Streaming and remote control architecture

Michael Sills-Lavoie
Department of Computer and Software Engineering
École Polytechnique de Montréal

August 9th, 2010
Tracing Mini-Summit, LinuxCon 2010

Content

1) **Remote control and streaming architecture**
2) **Needs for the tools**
3) **Overview of the functionalities and characteristics**
4) **Inside the architecture**
5) **Installation**
6) **Use case**
7) **Future work**

About me

- Michael Sills-Lavoie
- Graduate student at École Polytechnique de Montréal
- Several contributions to :
  - Streaming and remote control architecture
  - TCF binary transfer and plugin system
  - TCF LTTng agent/client plugin with streaming support
  - TMF remote control interface for LTTng tracepoints activation
- Work on dependency analysis

Remote control and streaming architecture

Needs for the tools

Overview of the functionalities and characteristics

Inside the architecture

Installation

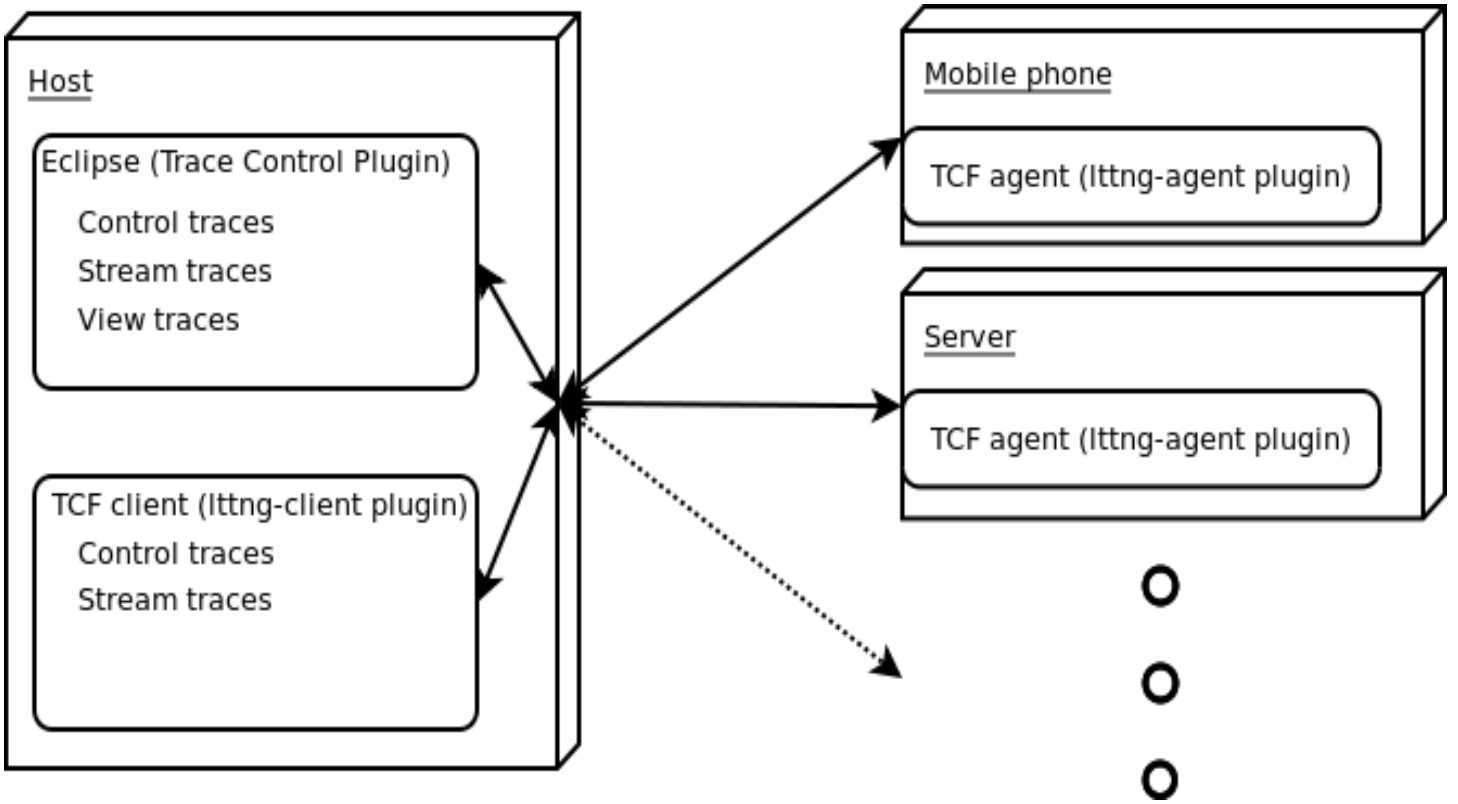
Use case

Future work

1. Remote control and streaming architecture (1 of 2)

- Tools developed to stream and control LTTng and UST traces remotely
- Consists of :
  - A **server** called lttng-agent (TCF agent plugin) installed on the traced systems
  - A **client** called lttng-client (TCF client plugin) installed on the system used to control the tracing
    - Graphical client integrated with Eclipse (LTTng only)
    - Command line client

1. Remote control and streaming architecture (2 of 2)



Remote control and streaming architecture

Needs for the tools

Overview of the functionalities and characteristics

Inside the architecture

Installation

Use case

Future work

2. Needs for the tools

- Developers often need to trace :
  - Small underpowered systems (ex. mobile phone)
    - No screen or small screen
    - Very small or no on-board storage
  - Servers in corporate environment
    - No screen or direct interface
    - Security constraints (ex. impossible to open many firewalls or router ports exclusively for tracing)
- Most developers like to have everything included in the same IDE (Eclipse)

Remote control and streaming architecture

Needs for the tools

Overview of the functionalities and characteristics

Inside the architecture

Installation

Use case

Future work

3. Overview of the functionalities and characteristics (1 of 2)

3.1 Characteristics

- TCF agent
  - Small program written in C
  - One daemon integrates many services (file system, process management, ...)
  - Only the required services are compiled
  - Supports plugins (ex. LTTng-agent)
  - Standard, simple and efficient communication protocol
  - Many services multiplexed into the same channel (port)
  - Easy to extend
  - Easy to interface with Eclipse or a C client

3. Overview of the functionalities and characteristics (2 of 2)

3.2 Functionnalities

- On par with the local LTTng and UST tools, it allows to:
  - Create/destroy traces
  - Control each channel individually
    - Enable/disable
    - Enable/disable flight recorder mode
    - Set the number of subbuffers
    - Set the size of the subbuffers
    - Set the periodical flush timer
  - Enable/disable markers
  - Stream LTTng and UST traces

Remote control and streaming architecture

Needs for the tools

Overview of the functionalities and characteristics

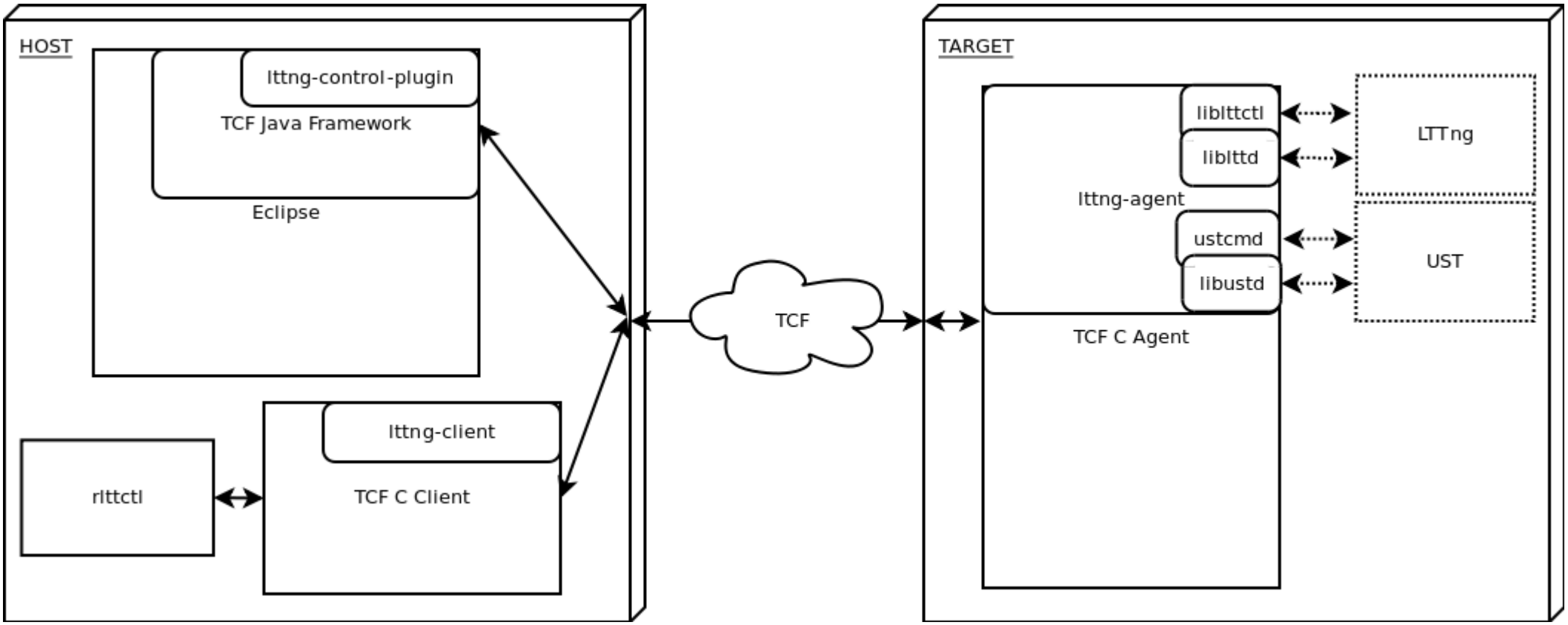
Inside the architecture

Installation

Use case

Future work

4. Inside the architecture



Remote control and streaming architecture

Needs for the tools

Overview of the functionalities and characteristics

Inside the architecture

Installation

Use case

Future work

5. Installation (1 of 3)

5.1 Server (on the traced system)

1) Install LTTng, UST or both (see *lttng.org*)

2) For LTTng, checkout and install lttcontrol (see *lttng.org*)

3) Checkout and install TCF from svn :
http://dev.eclipse.org/svnroot/dsdp/org.eclipse.tm.tcf/trunk/agent/

4) Checkout and install lttng-agent from git :

git://git.dorsal.polymtl.ca/git/lttng-agent.git

5. Installation (2 of 3)

5.2 Command line client (on the host)

1) Checkout and install TCF from svn :

http://dev.eclipse.org/svnroot/dsdp/org.eclipse.tm.tcf/trunk/agent/

2) Checkout and install lttng-client from git :

git://git.dorsal.polymtl.ca/git/lttng-agent.git

5. Installation (3 of 3)

5.3 Graphical client (on the host)

1) Install Eclipse (*http://eclipse.org/*)

2) Install Remote System Explorer (RSE) from the Eclipse package manager

3) Checkout and install the TCF Eclipse plugins :

   http://dev.eclipse.org/svnroot/dsdp/org.eclipse.tm.tcf/trunk/agent/

4) Checkout and install the lttng eclipse control plugin from git :

   git://git.dorsal.polymtl.ca/git/lttng-eclipse-control.git

Remote control and streaming architecture
Needs for the tools
Overview of the functionalities and characteristics
Inside the architecture
Installation
Use case
Future work

6. Use case (1 of 4)

6.1 Trace a remote system with command line client

1) Create the trace and start the transfer

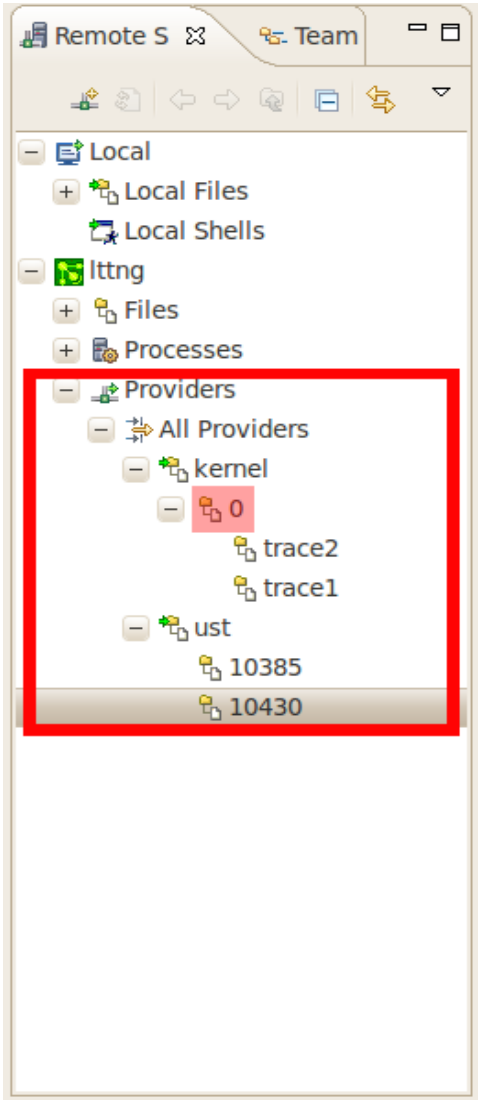
```
rlttctl -H test.domain.org -C -w /tmp/trace1 trace1
```

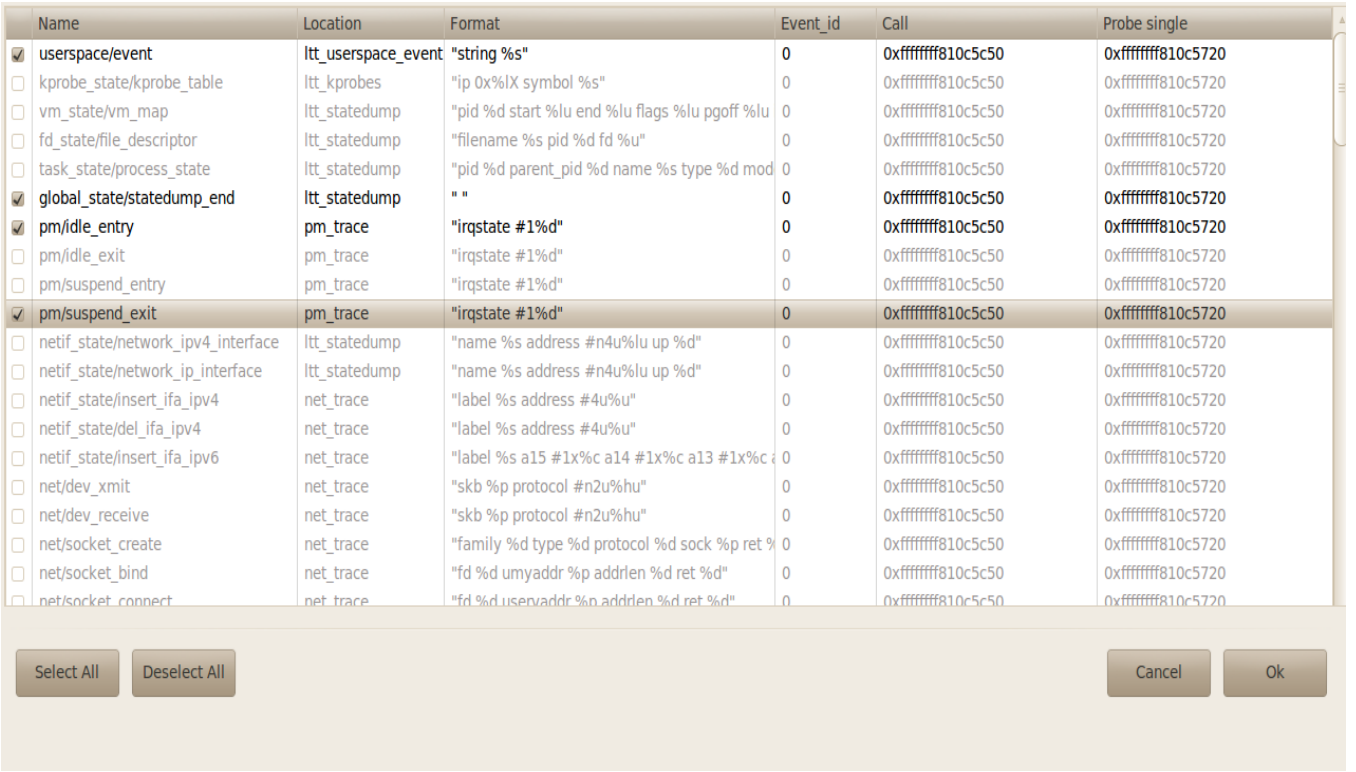
2) Destroy the trace

```
rlttctl -H test.domain.org -D trace1
```

6. Use case (2 of 4)

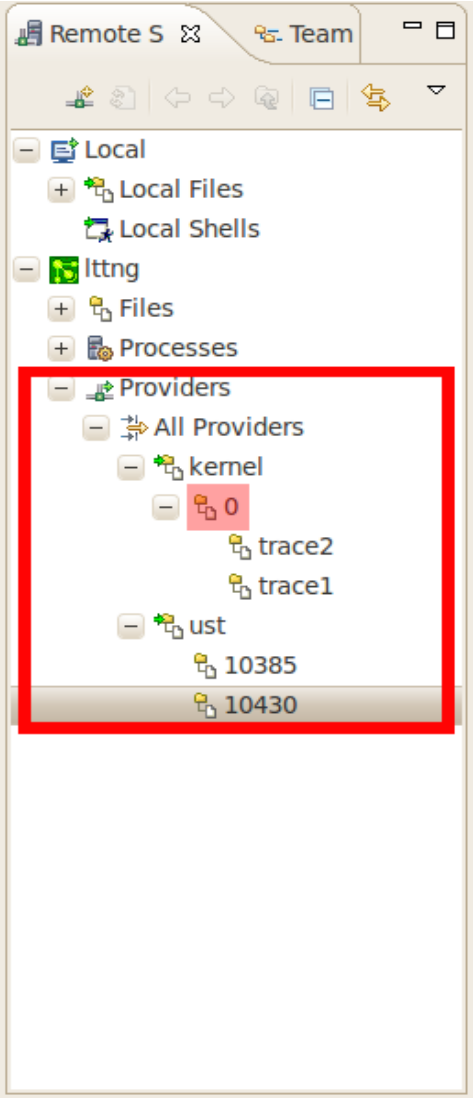
6.2 Trace a remote system with the graphical client

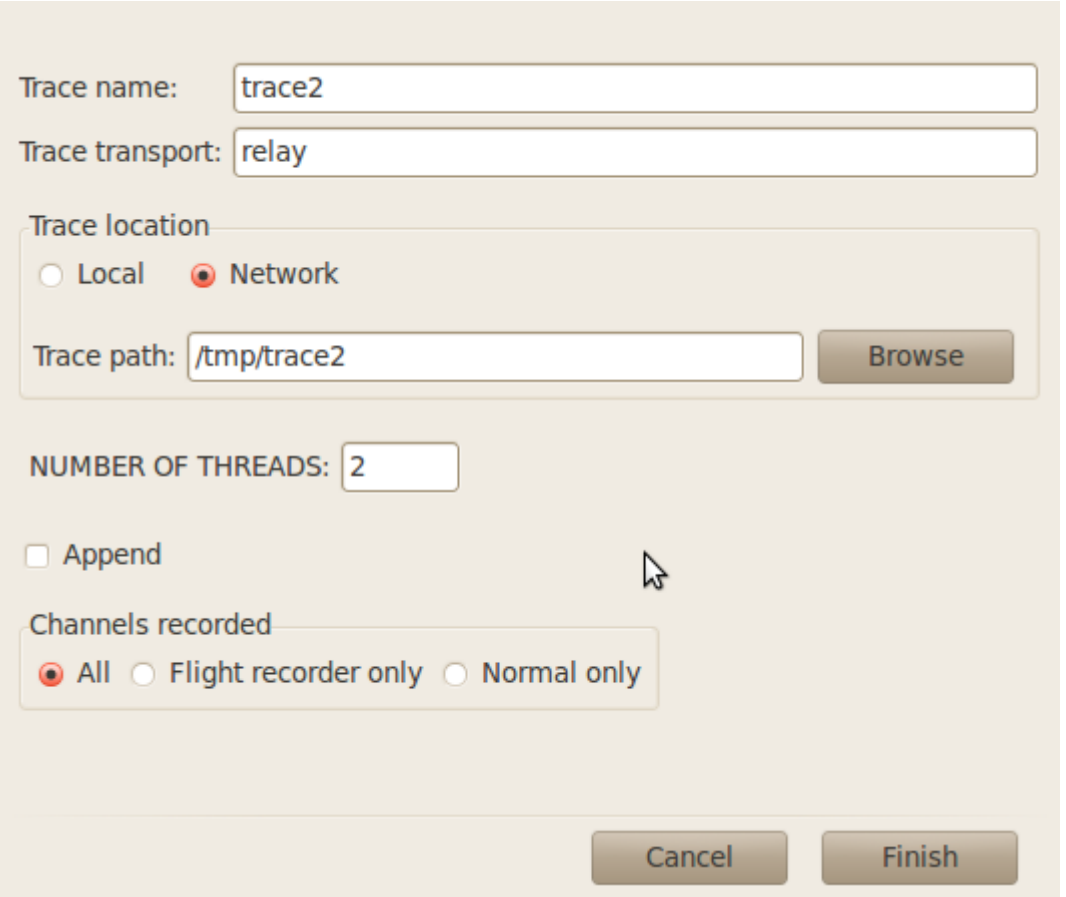




6. Use case (3 of 4)

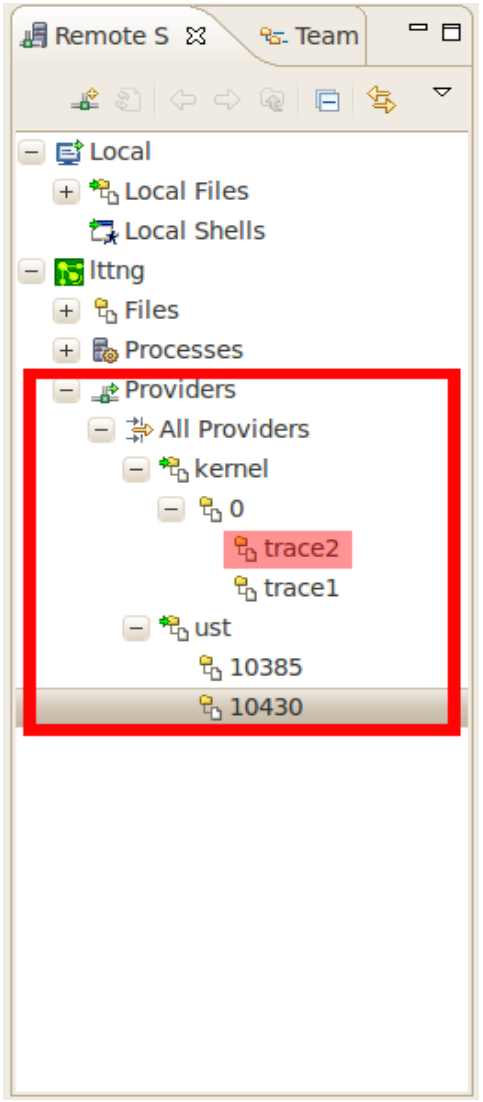
6.2 Trace a remote system with the graphical client

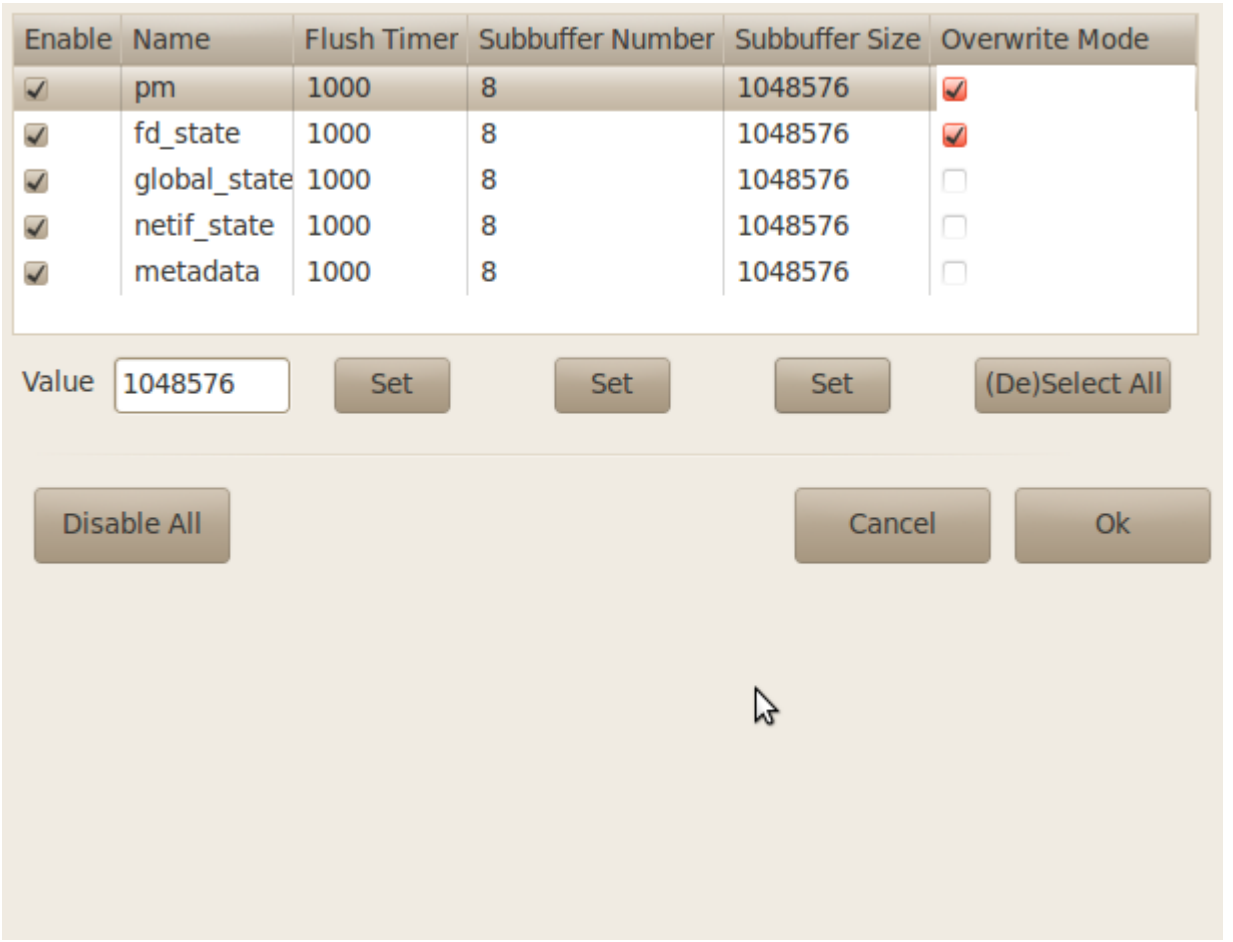




6. Use case (4 of 4)

6.2 Trace a remote system with the graphical client





Remote control and streaming architecture

Needs for the tools

Overview of the functionalities and characteristics

Inside the architecture

Installation

Use case

Future work

7. Future work

- Finish the integration of UST with the Eclipse plugin
- Provide an autotools based build system for TCF
- Integrate the Eclipse Trace Control plugin with TMF
- Update the viewing and analysis tools to take advantage of streaming traces

Questions ?