Instruction Sampling and Profiling

CScADS, July 20, 2009

Marty Itzkowitz, Analyzer Project Lead
Sun Microsystems Inc.
marty.itzkowitz@sun.com
Outline

• Objectives
• Instruction Sampling Events
  > Data delivered
• Instruction Sampling Initiation
  > How is sampling specified
• Instruction Sampling Event Delivery
  > Signal with context delivery
    – Synchronous to thread
  > File I/O
Objectives

- Use instruction sampling to get metrics of performance
  - Statistical sample of instructions is valid statistic over run
  - Data to provide detailed cost breakdown of execution
  - Attribute costs to instruction, source-line, function
    - If event is synchronous-enough, attribute costs up callstack

- Develop APIs to allow Instruction Sampling
  - Chip-independent description
    - Sample selection
    - Event delivery
What Tools (ours, anyway) Want

- **Introspection** – triggered by process itself
- **Per-thread sampling**
  - Collect data about the instruction execution
  - Deliver each event with a signal
    - As soon as possible; synchronous to thread
  - We'll try to generate events at ~ 0.1 – 1 event/millisecond
  - Propagate to new threads? Intercept thread start?
- **Collect callstacks at signal**
  - Verify context in event, and current context “close enough”
    - Replace leaf in stack with event PC
Instruction Sampling Events

• Data delivered
  > Want detailed information, including
    – Times at various parts of pipeline
      – Cost or costs of that instruction
    – Virtual (and physical) addresses
    – Instruction PC
  > Each chip is different

• Need API to give description of events on that system
  > Define all possible fields?
  > Special values for data not supported on that system
Instruction Sampling Initiation

- Need API to give description
  - How are instructions selected?
  - What filtering criteria are supported?
  - What delivery mechanism is supported?
Instruction Sampling Event Delivery

• Best: signal-based, with context containing event data
  > Or syscall to fetch event data
    – Must be async-signal safe
  > As synchronous as possible
    – So we can unwind stack, and get program context

• Not as convenient: I/O based
  > Can't be very synchronous
  > Takes file descriptor from user space
Thank You

Marty Itzkowitz, Analyzer Project Lead
Sun Microsystems Inc.
marty.itzkowitz@sun.com