OSv: probably the Best OS for Cloud workloads ~~you've never heard of~~

Roman Shaposhnik, Director of Open Source @Pivotal, rvs@apache.org

APACHECON EUROPE

CORINTHIA HOTEL
BUDAPEST, HUNGARY
NOVEMBER 17-21, 2014

# whoami

- Director of Open Source @Pivotal
- ASF junkie
  - Member, VP of Incubator
  - Co-founder of Apache Bigtop
- Used to work for Sun micro around Solaris
- Used to be a kernel hacker
  - Linux, Plan9

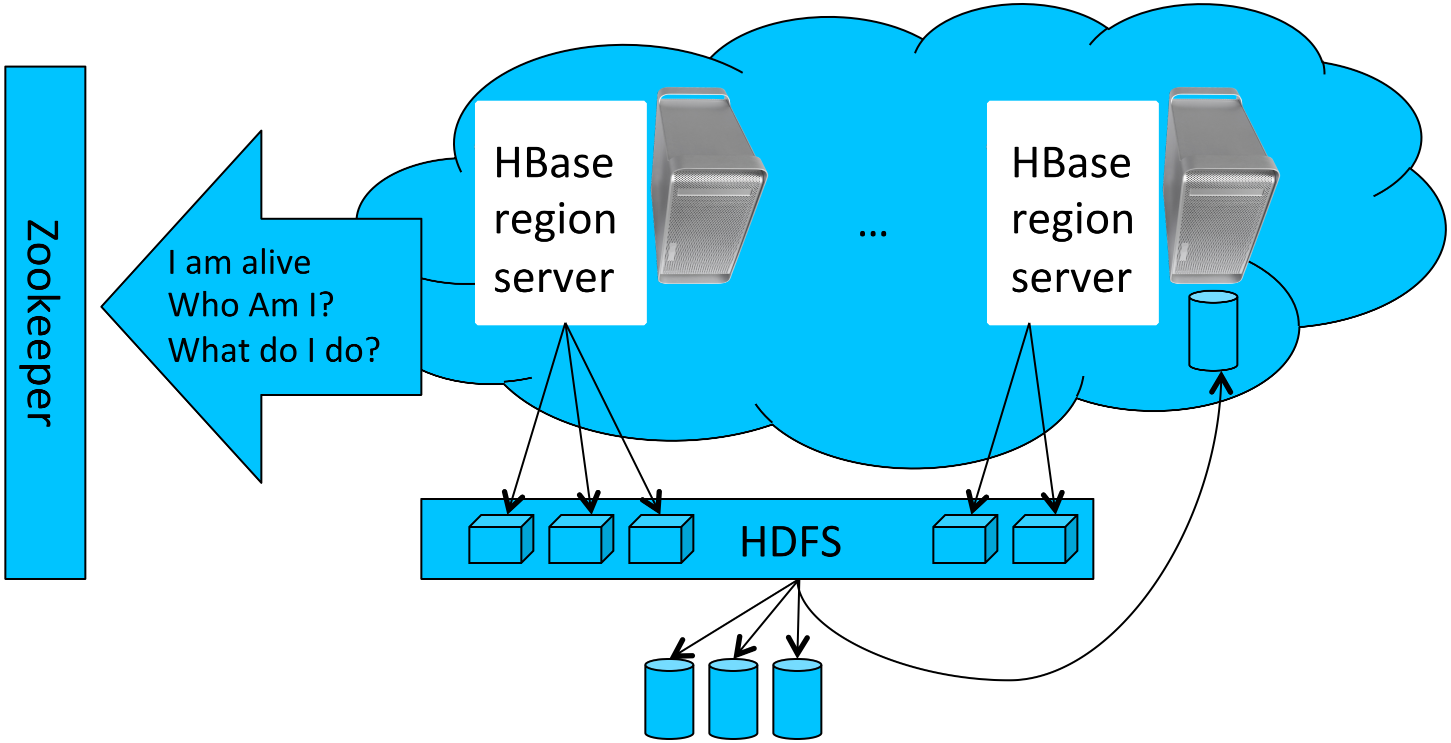
Why am I talking about OSv

- The most exciting development in kernel/ OS space in a long time
- How distributed systems and µservices were meant to be deployed
- A non-DOA way to run JVM on “bare metal”
- A very exciting building block for PaaS

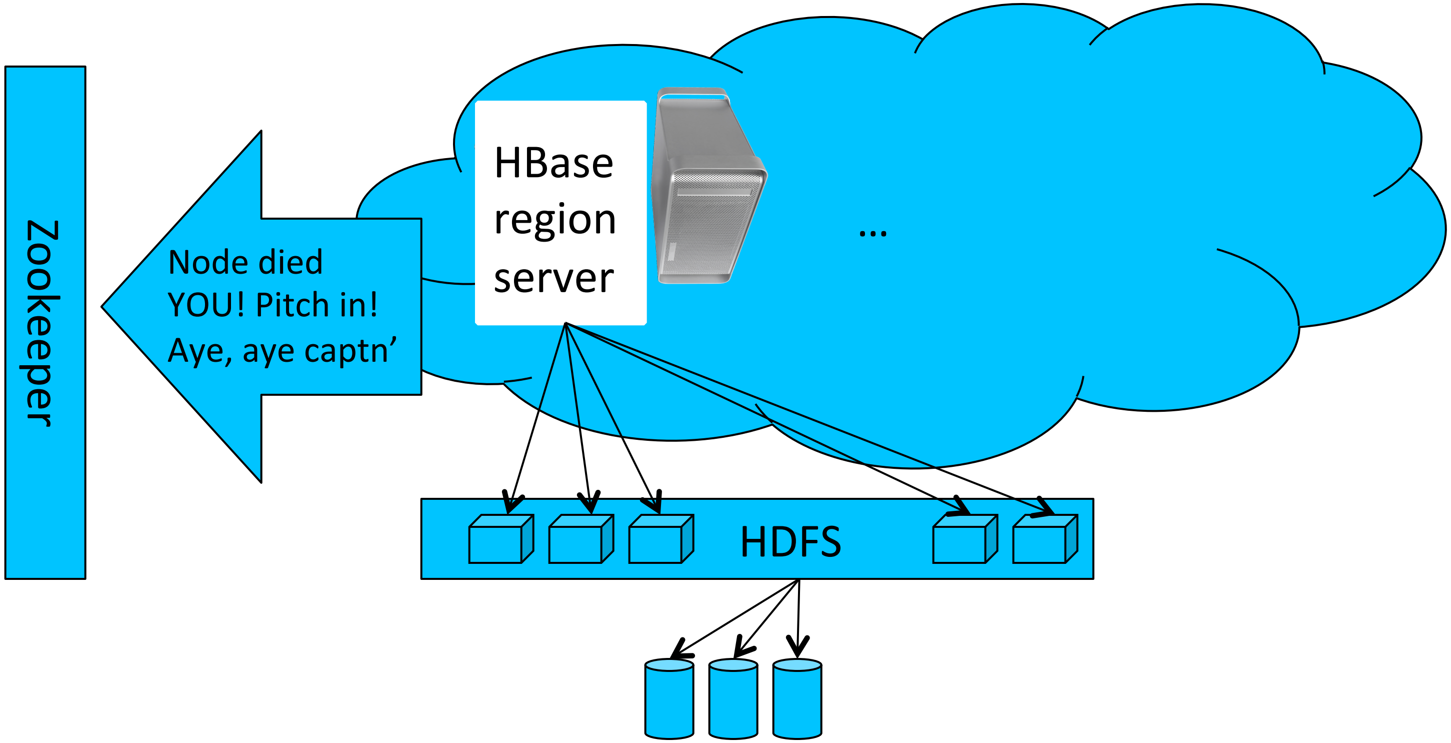
Why am I talking about OSv

- The most exciting development in kernel/ OS space in a long time
- How distributed systems and µservices were meant to be deployed
- A non-DOA way to run JVM on “bare metal”
- A very exciting building block for PaaS
- Expecting a beer from Cloudius Systems

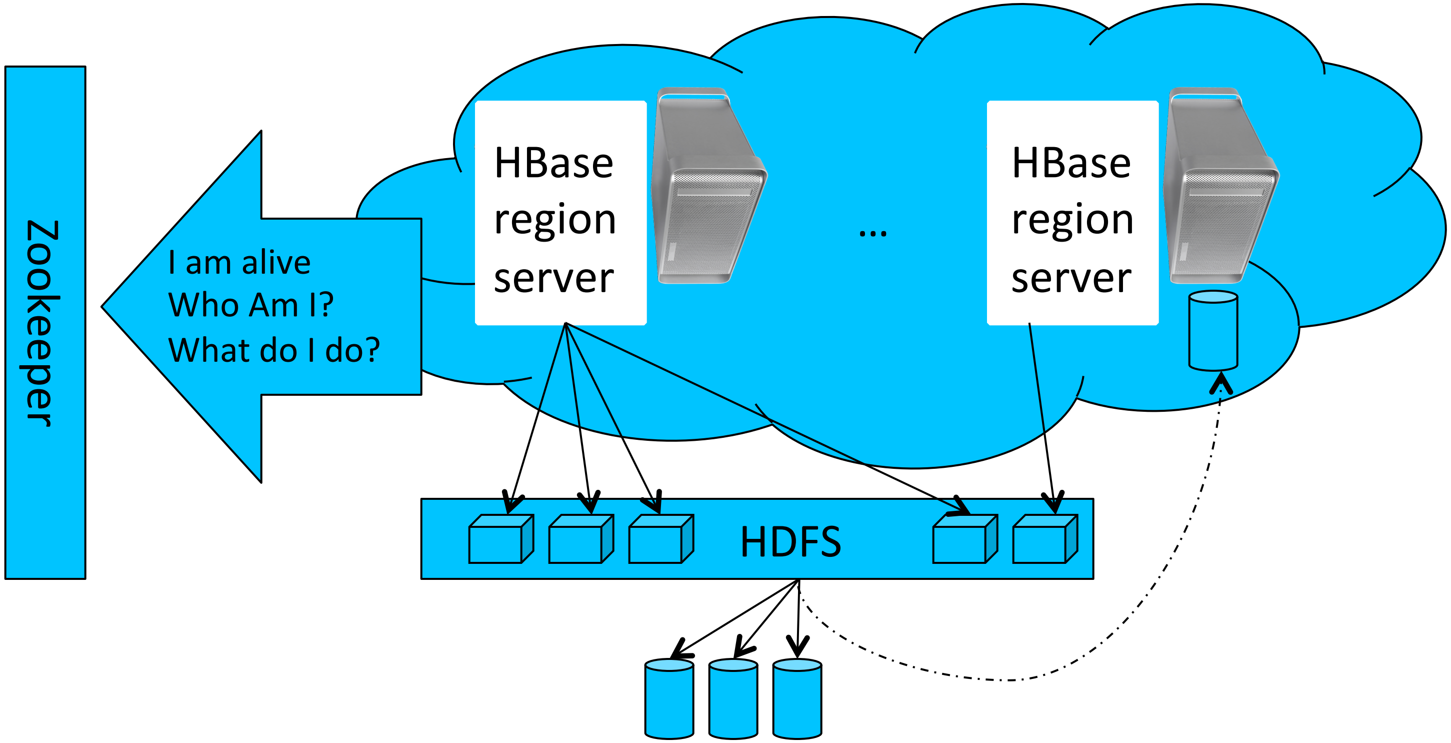
Distributed system architecture in 30s



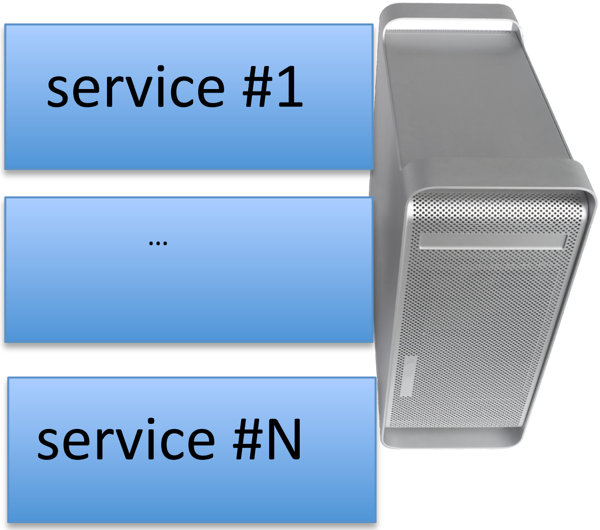
Failure scenario



Failure recovery



Zooming in on each node

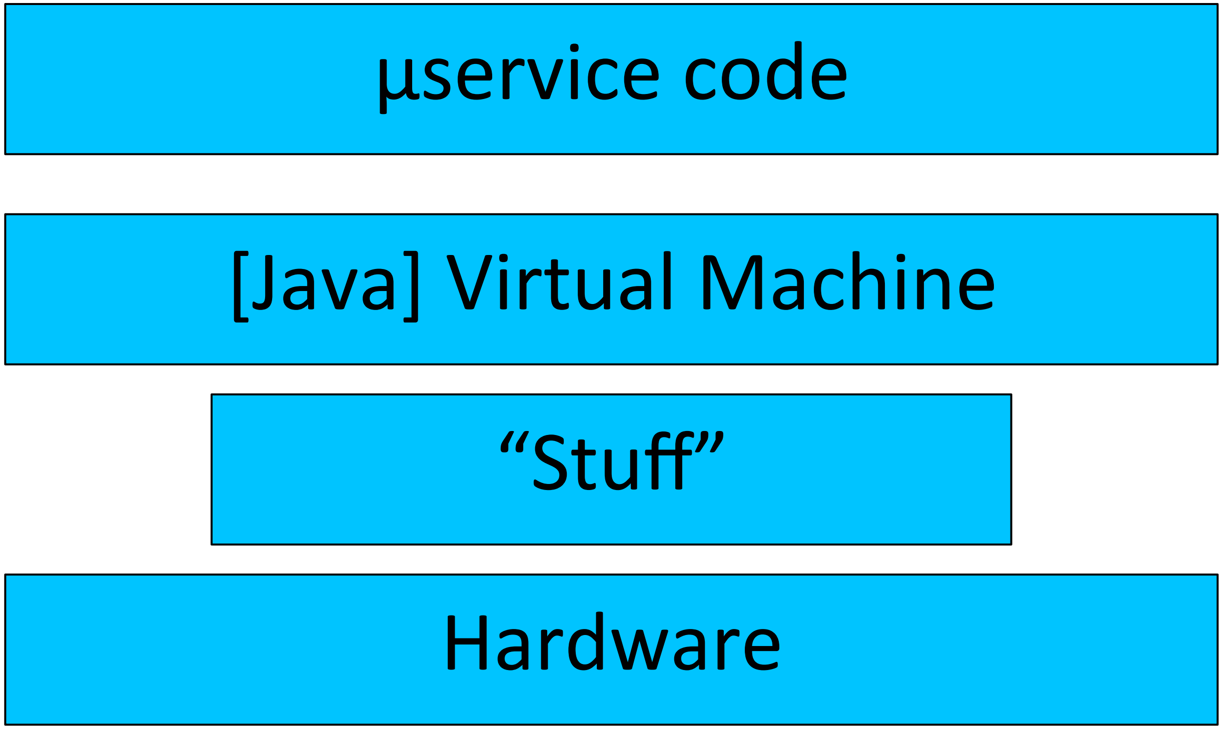


service #1

…

service #N

Anatomy of a µservice



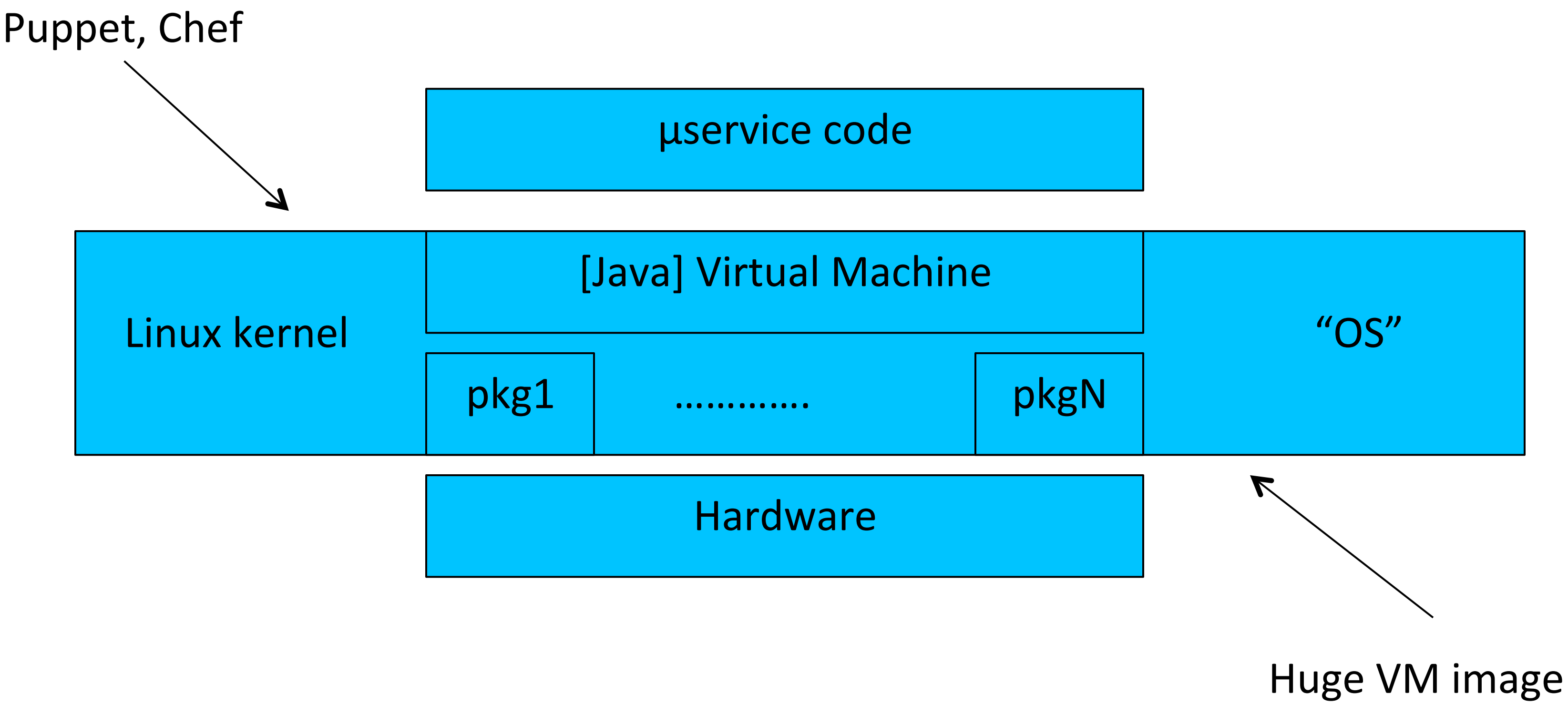
µservice code

[Java] Virtual Machine

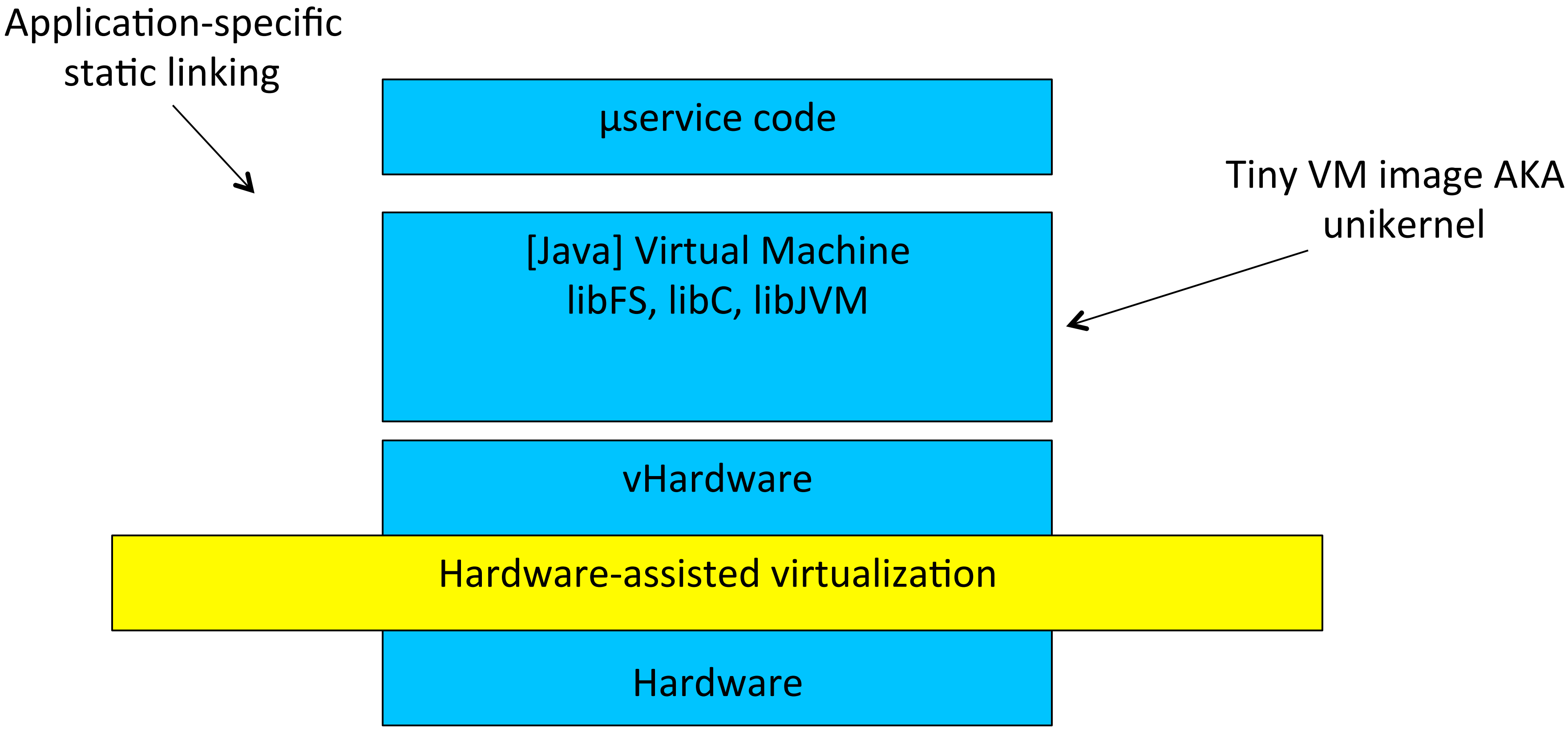
“Stuff”

Hardware

How are we doing it today?



Is there a better way?



Application-specific static linking

µservice code

[Java] Virtual Machine
libFS, libC, libJVM

vHardware

Hardware-assisted virtualization

Hardware

Tiny VM image AKA unikernel

What the heck is a FOOkernel?

- What OS design courses have taught us?
  - microkernels vs. monolithic kernels
- What did they left behind?
  - exokernels, nano
- What they should've taught us instead:
  - unikernels, anykernels

Unikernels

- “Unikernels: library operating systems for the cloud” came out in 2013
- A “library” operating system
- A kernel that can only support one process

Anykernels

- Programming discipline for kernel code reuse
- “The Design and Implementation of the Anykernel and Rump Kernels” by A. Kantee
- Capabilities
  - NetBSD filesystems as Linux processes
  - User-space TCP/IP stack

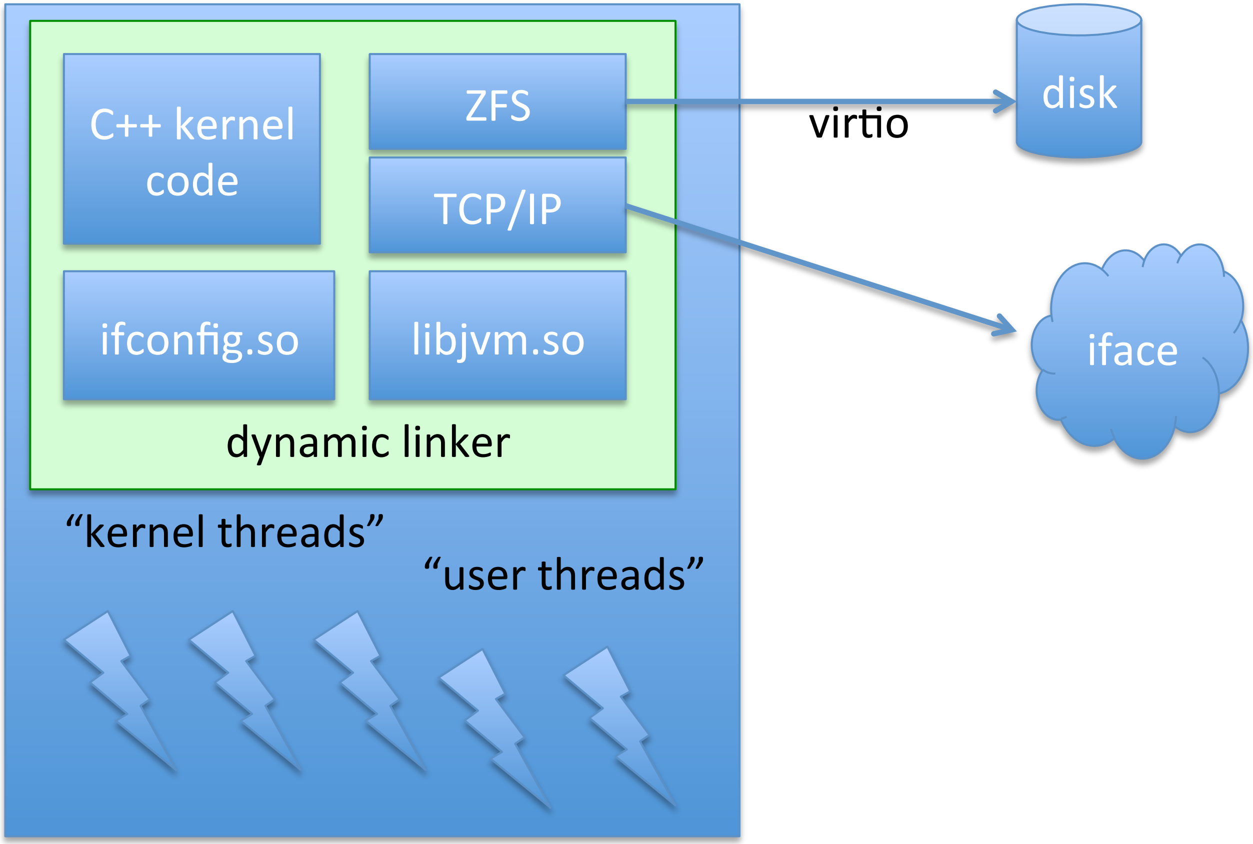
OSv from Cloudius Systems

- A unikernel for “POSIX” and memory managed platforms (JVM, Go, Lua)
- Anykernel’ish
  - E.g. ZFS
- Runs on top of KVM, Xen, VirtualBox, VMWare
- Looks like an app to the host OS
- Small, fast and easy to manage at scale

OSv manifesto

- Run existing Linux applications
- Run existing Linux applications faster
- Make boot time ~= exec time
- Explore APIs beyond POSIX
- Leverage memory managed platforms (JVM, Go)
- Stay open

What’s inside?



single address space in “kernel mode”

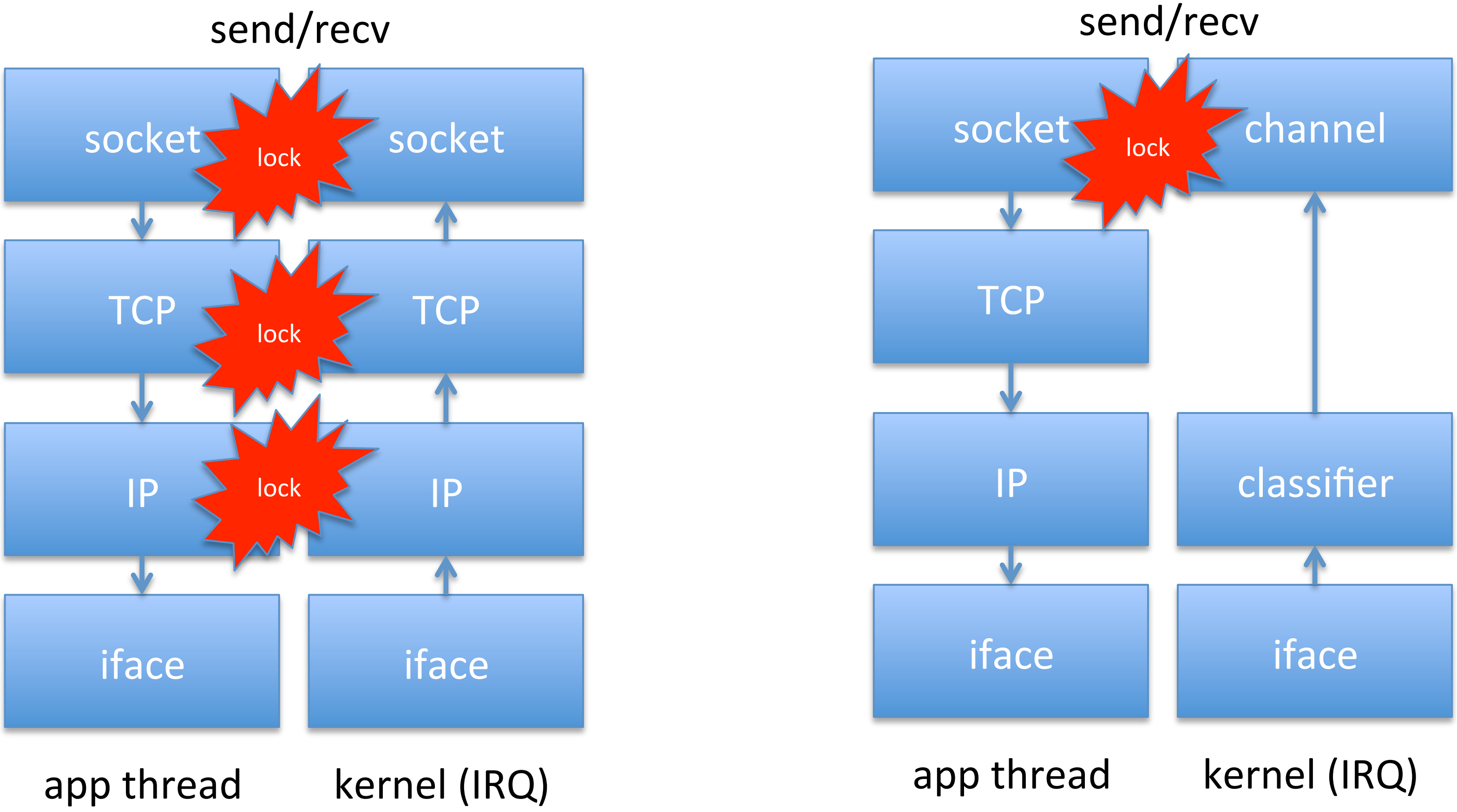
Anything it can't do?

- A 100% replacement for a Linux kernel
  - No fork()ing
- No process isolation
- The least amount of device drivers ever

Virtualization vs. performance

- Network-intensive apps:
  - unmodified: 25% gain in throughput
    47% decrease in latency
  - non-POSIX APIs use for Memcached:
    290% increase in performance
- Compute-intensive apps:
  - YMMV

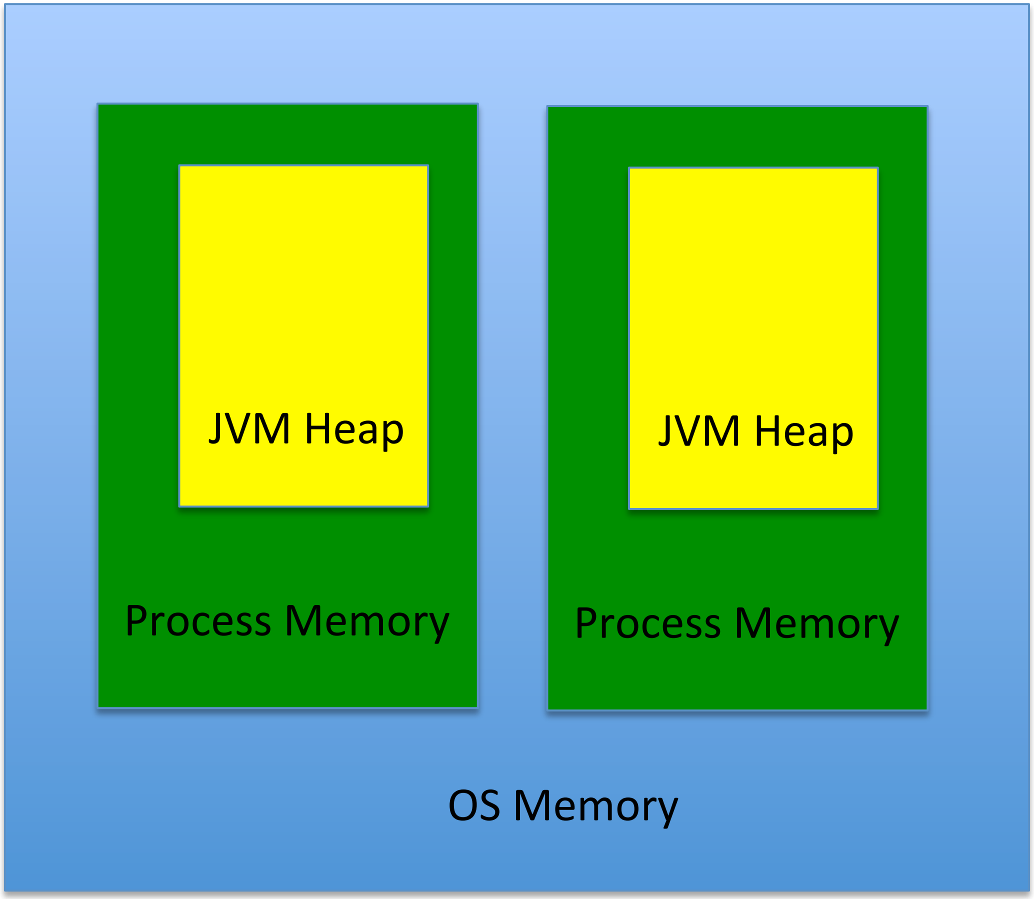
Van Jacabson’s net channels



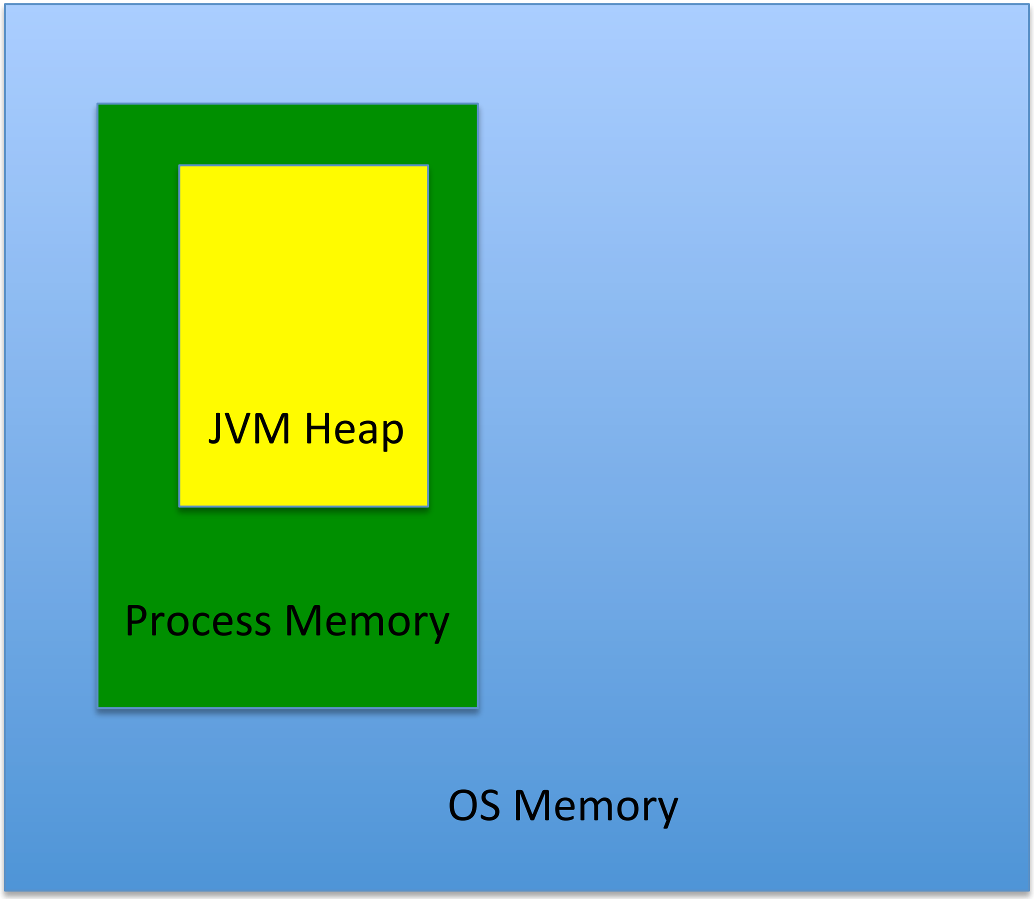
Traditional TCP/IP stack

OSv TCP/IP stack

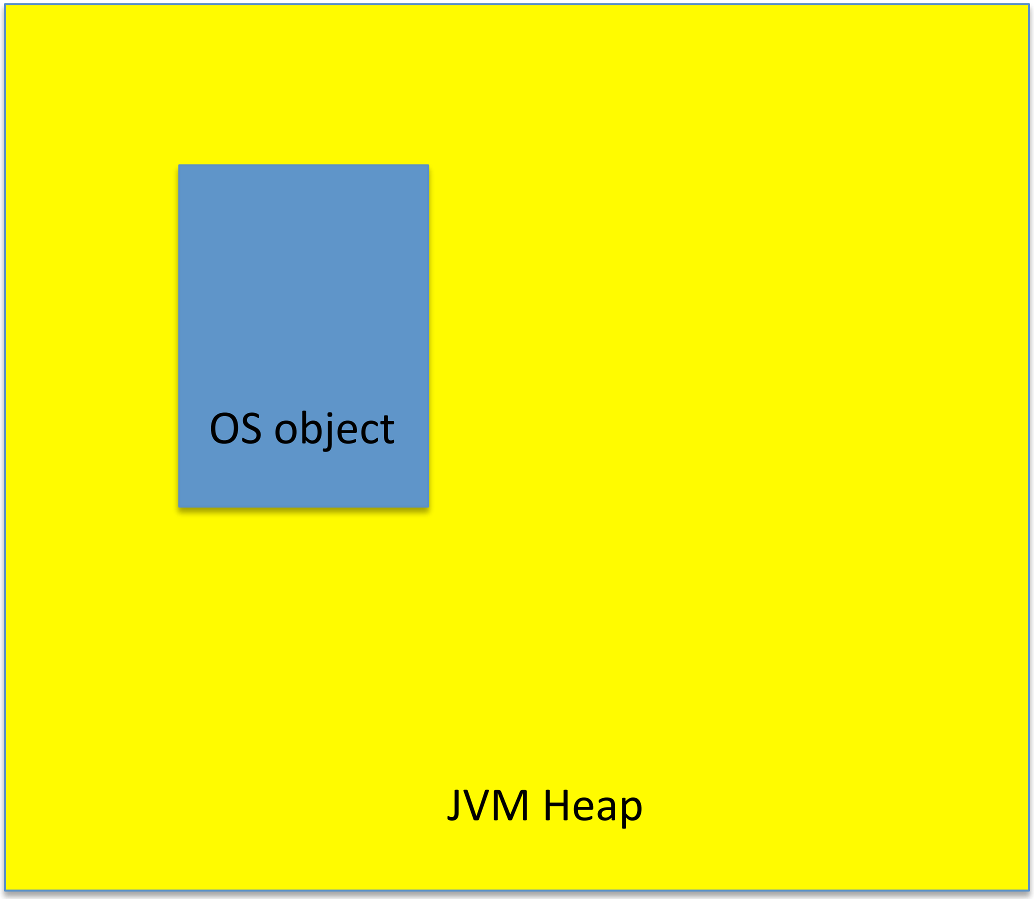
Memory management in UNIX



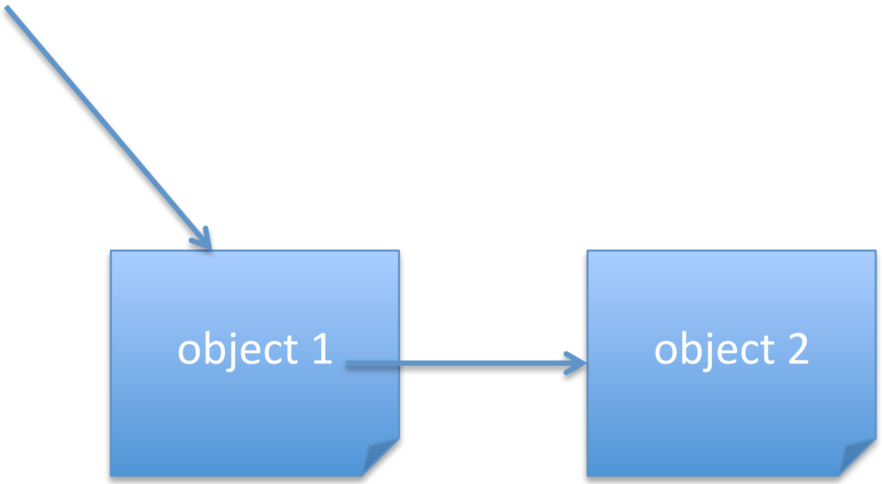
Memory management in OSv



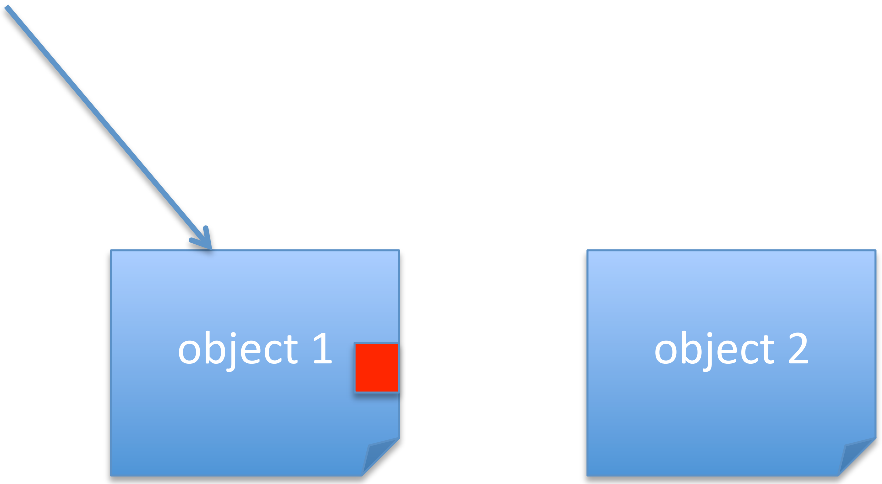
JVM balooning (no more -Xmx)



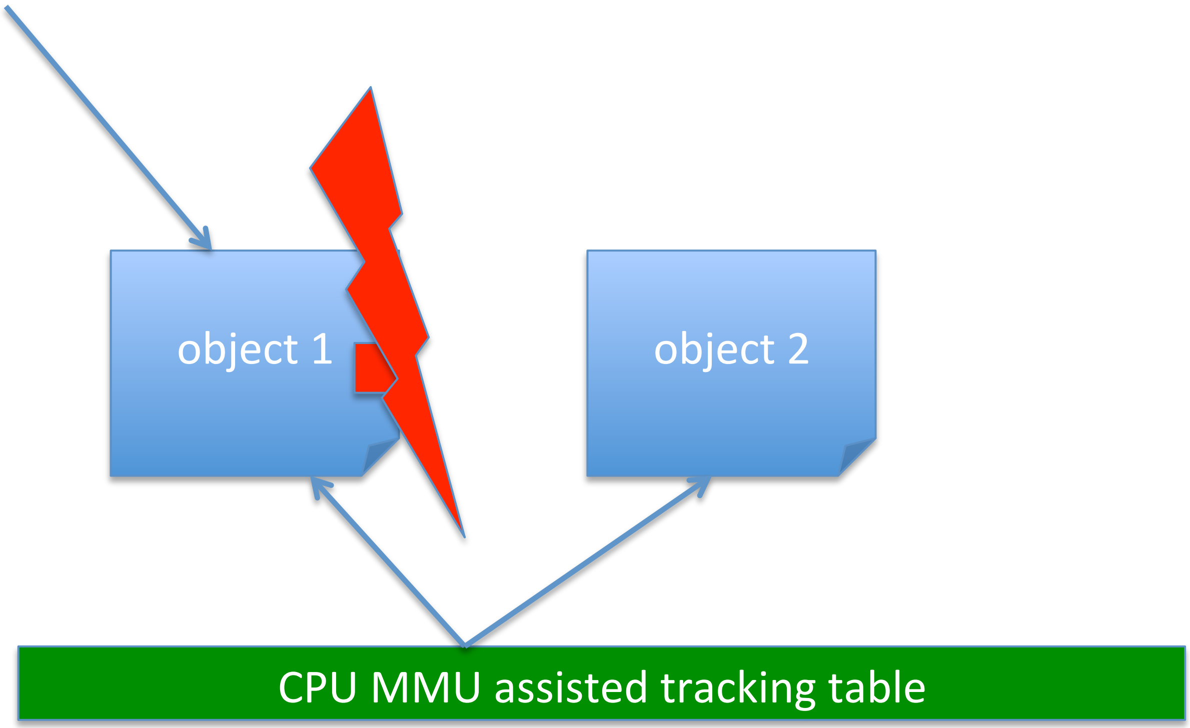
Turbo charging JVM GC



Turbo charging JVM GC



Turbo charging JVM GC



Apache Cassandra

- The good:
  - mostly working
  - tons of potential performance improvements
  - JVM balooning could do very nicely
- The bad:
  - performance on-par with bare metal
- The ugly:
  - OSv’s mmap() pecularities

Apache Zookeeper

- The good:
  - mostly working
  - no modifications: Bigtop rpm -> OSv image
- The bad:
  - no cycles to benchmark/validate
- The ugly:
  - no canonical image (Bigtop to the rescue!)

Apache Hadoop

- The good:
  - HDFS seems to be working
- The bad:
  - not sure what to do abot YARN
  - no cycles to benchmark/validate
- The ugly:
  - need to patch Hadoop common

Didn’t I tell you ‘no forking’?

From org.apache.hadoop.fs.DF:

```
public String getFilesystem() throws IOException {
-    if (Shell.WINDOWS) {
+    // if (Shell.WINDOWS) {
       this.filesystem = dirFile.getCanonicalPath().substring(0, 2);
       return this.filesystem;
-    } else {
-      run();
-      return filesystem;
-    }
+    // } else {
+    //  run();
+    //  return filesystem;
+    // }
  }
```

Apache HBase

- Next thing on the list
  - short of hiring an intern, not sure when it is happening
- First attempt at general Bigtop-based OSv packaging
- A case of 100% stateless application
- A good companion to Cassandra work

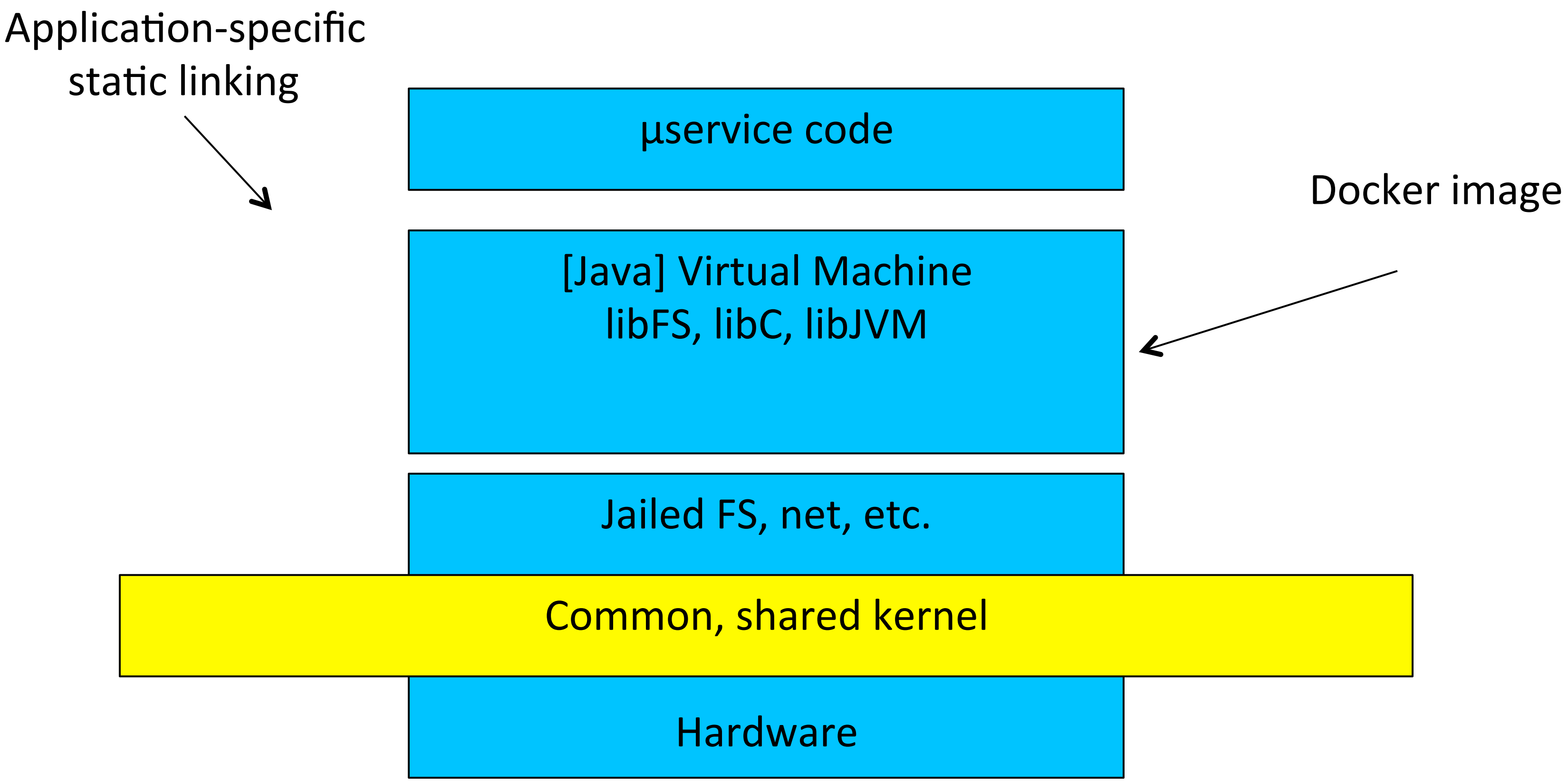
Bigtop's perspective

- "Apache Bigtop is to Hadoop what Debian is to Linux"
- Linux Packaging/Integration
  - init.d hooks (start/stop/restart)
- OSv Packaging/Integration
  - no way to prepare a java command line

Bigtop <3 Docker

- Assume 100% Docker integration
  - OSv as Docker “accelerator”
  - Universal containers
- IaaS players entering the market
  - Joyent’s Smart Data Center (now Open Source!)

But what about Docker?



Application-specific static linking

µservice code

Docker image

[Java] Virtual Machine
libFS, libC, libJVM

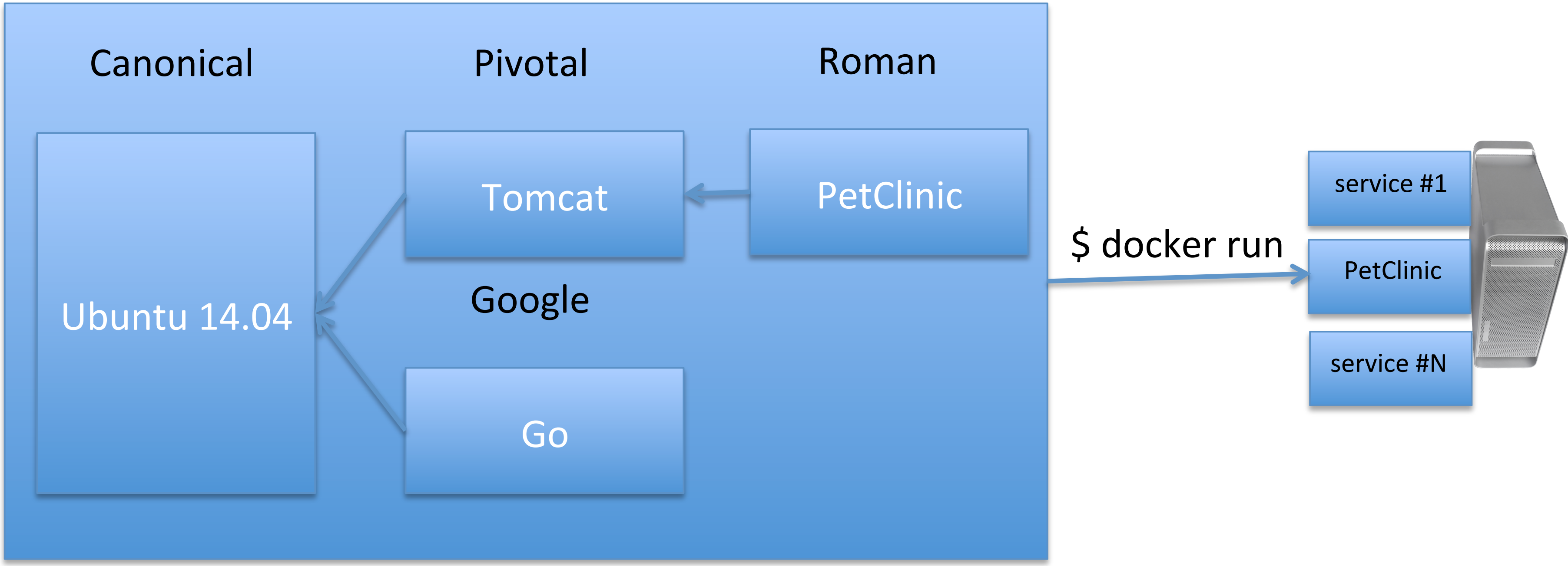
Jailed FS, net, etc.

Common, shared kernel

Hardware

Docker != LXC

- $ docker run roman/PetClinic
- Socially-driven image sharing



Bigtop as a universal packaging platform

- What is a package?
  - a [partial] image of a filesystem
- What is a docker container?
  - an image of a filesystem
- What is an OSv image?
  - an image of a filesystem (on-disk ZFS)

Why should it work this time?

- ~~Unikernels/exokernels back in ’90~~
- ~~JVM-on-bare-metal (Azul, BEA, etc.) back in ‘00~~
- Things they didn’t have back then
  - HW-assisted virtualization (KVM, XEN, etc.)
  - Elastic infrastructure oriented architectures
  - CloudFoundry

Elastic, next generation datacenter

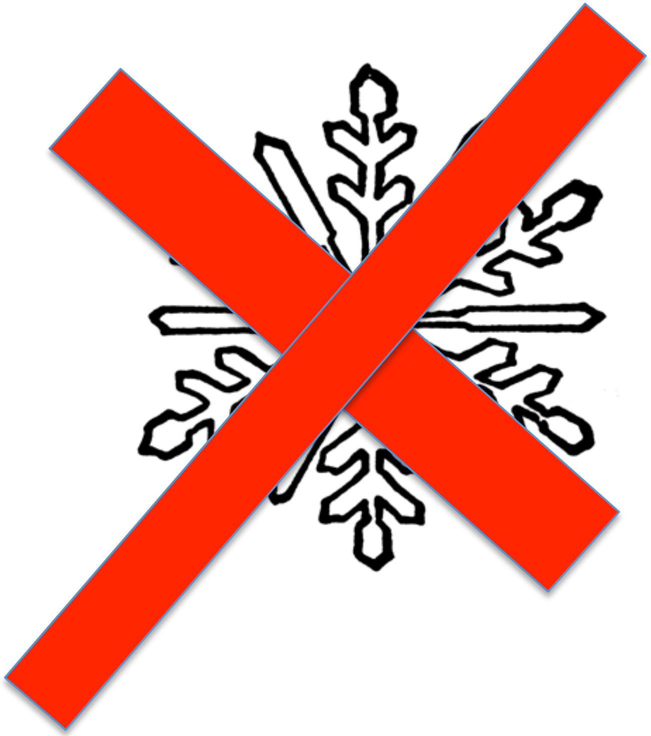
- Commodity, rack-provisioned Hardware
- Commodity, JeOS to get to Docker++
  - CoreOS, SmartOS
- Docker++ as a common backed
- OSv (really KVM, XEN)
- “GitHub” for μservies images

Finally killing DevOps

- Ops (IT) maintains the bare OS
- Devs maintain the images

Finally killing DevOps

- Ops (IT) maintains the bare OS
- Devs maintain the images

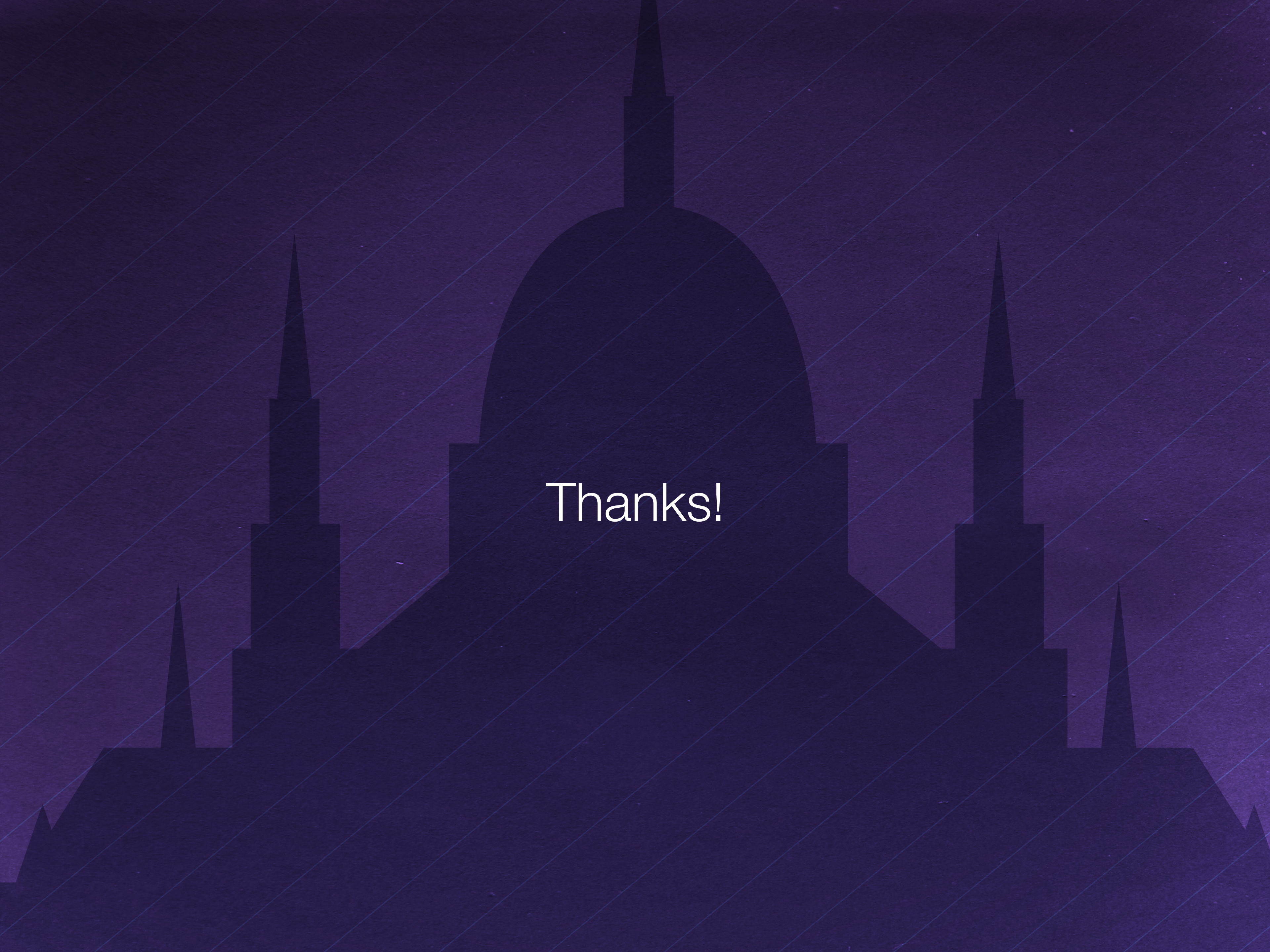


Questions?

By @cloud_opinion

Imagine no platforms
I wonder if you can
No need for PAAS or IAAS
A brotherhood of bare metal

Imagine there is no VM
It's easy if you try
No host below us
Above us only apps



Thanks!