Linux Support for Monitoring and Controlling Complex Executions

Working Group Outbrief

Scalable Tools Workshop 2025

Needs

- For applications to run at large scale, executables, their dynamic shared libraries need to load efficiently
    - Motivation for LLNL's Spindle
- Measuring the performance of complex applications requires intricate integration of tooling with application code
    - Wrapping functions in application and its libraries
    - Loading tool code into namespaces so applications and tools don't conflict
    - Dynamically loading only tool components needed for measurements desired
- Need support for multiple tools
    - At a minimum: Spindle + Performance Tool
    - Need support to integrate these functionalities
        - Don't want to prevent fast loading to use a tool
        - Would like fast loading help load the tools

Concerns

- LD_AUDIT supports observation and control of dynamic library loading and symbol binding
    - There still seem to be some bugs
        - Are symbind callbacks generated everywhere they are needed?
    - If you wrap dlopen, library constructors are invoked before you get control
        - Lack an interface for intercepting execution immediately after an object's constructor fires
- Glibc
    - Bugs
        - Support for thread-local storage with multiple namespaces
        - Support for thread-local storage for auditors
    - Missing features
        - Lacks some functionality that would help tools
        - Support for an interface to rebind symbols for interposition like LLNL's GOTCHA
            - Propose dlresym (for rebinding a symbol) as a building block for GOTCHA-like interposition
        - Support for attaching tools to running process, e.g. like thread_db for debuggers
- Libdl
    - Need a way to get a path to an object that doesn't have buffer overflow

Beyond LD_AUDIT

- Preliminary efforts for interfaces above LD_AUDIT
    - Audacious (Matt LeGendre)
        - Function wrapping
        - Library redirection: old path → new path
    - Ninlil (Jonathon Anderson)
        - High-level C++ layer on top of LD_AUDIT
        - Library tracking and redirection

Next Steps

Make progress in the presence of bugs

- Integrate code from GOTCHA into HPCToolkit to sidestep problem with pthread keys on Aurora
    - Locate pthread key routines without using libdl interfaces that create pthread keys!

Work towards a better future

- Assemble detailed list of bugs and key needs
    - With reproducers and test cases
- Engage Linux developers through Ben Woodard
- Rice and LLNL to collaborate on an API beyond LD_AUDIT
        - begin to formalize a basic interface

GLIBC Feature Requests

- First-party tool interface for notification on library loading
  - If you wrap dlopen, library constructors are invoked before you get control
- More notification of interesting events for tools
  - R_BRK is how debuggers intercept library loading
    - Be able to add a callback inside R_BRK rather than just having it as a place a debugger can put a trap
  - How to intercept file operations
    - API?
    - Wrapping?
      - Strengths of wrapping: easy to change or chain wrappers
    - Pad interfaces with nops so interception could be injected
- First-party interface for SDT points?
- Introspection API
  - If attaching to a process, want a thread_db-like interface to understand the threads present
- Should have a GOTCHA-like interface
  - Why? To support Pthread_keys using GOT rewriting
  - Wild-card style wrapping is useful
- LD_AUDIT
  - Needs hook for library constructors
  - la_symbind is missing notifications sometimes
    - && vs. ||
    - Also missing for data relocations
- Dlresym
  - Replace GOT entry for a symbol in the context of a library with this value
- Wrapping focused interface rather than a function symbol-focused interface

GOTCHA

- Library for function interposition
  - For example, enables a tool to inject wrappers for MPI functions inside an application without involving library preloading or using LD_AUDIT
    - Caliper uses this strategy
- Currently only works on x86 and power
- Writing a first party tool that is not loaded with LD_AUDIT. Want to know when libraries load
- What would it take for glibc to provide GOTCHA support?
  - Need a white paper
    - Problems it solves
  -

Caliper

- Wants to live in process namespace: needs to be visible to application
  - Can’t be an auditor
    - There is some disagreement about this
  - Supports user-configurable wrapping
    - MPI functions
    - I/O functions
    - Umpire measurements