An Approach to Characterization of Parallel Applications for DSM Systems

Darko Marinov, Davor Magdić, Aleksandar Milenković,
Jelica Protić, Igor Tartalja, and Veljko Milutinović

Department of Computer Engineering, School of Electrical Engineering, University of Belgrade
POB 35-54, 11120 Belgrade, Serbia, Yugoslavia

E-mail: {marinov,dav0r}@galeb.etf.bg.ac.yu, {emilenka,jeca,etartalj,vm}@etf.bg.ac.yu

Abstract

This paper concentrates on the problem of defining and measuring parameters that characterize behavior of parallel applications targeted to DSM (Distributed Shared Memory) systems. Results are based on the SPLASH-2 application suite. The developed characterization tool Scopa, along with applied simulation environment Limes, are publicly available, and appropriate for performing measurements on other parallel applications, as well. Only a short summary is presented here; for details, the interested reader is referred to the full-blown technical report given in the list of references.

1. Introduction

There are many studies partially concerning characterization of parallel applications aimed to DSM systems, but we are not aware of any that provides a systematic review of wide range of parameters. The essence of our approach to characterization was: (a) to define an exhaustive list of valuable parameters based on our experience in analyzing DSM systems; (b) to classify these parameters into several groups according to some common features; (c) to design Scopa (System for Characterization of Parallel Applications [1]) – a tool that extends previously developed tool Limes (LInux MEmory Simulator [2]), execution-driven simulator of multiprocessor systems; and (d) to perform experiments that measure parameter values on a subset of SPLASH-2 (Stanford ParalleL Applications for SHared memory [3]) benchmark suite.

Proceedings of the IEEE HICSS-31,
Kona, Hawaii, USA, January 1998.

The parameters presented in this paper are important for two main reasons. Firstly, they serve as input to analytical models of DSM systems and/or to generators of synthetic address traces used in simulation analysis of such systems [4]. Secondly, these parameters are to assist researchers who try to enhance performance of DSM systems by improving memory consistency protocols [5].

2. Conditions and assumptions

Definitions of terms used in the rest of this paper are the following: an *epoch* is a sequence of instructions executed by a processor between two successive barrier operations, and an *interval* is a sequence of instructions executed by a processor between any two synchronization points. There are two types of intervals: CSI (Critical Section Interval) is a sequence of instructions executed by a processor between lock (acquire) and unlock (release) primitives, and NCSI (Non-Critical Section Interval) is a sequence of instructions executed by a processor outside of a critical section. Assumption is that there are no explicitly nested critical sections, and it holds as condition in SPLASH-2 programs.

The PRAM (Parallel Random Access Machine) architectural model is simulated, since it best characterizes the inherent behavior of the applications. All the simulations were conducted for a 32-processor system, but the number can be easily changed. The assumptions concerning processors are that there are as many processors in the simulated system as there are threads in a parallel application, i.e., each thread is executed on a distinct processor, and that each thread is pinned to its processor, i.e., no migration is modeled. Furthermore, there are no additional threads except those created by parallel application, and the activities of the operating system kernel are neglected.

3. Parameters

The set of parameters measured in this work is divided into three semantic groups.

Scalar parameters group contains some general behavioral indicators. Each of the following parameters represents only one value per application:

- total number of locks used on all processors;
- total number of barriers used on all processors;
- average number of processors that access the same lock;
- frequency that given processor requires the same lock that it has just released;
- frequency that given lock is acquired again by the same processor that has last released this lock;
- frequency that given epoch has no critical sections;
- average number of CSI in epochs with at least one CSI;
- frequency of CSI with no shared data writes;
- frequency of NCSI with no shared data writes;
- average number of cycles between two consecutive acquire operations for the same lock.

Access-type-indexed array parameters group encompasses total and average numbers of instructions executed between particular synchronization points. Each of the following parameters represents an array of values separately measured for each type of operation (no memory access, private/shared reads/writes of ordinary shared variables, and synchronization primitives):

- total number of instructions executed by all processors;
- total number of instructions executed in all CSI on all processors;
- average number of instructions executed during one CSI;
- total number of instructions executed during all NCSI on all processors;
- average number of instructions executed during one NCSI;
- average number of instructions between two successive acquire operations for the same lock.

Block-size-indexed array parameters group is devoted to parameters considering physical coherence units of address space referred to as blocks. The following parameters are measured for wide range of block sizes: average/maximum/minimum number of blocks modified inside one CSI/NCSI with shared data writes, average/maximum/minimum number of blocks which are only read inside one CSI/NCSI with at least one read-only block, and frequency that CSI/NCSI contains at least one read-only block.

4. Characterization tools

In order to enable measurement of relevant parameters we used Limes as simulation environment, and originally developed characterization tool Scopa, which consists of two parts that behave like simulators. One of them is used for obtaining parameters of groups 1 and 2 that can be measured during execution-driven simulation, and the other one serves only as generator of reduced trace files, a kind of intermediate form used for collecting data relevant for parameters of group 3. Certain post-simulation examination of these files by additional program is needed to compute these parameters.

Reduced trace files can ease further analyses of application behavior, since they might be used to obtain values of parameters yet to be defined with less programming efforts and execution time. Also, these files may be input to characterization tools on another systems, independent of Limes, and even Linux.

All results generated during this characterization are available at: http://www.galeb.etf.bg.ac.yu/~marinov/

5. Conclusions

Our future plans include the development of models for evaluation of algorithms and architectures in the field of DSM systems based on acquired parameter values. We also plan to reevaluate memory consistency models and their implementations according to the results obtained in this study.

6. References

[1] Marinov, D., Magdic, D., Milenkovic, A., Protic, J., Tartalja, I., Milutinovic, V., "Characterization of Parallel Applications for DSM Systems," Technical Report TI-ETF-RTI-97-040, School of Electrical Engineering, University of Belgrade, Belgrade, Yugoslavia, September 1997. (http://www.galeb.etf.bg.ac.yu/~marinov/)

[2] Magdic, D., "Limes: A Multiprocessor Simulation Environment," *TCCA Newsletter*, March 1997, pp. 68-71.

[3] Woo S. C., Ohara M., Torrie E., Singh J. P., Gupta A., "The SPLASH-2 Programs: Characterization and Methodological Considerations," *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, Santa Margherita Ligure, Italy, June 1995, pp. 24-36.

[4] Protic, J., Tomasevic, M., Milutinovic, V., "Distributed Shared Memory: Concepts and Systems," *IEEE Parallel and Distributed Technology*, Vol. 5, No. 1, Summer 1996.

[5] Protic, J., Tartalja, I., Tomasevic, M., "Memory Consistency Models for Shared Memory Multiprocessors and DSM Systems," *Proceedings of the 8th Mediterranean Electrotechnical Conference melecon '96*, Bari, Italy, May 1996.