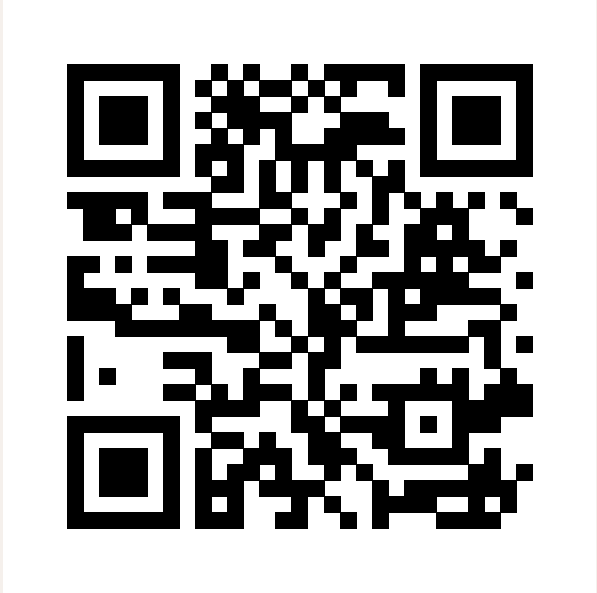
TinyRange: Next-generation Virtualization for Cyber and Beyond

Joshua D. Scarsbrook - The University of Queensland

July 10th, 2024

Slides

https://vbitz.github.io/presentations/2024/tinyrange2



$ whoami

Joshua Scarsbrook (j.scarsbrook@uq.edu.au)

Research Officer at The University of Queensland

Interested in...

- Operating Systems
- Virtualization
- Embedded Hardware

Working in the Computational Imaging Group of EECS

What is TinyRange?

Next-generation Virtualization for Cyber and Beyond

Long Term Goal: Make running and building *any* software effortless for all modern hardware.

Where are we today?: Super fast and easy `Linux` virtual machines.

TinyRange is Open Source!

License: Apache License 2.0

Source Code: https://github.com/tinyrange/tinyrange

Virtual Machines vs. Containers

- **Virtual Machines** emulate the entire computing stack down to the CPU. You can run any operating system in a virtual machine.
- **Containers** share the kernel and hardware resources with the host operating system. They are applications given a different view of the same operating system.
- The biggest limiting factor for virtual machines is how you build them.

How do we get files into `Virtual Machines`?

- **Block Devices:** Universal support but slow to build.
- **Network Filesystems:** Moderate support with moderate speed (e.g. `SMB`, `NFS`, `sshfs`).
- **VM Filesystems:** Limited support but very fast (e.g. `virtio-fs`, `virtio-9p`, Shared Folders).

Virtual `ext4` Filesystems.

- A write-only `ext4` driver in `Go`.
- Mapped to virtual machines using `NBD` as a regular block device.
- Performance comparable to a regular disk image or RAM disk.

How?

- A custom language that defines structures with safe byte-level indexing.

```
type DirEntry2 struct {
    inode                u32_le  // Number of the inode that this directory points to.
    rec_len              u16_le  // Length of this directory entry.
    name_len             u8      // Length of the file name.
    file_type            u8      // File type code, see ftype table below.
}
```

- Virtual memory mapping emulated in Userspace with byte-level granularity.
- ~2000 lines of `Go` to implement `ext4` support.

TinyRange Research Gaps

1. **Software Installation:** Now we can get files into a virtual machine how do we get software installed?
2. **Virtualization:** Currently using QEMU. A better replacement could enable running all this in a web browser.
3. **Bootloader:** Currently ties us to Linux.
4. **Alternate Guest Operating Systems:** Needs a driver for the filesystem and a bootloader.

Scripting

```
make_vm([
    define.plan(
        builder = "alpine@3.20",
        packages = [
            query("ifupdown-ng"), query("busybox"), query("busybox-binsh"),
            query("alpine-baselayout"), query("openrc"), query("docker"),
            query("docker-openrc"), query("hyperfine"),
        ],
        tags = ["level3"],
    ),
    vm_modfs,
    directive.add_file("/run/openrc/softlevel", file("")),
    directive.add_file("/etc/network/interfaces", file("")),
    directive.run_command("openrc"),
    directive.run_command("service docker start"),
    directive.run_command("""while (! docker version > /dev/null 2>&1); do
  sleep 0.1
done""" ),
])
```

What am I using this for?

NeuroDesk

- Headed by Dr. Steffen Bollmann with many other contributors.
- Currently built with generated `Dockerfiles` and two containers using TinyRange.
- Hoping to use TinyRange in the future.
- ~100 Neuroimaging tools distributed publicly with `Docker`, `Singularity`, `CVMFS`.
- Users all over the world.

Thanks for Listening

Source Code: https://github.com/tinyrange/tinyrange

Email: j.scarsbrook@uq.edu.au

Mastodon: @jscarsbrook@infosec.echange