Mid-Project Report
December 8th, 2010

Tracing and Monitoring Distributed Multi-Core Systems

Adaptative Fault Probing

E-mail:
mathieu.desnoyers@efficios.com

> Presenter

- Mathieu Desnoyers
- EfficiOS Inc.
  - http://www.efficios.com
- Author/Maintainer of
  - LTTng, LTTV, Userspace RCU
- Ph.D. in computer engineering
  - Low-Impact Operating System Tracing

> Generic Ring Buffer Library

- Derived from the LTTng ring buffer
  - Exists since 2005
  - LTTng ring buffer ported to user-space (UST)
- Goals
  - Generic and flexible
  - Clean API
  - Fast and compact
  - Reliable

> Genericity and Flexibility

- Target Perf, Ftrace, LTTng and drivers
- Not only tracer-specific
  - Ring buffer sits in /lib
- Achieve genericity without hurting performance
  - Ring buffer clients
  - Instantiate client-specific configurations
  - Express configuration into a constant client structure passed as parameter to inline functions

> Configuration

- Buffers per-CPU or global
- Overwrite or discard mode
- Natural or packed alignment
- Output
  - splice(), mmap(), read(), iterator, client-specific
- Memory allocation backend
  - page, vmap, static
- OOPS consistency, IPI barrier, wakeup

> Common Trace Format (CTF)

- Answer the need of
  - Embedded
  - Telecom
  - High-performance
  - Linux Kernel community
- Collaboration with the Multi-core Association Tool Infrastructure Work Group (guest member)
  - Aim is to create a standard format for both software and hardware-level tracing

> Common Trace Format (CTF) implementation effort

- BabelTrace trace converter
  - in progress
- LTTng/LTTV migration to CTF
  - planned for early 2011

> TRACE_EVENT()

- A set of preprocessor macros
- Holding the event field descriptions and tracepoint declaration at the same code location

```
TRACE_EVENT(sched_switch,

        TP_PROTO(struct task_struct *prev,
                 struct task_struct *next),

        TP_ARGS(prev, next),

        TP_STRUCT__entry(
                __array(        char,   prev_comm,      TASK_COMM_LEN   )
                __field(        pid_t,  prev_pid                        )
                __field(        int,    prev_prio                       )
                __field(        long,   prev_state                      )
                __array(        char,   next_comm,      TASK_COMM_LEN   )
                __field(        pid_t,  next_pid                        )
                __field(        int,    next_prio                       )
        ),

        TP_fast_assign(
                memcpy(__entry->next_comm, next->comm, TASK_COMM_LEN);
                __entry->prev_pid       = prev->pid;
                __entry->prev_prio      = prev->prio;
                __entry->prev_state     = __trace_sched_switch_state(prev);
                memcpy(__entry->prev_comm, prev->comm, TASK_COMM_LEN);
                __entry->next_pid       = next->pid;
                __entry->next_prio      = next->prio;
        ),
```

> Road ahead for LTTng

- Shrink the LTTng kernel tree for easier distribution
- Move LTTng ABI closer to Perf
  - Use file descriptors rather than DebugFS VFS
- Focus mainlining on tracing clock sources
- Integration of CTF and Generic Ring Buffer in LTTng
  - Integration of TRACE_EVENT() from mainline Linux kernels with LTTng

> EfficiOS 2011 LTTng Planning

- Complete LTTng refactoring by end of March 2011
  - ABI changes, shrink kernel tree, Common Trace Format (CTF) integration, Generic Ring Buffer integration, TRACE_EVENT integration.
- UST : bring these improvements from LTTng to user-space
- BSD : LTTng proof of concept on BSD

> Thank you!

EfficiOS

- http://www.efficios.com
  - LTTng Information

    - http://lttng.org
    - ltt-dev@lists.casi.polymtl.ca