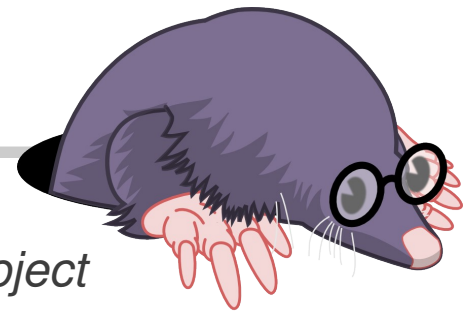


Live Trace Analysis Infrastructure

Yannick Brosseau
yannick.brosseau@polymtl.ca

December 8, 2010

Mid Project Meeting, Distributed Multi-Core Tracing Project



Content

1. Objectives
2. Tracer Requirements
3. Current Implementation
4. Future Work

Objectives

- Allow reading and analysing a trace as it is recorded while:
 - Maintaining a time ordered view with non synchronous flushing of different tracefiles
 - Allowing for new events definition in the stream
 - Synchronizing the time of multiple traces
 - Computing the statistics correctly

Tracer Requirements

- To allow the viewer to read a live trace, the tracer must write its buffer periodically
- You need to specify a flushing period to the LTTng or UST daemon

Implementation (libtraceread)

- Libtraceread changes:
 - Change open_trace API to add a is_live flag
 - In live mode:
 - No data available does not mean the end of the trace
 - Instead return a flag indicating that more data are to come
 - Currently, it's the client's responsibility to poll for new data

Implementation (viewers)

- Current status of the viewers:
 - LTTV text mode
 - Simply print new events
 - Graphic mode (LTTV)
 - Can open live trace
 - Need to change the time frame management to refresh time boundaries
 - Need to refresh displays, but with new data only

Challenge

- No notification from tracer to viewer
- Asynchronous file writing, with dependency between files
 - We need to read the metadata for an event before reading the event
 - The metadata file might be flushed after the event file

Solution

- Defer the reading of the newest parts of the trace files
 - Get the time of the latest event in the trace files at the current reading
 - Wait for some time. (Which should be the period of the flushing of the tracer buffers)
 - Read the event up to the time of the latest event previously read
 - (Now the metadata will have been flushed)

Conclusion and future work

- Allow the user to adjust the polling frequency
- Integrate inotify in LTTV to react to file changes on disk instead of polling
- New channels/CPU's to appear after the starting of the viewer
- Auto-detect if a trace is live or not
- Synchronisation between different traces