IISWC'07 Tutorial Call For Participation:
Using the Pin Instrumentation Framework for Workload Characterization

Saturday, September 29th, 2007
Boston, MA

With several new emerging application domains, understanding the requirements of newer applications is essential in designing future high performance processors. Such workload characterization and exploratory studies require fast and efficient techniques that can determine the behavior of these emerging workloads. This workshop will illustrate the use of Pin to conduct workload characterization and performance studies. Pin is a dynamic instrumentation system provided by Intel that has become widely used throughout academia and industry. Pin allows code (C/C++) to be injected at arbitrary places in an executable while it is running. The injected code is used to observe the behavior of the program, and can be used to write a variety of workload characterization tools such as application profilers and trace generators. Pin provides a rich API that abstracts away the underlying instruction set idiosyncrasies and allows context information such as register contents to be passed to the injected code as parameters. Pin automatically saves and restores the registers that are overwritten by the injected code so the application continues to operate normally. Pin makes it easy to do studies on complex real-life applications, which makes it a useful tool for enabling workload characterization studies.

This tutorial targets researchers, students, and educators alike, from the novice Pin user to the expert Pinhead. The tutorial will provide a brief background on Pin and describe how to build simple Pin tools that can help in workload characterization.

AGENDA

This tutorial consists of four presentations. The first presentation provides an introduction to the Pin API and the basic concepts of writing instrumentation tools in Pin. Detailed instructions on writing instrumentation tools useful for architecture research are also presented. The next three presentations consist of research projects that use Pin.

Time	Talks
1:30 – 2:15	"***Introduction to Pin"*** by Harish Patil
2:15 – 3:00	***"Micro-architecture-Independent Workload Characterization Studies Using Pin"*** by Kenneth Hoste/Lieven Eeckhout
3:00 – 3:30	---------- Break ----------
3:30 – 4:15	***"Sampling Techniques for Pin Based Simulation"*** by Harish Patil
4:15 – 5:00	***"Micro-architecture Studies Using Pin: Branch Predictors, Caches, & Simple Timing Models"*** by Aamer Jaleel

Questions? Please contact: aamer.jaleel@intel.com or harish.patil@intel.com

Introduction To Pin, *Harish Patil, Intel*

Pin is a dynamic instrumentation system that allows users to write their own program analysis tools using a specified application programmer's interface. This section covers the basics of Pin and writing Pin tools.

Micro-architecture-Independent Workload Characterization Studies Using Pin
Kenneth Hoste and Lieven Eeckhout, Ghent University, Belgium

This session presents MICA: Micro-architecture-Independent Characterization of Applications. MICA is a Pin tool for measuring run time behavioral program characteristics. In contrast to common workload characterization, these characteristics are completely independent of the micro-architecture (e.g. cache configuration, branch predictor, number of functional units, etc.) on which the program runs. This session will show how to use the Pin instrumentation framework to obtain these characteristics. We will explain how to extract the needed information from the dynamic instruction stream using Pin, process the data collected and gain insight into the runtime behavior of an application using the obtained characteristics.

Sampling Techniques for Pin Based Simulation, *Harish Patil, Intel*

Detailed analysis of large programs can be very time-consuming. It is therefore desirable to be able to focus on certain portions of the applications for analysis. This section covers the hooks pin-tool writers can use to efficiently process only certain regions in the applications. A toolset that helps find representative application regions (PinPoints) will also be described.

Micro-architecture Studies Using Pin: Branch Predictors, Caches, & Simple Timing Models, *Aamer Jaleel, Intel*

This session of the tutorial demonstrates the use of PIN as an alternative to execution-driven and trace-driven simulation methodologies. We illustrate the use of Pin to do micro-architectural studies such as branch predictor simulation, cache simulation, and even simple-timing models. The session also describes CMP$im, a Pin based CMP model. CMP$im can be used to understand overall memory behavior of a workload as well as characterize instruction profile, cache performance, and data sharing behavior of workloads at speeds of 4-10 MIPS.