

High-performance tracing of many-core systems with LTTng

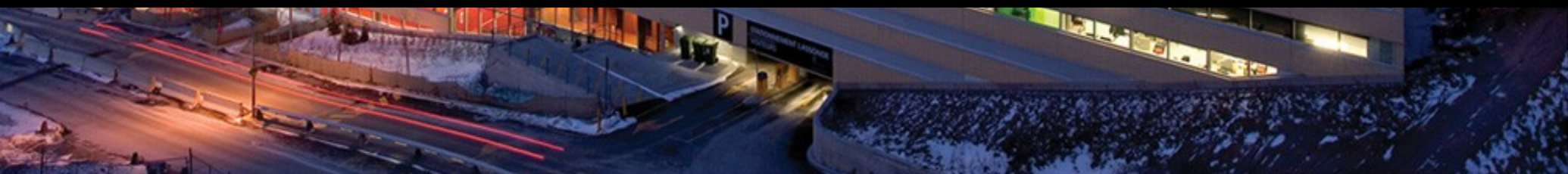
Simon Marchi

Laboratoire DORSAL
Département de génie informatique

POLYTECHNIQUE
MONTRÉAL



AFFILIÉE À
L'UNIVERSITÉ DE MONTRÉAL



Outline

- Work done
- Benchmarks
 - memcached
 - dragon
 - hello
 - packet processor
- Possible routes for improvement



Work, done and current

- Completed port of LTTng to Tileria
 - Added support for system call tracing
- Designed benchmarks to evaluate future improvements
- Performance analysis to find bottlenecks



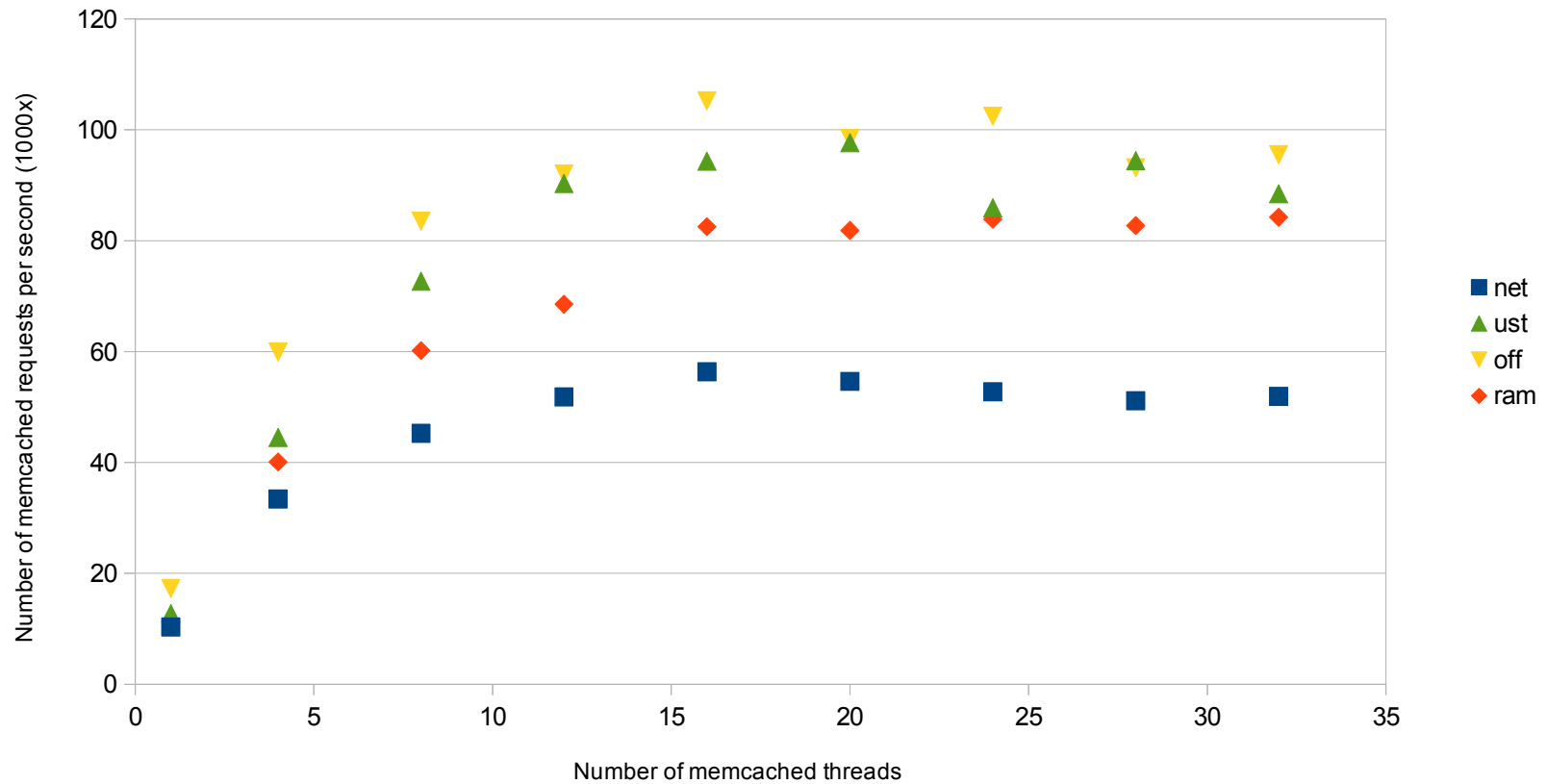
Benchmark: memcached

- Network application, a lot of interaction with the kernel: a lot of kernel events
- It is one of the featured applications on Tiler's website, so it must work pretty well...

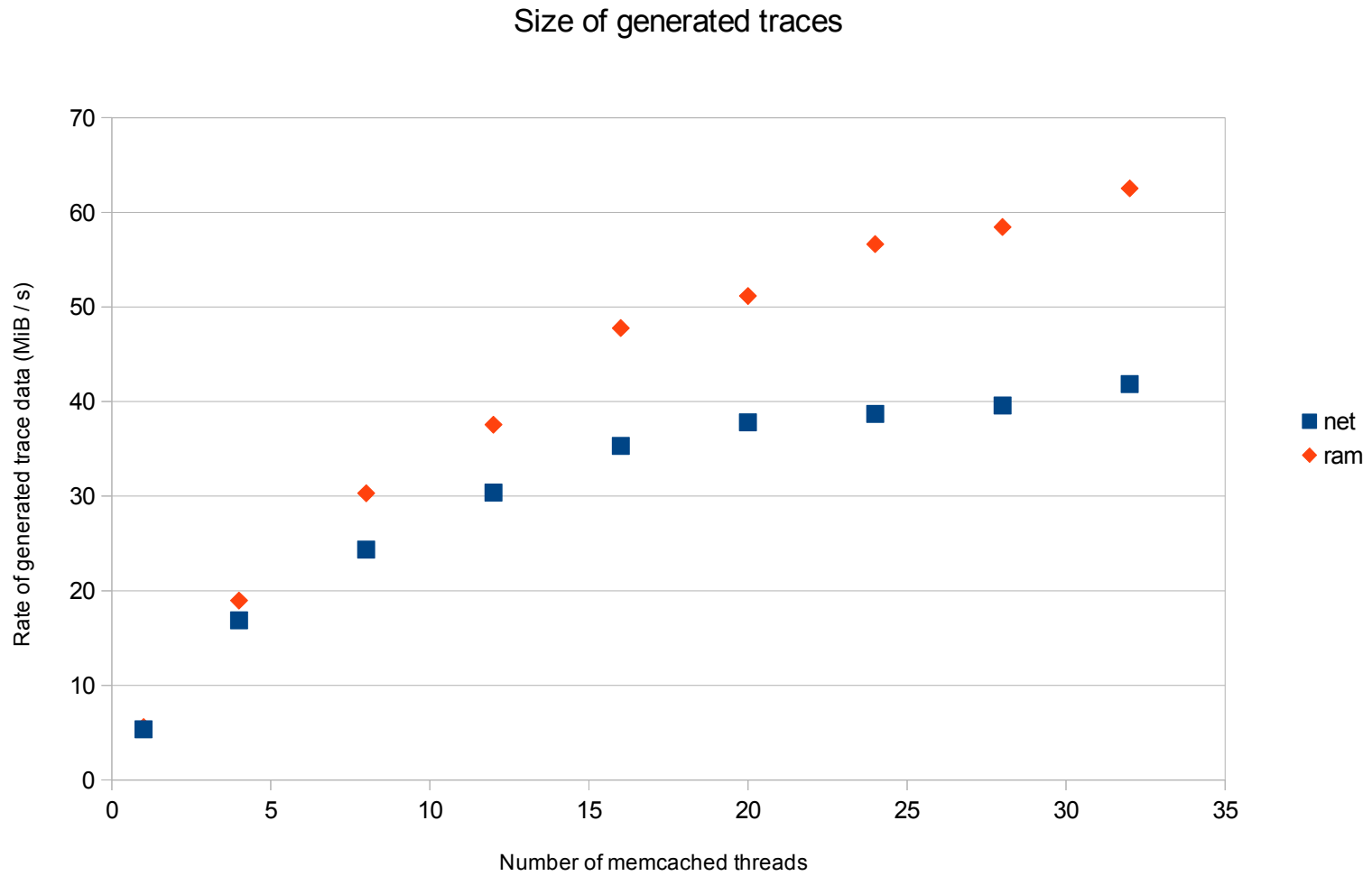


Benchmark: memcached + tracing

Number of requests served by memcached with different tracing modes



Big traces !



Network streaming hurts

- Saving to ram:

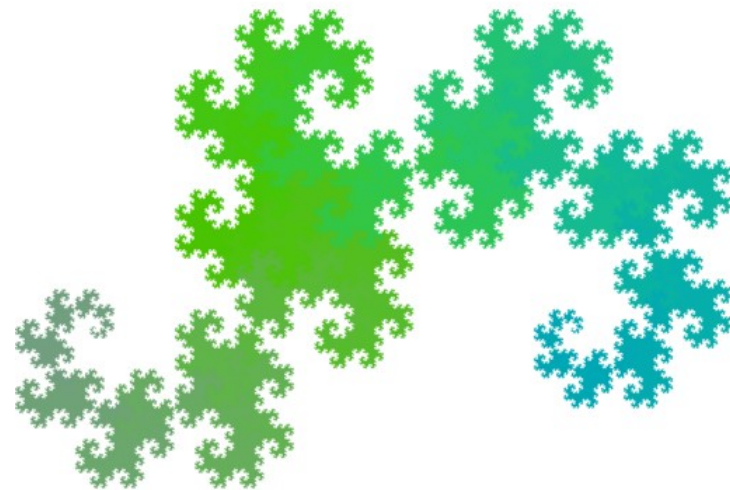


- Streaming on network:



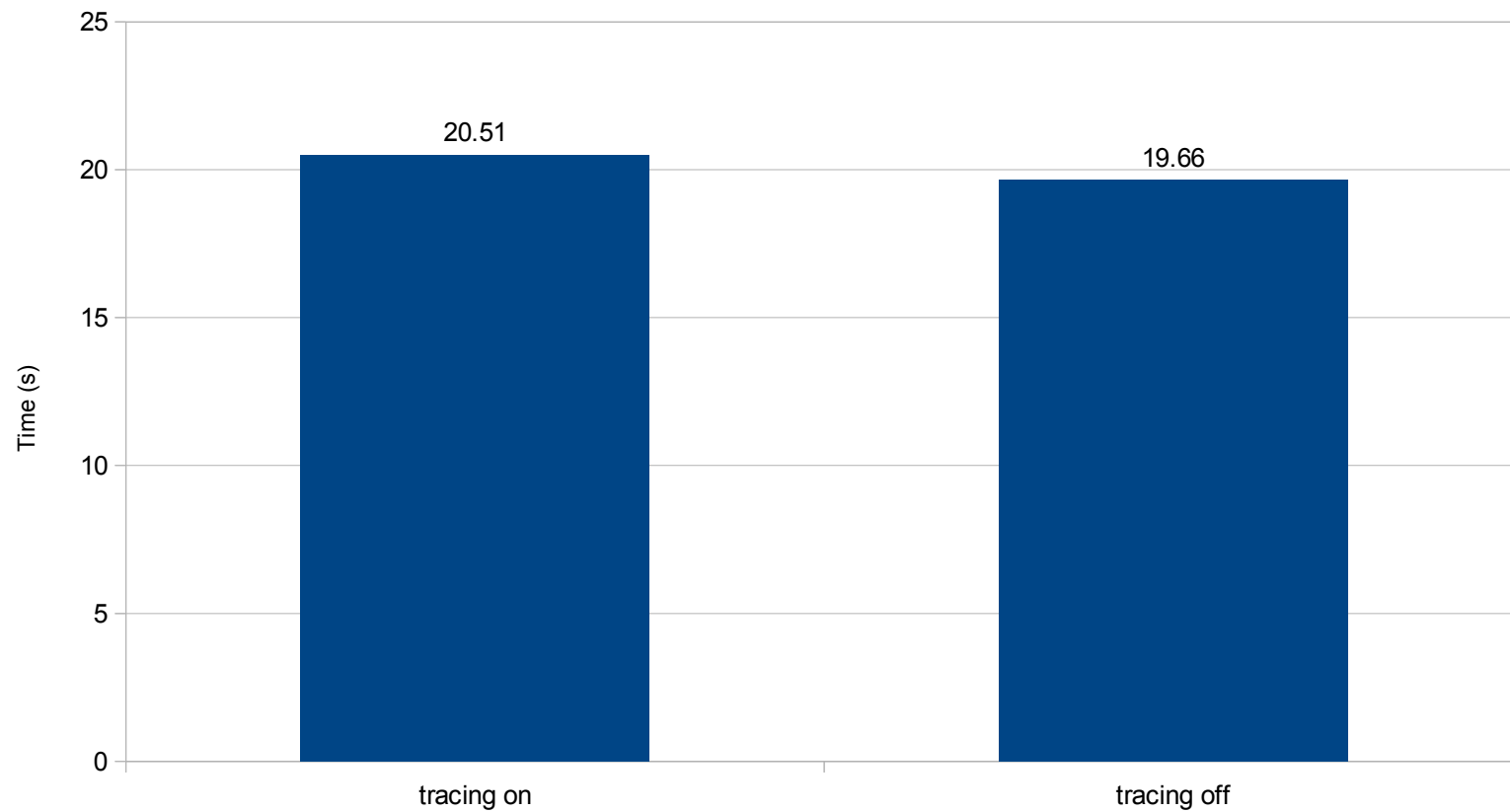
Dragon

- Highly parallel, academic application that draws a fractal.
- Essentially userspace-only application, very little interaction with the kernel.



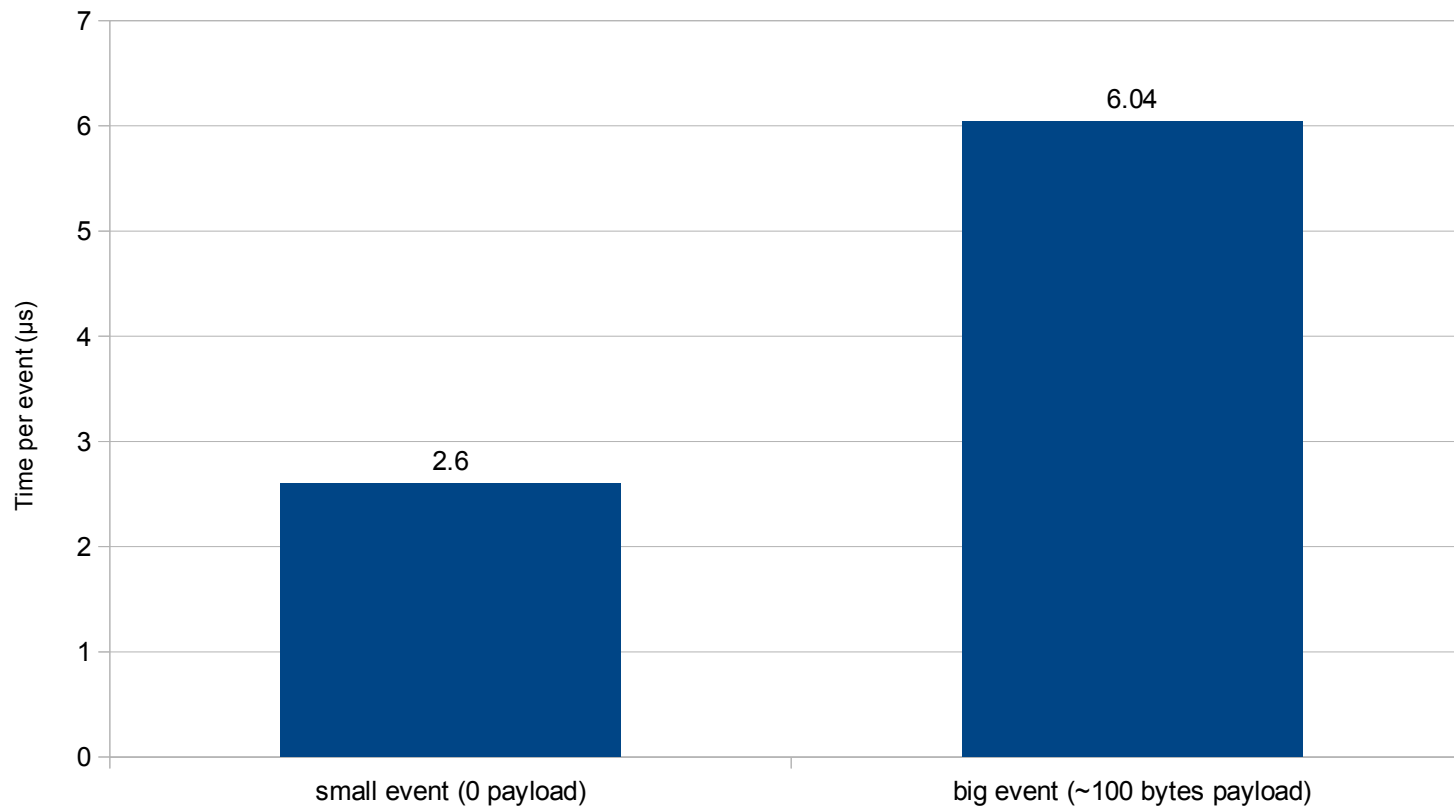
Dragon + kernel tracing

Execution time of "dragon" (10 executions) using 32 threads



Hello UST test case

Overhead per UST event (amortized on 10^7 events)



Hello UST test case

- Two system calls for each UST event
 - getcpu
 - clock_gettime

Process	pid	tid	time	kernel	syscall
dropbear	1733	1673	14:59:34.97364427	kernel	
sh	1734	1733	14:59:34.973649200	kernel	
ltnng	1902	1734	14:59:34.973826046	kernel	
ls	1978	1734	14:59:43.812517080	kernel	
hello	1979	1734	14:59:48.861891098	kernel	clo get cloc get cloc
hello	1980	1979	14:59:48.945762949	kernel	
hello	1981	1979	14:59:48.946519452	kernel	
hello	1982	1981	14:59:48.950836855	kernel	



Planned: Packet processor

- Use the hardware network packet classifier available on the processor
- Write a sample application that analyzes incoming network packets
- Should be a good way to fully utilize all 36 cores



Possible routes for improvement

- Adapt caching strategy for trace data
 - Evaluate impact of tracing on the app's cache hit rate.
- Use huge pages
 - Reduce TLB stress
- Analyze memory controllers balance



Questions ?

