From Network to Application: Understanding Your Distributed System with Trace Compass

Geneviève Bastien
Research Associate
Dorsal Laboraty
École Polytechnique de Montréal

Tracing Summit,
Dusseldorf, Germany,
October 13, 2014
Trace Compass

- Formerly known as TMF, the Linuxtools LTTng Eclipse plugin.
- **Trace visualization tool**
  - Standalone Rich Client Platform (RCP) application.
  - Also available as Eclipse plugins.
- **Extendable framework**
  - Add support for new trace types
  - Build trace analysis
  - With data-driven analysis, it's now easier than ever
Trace Compass

- Now goes beyond Linux-only
  - Trace types:
    - LTTng / CTF
    - BTF
    - Custom text and XML
    - GDB
    - PCAP
    - Windows! (prototype with CTF converter)
  - Analysis:
    - LTTng Kernel: Control Flow View, Resources View, CPU usage
    - LTTng UST: Memory Usage (liblttng-libc-wrapper), CallStack View (-g -finstrument-functions)
    - PCAP: **Network Stream lists**
    - META: **Data-driven analysis, Network trace synchronization, Virtual Machine analysis**, Critical path analysis
• 1 distributed application : 3 use cases
  – Local only (show data-driven analysis)
  – On 2 machines on the network (show network analysis)
  – On 2 virtual machines on the same host (show virtual machine analysis)
Demo Application

- MPI application: 5 worker threads + 1 server sending imbalanced workload to workers.
Data-Driven Analysis

- MPI application: 5 worker threads + 1 server sending imbalanced workload to workers.
State change:
Worker/\langle worker\_id \rangle = \text{WORKER\_AT\_WORK}
Data-Driven Analysis

- Visualization of the thread's states: time graph views or XY views

```
<timeGraphView id="mpi.imbalance.view.timegraph">
  <definedValue name="WORKER_AT_WORK" value="2" color="#66FF33" />
  <definedValue name="WORKER_WAIT" value="3" color="#FF3300" />
  <definedValue name="WORKER_IDLE" value="4" color="#CC66FF" />
  <entry path="Worker/*">
    <display type="self" />
  </entry>
</timeGraphView>
```
Future work

• Data-driven analysis:
  − Define visually, with state diagrams
  − Smart filters and user-defined actions on those filters
  − And much much more!

• GPU traces and analysis

• Compare traces from different executions, for CPU/Memory usage, etc.

• Live tracing

• Improve performances with large experiments
Questions

Resources

- Home Page: http://www.eclipse.org/tracecompass
- Mailing List: https://dev.eclipse.org/mailman/listinfo/tracecompass-dev
- Trace Compass standalone application used in this presentation: http://secretaire.dorsal.polymtl.ca/~gbastien/TracingRCP/DorsalExperimental/
- Sources:
  - TMF in Linuxtools: (under the Lttng folder)
    git://git.eclipse.org/gitroot/linuxtools/org.eclipse.linuxtools.git
  - Experimental: branch dorsal_experimental
    http://git.dorsal.polymtl.ca/~gbastien?p=linuxtools-tmf.git;a=summary
- Used in this demo:
  - Sample MPI traces and XML analysis: http://secretaire.dorsal.polymtl.ca/~gbastien/tracingSummit2014/
  - MPI-imbalance source code: branch cluster (folder cluster/mpi-imbalance)
    http://git.dorsal.polymtl.ca/~gbastien?p=workload-kit.git;a=summary
- IRC: #Lttng on oftc
- More doc and links: http://Lttng.org/eclipse
Annexes

(Screenshots in case Eclipse refuses to cooperate)
Experiment 1: Local: Control Flow View
Experiment 1: Local: Thread View
Experiment 2: Distributed Network: Control Flow View and Worker View
Convex-Hull Synchronization Algorithm
## Experiment 2: Distributed Network: Synchronized View

![Graphical representation of distributed network synchronization](image-url)

### Control Flow
- Processes: mpi-imbalance, systemd, sshd, orted, mpi-imbalance
- Threads: TID 2836, 2871, 2834, 2837, 2838, 2872, 502, 950, 956, 957, 958, 980
- Traces: distributed/localhost/kernel

### Thread View
- Threads: 1, 2, 3, 4, 5, distributed/132.207.169.248/kernel
- Time stamps: 14:51:32.000, 14:51:32.500, 14:51:33.000, 14:51:33.500, 14:51:34.000, 14:51:34.500

### Histogram
- Name: distributed/localhost, distributed/132.207.169.248
- ID: 2, 3, 4, 5

---

The image shows a synchronized view of a distributed network experiment, with control flow, thread view, and histogram data. The threads and processes are visualized with time stamps and event markers.
### Experiment 2: Distributed Network: PCap traces

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Source</th>
<th>Destination</th>
<th>File</th>
<th>Protocol</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:51:32.770 558 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>56547 &gt; 22 [ACK] Seq=4048612384 Ack=383585902 Len=32</td>
</tr>
<tr>
<td>14:51:32.771 089 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>22 &gt; 56547 [ACK, PSH] Seq=383585902 Ack=4048612407 Len=32</td>
</tr>
<tr>
<td>14:51:32.784 885 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>22 &gt; 56547 [ACK, PSH] Seq=383585902 Ack=4048612407 Len=32</td>
</tr>
<tr>
<td>14:51:32.784 980 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>56547 &gt; 22 [ACK] Seq=4048612407 Ack=383585925 Len=32</td>
</tr>
<tr>
<td>14:51:32.785 382 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>56547 &gt; 22 [ACK] Seq=4048612407 Ack=383585925 Len=32</td>
</tr>
<tr>
<td>14:51:32.785 386 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>56547 &gt; 22 [ACK, PSH] Seq=383585925 Ack=4048612407 Len=32</td>
</tr>
<tr>
<td>14:51:32.785 738 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>22 &gt; 56547 [ACK, PSH] Seq=383585925 Ack=4048612407 Len=32</td>
</tr>
<tr>
<td>14:51:32.786 766 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>56547 &gt; 22 [ACK] Seq=4048614375 Ack=383587573 Len=32</td>
</tr>
<tr>
<td>14:51:32.786 769 0CC</td>
<td>08:24:bd:80:00:40/132.207.169.248/22</td>
<td>db:24:bd:90:00:40/132.207.72.9/56547</td>
<td>tcpdump.out</td>
<td>TCP</td>
<td>22 &gt; 56547 [ACK] Seq=383587573 Ack=4048614375 Len=32</td>
</tr>
</tbody>
</table>

---|-----------------------------|-------------------------------|--------------------------
ID | Endpoint A | Endpoint B | Packets | Bytes | Packets A -> | Bytes A -> | Packets B -> | Bytes B -> | Start Time |
0  | 00:22:4d:86:a8:09/132.207.72.9 | d8:24:b9:80:00:40/132.207.169.248 | 361 | 42848 | 190 | 20839 | 171 | 22159 | 14:51:32.769 934 00 |
1  | 00:22:4d:86:a8:09/132.207.72.9 | d8:24:b9:80:00:40/132.207.169.248 | 2 | 132 | 1 | 66 | 1 | 66 | 14:51:33.808 411 00 |
2  | 00:22:4d:86:a8:09/132.207.72.9 | d8:24:b9:80:00:40/132.207.169.248 | 2 | 132 | 1 | 66 | 1 | 66 | 14:51:34.331 755 00 |
3  | d8:24:b9:80:00:40/132.207.169.248 | 00:22:4d:86:a8:09/132.207.72.9 | 6 | 540 | 3 | 228 | 3 | 312 | 14:51:34.483 783 00 |
Experiment 2: Distributed Network: PCap stream filter

FILTER stream (ip: d8:24:bd:90:00:40/132.207.169.248 <> d8:24:bd:90:00:40/132.207.169.248)

name: TCP between the 2 hosts

FILTER TCP between the 2 hosts

- AND
  - Internet Protocol Version 4 CONTAINS
- OR
  - AND
    - packetsource: CONTAINS "d8:24:bd:86:a8:09/132.207.72.9"
    - packetdestination: CONTAINS "d8:24:bd:90:00:40/132.207.169.248"
  - AND
  - WITH EVENTTYPE Common Trace Format : LTTng Kernel Trace
    - OR
      - type: CONTAINS "inet_sock_local_in"
      - type: CONTAINS "inet_sock_local_out"
Experiment 3: Virtual Machines: Control Flow View and Thread View
Experiment 3: Virtual Machines: qemu processes view
Experiment 3: Virtual Machines: VCPUs view
Experiment 3: Virtual Machines: 1 VCPU view
Experiment 3: Virtual Machines: VM Preempt View