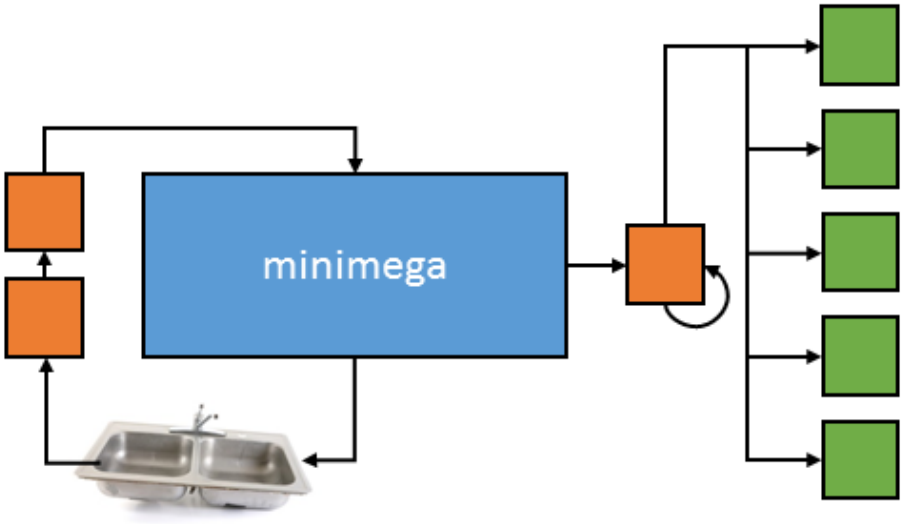
SAND2017-1449PE

Introducing miniplumber

David Fritz
7 Feb 2017

Introduction

miniplumber provides uni- and multi-cast experiment communication to VMs, minimega instances, and external programs anywhere in an experiment.



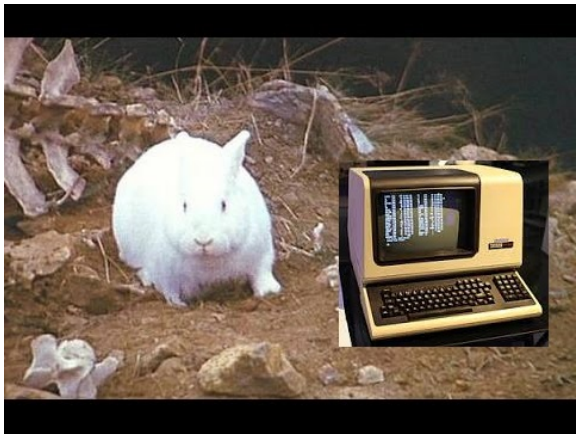
Motivation

How do we plumb non-network based connectivity?

- Emulytics helps study many network-centric problems
- minimega builds networks easily
- Great support for describing machine-to-machine structure/behavior
- **Not great** support for describing machine-to-anything-else structure/behavior

Plumbing the kitchen sink

- Buttons
- Serial connections
- Vehicle radars
- **physics** - sound, heat, light, rabid bunnies
- cyber-physical interactions of all kinds
- measurement data-planes



Plumbing

Enter *miniplumber*

- A networkless, out-of-band, inter-process communication layer
- Quick specification of communication pathways (pipelines)
- uni- or multi-cast experiment communication
- Supports any number of clients (scales along with the rest of minimega)
- Similar to unix pipelines (though not limited to linear pipelines)
- Borrows concepts from the plan 9 plumber
- Newline delimited messages (as opposed to unix byte streams)
- Works on host, in minimega, and any miniccc client (ARM, x86-64 / *bsd, linux, windows)

A quick example

```
# create a pipeline connecting foo and bar
plumb foo "sed -u s/foo/moo/" bar

plumb

# write something to foo
pipe foo "the cow says foo"

pipe

# works from the command line too
# echo "hello plumber!" | minimega -pipe foo

clear plumb
clear pipe
```

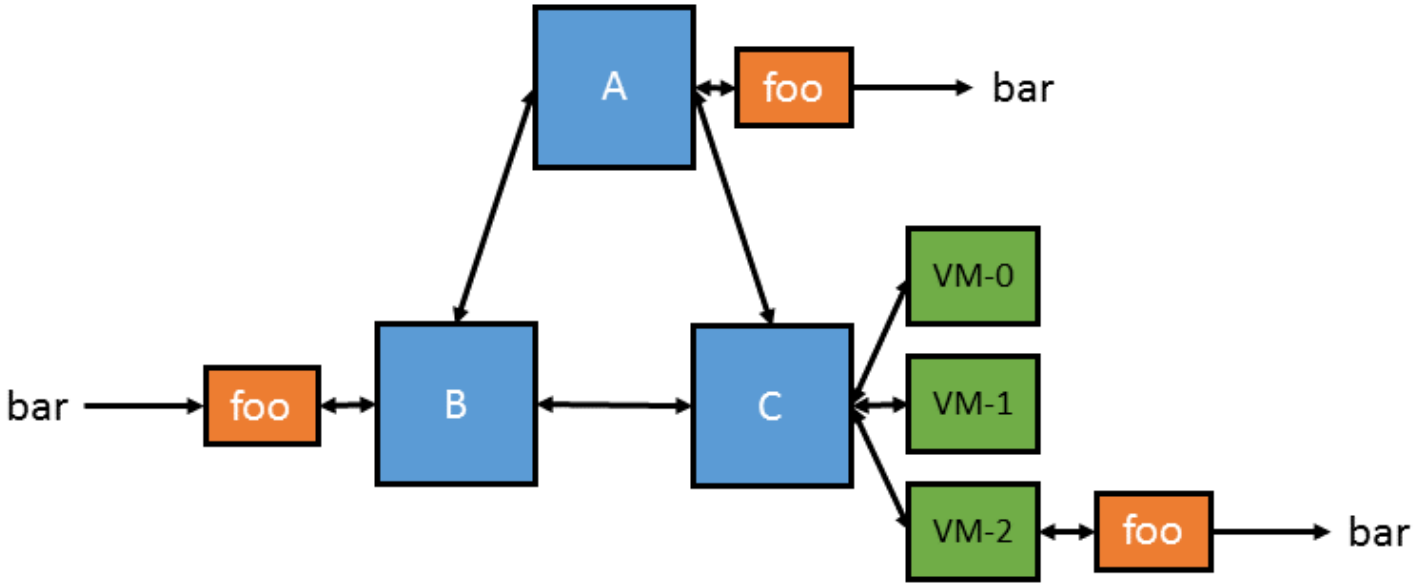
Plumbing objects

- Simple standard I/O based communication
- Easily plumbed to existing unix tools
- Several minimega-supplied tools (minevent, distribution functions, etc.)
- Pipelines are plumbed locally, pipe data is forwarded everywhere

```
# plumbing productions are similar to unix pipelines
plumb "cat data.txt" "grep -i foo" lines_with_foo

# named pipes can attach directly
plumb this is a valid production

# pipelines live until any stage of the pipeline is destroyed
plumb foo "grep -i foo" bar

# kill the above pipeline by closing foo
clear pipe foo
```

minimega distribution

Fully distributed via meshage



Out-of-band communication

- Works over meshage for node-to-node communication
- Uses miniccc/ron for node-to-VM communication
- Fully out-of-band when using miniccc networkless backchannels

Message scheduling

- Support for most communication use cases
- one-to-many
- one-to-one (round robin, random)

```
# create a pipe with round-robin scheduling
pipe foo mode round-robin

pipe
```

Use cases

- cyber-physical experiments
- training
- data forwarding

Summary

- *miniplumber* - simple, out-of-band communication framework
- Growing library of communication primitives, support for any external programs
- Bring cyber-physical / simulation layers to emulytics

Thank you

David Fritz
7 Feb 2017