LinuxCon 2010

Efficient Trace Format for System-Wide Tracing

Presentation at:
http://www.efficios.com/linuxcon2010

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

> Plan

- Why we need a common trace format
- Linux kernel tracing today
- End user use-cases
- User requirements
- Trace format proposal outline
- Reference implementation

> Why we need a common trace format

- Interoperability between tracers and analysis tools
  - LTTng, Ftrace, Perf, LTTV, Eclipse Linux Tools LTTng viewer, Kernelshark, ...
- Analysis of heterogeneous systems

> Linux kernel tracing today

- Shared instrumentation
  - Static tracepoints (TRACE_EVENT())
  - Dynamic probes
  - Function tracer
  - Performance counters
- Perf
- Ftrace
- LTTng (external patch)

> State of Linux tracers

- Ftrace, Perf
  - Opening the Linux kernel developer community to tracing
  - Centered on kernel developers requirements
  - Still missing the point for companies developing on top of Linux (end users)
    - Telecommunication companies
    - Embedded systems
    - Enterprise servers
    - And many more !

> End user use-cases: telecom

- Monitoring of telecommunication systems
  - Enhance error reports with trace data
  - Configured and used by engineers and operators
  - Always-on trace data collection
  - Reboot time is critical
  - Limited trace extraction bandwidth, storage and memory
  - Traces gathered over a large collection of nodes, viewed on different hosts

> End user use-cases: RTOS

- Small footprint RTOS
  - Limited memory
  - Bounded tracer execution time
  - In some cases, heterogeneous system with both Linux and RTOS interacting

> End user use-cases: servers

- Performance analysis and debugging of enterprise servers
  - System-wide problem scope
  - Rare occurrence of problems
  - Very large traces generated
  - Delay between end of tracing and trace analysis availability directly affects users
  - Traces gathered over a large collection of nodes, viewed on different hosts

> User requirements: user classes

- Telecommunication
- Embedded
- Enterprise servers
- High-performance computing

> User requirements: users

Reflects the needs of the following users:

- Google
- IBM
- Ericsson
- Samsung
- Nokia
- Siemens
- Freescale
- MCA TIWG members
- Wind River
- Monta Vista
- Autodesk
- Cisco
- Mentor Graphics
- Texas Instruments
- Fujitsu

> User requirements (1)

- Compactness of traces
- Scalability to multi-core and multi-processor
- Low-overhead is key
- Production-grade tracer reliability
- Flight recorder mode
- Availability of trace buffers for crash diagnosis
- Support multiple trace sessions in parallel

> User requirements (2)

- Heterogeneous environment support
  - Portability
  - Distinct host/target environment support
  - Management of multiple target kernel versions
  - No dependency on kernel image to analyze traces (traces contain complete information)

> User requirements (3)

- Network streaming support
- Live view/analysis of trace streams
- System-wide (kernel and user-space) traces
- Scalability of analysis tools to very large data sets

> Trace Format Proposal Outline

- Architecture
- Linux-specific model

> Architecture

- High-level model aiming at industry-wide approval
- 3 constituents:
  - Event
  - Section
  - Metadata

> Event

- Physically ordered within a section
- Basic structure
  - Event type: numeric identifier
  - Event context
  - Event payload

> Event context (all optional)

- Ordering identifier
  - Sequence number or time-based
- Current time
- Execution context
  - IRQ, bottom half, thread context...
- Hardware performance counter information
- Thread, Virtual CPU, CPU, board, node ID
- Event payload size

> Event payload

- Variable event size
- Maximum event size configurable
- Payload size information available through metadata (and optionally in event context)
- Supports various data alignment, e.g.
  - Natural alignment
  - Packed alignment

> Section

- Similar to ELF sections
- Has a multi-level section identifier
- Contains a subset of event types
- Section context (all optional)
  - Apply to all events contained in that section
  - Thread, Virtual CPU, CPU, board, node ID
  - Execution context
    - IRQ, bottom half, thread context...

> Metadata

- Describes
  - Application environment setting
  - Basic types available, byte ordering
  - Event type to (section, event ID) mapping
  - Section context fields
  - Event context fields (per section and per event)
  - Per-event payload fields
- Scope: whole trace

> Metadata (basic types)

- Types available
  - Integer
  - Strings
  - Arrays
  - Sequence
  - Floats
  - Structures
  - Maps (a.k.a. Enumerations)
  - Bitfields
  - ...

> Metadata (3)

- Describes invariant properties of the environment generating the trace
- Architecture-agnostic (text-based)
- Trace version
- Trace capabilities
  - Event ordering, time flow, ...

> Linux-specific Model

- Event payload
  - Support ISO C naturally aligned and packed type layouts
- Require events to be ordered by time-stamps
  - Both ordering and time capabilities
- Payload size encoded within metadata
- Each section is represented as a trace stream
  - For the kernel, map each event group / CPU ID to a stream

> Linux-specific Model

- Store metadata in a section, along with the trace
  - Extract metadata from TRACE_EVENT() data
- Use target endianness
- Should allow 1 to 1 mapping between memory buffers and generated trace files
  - Zero-copy with splice()

> Reference implementation

- Conversion library
  - To standard format
  - From standard format
  - LGPL
- Providing format conversion as first integration step
- Will be usable as reference implementation to generate the format natively from the tracer
- Ongoing work

> Funding

- Thanks to Ericsson and the Embedded Linux Forum for funding parts of this work.
- Thanks to the Multi-Core Association Tool Infrastructure Work Group for their collaboration on the creation of this trace format.

> Questions ?

?

EfficiOS

  - http://www.efficios.com

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