Enhanced filtering of data using data-driven analysis

Jean-Christian Kouamé
Michel Dagenais

Progress Report Meeting
Dec 2014
• Introduction
• Motivation
• Filter Tool
• Interested cases
• Further works
• Demo
Introduction

- **Trace Compass**
  - Trace visualizer
  - Allow a lot of analysis (Kernel, CPU usage, control flow, ...)
  - Analysis deals with a huge amounts of events

- **XML part**
  - Previous works (Florian Wininger)
  - XML analysis

- **Contribution**
  - XML pattern description language
  - Filter analyzer
  - Event filterer
  - Filter views
Motivation

- Why using Trace Compass?
  - Analyze amount of information inside the trace
  - Many angle of analysis
- Problems
  - Too much information
  - Difficulty to move into the trace
  - Difficulty to see recurrent small problems
- What users want?
  - Interest for some sequences of events
  - Interest for few types of events with specific values
Filter Tool

• Utility
  • Filter data
  • Detect complex default
  • Follow mecanism
  • Generate high-level events
• Description language
  • Why using XML?
    • Already support in Trace Compass
    • Simplicity
    • Extensible
Filter Tool

- Description language
  - 3 main entities
    - Finite state machine (FSM)
      - Describe the pattern (scenario)
      - Support preconditions and preactions
    - Transitions
      - Conditions that trigger the state transitions
      - Conditions based on events or on the time
  - Actions
    - Action to execute
    - Supported Actions:
      - State changes
      - Generate synthetic events
      - Start a new FSM
    - Possibility to combine actions
Filter Tool

- XML structure

```xml
<filterHandler filterName="sched_switch">
    <initialFsm id="sched_switch"/>
    <transitionInput id="sched_switch">
        <event eventName="sched_switch"/>
    </transitionInput>
    <action id="update Current_thread">
        <stateChange>
            <stateAttribute type="location" value="CurrentCPU"/>
            <stateAttribute type="constant" value="Current_thread"/>
            <stateValue type="eventField" value="next_tid"/>
        </stateChange>
    </action>
    <fsm id="sched_switch" multiple="false">
        <precondition input="sched_switch"/>
        <stateTable>
            <stateDefinition name="sched_switch">
                <transition input="sched_switch" next="sched_switch" action="update Current_thread"/>
                <transition input="#other" next="sched_switch"/>
            </stateDefinition>
        </stateTable>
        <initialState id="sched_switch"/>
    </fsm>
</filterHandler>
```
Filter Tool

- Debugging the patterns
  - Time graph view
  - Follow the scenarios execution
  - Show the status, the state and the variables
Interested Cases

• System calls (kernel)
  • Abstraction of all system calls
  • Zoom-in
    • Follow process
    • Follow file
    • Look into irq
    • Follow socket connection and data transfer
Interested Cases

• SYN Flood Attack
  • 2 steps
    • Half TCP connections
    • Threshold

• Tools
  • Apache
  • Hping3
  • LTTng-modules (Francis Giraldeau github)
Further Works

- Optimisation
- Choose what filter to run
- Event criteria filterer