GDB User Space/Kernel Tracepoints

Rafik Fahem
Michel Dagenais
Department of Computer and Software Engineering

December 9, 2010
École Polytechnique, Montreal

Content

- GDB tracepoints
- KGTP kernel module
- New developments
- Results
- Future developments
- Conclusion

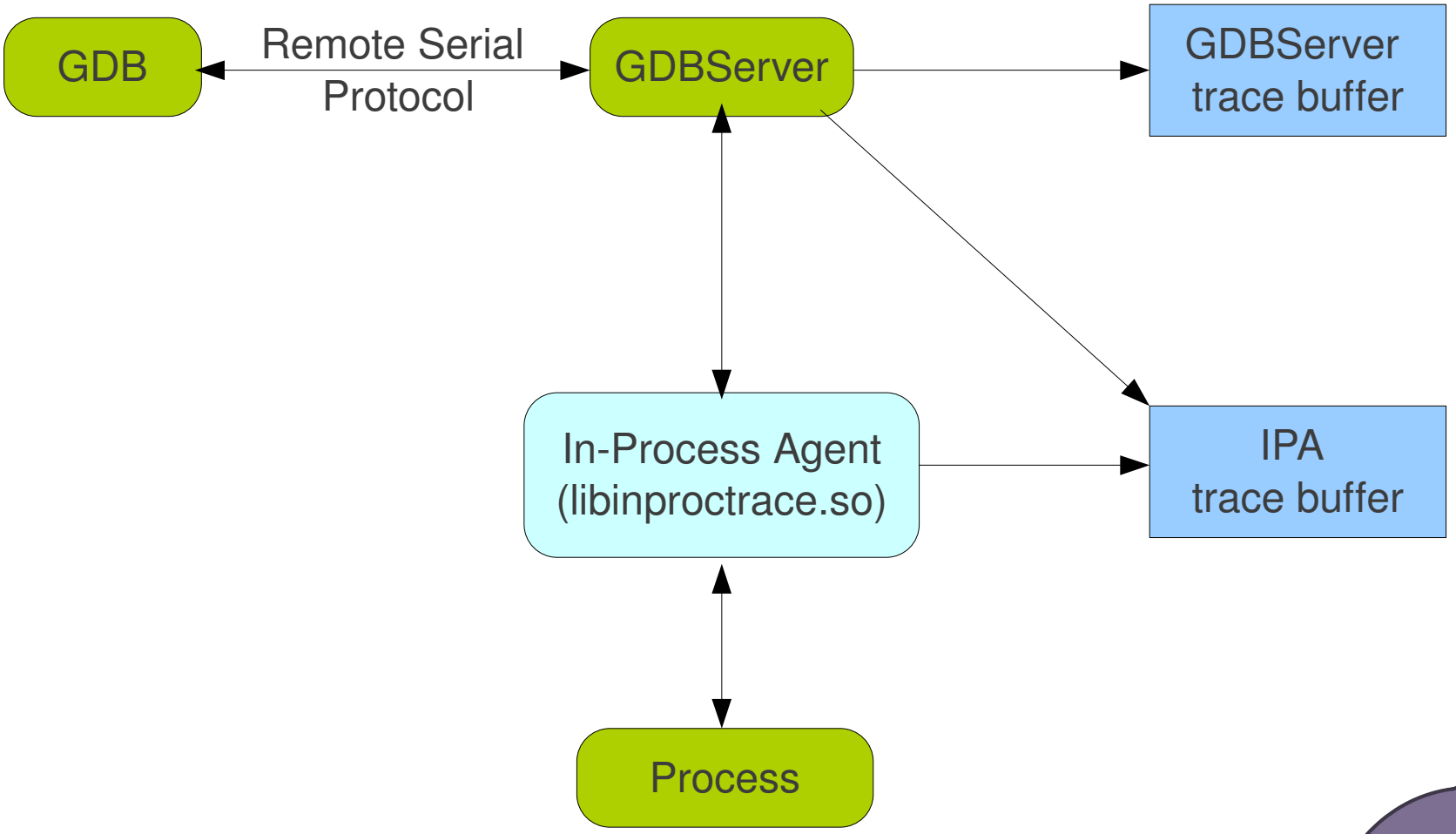
GDB tracepoints

- Debug a program without interrupting it
- Locations in the program to which data collecting probes are associated
- Collected data: $regs, $args, user defined expressions...
- Tracepoints may be conditional
- 3 types:
  - Slow tracepoints
  - Fast tracepoints
  - Static tracepoints

GDB fast tracepoints

- Implemented using a jump instead of a trap
- Conditions and actions specified using expressions
- Expressions translated into bytecode
- Conditions bytecode is converted to native code to increase performance
- Actions bytecode is interpreted when the tracepoint is hit

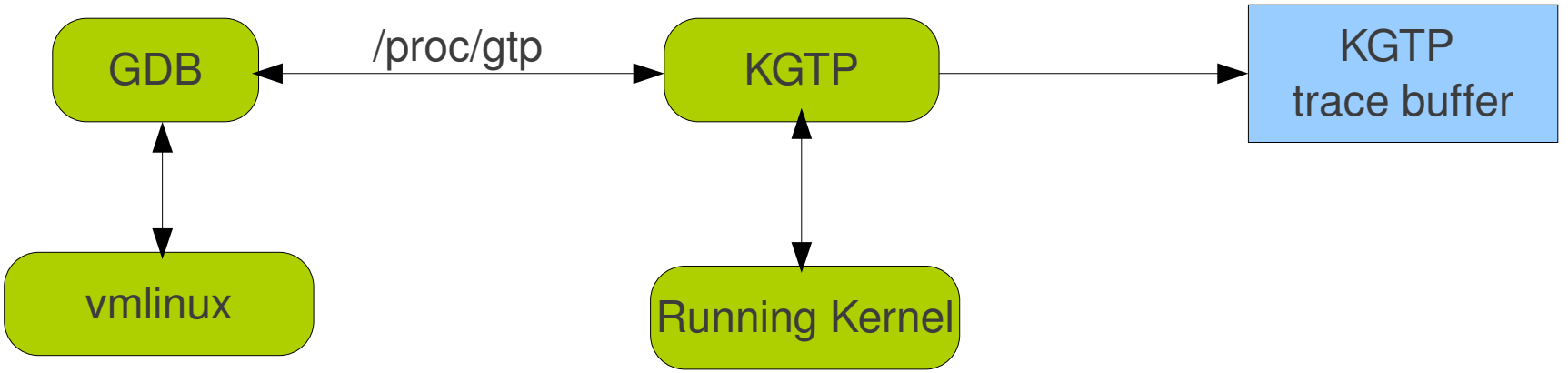
GDB fast tracepoints architecture



KGTP module

- Equivalent of GDB tracepoints in kernel space
- Implements the GDB Remote Serial Protocol
- GDB and module communicate via /proc/gtp
- GDB kernel tracepoints implemented using Kprobes
- Conditions and actions are executed using bytecode
- Native code not supported

KGTP architecture



Tracepoint compiled actions

- Forty opcodes in the bytecode language
- Some of them are not supported in the translator, including tracing instructions
- Added tracing instructions support
- Implementation of native code functions is the same as those used with conditions
- Executing native code should improve performance

KGTP Native Code Support

- The bytecode to native code translation mechanism was ported to KGTP module.
- Conditions and actions converted to native code
- Feature still under development

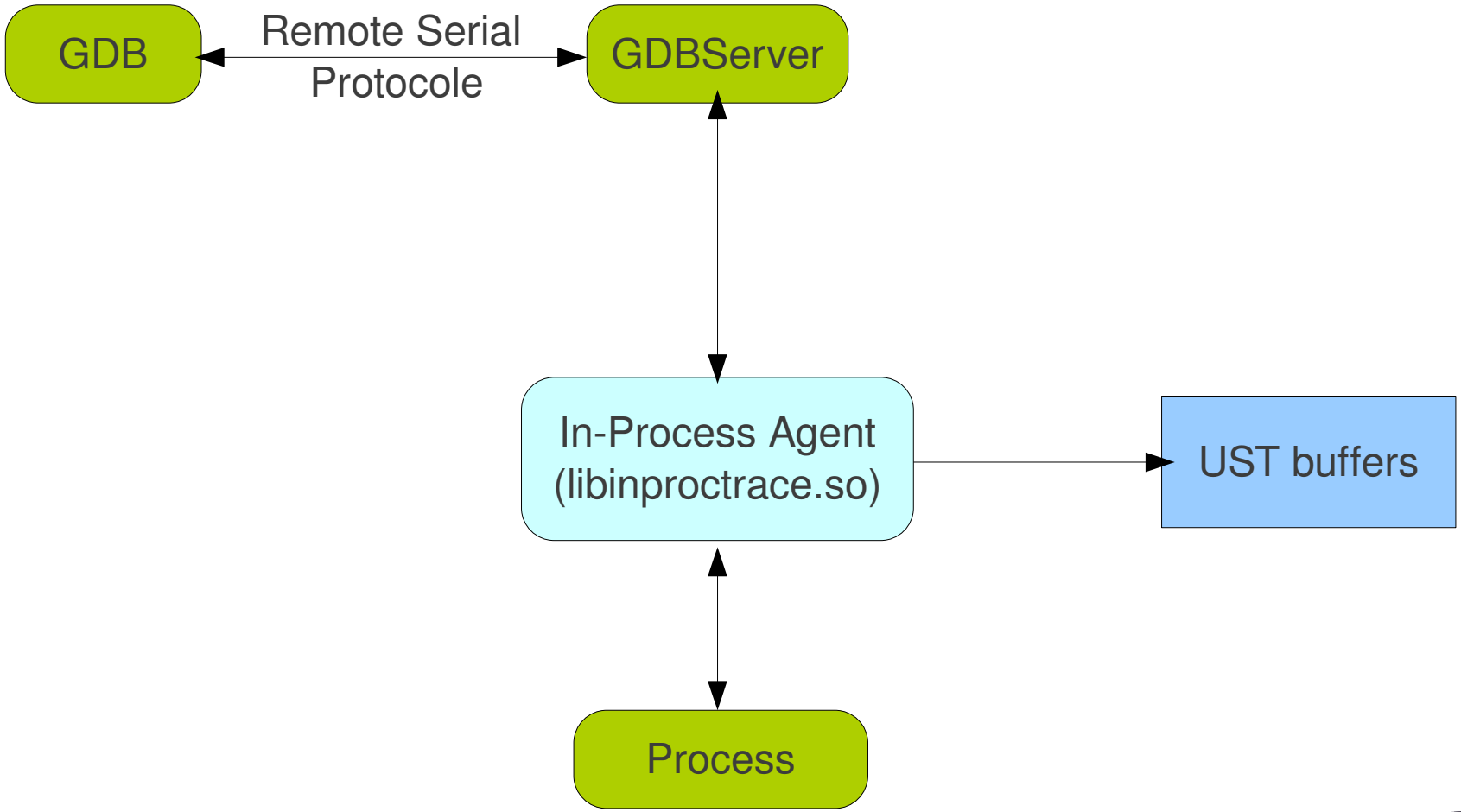
Results: conditions

Iterations		50,000,000	Time Per Iteration
Without Instrumentation		0.227 s	5 ns
Slow Tracepoints/Simple condition		> 10 minutes!	-
Simple Condition	Bytecode	9.21 s	184 ns
	Native Code	4.56 s	91 ns
Complex Condition	Bytecode	22.22 s	444 ns
	Native Code	5.74 s	115 ns

Results: actions

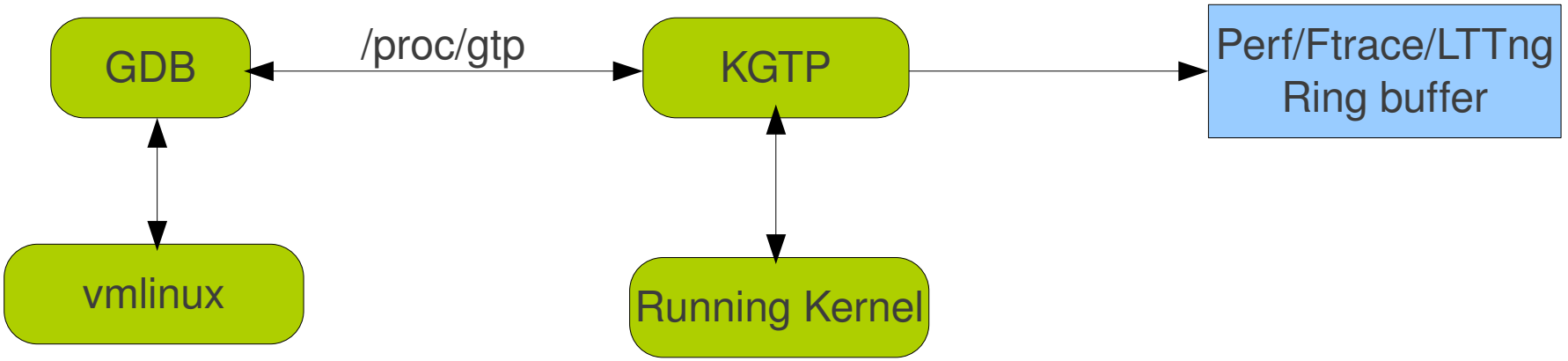
Iterations		400,000	Time Per Iteration
Slow tracepoints/Simple condition		21.75 s	54.375 µs
Simple Action	Bytecode	0.87 s	2.175 µs
	Native Code	0.85 s	2.125 µs
Complex Action	Bytecode	1.23 s	3.075 µs
	Native Code	1.16 s	2.9 µs

Future Development



New Fast Tracepoints architecture

Future Development



New KGTP architecture

Conclusion

- GDB tracepoints provide a quick solution to trace in user space and kernel mode
- Some improvements are needed to take advantage of native code performance
- UST and LTTng efficient buffers are a possible alternative