Knowledge Engineering for Model-based Parallel Performance Diagnosis

Li Li and Allen D. Malony, {lili, malony}@cs.uoregon.edu
http://www.cs.uoregon.edu/research/tau
Computer and Information Science Department, University of Oregon, Eugene, OR

Abstract

Scientific parallel programs often undergo significant performance tuning before meeting performance expectations. Performance tuning naturally involves a diagnosis process - locating performance bugs that make a program inefficient and explaining them in terms of high-level design. We present a systematic approach to generating performance knowledge automatically diagnosing parallel programs. Our approach exploits program semantics and parallelism embedded in computational models to search and explain bugs. We first describe how to extract performance knowledge from parallel models. Second, we represent the knowledge in a way such that diagnoses can be carried out in an automatic manner. We then present Hercule - a prototype automatic performance diagnosis system - and our evaluation system for validating diagnosis results. Our experience diagnosing Master-Worker programs shows that model-based performance knowledge can provide effective guidance for locating and explaining performance bugs at a high level of program abstraction.

Introduction

Two observations of existing important performance measurement and analysis tools particularly motivate our work:
- The performance feedbacks provided by the tools tend to be descriptive information about parallel program execution at low level of program abstraction.
- The design of performance experiments, examining performance data, and evaluating performance against the expected to identify performance bugs are not well automated and not naturally guided by a diagnosis strategy.

Our approach to performance diagnosis is based on parallel computational models. A computational model is a recaing algorithmic and communication pattern in parallel computing, e.g., Master-Worker, Pipeline.

Knowledge Base

- Performance metric set
- Metric data/trace
- Compute performance metrics
- Evaluate against the expected
- Exploitable performance factors
- Normalize performance factors
- Select targeting metric set
- Specify performance metrics
- Generate performance data/profile
- Evaluate performance factors

A high-level view of model-based iterative performance diagnosis process. The use of model knowledge is annotated on the process steps.

Representation of Performance Knowledge

How do we represent performance knowledge in a knowledge base so that diagnosis process can be performed with minimum user intervention and algorithms- and implementation-specific performance information can be readily derived and incorporated into the system?

We create an inference tree that represents our bottom-up performance diagnosis approach and formalizes a structured knowledge invocation process.

- The root of the tree represents the symptom that we are going to diagnose.
- Branch nodes represent intermediate observations that we have achieved so far and need further performance evidences to explain.
- Leaf nodes represent an explanation of the root symptom in terms of high-level performance factors.
- Rapidly incorporate knowledge generated from algorithm variants through adding branches at appropriate tree levels.
- An example inference tree of a Master-Worker model is presented in the figure below.

Experiments and Results

A problematic Master-Worker program run with 7 processors on a SMP cluster with 16-dual processor nodes connected by gigabyte switches.

Conclusions

We describe a systematic approach to generating and using model-based performance knowledge for the purpose of automatic performance diagnosis. The methodology makes use of operation semantics and parallelism found in parallel computational models to diagnose for performance bug search and explanation. In the approach, we present an intermediate step that identifies abstract events that describe model knowledge, i.e., the performance modeling, model-specific metrics, and performance cause inference steps.

A knowledge base is engineered by interpreting expert information in each category for parallel models of interest. The methodology also addresses the adaptability of knowledge generation to diagnostic task and implementation variants. One important objective of our work is to study how model knowledge is represented. In this work, we showed the use of inference tree for formalizing a structured knowledge invocation process for automatic performance diagnosis. We also demonstrate the use of prototype Hercule parallel performance diagnosis system in representative programming paradigms, Master-Worker. Our preliminary results show that model-based performance knowledge provides effective guidance for performance bug search and explanation process at a high level of program abstraction.