EXTENSIBLE SOURCE-LEVEL INSTRUMENTATION TOOL FOR PERFORMANCE MEASUREMENTS

By

Jonathan Chen

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF COMPUTER SCIENCE

Approved:

_____

Dr. Boleslaw K. Szymanski
Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

May 2007
(For Graduation August 2007)

ABSTRACT

In the area of parallel computing, the complex and computationally demanding nature of scientific applications as well as the increasing prevalence of parallel computers has dramatically increased the number of parallel applications. As optimizing parallel code can be time consuming, there are often ineffiencies that are left unresolved in parallel application codes. The purpose of this thesis is to describe the automated instrumentation tool that the author contributed to the Instrumentation Database (IDB) [**?**] framework.

The IDB framework is a scalable approach to collect performance data in such a way that the problem size and run-time environment do not affect the amount of information collected. In order to collect data, IDB probes must be added to the program being instrumented. These probes collect information about the control flow of the program as well as various performance data. The data collected by the probes is subsequently stored in a database for further visualization and analysis.

Manually inserting probes into existing code can easily be the most tedious task in the process of using the IDB framework. This thesis focuses on improvements to the IDB framework by automating the task of source code instrumentation. We present our reasons for instrumentation at the source code level and demonstrate its effectiveness through an implementation of the automated instrumentation tool. We show that our instrumentation tool, when coupled with the IDB framework, enables easy analysis of parallel codes.