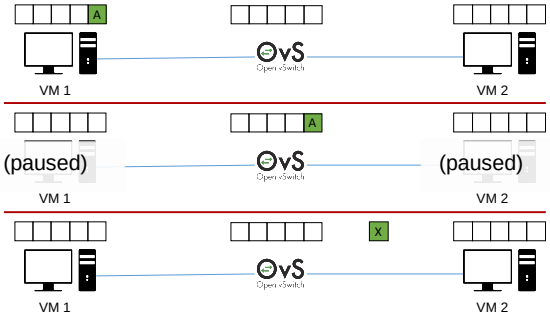
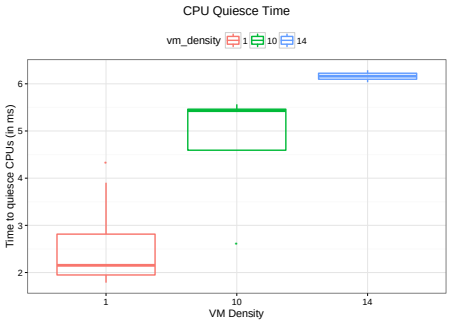
Exceptional service in the national interest

Sandia National Laboratories

FIREWHEEL





Staghorn

An Automated Large-Scale Distributed System Analysis Platform

Kasimir Gabert (5638), Ian Burns (9526), Steven Elliott (5634), Jenna Kallaher (5632), Adam Vail (5634)

U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration

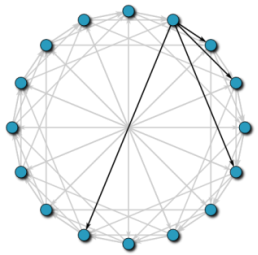
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2017-8419 C

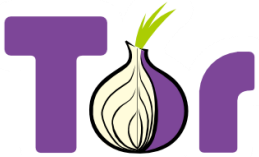
Problem

Sandia National Laboratories

- Large, distributed systems have become ubiquitous
- A common method for understanding their behavior is to simply run them and observe / experiment (Emulytics)
- This necessarily competes with the model for CPU time, and the model and analysis must run at clock rate
- We built a way to “stop time” within a model, opening the door to the larger world of offline model analysis

openstack™
CLOUD SOFTWARE





TorProject.org

A Few Use Cases

Sandia National Laboratories

- Vulnerability analysis
- Debugging systems
- Optimizing tests
- Experimental repeatability
- Training

Key Contributions

Sandia National Laboratories

- A full-system snapshot and restore capability for Sandia's large-scale emulation-based model environments which preserves network and I/O state
- A network modification system that allows for modification of Ethernet frame contents and delivery, or the introduction and removal of frames, during a snapshot
- The evaluation of this capability on real-world use-cases

Design Requirements

Sandia National Laboratories

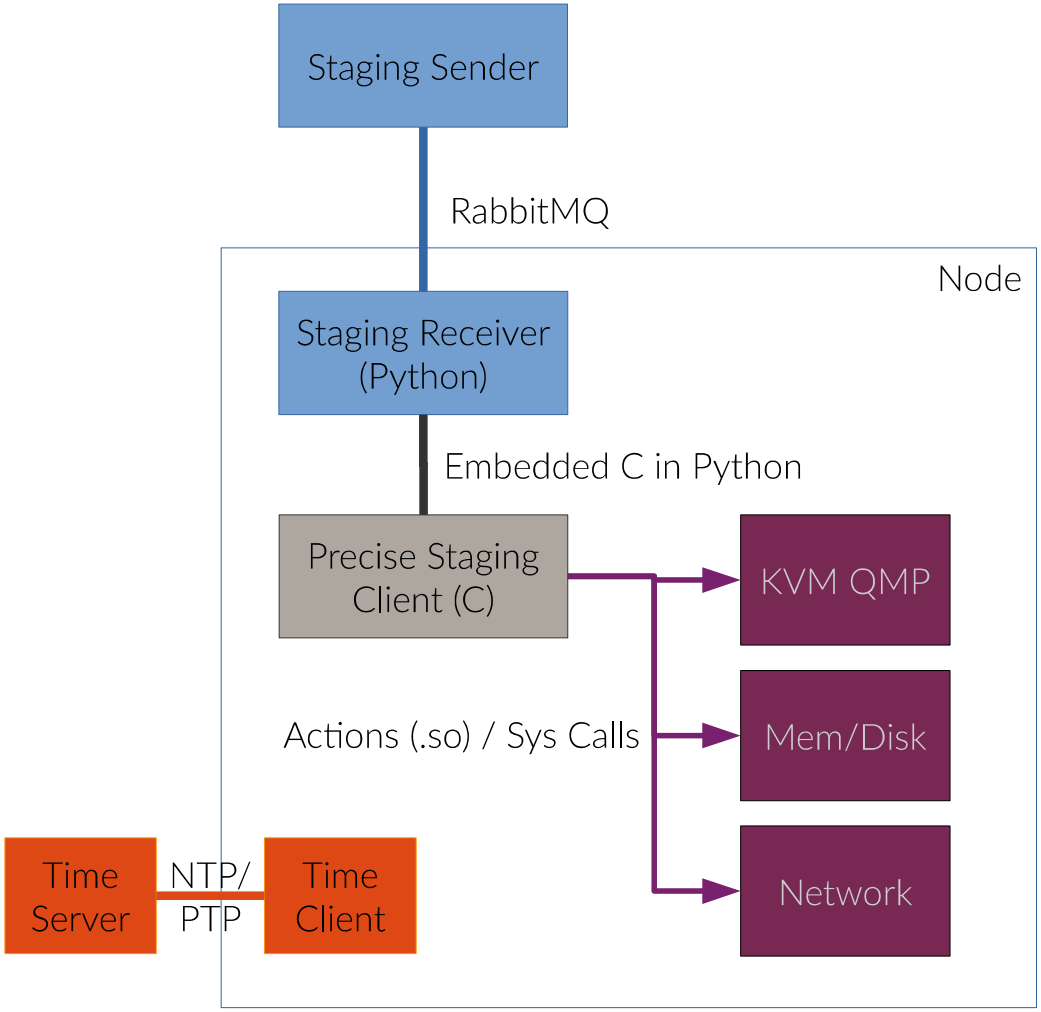
- The system must not perceive that a snapshot has occurred
- Staghorn must preserve machine and network state
- Staghorn must snapshot quickly so that each virtual machine is snapshotted within a tight time window

Firewheel

Sandia National Laboratories

- Staghorn is built on top of Firewheel, Sandia-developed tool for automating the challenging parts in Emulytics
- Two big technologies Firewheel brings:
  - Graphs to represent models
  - Plugin architecture to make automation extensible
- Firewheel is scalable: to 75,000 VMs booting in 13 minutes

Staging Architecture



VM State Snapshots

Sandia National Laboratories

- Currently using QEMU migration-based snapshots
  - Straightforward to implement because they utilize existing QEMU mechanisms.
- Explored two other approaches:
  - Process-level snapshots
  - QEMU fork-based snapshots

Network Snapshots

Sandia National Laboratories

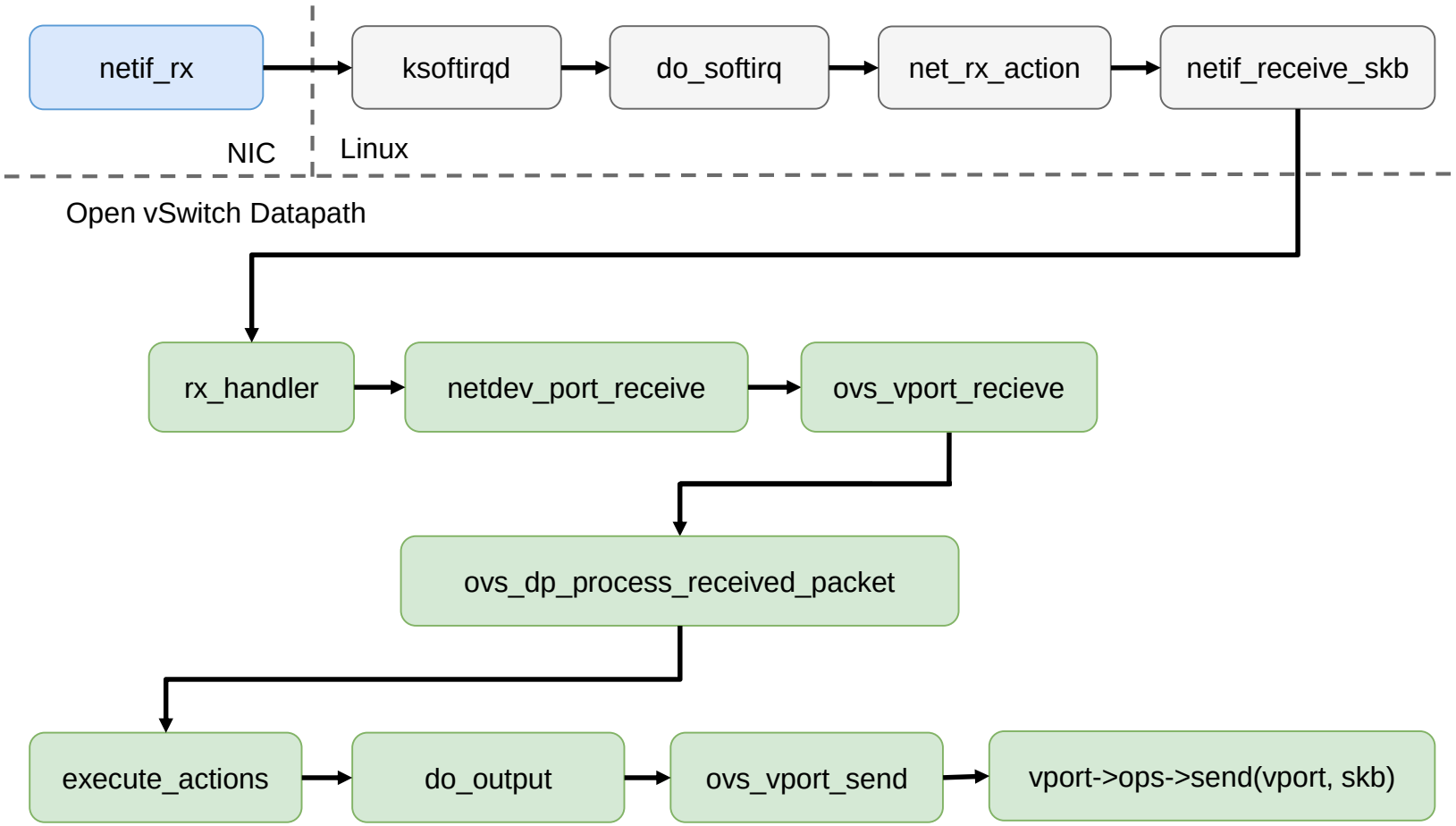
- Design decisions:
  - Should we prioritize packet latency or packet ordering?
    - Choose packet ordering but minimize queuing delay as much as possible
  - How to pass information to/from the kernel?
    - Netlink, it is quick, asynchronous, and easy to implement
  - Where should we place our modifications?
    - Open vSwitch

Why OVS

Sandia National Laboratories

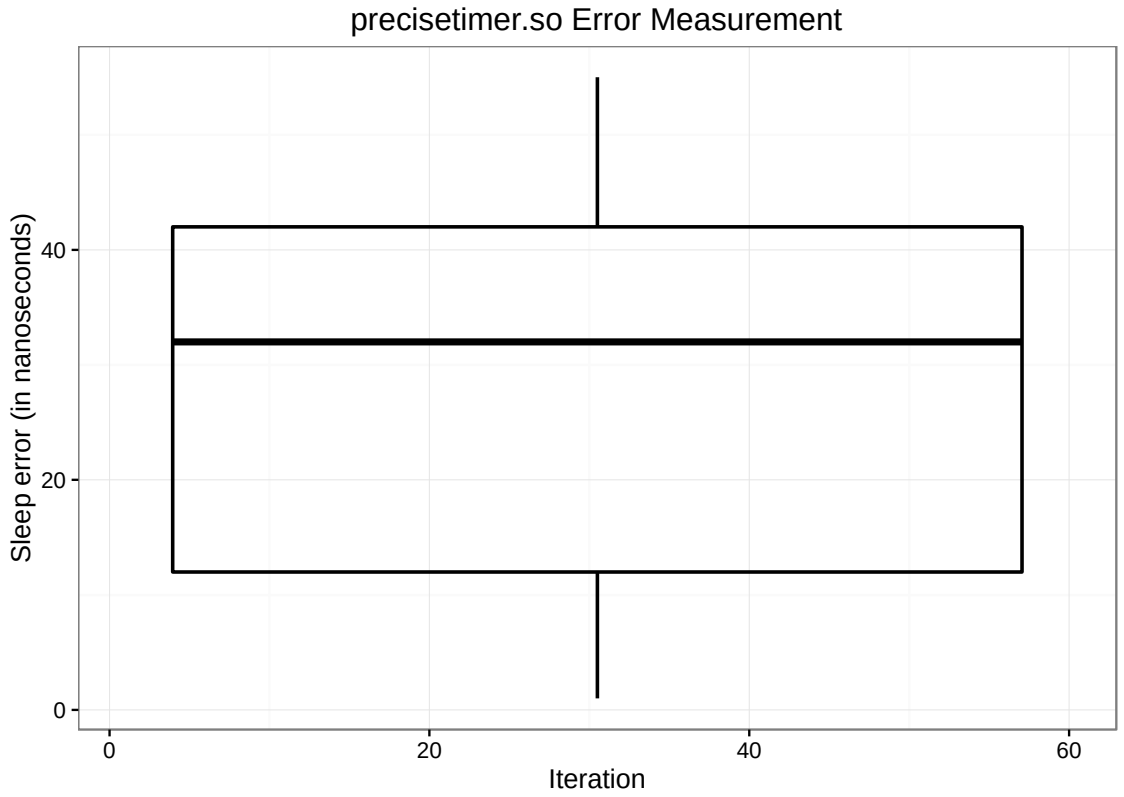
- Can capture packets between cohosted VMs
- Easy to install and actively developed
- Compatible with virtualization platforms (KVM, Xen, etc.)
- Already works with both Minimega and Firewheel

Network Snapshot Architecture

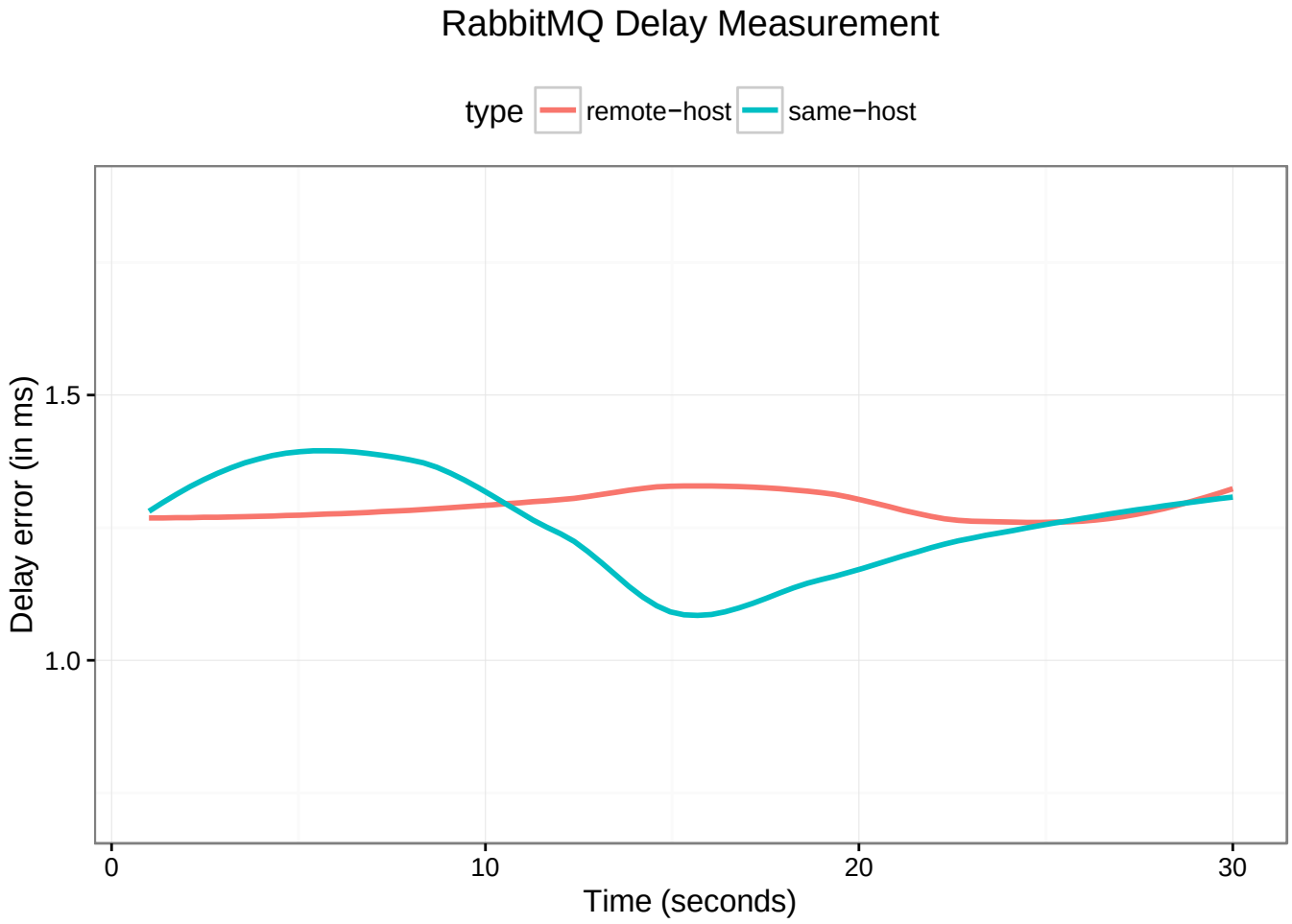


Evaluation – precisetimer.so

- Tried to sleep 1 second into the future 60 times and measured how close the sleep was to the desired time.
- Results ranged from 1 – 55 ns with mean of 28.05 ns

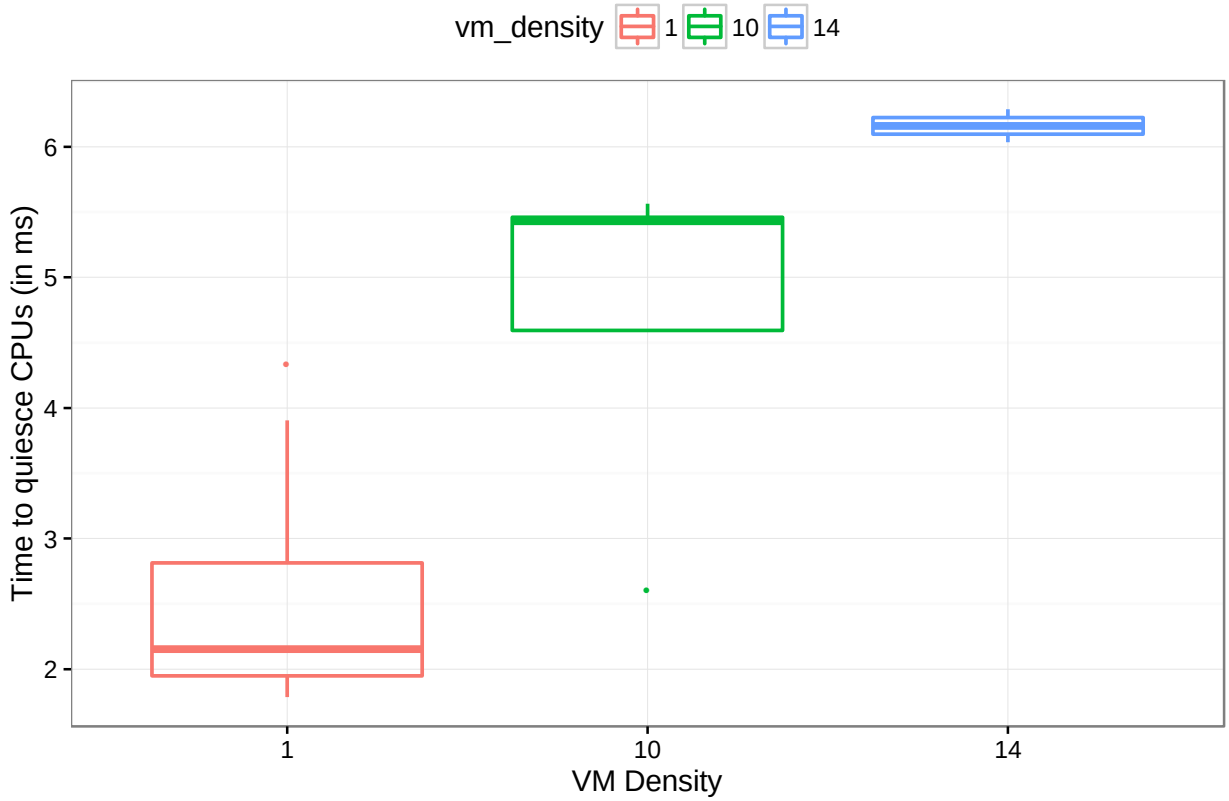


Evaluation - RabbitMQ

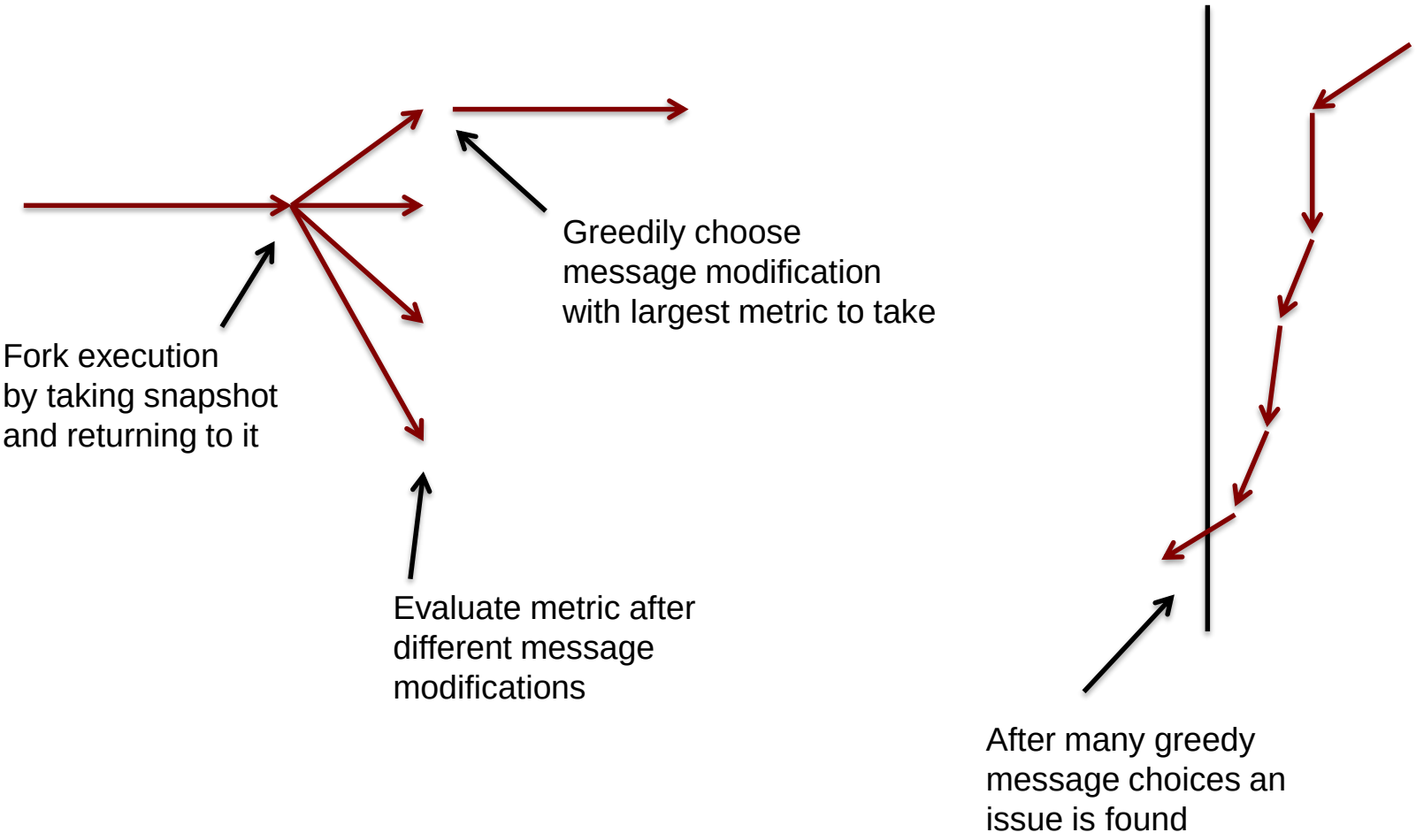


Evaluation – Snapshot Timing

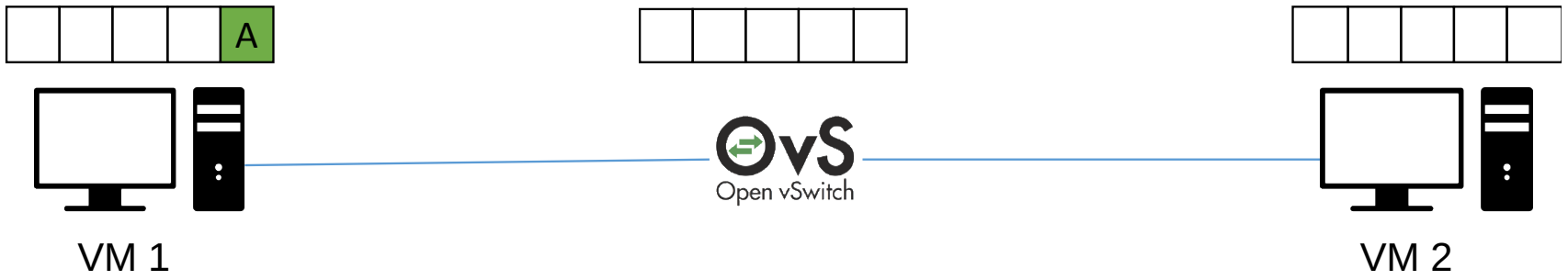
- One of the most critical timing aspects of Staghorn is the performance of quiescing the virtual CPUs on each VM



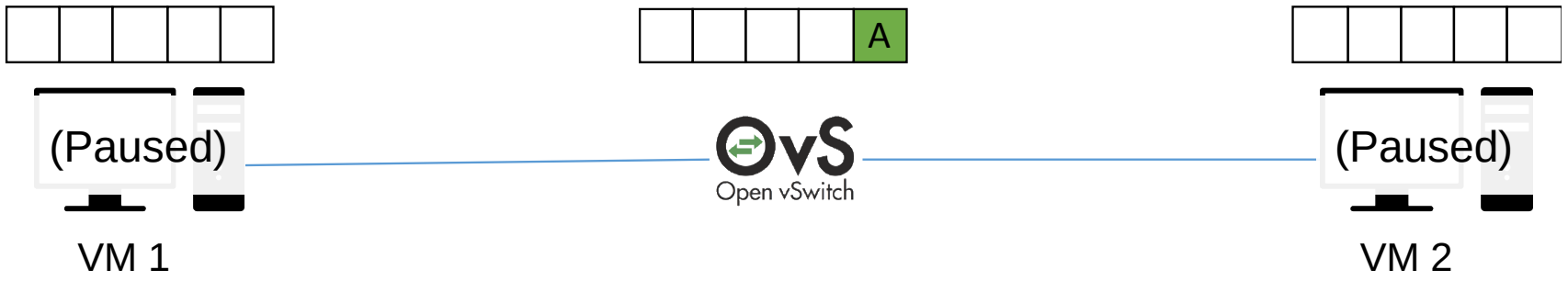
Use Cases – Distributed Fuzzer



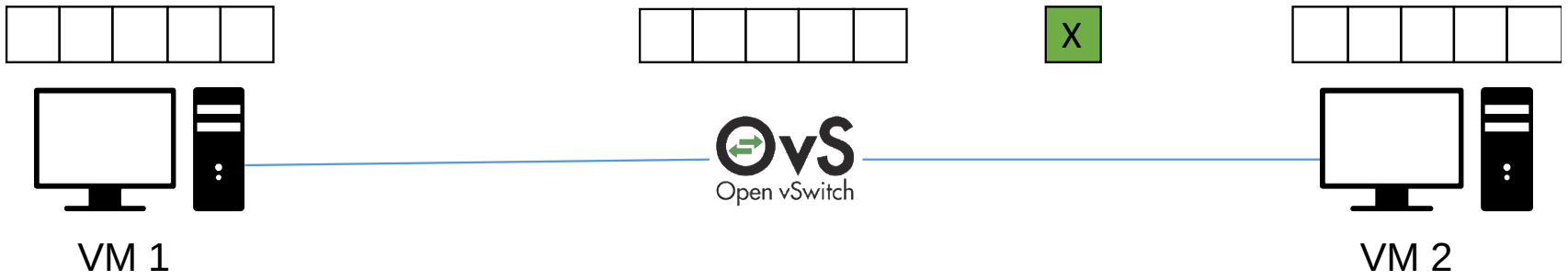
Use Cases – Distributed Fuzzer



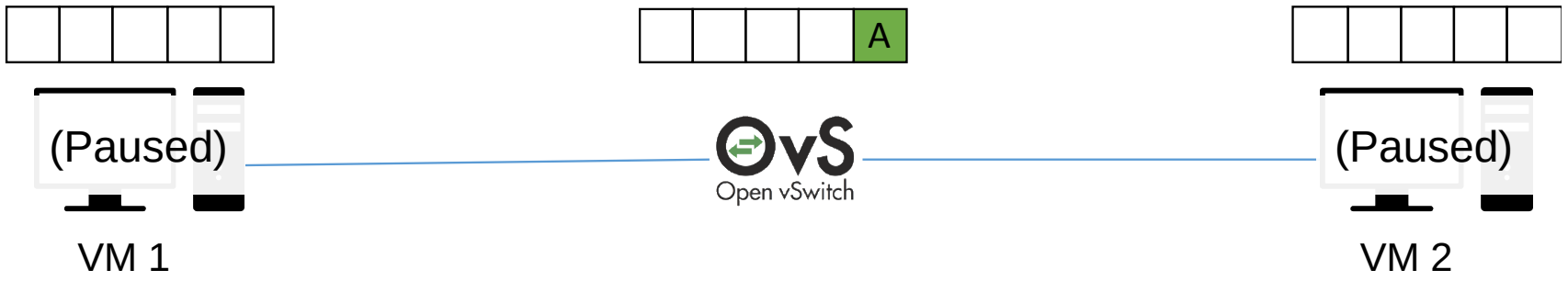
Use Cases – Distributed Fuzzer



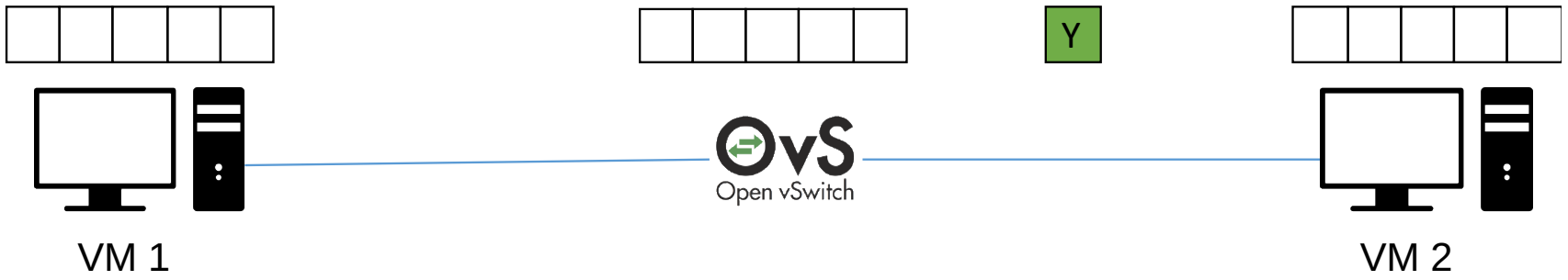
Use Cases – Distributed Fuzzer



Use Cases – Distributed Fuzzer



Use Cases – Distributed Fuzzer



Use Cases – Distributed Debugger

Sandia National Laboratories

1. Set breakpoint
2. Install Staghorn Trigger
3. Staghorn will wait until the breakpoint is hit to snapshot the system.

```
$ jdb -attach 198.128.0.1:8888
Set uncaught java.lang.Throwable
Set deferred uncaught java.lang.Throwable
Initializing jdb ...
> stop in Example.math
Set breakpoint Example.math
>
```

Trigger

Offset	: 75
Data	: 4064020000000102

```
Set breakpoint Example.math
>
Breakpoint hit: "thread=main", Example.math(), line=5 bci=0

main[1]
```

Use Cases – Debug Experiments

Sandia National Laboratories

- Firewheel user's experiment failed after about 8 hours.
- An 8 hour debug cycle is unacceptable.
- Staghorn was used to snapshot before the crash enabling the user to quickly test various fixes.

Conclusion/Future Work

Sandia National Laboratories

- Conclusion
  - We have opened the door to offline analysis and modification for our large-scale emulation based models
- Follow-on work:
  - Improve our performance
  - Implement/productize more use cases
  - Better identify how long it takes for CPUs to quiese and improve this time
  - Improve the stability of process-level snapshots and QEMU fork-based snapshots

Any Questions??

Sandia National Laboratories

- Paper: www.sandia.gov/emulytics/staghorn-report.pdf
- Contact info: Steven Elliott (selliot@sandia.gov)