARTICLE INFO

Keywords:
Support vector regression
Multiprocessors
Distributed shared memory
Interconnection networks

ABSTRACT

Recent advances in the development of optical technologies suggest the possible emergence of optical interconnects within distributed shared memory (DSM) multiprocessors. The performance of these DSM architectures must be evaluated under varying values of DSM parameters. In this paper, we develop a Support Vector Regression (SVR) model for predicting the performance measures (i.e. average network latency, average channel waiting time and average processor utilization) of a DSM multiprocessor architecture interconnected by the Simultaneous Optical Multiprocessor Exchange Bus (SOME-Bus), which is a high-bandwidth, fiber-optic interconnection network. The basic idea is to collect a small number of data points by using a statistical simulation and predict the performance measures of the system for a large set of input parameters based on these. OPNET Modeler is used to simulate the DSM-based SOME-Bus multiprocessor architecture and to create the training and testing datasets. The prediction error and correlation coefficient of the SVR model is compared to that of Multiple Linear Regression (MLR) and feedforward Artificial Neural Network (ANN) models. Results show that the SVR-RBF model has the lowest prediction error and is more robust. It is concluded that SVR model shortens the time quite a bit for obtaining the performance measures of a DSM multiprocessor and can be used as an effective tool for this purpose.

1. Introduction

Large-scale DSM multiprocessors are the most feasible way of achieving the enormous computational power required in many science and engineering applications. DSM systems provide a shared address space by physically distributing the memory among different processors (Culler, Singh, & Gupta, 1997). The key strength of DSM systems is that communication occurs implicitly as a result of conventional memory access instructions (i.e. loads and stores), which makes them easier to program. Their success, however, is highly dependent on the efficiency of the underlying interconnection network, which allows the processing nodes to communicate with each other. The interconnection network has a direct effect on the remote memory latency, which is caused by accessing a memory location in a processor other than the one originating the request. A remote memory access takes 3–5 orders of magnitude longer than the local memory access (Hagersten & Koster, 1999; Laudon & Lenoski, 1997) with most of the time consumed in communication over the interconnection network of the system. Although DSM systems use latency reducing/hiding techniques (Gharachorloo et al., 1990) to reduce remote memory latency, these techniques require extra bandwidth and greatly increase memory traffic by fetching more data than needed (Lenoski & Weber, 1995). Additionally, every transaction in a DSM system consists of a request, response (data), several acknowledge and coherence messages. As the system size increases, more processors are injecting more messages (both transaction related messages and latency tolerating requests) into the network that causes network contention (Pai & Panda, 1997) for various shared resources.

Smaller DSM systems ranging from 4 to 8 nodes usually interconnect via a single switch. An enlarged system requires a hierarchy of switches, which causes a significant routing or switching delay in the additional switching stages, which in turn increases remote latency (Cray, 2004). Huang, Sze, Landin, Lytel, and Davidson (2003) reported that scaling from a medium to a large-scale multiprocessor increases memory access latency by 60%. Future high performance DSM systems will utilize commercial off-the-shelf processors that require aggregate computational and communication bandwidths on the order of 4–40 terabits per second (Lemoff et al., 2004). Thus, lack of sufficient bandwidth will be the fundamental obstacle to future scalable DSM systems.

One technology that has the potential for providing higher-bandwidths and lower latencies than current electronic-based interconnects is optical interconnects (Collet et al., 2000; Pu

* Corresponding author.
E-mail address: mfakay@cu.edu.tr (M.F. Akay).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.02.092

et al., 1999). Optical fibers exhibit extremely high-bandwidth and can be multiplexed to provide a large number of independent communication channels. Recently, there have been significant developments in optical and optoelectronic devices which make optical interconnects a viable and cost-effective option for building high-bandwidth, low-latency and scalable optical interconnection networks. One such optical interconnect is the SOME-Bus, which incorporates optoelectronic devices into a very high performance processing architecture. It is a low-latency, high-bandwidth, fiber-optic interconnection network, which directly connects each node to all other nodes. One of its key features is that each of N nodes has a dedicated broadcast channel operating at 20–30 GB/s, realized by a group of wavelengths in a specific fiber, and an input channel interface based on an array of N receivers, which simultaneously monitors all N channels, resulting in an effectively fully connected network. Further details about the SOME-Bus interconnection network can be found in Akay and Katsinis (2008).

The performance analysis of network architecture is a very crucial factor in designing DSM multiprocessor systems. Very often, simulation is the only feasible method because of the nature of the problem and because analytical techniques become too difficult to handle. Simulation occurs at many levels, from circuit to system, and at different degrees of detail as the design evolves. Execution-driven and trace-driven multiprocessor simulations (Sendag, Yilmazer, Yi, & Uht, 2007; Thiele, Wandeler, & Chakraborty, 2005) have been extensively used in order to obtain a reliable and accurate prediction of the final design. One of the problems with simulation is that although these simulations can be done at a high level of abstraction, they still are extremely time consuming. There are several reasons why this is the case. First, the benchmarks that need to be simulated typically consist of several hundreds of billions of dynamically executed instructions. Second, multiple of these benchmarks need to be simulated in order to cover a representative set of applications. Third, the complexity of the target system reflects itself in the complexity of the simulator making the simulator at least four orders of magnitude slower than native hardware execution. Fourth, during design space exploration all benchmarks need to be simulated multiple times in order to identify the optimal design for a given cost function covering performance, power, area, cost, reliability, etc.

With the objective of reducing simulation time without loosing accuracy, some interesting proposals have appeared in the last years. The first one is the sampled simulation, which chooses in an intelligent way a small portion of the program trace to simulate (Wenisch, Wunderlich, Falsafi, & Hoe, 2006). The second one is using a reduced set of the inputs of a benchmark (Eeckhout, Sampson, & Calder, 2005). Finally, there is statistical modeling and simulation, which characterizes the behavior of the program and architecture with some probability distributions (Genbrugge & Eeckhout, 2007; Nussbaum & Smith, 2002). However, although statistical simulation is a powerful tool in multiprocessor design, it can still be time consuming especially when the DSM multiprocessor system to be simulated has many parameters and these parameters have to be tested with different probability distributions or values. For instance, the run time of our statistical SOME-Bus simulator on a high performance computer is about 25 min when we would like to collect 500 performance measures with different values of five input parameters. Due to this problem, we propose to apply intelligent techniques for predicting the performance of a DSM multiprocessor in a faster way. The basic idea is to collect a small number of multiprocessor performance measures by using a statistical simulation and predict the performance of the system for a large set of input parameters based on these.

SVR has been proposed as an effective statistical learning method for regression problems. The foundation of SVR was developed by Vapnik (2000), and it became more popular due to the generalization capabilities it demonstrated. Recently, several studies (Cherkassky & Ma, 2004a; Wu, Ho, & Lee, 2003; Yang, Chan, & King, 2002; Yang, King, & Chan, 2002) have successfully applied SVR for function estimation. However, to the best of our knowledge, SVR has not been used for predicting the performance measures of a DSM multiprocessor.

In this paper, OPNET Modeler (OPNET Inc., 2007) is used to develop a statistical simulator of a multiprocessor architecture interconnected by the SOME-Bus optical network. All possible sequences of events that can occur during the processing of a memory reference are taken into account in the simulation environment. The simulator has been run for different values of the following DSM parameters: ratio of the mean message channel transfer time to the mean thread run time (T/R), probability that a block can be found in modified state $\{P(M)\}$, probability that a data message is due to a write miss $\{P(W)\}$, probability that a cache is full $\{P(CF)\}$ and probability of having an upgrade ownership request $\{P(UOR)\}$. Performance measures such as average processor utilization (i.e. average fraction of time that threads are executing), average response time (i.e. the time interval between the instant when a cache miss causes a message to be enqueued in the output channel until the instant when the corresponding data or acknowledge message arrives at the input queue) and average channel waiting time (i.e. the time interval between the instant when a packet is enqueued in the output channel until the instant when the packet goes under service) have been measured. One hundred and sixty two data points are collected to form a training dataset. SVR, Artificial Neural Networks (ANN) and Multiple Linear Regression (MLR) have been applied separately on the training dataset to build regression models. The regression models are then used to predict the performance measures of the SOME-Bus multiprocessor. Several metrics such as the mean absolute error (*MAE*), root mean squared error (*RMSE*), relative absolute error (*RAE*), root relative squared error (*RRSE*), correlation coefficient and absolute percent error are used to evaluate the performance of each regression model. The results show that the SVR-RBF model has the lowest prediction error and can be used as an effective tool for predicting the performance measures of a DSM multiprocessor architecture.

Section 2 summarizes the optical SOME-Bus interconnection network. Section 3 presents overview of linear and nonlinear SVR. Section 4 gives details of the OPNET-based simulation framework and the SVR model for predicting the performance of the SOME-Bus multiprocessor. Two kernel functions are examined during the selection of the model. Section 5 presents the prediction accuracy of the SVR models and compares it with the prediction accuracy of the models of MLR and ANN that, like the SVR model, can learn the relationship between performance measures and input parameters. Discussion of the results is also presented in this section. Finally, Section 6 concludes the paper along with outlining future directions.

2. Overview of the SOME-Bus interconnect

The SOME-Bus incorporates optoelectronic devices into a high performance processing architecture. It is a low-latency, high-bandwidth, fiber-optic interconnection network, which directly connects each node to all other nodes. One of its key features is that each of N nodes has a dedicated broadcast channel operating at 20–30 GB/s, realized by a group of wavelengths in a specific fiber, and an input channel interface based on an array of N receivers which simultaneously monitors all N channels, resulting in an effectively fully connected network (Fig. 1). No node is ever blocked from transmitting by another transmitter or due to contention for shared switch logic.