Tracing the Google Platform

Montreal Tracing Summit
January 2008

Mathieu Desnoyers
École Polytechnique de Montréal

Summary

- Speaker presentation
- Why Google needs tracing
- Tracing requirements
- Ktrace vs LTTng
- Development plans

Speaker presentation

- Mathieu Desnoyers
  - Ph.D. student at École Polytechnique de Montréal
- LTTng maintainer since 2005
- IBM Research intern summer 2006
- Google intern winter 2007

Why Google needs tracing ?

- Real-life examples OLS2007

  – Occasional poor latencies for I/O write requests

  – Race condition in OOM killer

Tracing requirements

- Tracing on production systems
- Very low impact when disabled
- Low impact when enabled
- Disk dump and flight recorder modes
- Triggers to gather the data
- Use memory as efficiently as possible

Ktrace vs LTTng

- Ktrace is a home-made tracer made to fit Google's needs
  - Compact buffers
  - Low impact syscall tracing
- LTTng is a more generic and efficient alternative
  - Suitable for mainlining
  - Port Ktrace features to LTTng

Development plans

- Integrate LTTng to the Google platform
- Aim at LTTng mainlining
  - Markers (already in 2.6.24)
  - Tracer

LTTng mainlining plans

- Local cmpxchg
  - in mainline / -mm
- Markers support for multiple probes
  - in -mm
  - Steven Rostedt latency tracer (-rt tree)
- Text Edit Lock
- Immediate values

LTTng mainlining plans

- Instrumentation
- Kernel Trace thread flag
- Timestamping
- LTTng buffering
- LTTng format string parser
- Statedump
- Linux Kernel Markers /proc interface

Questions ?

Mathieu Desnoyers