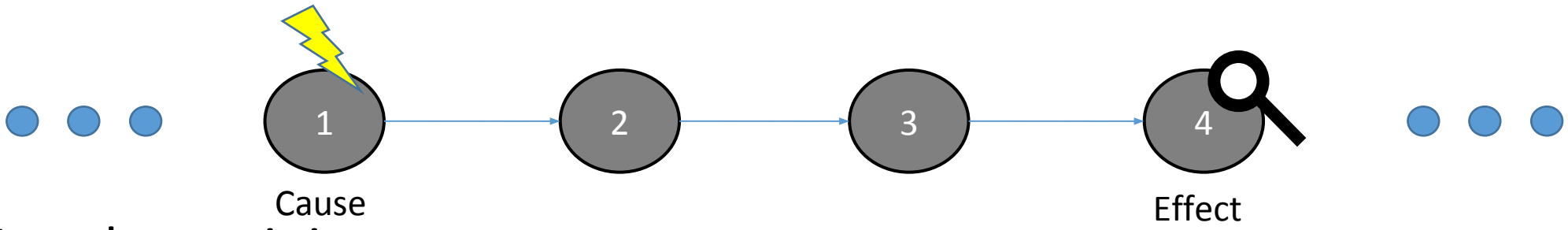
Time Travel for C++ Developers

Intro to the RR debugger

Motivation: why “time travel” debugging?

- Debugging: observe “effect”, but want “cause”!



- Non-determinism
  - Effect shows/disappears randomly
  - In the next faulty run, setup (memory addresses, threads …) might be different

RR Project: Description and Goals

- Record a “run” of an application under test, replay in debugger
  - Possibly on remote systems
- “Exact copy”
  - memory addresses etc. stay exactly the same
  - All sys-calls simulated
- Reverse execution
  - set breakpoint on effect, run backwards till cause fires!
- Fast
  - use CPU acceleration features, overhead for record / replay usually < 2x

Preparations

- Might need some additional setup (for hardware support):
  - echo 1 > /proc/sys/kernel/perf_event_paranoid (or add kernel.perf_event_paranoid to /etc/sysctl.conf)
  - Settings “0” and “-1” allow better debugging but are more dangerous !
- Install the “rr” packages from https://github.com/mozilla/rr/releases
  - Rpm or deb
  - Works “out of the box” on relatively recent Ubuntu/Debian, Fedora/Redhat boxes
- Or build from source:
  - https://github.com/mozilla/rr/wiki/Building-And-Installing

Recording a Session

Replaying a Session

Special Options

Traveling back in time

- Reverse-cont example

Additional commands

Threading example

Integrate into IDE

- See https://github.com/mozilla/rr/wiki/Using-rr-in-an-IDE

- Basics: use GDB remote server protocol, e.g. Qt Creator:
  - “Debug -> Start Debugging -> Attach to Running Debug Server …”

  - How to “reverse” (no buttons!):
  - “Window -> Views -> Debugger Log”
  - Enter “rs”(reverse-step) / “rc”(reverse-continue) in “Debugger Console”

Limitations

- Single-core only (multi threaded applications serialized)
- Shared-memory connections with non-debugged processes cannot be traced
- Some syscalls not implemented
- Does not work on some Virtualization platforms (Virtualbox, Hyper-V…)

- x86(-64) / Linux only
  - Windows: Visual Studio debugger can do something similar
  - ARM/general Linux: UNDO DB
    - slower but more versatile approach compared to RR

Image see

http://rollsoffthetongue.tumblr.com/post/65441796803/what-do-we-want-time-travel-when-do-we-want-it

Questions?

Via Mail: Volker Aßmann (volker.assmann _at_ gmail.com)
Code: https://github.com/volka/talks