

dyntrace: Fast & Flexible Dynamic Tracing

Christian Harper-Cyr

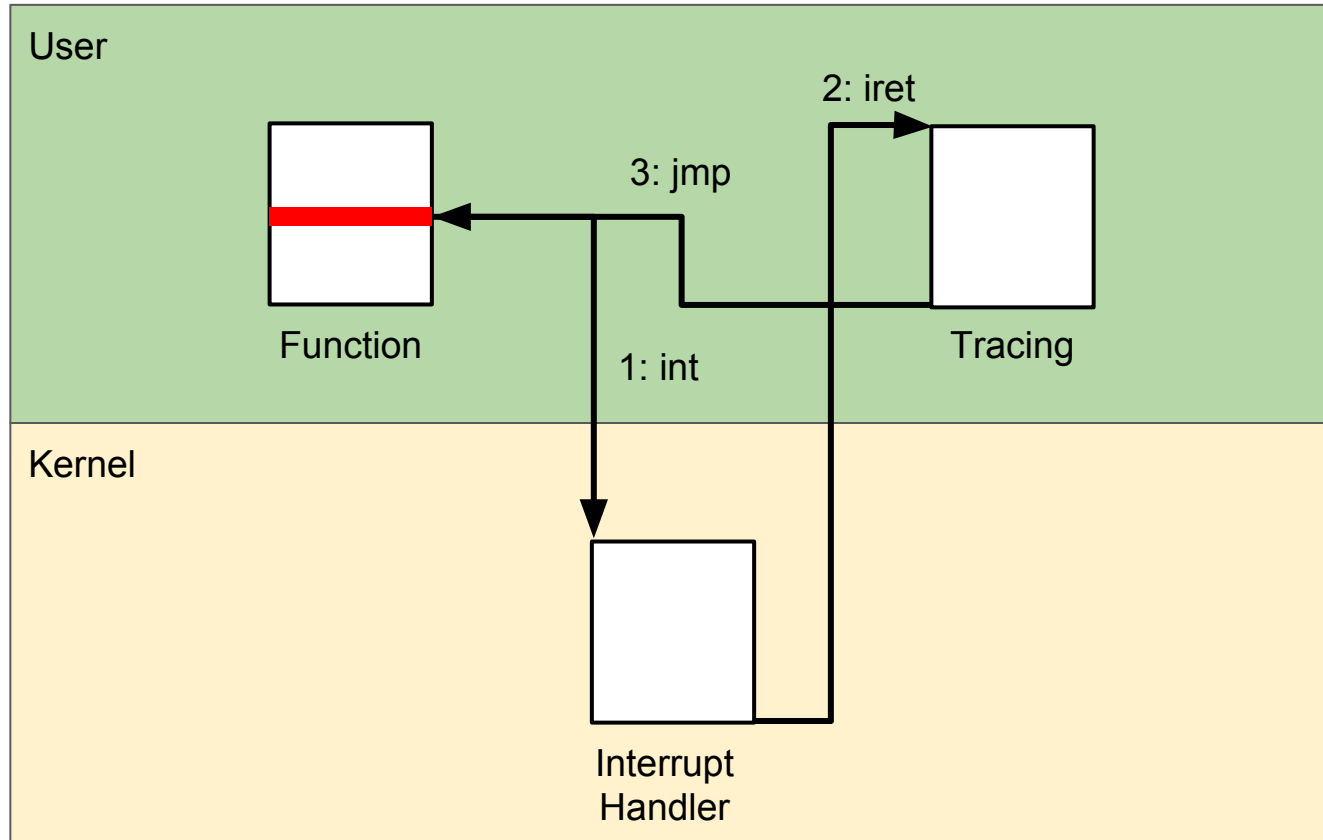
École Polytechnique de Montréal

DORSAL

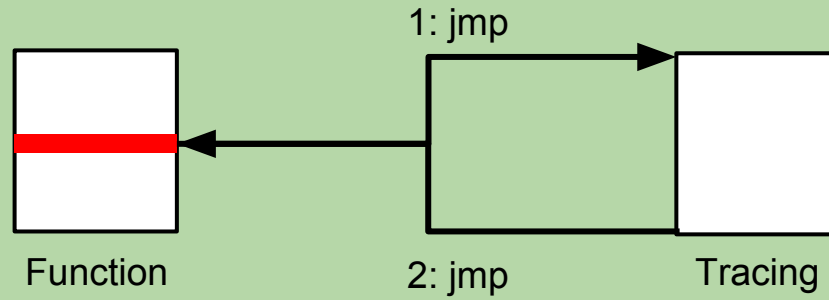
In this presentation

- Dynamic tracepoints
- dyntrace
 - dyntrace
 - x86 specifics
 - ARM specifics
- dyntrace performances
- Future Work

Dynamic Tracepoints



User



Kernel

Dynamic tracing tools

- Kernel
 - ftrace
 - kprobe
- User
 - uftrace
 - uprobe
 - GDB
 - dyntrace

dyntrace

dyntrace

- Prototype
- Full dynamic tracing ecosystem
- Architecture independent
- Easily extensible
 - Architecture
 - Tracers
 - Clients
- Attach to running processes
- Dynamic tracing library
- Injection library

Example

```
sudo dyntraced --daemonize
```

```
dyntrace attach nano
```

```
dyntrace add -r 'nano:do_(home|end)' -n my-tp -e lttng
```

```
lttng create
```

```
lttng enable-event -ua
```

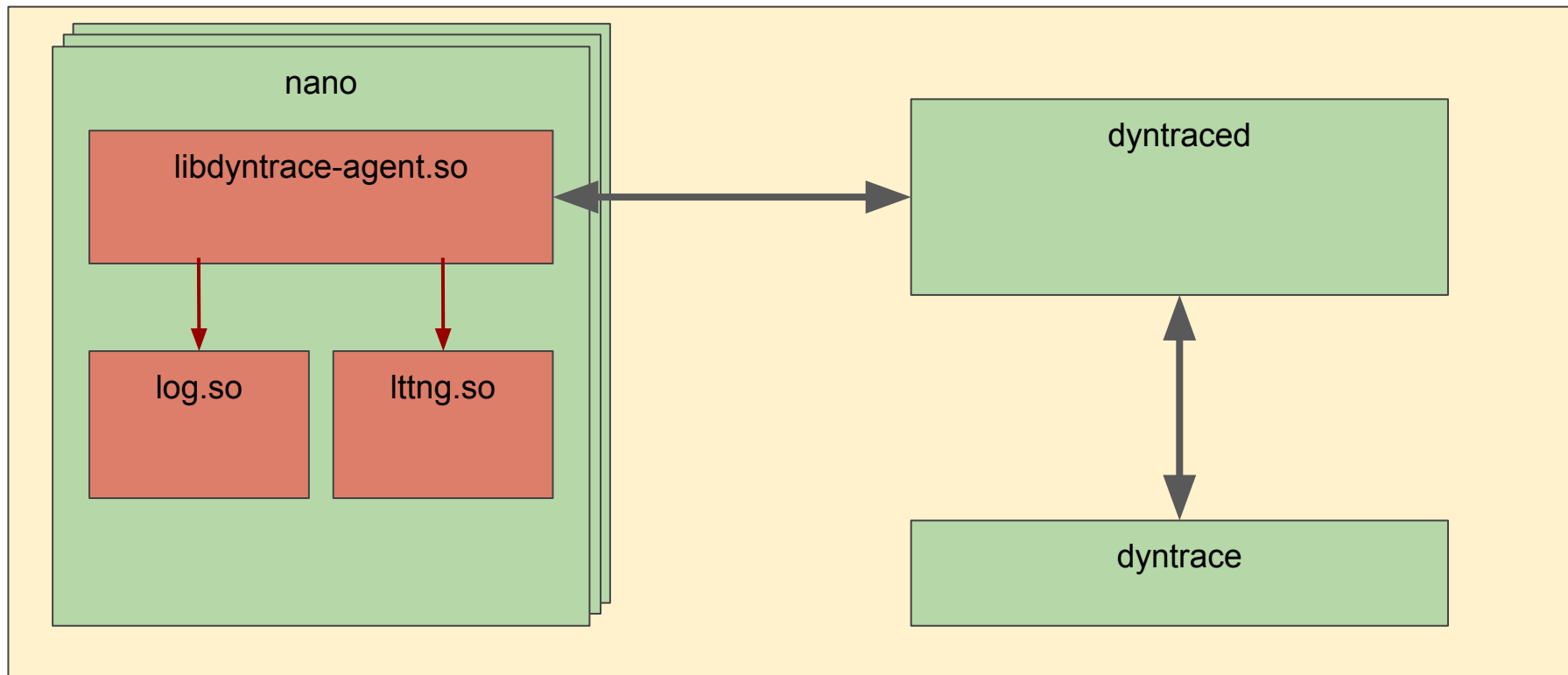
```
lttng start
```

```
...
```

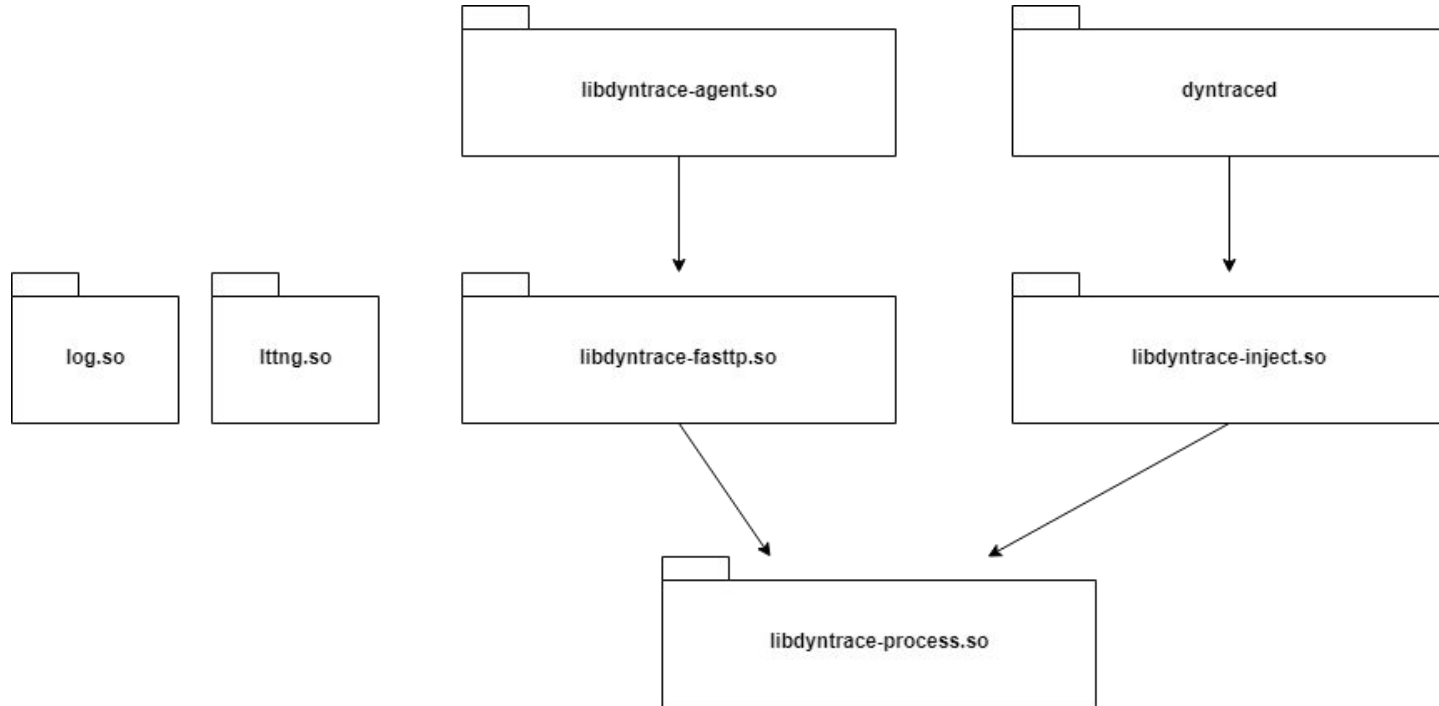
```
lttng stop
```

```
dyntrace rm nano:my-tp
```

dyntrace architecture



dyntrace packages



Thread-safety

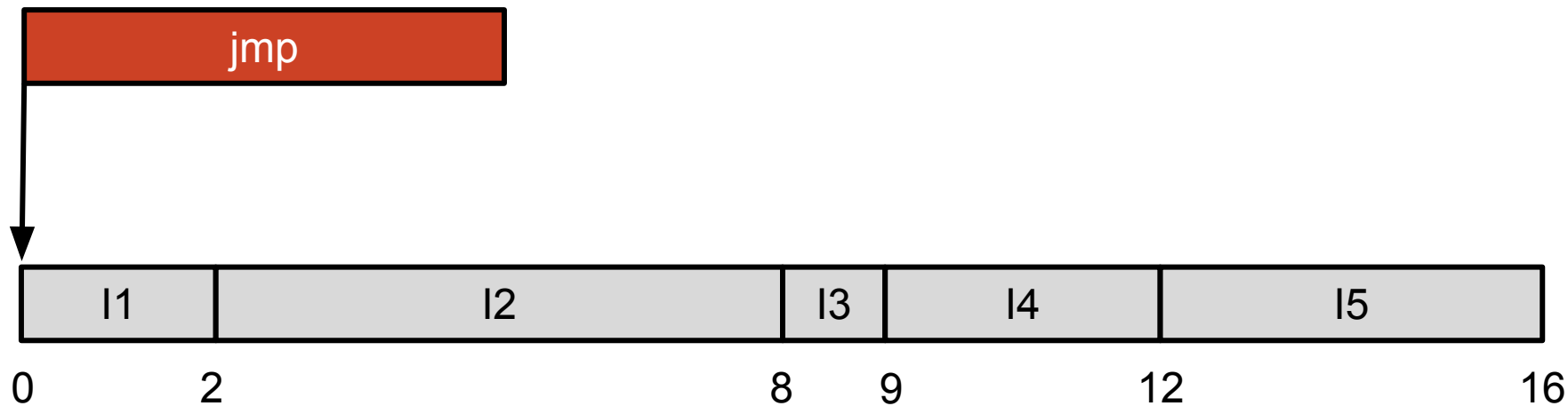
- Lock-free (atomic) tracepoint
 - Insertion
 - Execution
 - Activation
 - Deactivation
- Delayed tracepoint deallocation
 - Periodic reclaim period

x86 specifics

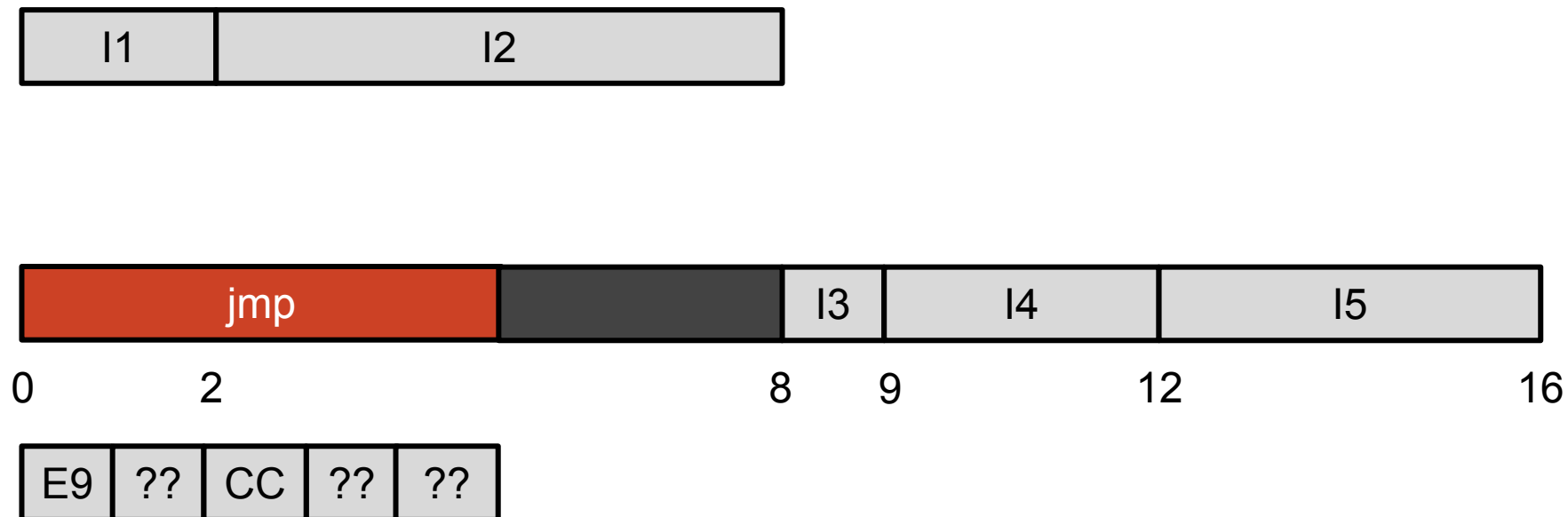
x86 specifics

- 32 and 64-bit support
- Needs a free page 2GB from tracepoint location
- Offset in jump has some bytes fixed (more on that later)

x86 tracepoint insertion



x86 tracepoint insertion



ARM specifics

ARM specifics

- 32 bit only
- Needs a free page 32MB from tracepoint location

dyntrace performances

Test environment

- Timed using the `std::chrono` library
- Timed over 50000 iterations per run
- Results averaged over 100 runs
- gcc 7.3.1
- libstdc++ 6.0.24

x86_64

Tool	Type	Time (ns)	Overhead (ns)	Overhead (%)
none		1070		
dyntrace	Point	1181	111	10
	Entry-Exit	1302	232	22
uprobe	Point	2560	1490	139
	Entry-Exit	3090	2020	188
Intel i7-6700k @4.2GHz 32 GB RAM ArchLinux 4.16.3-1-ARCH				

ARM

Tool	Type	Time (ns)	Overhead (ns)	Overhead (%)
none		10567		
dyntrace	Point	10621	50	0.5
	Entry-Exit	10783	220	2.0
uprobe	Point	13670	3100	29.4
	Entry-Exit	16200	5630	53
ARM Cortex-A53 @1.2GHz (Raspberry-Pi 3 Model B) 1 GB RAM ArchLinux 4.14.34-1-ARCH				

Future Work

- Integration into LTTng and ufttrace.
- Improve stability
- Revise atomics and thread-safety
- Reduce memory usage
- Optimizations
- ARM 64-bit
- PowerPC

Thanks for listening !
Questions ?

Code and documentation available at

<https://github.com/charpercyr/dyntrace>