Mermaid: modelling and evaluation research in MIMD architecture design

Pimentel, A.D.; van Brummen, J.; Papathanassiadis, T.; Sloot, P.M.A.; Hertzberger, L.O.

Publication date
1995

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Modelling and Evaluation Research in MIMD Architecture Design (MERMAID)

A.D. Pimentel    L.O. Hertzberger    P.M.A. Sloot
Dept. of Computer Systems, University of Amsterdam
October 10, 1995

Keywords: performance evaluation, simulation, MIMD architectures

Introduction

The Mermaid project focuses on the construction of simulation models for MIMD multi-computer architectures with the purpose of performance evaluation [Pimentel95]. At the beginning of the project two guiding principles were formulated. First, the level of architecture abstraction should allow simulation within reasonable time, preferably avoiding low-level (bus-cycle) emulation. And second, architecture choices are important design options which must be simulated without too much remodelling effort.

The MIMD architectures modelled within Mermaid are the GCel, the PowerXplorer and future PowerStone architectures of Parsytec [Langhammer93].

The Mermaid simulation methodology

To evaluate MIMD multi-computers, a parameterized algorithmic model was created. It is capable of supplying the simulator with a trace of events, called operations, representing processor activity, memory I/O, and communication message passing.

Simulation of an application load on an architecture takes place at three different levels, as depicted in Figure 1. The application level contains application descriptions used as input to our simulation models. This is done by either stochastically describing the behaviour of programs using probabilities or by instrumenting real programs with annotations representing the exact execution behaviour.

At the generation level a trace of operations is generated from the application descriptions. This generation process exploits knowledge of the target architecture and runtime model in order to tune these operation traces.

Figure 1: Simulation scheme.
Finally, the architecture level consists of operation trace-driven simulation models. Every model has a set of machine parameters that has been calibrated by either published information or benchmarking. To allow simulation within reasonable time, the simulation models do not fully emulate the hardware and keep limited state information during simulation.

Architecture modelling

The architecture models, implemented in Pearl [Muller93], are generic in the sense that the different Parsytec architectures can be represented by means of parameterization. The computation module of a MIMD node is modelled as a number of processors and caches, a bus and a memory. The model for a complete node is constructed by combining the computation module with a router component and four communication links (shown in Figure 2). All node components are fully parameterized to be able to evaluate design options at board-level and to evaluate different routing strategies.

The nodes are linked together resulting in a multi-node model to reflect the platform’s interconnection scheme. Application loads result in an operation trace for each node in the multi-node model. This approach allows for all kinds of synchronization and load balancing scenario’s.

An example experiment

This section presents the validation results for an application load running on the Parsytec GCell architecture model. The application is a parallel solver of linear equations using Gaussian elimination. Figures 3.a and 3.b depict the execution time and estimated (simulation) time for a range of runs with different equation-matrix sizes.

From the curves can be seen that the estimated times are slightly more optimistic than real execution. However, with a worst-case error of 12 percent, the simulation estimates are reasonable accurate. Another important aspect is that simulation closely follows the execution trend. Even for the small peak at a problem size of 100x100 with 32 processors, this trend is perfectly followed.

Future work

Much effort has been put in validating the GCell architecture model. This work will proceed for the PowerXplorer architecture model. Further, application modelling will continue in order to extend the diversity of application loads. Finally, models of future architectures will be built for evaluation purposes after which feedback can be provided to the designing team.
Figure 3: Simulation of parallel Gaussian elimination.

References


Contact address

Contact: A.D. Pimentel
Institution: Computer Systems Department
University of Amsterdam
Mail: Faculty of Mathematics & Computer Science
University of Amsterdam
p/a A.D. Pimentel
Kruislaan 403
1098 SJ Amsterdam
Phone: +31 20 525 7578
Fax: +31 20 525 7490
Email: andy@fwi.uva.nl
WWW: http://www.fwi.uva.nl/fwi/research/vg4/arch/