Large-scale performance monitoring framework for cloud monitoring

Live Trace Reading

Julien Desfossez
Michel Dagenais

December 2013
École Polytechnique de Montreal

LTTng features for Cloud Providers

- LTTng 2.1 (12/2012): trace streaming
- LTTng 2.2 (06/2013): trace-file rotation
- LTTng 2.3 (09/2013): snapshots
- LTTng 2.4 (RC2 released yesterday): live trace reading

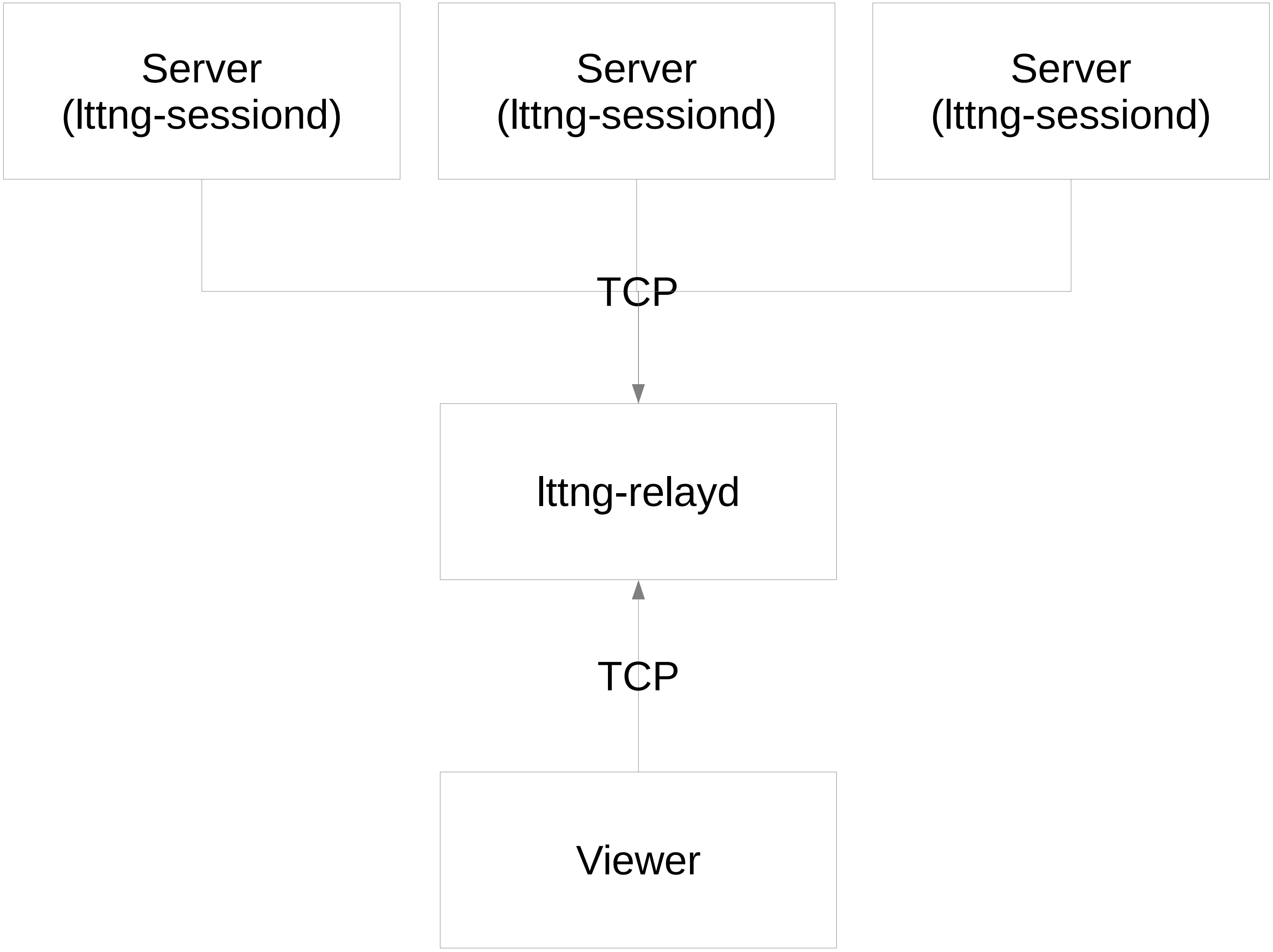
Flight recorder session + snapshot

```
$ lttng create --snapshot
$ lttng enable-event -k sched_switch
$ lttng enable-event -k –-syscall -a
$ lttng start
$ ...
$ lttng snapshot record
Snapshot recorded successfully for session auto-20131019-113803
$ babeltrace /home/julien/lttng-traces/auto-20131019-113803/snapshot-1-20131019-113813-0/kernel/
```

Live Trace Reading

- Read the trace while it is being recorded
- Local or remote session
- Configurable flush period

Infrastructure integration



Server (lttng-sessiond)

Server (lttng-sessiond)

Server (lttng-sessiond)

TCP

lttng-relayd

TCP

Viewer

Live streaming session

On the server to trace :

```
$ lttng create -–live 2000000 -U net://10.0.0.1
$ lttng enable-event -k sched_switch
$ lttng enable-event -k –-syscall -a
$ lttng start
```

On the receiving server (10.0.0.1) :

```
$ lttng-relayd -d
```

On the viewer machine :

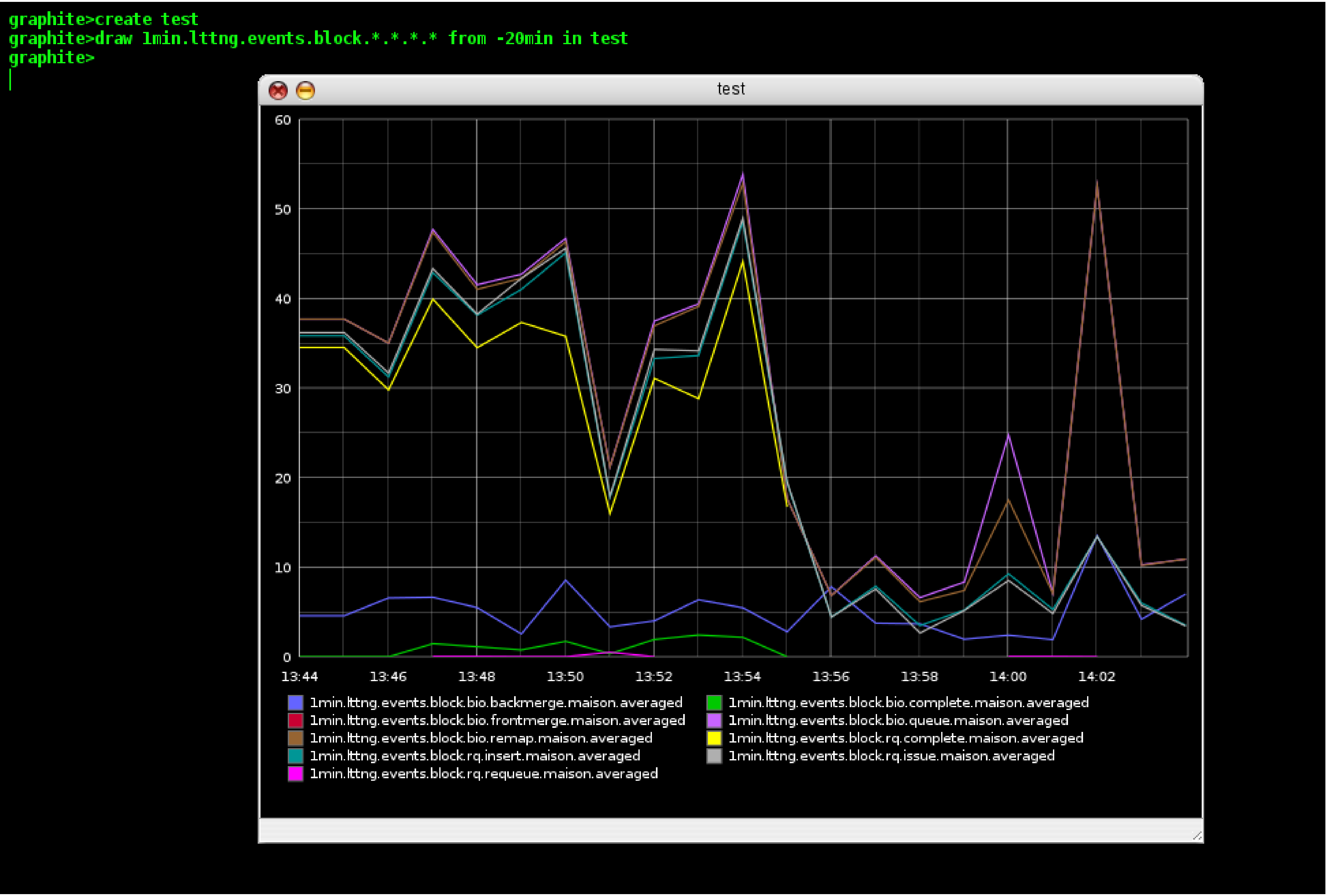
```
$ lttngtop -r 10.0.0.1
```

Or

```
$ babeltrace -i lttng-live net://10.0.0.1
```

LTTngTop

- Top-alike interface to read LTTng kernel traces
- CPU usage, per-process file activity, kprobes hit, per-process perf counter display
- Navigate in the trace second-by-second
- Read offline traces or connect to a relay for live-streaming
- Experimental in-memory live-reading



```
graphite>create test
graphite>draw 1min.lttng.events.block.*.*.*.* from -20min in test
graphite>
```

test
60
50
40
30
20
10
0
13:44
13:46
13:48
13:50
13:52
13:54
13:56
13:58
14:00
14:02
🟦 1min.lttng.events.block.bio.backmerge.maison.averaged
🟩 1min.lttng.events.block.bio.complete.maison.averaged
🟥 1min.lttng.events.block.bio.frontmerge.maison.averaged
🟪 1min.lttng.events.block.bio.queue.maison.averaged
🟫 1min.lttng.events.block.bio.remap.maison.averaged
🟨 1min.lttng.events.block.rq.complete.maison.averaged
🟦 1min.lttng.events.block.rq.insert.maison.averaged
⬜ 1min.lttng.events.block.rq.issue.maison.averaged
🟪 1min.lttng.events.block.rq.requeue.maison.averaged

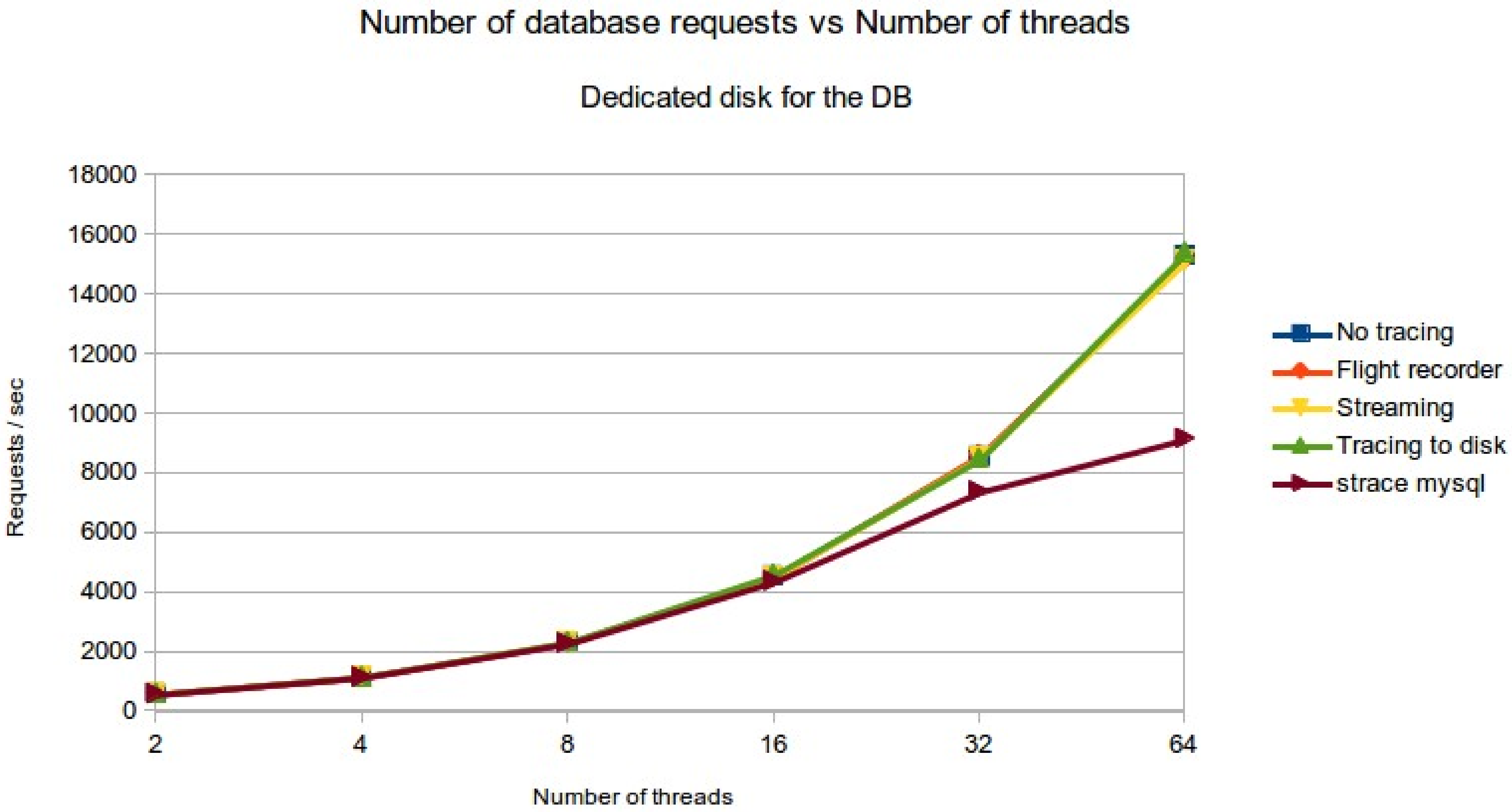
Performance results

- sysbench MySQL benchmark with increasing number of threads on a quad-core i7, 6GB RAM, 7200 RPM
- Tracing all system calls and sched_switch with LTTng in different modes :
  - Flight recorder with a snapshot recorded every 30 seconds
  - Streaming the trace to a remote server
  - Writing the trace on a dedicated disk
- Tracing all the threads of MySQL with strace to a dedicated disk

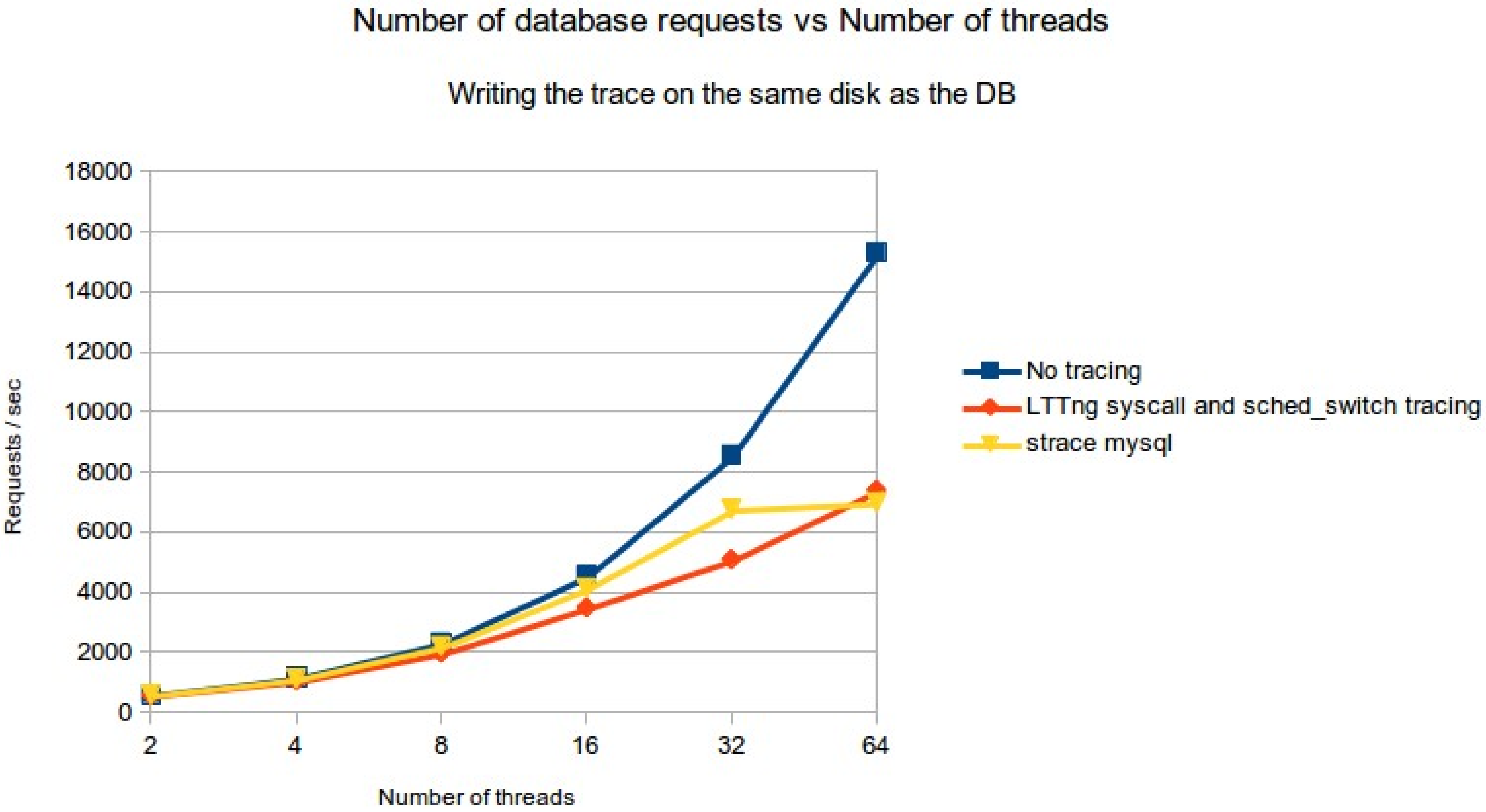
Performance results

- The test runs for 50 minutes
- Each snapshot is around 7MB, 100 snapshots recorded
- The whole strace trace (text) is 5.4GB with 61 million events recorded
- The whole LTTng trace (binary CTF) is 6.8GB with 257 million events recorded with 1% of lost events

Performance results



Sharing the disk with DB and trace



Performance result with virtualization

- 2 KVM VMs on the same host
- One is an apache web server
- The other one downloads a 5GB iso file from the first with wget
- Same LTTng instrumentation and setup (syscalls and sched_switch)
- No noticeable overhead when recording the trace on an external disk, network or snapshots.

Conclusion

- Snapshots and live trace reading create new use-cases for using tracing
- Production continuous monitoring is now possible with tracing
- Performance results are encouraging

Future Work

- Integrate with already existing monitoring tools (graphite, Nagios, etc), beta already working
- Integrate with Zipkin as part of the new effort to monitor OpenStack at Yahoo
- Filter and pre-process the trace before sending
- Distribute the analysis
- Remote control of the tracer
- More advanced triggers to collect snapshots, start/stop tracing, etc.

Install it

- Packages for your distro (`lttng-modules, lttng-ust, lttng-tools, userspace-rcu, babeltrace`)
- For Ubuntu : PPA for daily build (`lttngtop`)
- Or from the source, see `http://git.lttng.org`