

PMaC Prediction Framework

Table of contents

1 Overview.....	2
2 Obtaining a Machine Profile.....	2
3 Obtaining an Application Signature.....	3
4 Convolving.....	3

1. Overview

The PMaC Prediction Framework is an implementation of an automated prediction model. A **prediction model** is a calculable expression that takes as parameters attributes of application software, input data, and target machine hardware (and possibly other factors) and computes, as output, expected performance. The expected performance in this case is the expected runtime of the application.

The PMaC Framework operates by using the following three elements.

Machine Profile

Characterization of the rates at which a machine can (or is projected to) carry out fundamental operations abstract from any particular application.

Application Signature

Detailed summaries of the fundamental operations carried out by the application independent of any particular machine.

Convolution Methods

Algebraic mappings of the Application Signatures on to the Machine profiles to arrive at a performance prediction.

framework overview

Figure 1. An outline of the performance prediction model

The following sections outline the prediction process in more detail.

2. Obtaining a Machine Profile

For the purposes of the PMaC prediction framework a **Machine Profile** is a characterizations of the rates at which a machine can (or is projected to) carry out fundamental operations, abstract from any particular application.

Memory and network performance are measured by simple benchmarks run on 1 to 2 nodes of a target system to create the machine profile.

Memory Operations

The performance of a machines memory sub-system is measured using PMaC's [MultiMAPS](#) Benchmark. MultiMAPS measures the bandwidth achieved by the machine while retrieving a variety of data sizes using different strides and line sizes. MultiMAPS is only run on a single node of the system. Figure 2 shows a sample of the data given by MultiMAPS for 3 different systems.

Network Operations

Network performance is measured by a simple mpi ping-pong benchmark run on 2 nodes of the system in various ways.

Sample data collected from MultiMAPS

Figure 2: Sample data collected from MultiMaps

3. Obtaining an Application Signature

Two categories of operations are observed in order to obtain an application signature, memory and network operations, collected by PMAcInst and MPIDTrace respectively.

PMAcInst

PMAcInst is a binary re-writer that instruments memory and floating-point operations. The memory address stream of those operations is processed on-the-fly through cache simulator. The cache simulator is capable of simulating >20 different cache structures. Output is a summary for each cache structure

PSiNS Tracer

An instrumentation tool (link time) to capture information about MPI calls including the time spent in each call, argument values to the call, and hardware counter information about the execution. PSiNS Tracer combines the minimal features of MPIDTrace and IPM tools.

MPIDTrace

Network operations (mpi calls) are recorded by a tool called MPIDTrace. MPIDTrace is part of the Dimemas project, developed by the [European Center for Parallelism at Barcelona](#). MPIDTrace creates an event trace file, which is used by Dimemas for simulation in one of the convolution steps.

4. Convolution

There are two convolution steps. The first step, the PMAc Convolver, models the work done on the processor in between communication events. This model work or time is used by Dimemas to simulate the execution of the application on the target system.

PMAc Convolver

This convolution step uses the trace results from PMAcInst and the memory profile data to produce an estimated time for the work done on the processor in between communication events on the target machine for that application.

PSiNS Simulator

Similar to Dimemas, PSiNS Simulator uses the estimated time from the PMAc convolver along with MPI call trace produced by different tools (PSiNS Tracer, MPIDTrace) to simulate the execution of the application on the target system and emits the execution time as well as significant information on the decomposition of execution time to program components similar to the information given by IPM.

Dimemas

Dimemas uses the estimated time from the PMaC convolver along with the event trace produced by MPIDtrace to simulate the execution of the application on the target system.