Performance Evaluation for Marker-Propagation Parallel Processing Systems [1]

Ronald DeMara[2] and Dan Moldovan

Parallel Knowledge Processing Laboratory
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-1115
Tel: 213-740-9127, Fax: 213-740-4449

Abstract

The need for fast, scalable, and efficient computer architectures for Artificial Intelligence (AI) applications is well recognized. However, relatively little is known about the performance of parallel AI architecures. In this paper, we present techniques to evaluate and improve *marker-propagation architectures* which utilize the massive parallelism inherent in many AI applications. Based on an analysis of marker-passing programs and knowledge bases, we developed a set of performance indices by defining concepts such as marker *Power* and *Dispersion*. These indicies and a classification of workloads were used to construct a suite of performance benchmarks. We then developed a 160-processor marker-passing supercomputer called SNAP-1 which was tailored to provide visibility into the performance-critical features of marker-propagation architectures. Finally, we devised a hybrid hardware/microcode tracing methodology to collect and interpret the results in terms of the metrics we have defined.

Index Terms: Knowledge Processing, Marker-Passing, Multiprocessor Architecture, Performance Measurement, Semantic Networks

---

[1]This research funded by National Science Foundation Grants MIP-89/02426 and MIP-90/09109 and the Texas Instruments' University Program.

[2]Electronic mail address: demara@gringo.usc.edu

1 Introduction

During the last decade, advances in parallel processing and VLSI technology have had considerable impact on computers for numeric computation, but neglible impact on those for Artificial Intelligence (AI) applications. Yet a broad class of AI applications, such as real-time speech-to-speech translation, robotic control, high-level reasoning, and computer vision require systems with orders of magnitude increases in performance over currently available machines. Several parallel AI architectures have been proposed and a handful of systems have been built. However, their performance remained insufficient for many complex AI tasks.

A fundamental problem is that parallel processing of AI problems is not well understood. In particular, more research is needed to analyze the parallelism in AI applications and to identify the factors affecting system performance. In this paper, we discuss *methods* and *tools* for evaluating the performance of *marker-passing* programs on loosely-coupled multiprocessors. We are interested in marker-passing because it is a powerful and intrinsically parallel programming model for AI problems. Our goals are to *quantify* the parallelism in marker-propagation algorithms and to *improve* the performance of computer architectures for marker-passing. Ultimately, we seek answers to the following questions:

- How can marker-passing algorithms be characterized and described?
  - What is the "size" of a marker-passing algorithm?
  - How much and what types of parallelism exist in a marker-passing programs?
  - What is the effect of performance degradation due to serial sections of the algorithm?
- How well does a parallel AI architecture fit the needs of the algorithms?
  - How can a parallel marker propagation architecture be balanced for optimal performance?
  - What are the costs of synchronization and communication?
  - What grain size realizes the most efficient computation?
  - How do interconnection strategies affect performance?
  - What factors affect the speed of marker propagation?
  - Of the primitive operations most heavily used, which are the most time consuming?

1.1 Marker-Passing Paradigm

Marker-Passing is a promising approach to utilizing the parallelism inherent in many AI applications. Reasoning operations are accomplished by wavefronts of activation that *markers* spread in parallel throughout a knowledge base. Each marker is implemented as a message containing a set of binary flags, numeric values, propagation rules, and activation source-IDs. Waves of marker activations are used to change the state of the concepts in the knowledge base. At the end of the propagation, certain concepts obtain global or local maximum activation strengths. These marked concepts represent active hypotheses.

Typically, the knowledge base is represented as a *semantic network*. A semantic network is a knowledge representation scheme which codifies information in the form of a directed graph. A semantic network node is capable of storing a single fact, concept, rule, pattern, etc. Programmable interconnections between nodes are called *relations*. Each relation type denotes a different attribute or relationship between nodes.

Weights or probabilities are attached to both the individual markers that make up the activation wave and the relations they travel through to permit *cost calculation* and *probabilistic reasoning*. An efficient marker propagation architecture should propagate multiple markers in parallel. In this way, many alternative hypotheses are evaluated simultaneously, such as which step of a plan a robot should execute next or the intended meaning of some natural language utterance.

The marking is done by propagating messages from marked nodes to other nodes as dictated by the *propagation rule*. The propagation rule avoids marking of incorrect (semantically unrelated) nodes by passing markers only along relations of specified types. It also allows the marker passing operations to occur under distributed rather than centralized control.

1.2 Performance Approach

In comparison to other domains, performance analysis for knowledge processing lacks several prerequisites. In particular, few meaningful workload classes, measures of application size, benchmarks, or metrics have been defined. Thus our approach to performance evaluation spans both *application-oriented* and *architecture-oriented* aspects, as shown in Figure 1.

On the application side, we analyzed marker-passing subroutines to classify typical marker-passing *workloads*. This identified the basic operations

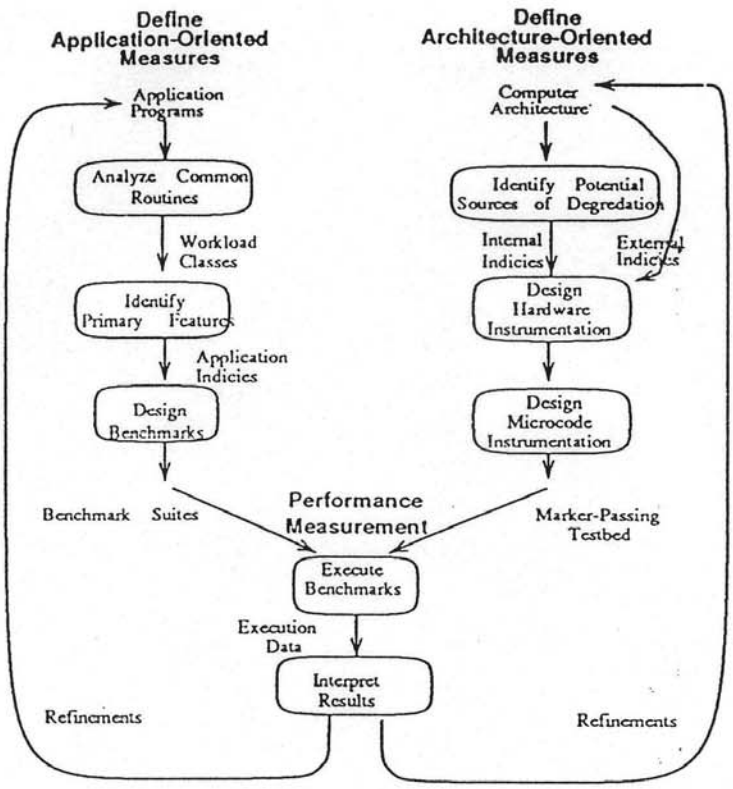

Figure 1: Performance Evaluation Process

performed and the range of algorithms encountered. For each workload class, we defined representive *performance indicators* to quantify the salient features of the algorithms and knowledge bases. Finally, we defined benchmark suites similar to those that already exist for numeric supercomputers, database machines, or RISC workstations. We concentrated on making this set of benchmarks portable and reflecting the relevant indicators for each class of algorithms.

On the architecture side, we identified metrics for several sources of performance degradation in marker-propagation architectures. Currently, little qualitative information and no quantitative measurements for communication, synchronization, latency, and starvation overheads have been obtained for marker-propagation architectures. Thus, we developed methods to collect this low-level data and relate it back to the overall performance of the machine.

Furthermore, experience has shown that when developing a novel com-

puting architecture, theory alone is not sufficient for proving capability and ensuring optimal performance. Performance in massively parallel AI architectures is dependent on the intricacies of the algorithm and its implementation, the structure knowledge base, the low-level software support and allocation management, as well as the underlying parallel hardware. It is typically infeasible to accurately capture all of these interacting effects in an analytical or simulation model. Thus, our approach to measurement has been to execute benchmark programs on an instrumented processor to permit refinement of the algorithms and architecture.