Update on the BPF support in the GNU Toolchain

Jose E. Marchesi

Oracle Inc.

GNU Tools Track @ LPC 2020

ORACLE

Contents

ORACLE

The Project

BPF: in-kernel virtual machine.

- **Phase 1:** add BPF target to the toolchain
- **Phase 2:** make the generated programs palatable for the kernel loaders and verifier, and **keep it that way**.
- **Phase 3:** provide other development goodies for BPF developers (simulator, debugger, tracer, etc.)

The Port

`bpf-unknown-none`

- **binutils port**
  - Upstream since Aug 2019.
  - Debian: `binutils-bpf`.
  - Oracle Linux 8: `cross-binutils`.
- **GCC backend**
  - Upstream since Sep 2019.
  - Debian: `gcc-bpf`.
  - Oracle Linux 8: `cross-gcc`.
- **GDB port**
  - Upstream since Aug 2020.
- **Simulator**
  - Upstream since Aug 2020.
- **Dejagnu board**
  - bpf-sim

ORACLE

xBPF

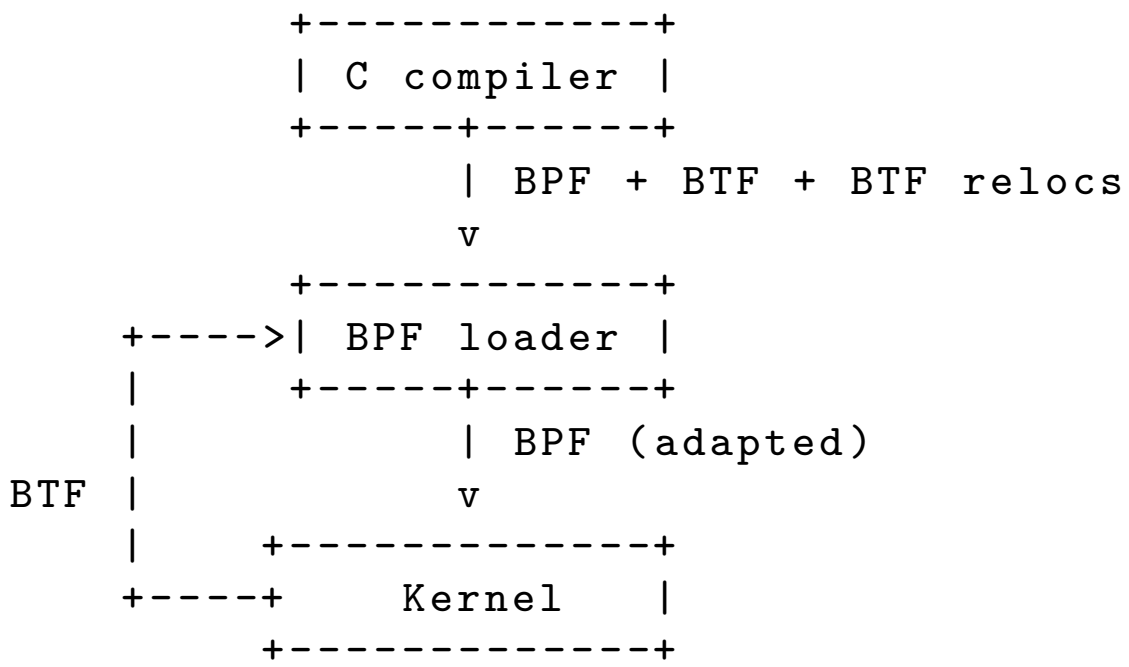
- Experimental BPF.
- Purpose: compiler testing, BPF debugging, userland.
- `-mxbpf` in GCC and GAS.
- Current extensions:
  - Save/restored callee-saved used registers.
  - Indirect calls: `callr %reg`
- Coming extensions:
  - Signed division instruction.
  - Zero register?
  - Indirect jumps.
  - `%fp` relative addressing.
  - Remove limit on stack frame size.

BTF

- Debugging format used by eBPF.
- Similar to CTF.
- `bpf-unknown-none-gcc -g` should generate BTF, not DWARF.
- Instrumental for CO-RE.

BPF Compile-Once, Run-Everywhere

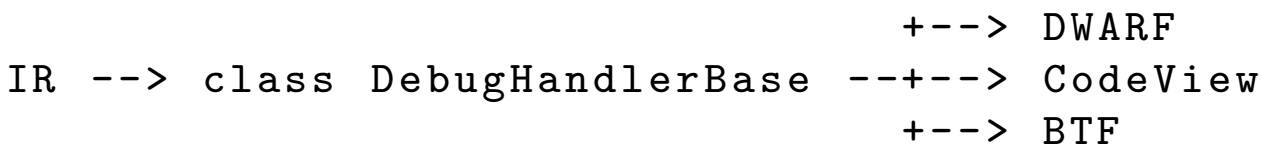
- BTF and field offset relocs (`__builtin_preserve_access_index`)
- Kernel headers: `/sys/kernel/btf/vmlinux` → `vmlinux.h`



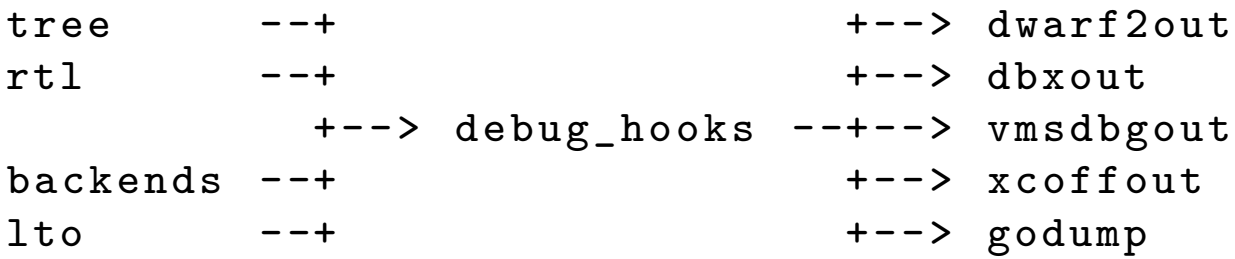
ORACLE

Diversity of debugging formats in GCC

- LLVM



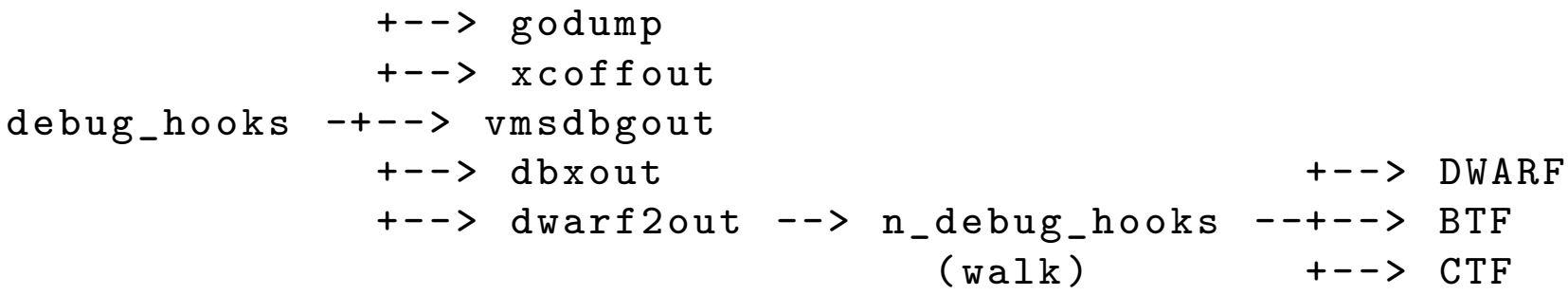
- GCC



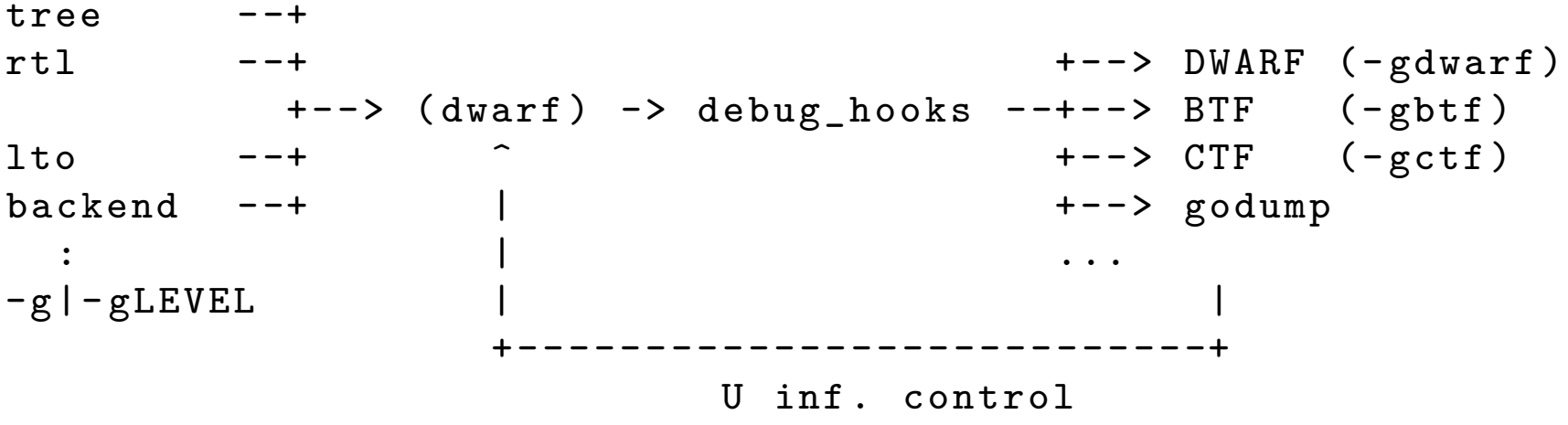
ORACLE

New debug hooks in GCC?

- Step 1:



- Step 2:



- Step 3: everyone is happy :-)

ORACLE

Idea: xBPF based Infinity

Note compiler	`i8c`	bpf-unknown-none-gcc
Note testing framework	`i8i`	bpf-unknown-none-run
Client library	`libi8x`	libxbpf (TBW)

`http://infinitynotes.org`