Multicore Debugging with GDB

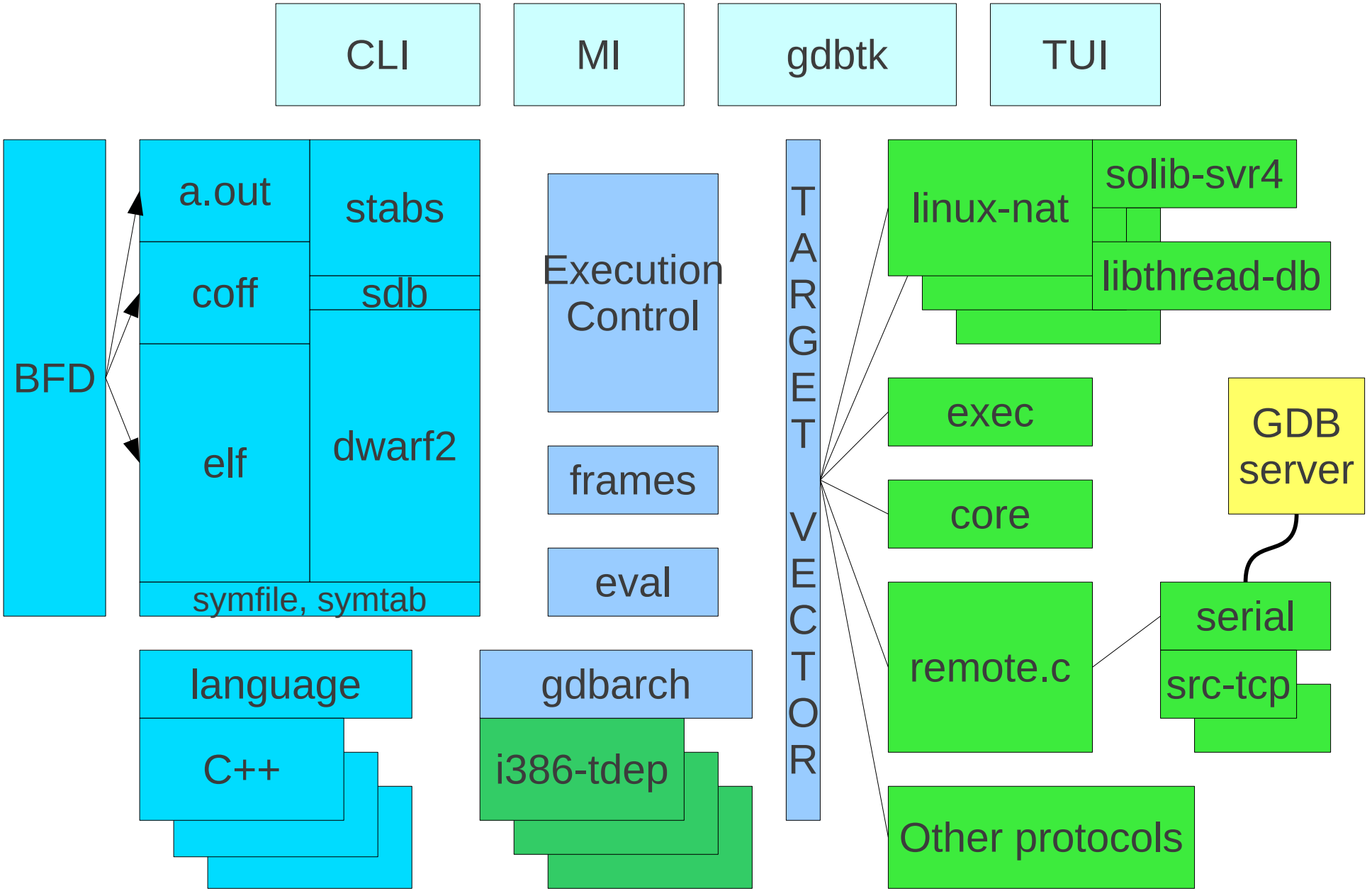
Stan Shebs
Mentor Graphics

April 2013

GDB, the GNU Debugger

- A component of the GNU system since 1986
- Initially native debugging only, but soon extended for cross-debugging
- Several major redesigns / rewrites
  - Target vector, frame objects, gdbarch, …
  - Only a handful of lines remain from early versions
- Default debugger for Linux
    - *(LLDB a possible successor?)*

GDB: The Big Picture



Target System Trends

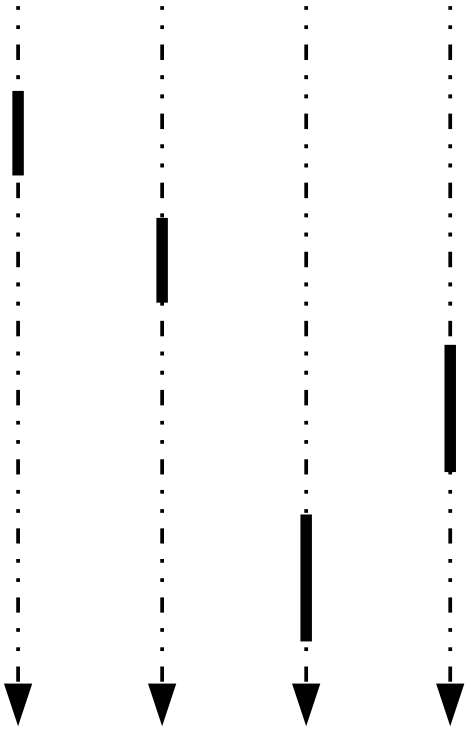
- Widespread Linux usage & growing
  - Low end: phones, tablets, gadgets
  - Middle end: automotive, “infotainment”
  - High end: backbone switches, compute farms
  - Desktop: meh
- Multiple cores are common
  - 2-16 cores in the field
  - 32-100+ cores in the lab

Approaches

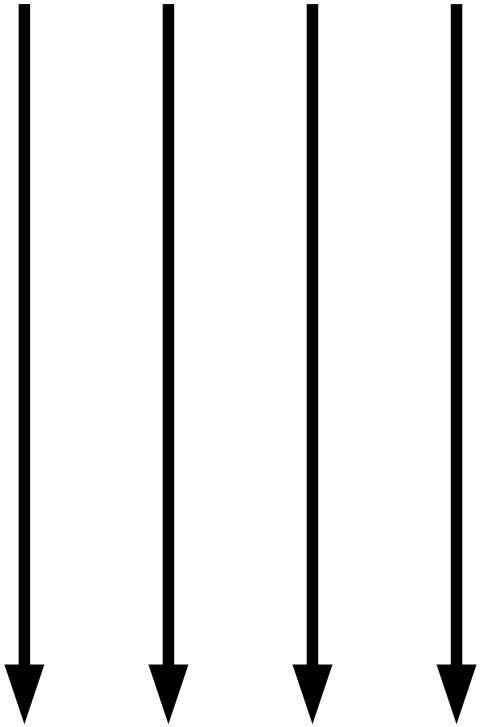
- Do nothing
  - Kernel conceals cores, GDB does threads already
- Per-core commands
  - Useful if app has core affinities
  - Useful if hardware has heterogeneous cores
- Shifting work to the target
  - In many-core systems, network becomes bottleneck

Effect of Simultaneity

Single core, time-sliced



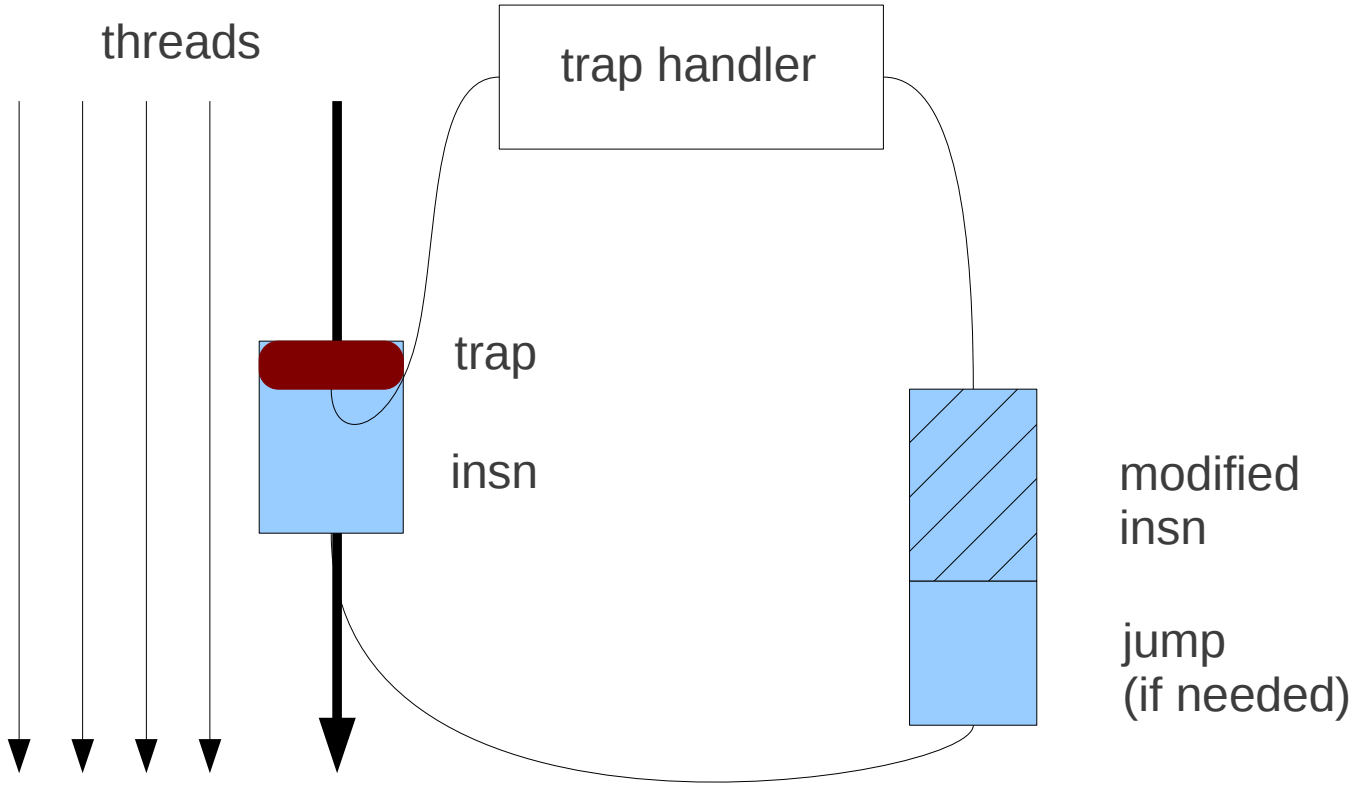
Four cores, all running



Non-stop Debugging

- Traditional behavior is “all-stop”
  - Entire process frozen until step/continue
- In non-stop mode, can stop & step one thread while others continue to run
  - “continue” resumes only the current thread
  - “continue -a” to resume all threads
  - “interrupt” to stop the current thread manually
  - “interrupt -a” to stop all threads
- Non-stop available since GDB 7.0 (2009)

How Displaced Stepping Works



From One to Many

- Old way operates on one thread at a time
  - (gdb) thread 1
  - (gdb) step
  - (gdb) thread 2
  - (gdb) step
  - [hit breakpoint in thread 4]
  - (gdb)
- OK for 2-5 threads, inefficient for more

Sets

- Extend GDB commands to work on sets of processes, threads, and cores
- General form is processes.threads@cores
- Options include
  - explicit
  - ranges
  - predicates
  - union, intersection, complement
  - named sets

Set Examples

- 4563.*
  - All threads of pid 4563
- 4563.*@1-20
  - All threads of 4563 on cores 1 through 20
- *.signalshaper
  - All threads named “signalshaper” on any process
- .1-20
  - 20 threads of the current process

Commands Using Sets

- step *.*@6
  - Single-step all threads of all processes currently running on core 6
- continue .1-5,worker
  - Resume any threads numbered 1 to 5, and any named “worker”
- break myfun thread 100-1000
  - Break in any thread numbered 100 or above

The Host as Bottleneck

- GDB is like a duck
  - Single user command may result in dozens of interchanges with target – get a register, decide to dereference it and read memory, step one instruction, get program counter, etc etc
- On a single-core target with multiple threads, only one thread at a time can need attention, other threads are suspended
- On a multi-core system, one hundred threads can hit the same breakpoint at the same instant

Target-side Operations

- Get host out of the critical path by moving work to the target
- Assumes debugging agent(s) built into the target program somehow
  - Static link, running in dedicated thread
  - Dynamic link, GDBserver handling threads
- Z (breakpoint) packets move breakpoint trap management to target

Target-side Breakpoint Conditions

- Break foo.c:45 if globvar > 92
- Target does the comparison itself, only notifies GDB if the test is true, else continues on
- GDB translates conditional expression to a bytecode sequence, using simple compiler and download mechanism originally developed for tracepoints
- In GDB 7.5

Towards a New Remote Protocol

- Current protocol designed for debugging 68k over serial line, using several kilobytes of memory on target
- Simple commands
  - “g” gets all registers, returns long hex string
- May send hundreds of packets for a backtrace
  - Network latency, context switching, swapping

Multicore Services Framework

- Multicore Association initiative for a generic debugging / tracing mechanism
- Requirements
  - Cross-platform, cross-technology
  - Discovery of targets and their characteristics
  - High performance (multi-GB traces)
  - Coordinate multiple host tools (GDB, LTTng)

GDB in the Services Framework

- Add URIs to target command
  - target mcsf://lab2/router13/arm4core?456.*
- Four services:
  - Get (reg0-7, loc2, loc3:16 bytes, ...)
  - Set
  - Exec control
  - Instrumentation

Reactions?

- TCF!
- Uh, do we really need all this?
- Linus says to debug with psionic powers...
- You're asking for money, aren't you
- Hey, isn't kgdb like a target agent?
- ...