

MULTIVERSE

Automatic Hybridization of Runtime Systems

Kyle C. Hale, Conor Hetland, Peter Dinda



Northwestern
University

HOBBS
xstack.sandia.gov/hobbes

ILLINOIS INSTITUTE
OF TECHNOLOGY 

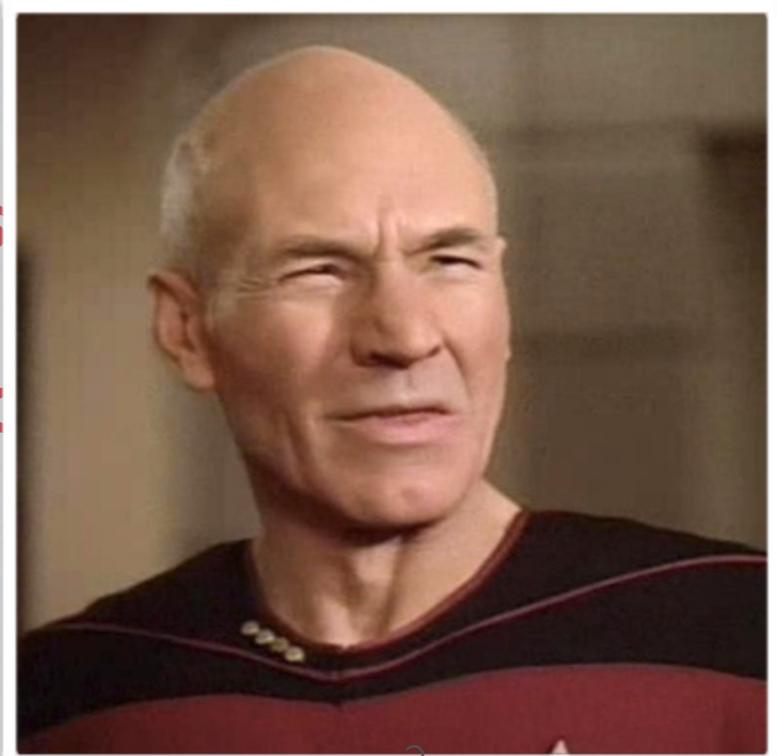
HYBRID PARALLEL RUNTIMES

LONG-TERM GOAL:

let's reimagine systems S/W stack for parallel runtimes
(esp. for high-level parallel languages)

why in the world?

imposed abs
avoid duplicated func
explore



W capabilities
runtime system and OS
g things!

A THOUGHT EXPERIMENT



2 ASSUMPTIONS

1. YOU ARE A **HARDCORE OS HACKER**

expert **parallel programmer**

elite **performance engineer**

you can squash bugs like no one's business

you have **masochistic tendencies**

2 ASSUMPTIONS

2. YOU HAVE **INFINITE TIME** ON YOUR
HANDS

NOW, GO BUILD A PARALLEL RUNTIME SYSTEM

*on raw hardware



HOW **DIFFERENT** IS YOUR RUNTIME
FROM A TYPICAL OS KERNEL?

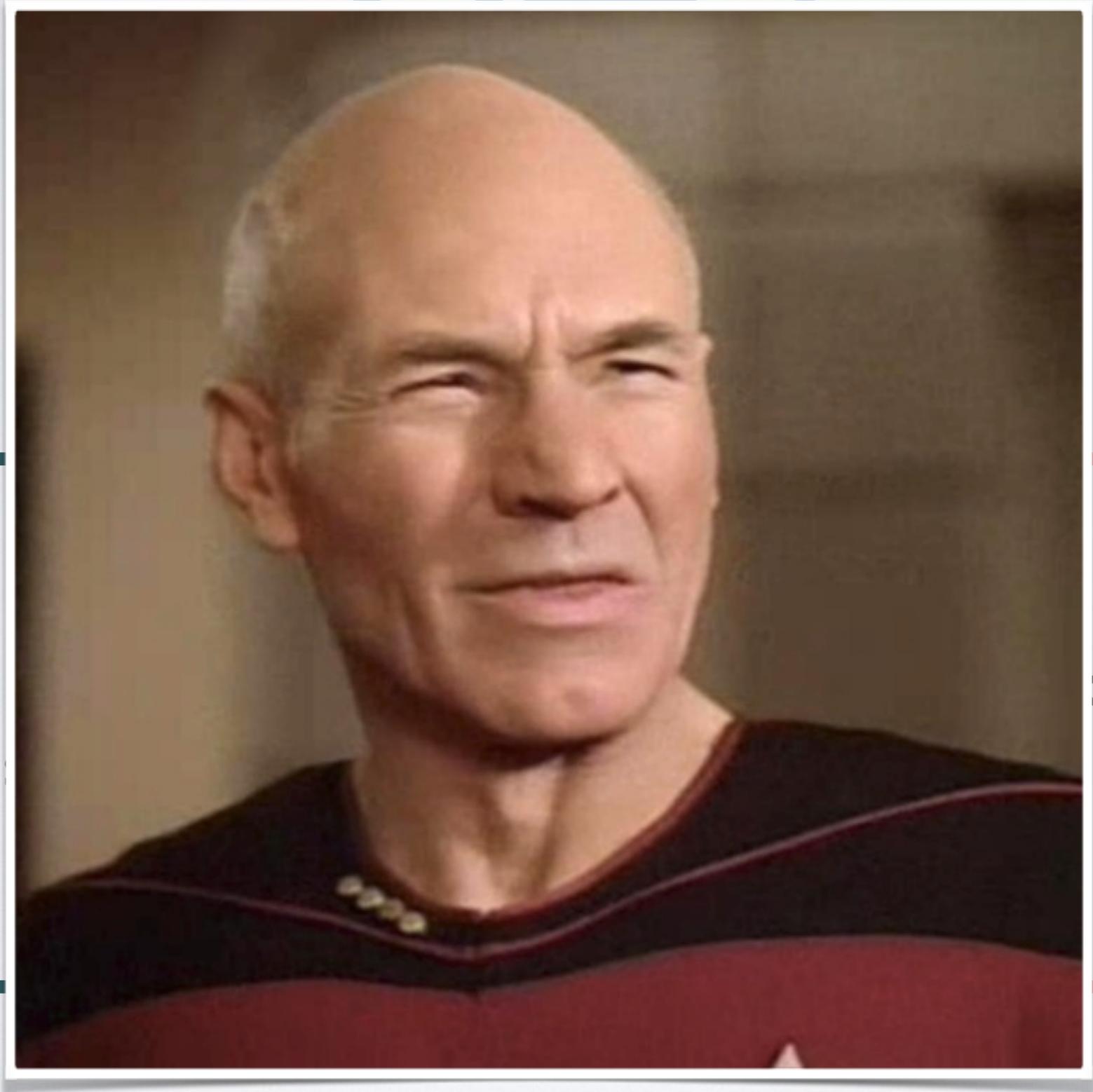
[Hale et al., HPDC '15]

[Hale et al., VEE '16]

PROBLEM:

you've **already built a runtime for Linux**

building it from scratch is hard



user-

SO F
I

OKAY, LESS AMBITIOUS

I'll give you a kernel framework

Now you just have to **port**, probably **add**
some functionality

***instead of starting from
scratch, we can port to
kernel mode:***

DIFFICULT

TIME-CONSUMING

ERROR-PRONE

development cycle:

do {

ADD FUNCTION

REBUILD

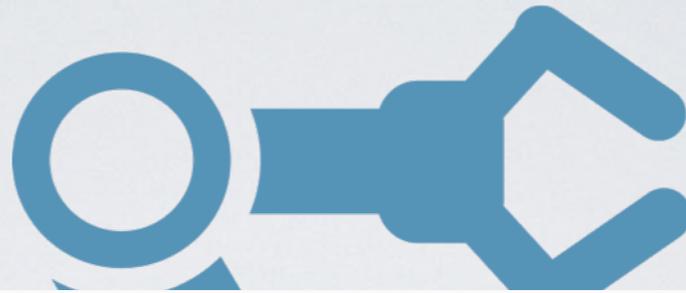
BOOT specialized kernel

} while (kernel falls over)

FURTHERMORE

*much of the
functionality is*

NOT ON THE CRITICAL PATH



user-s

ed OS
nel



MULTIVERSE

C/C++ source tree for
runtime system

point Multiverse to runtime's
source tree

rebuild

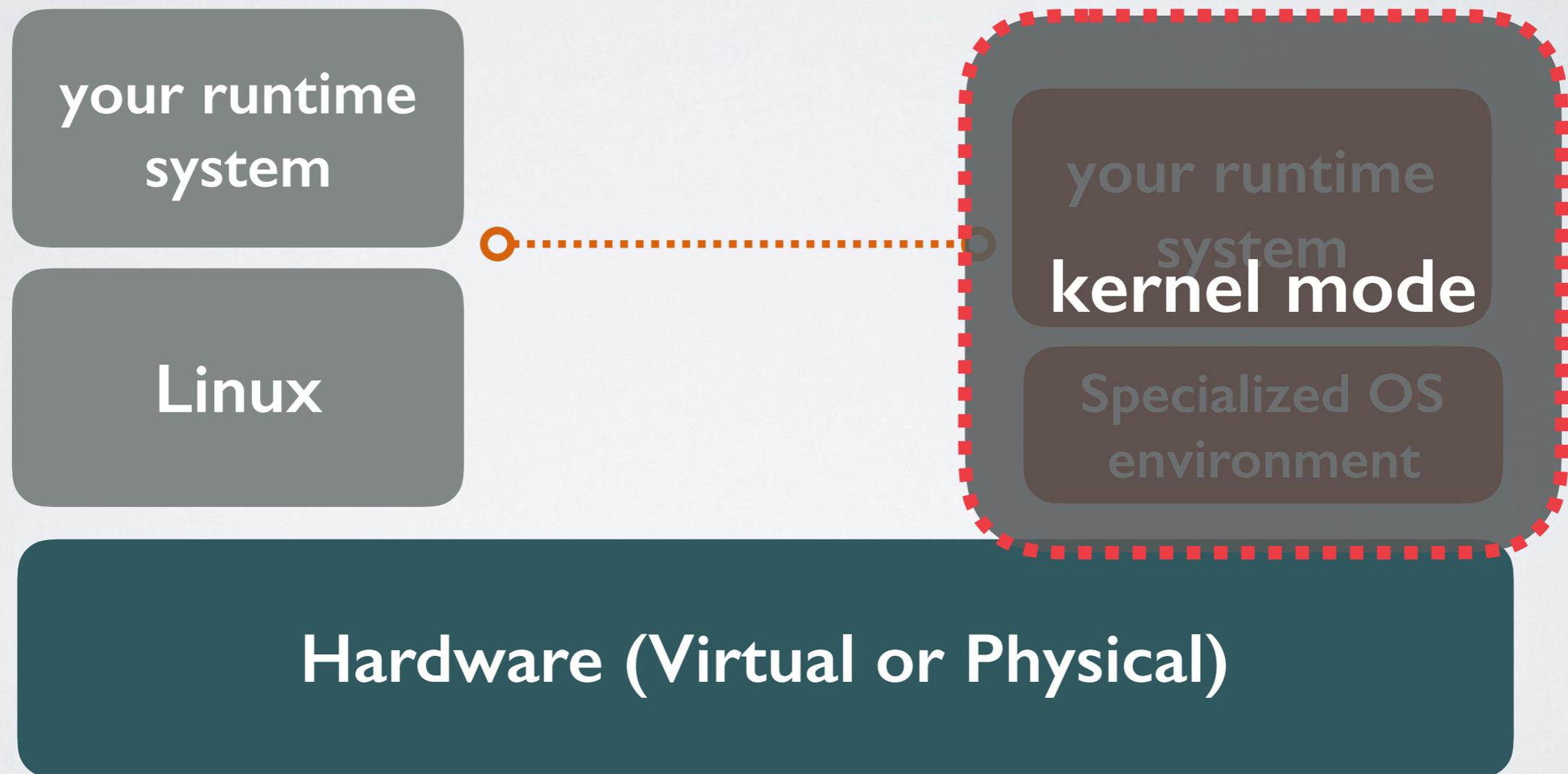
Automatic Hybridization
run it (boots as kernel)

identify hotspots

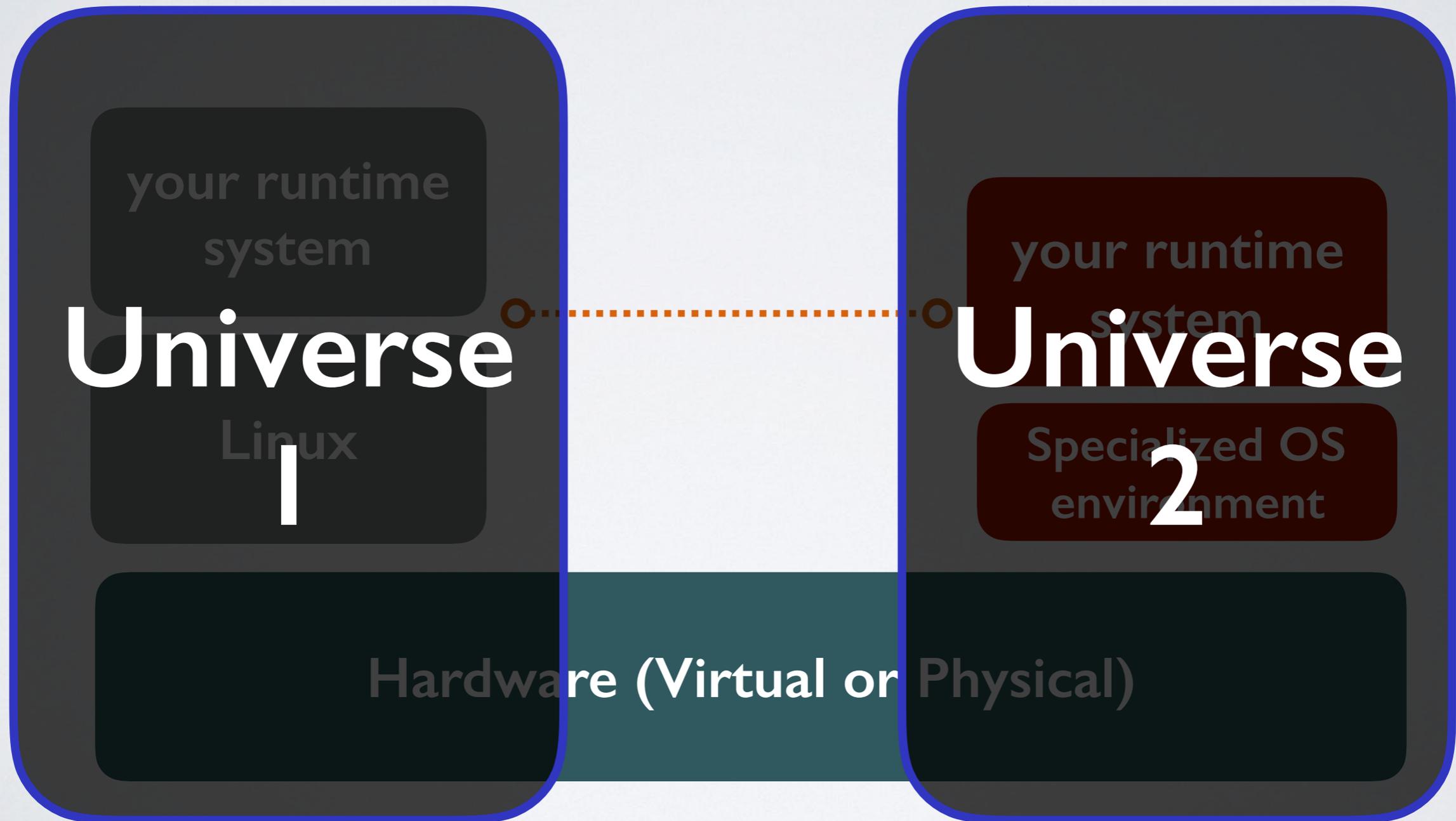
bring functionality into your
kernel

WHAT YOU GET

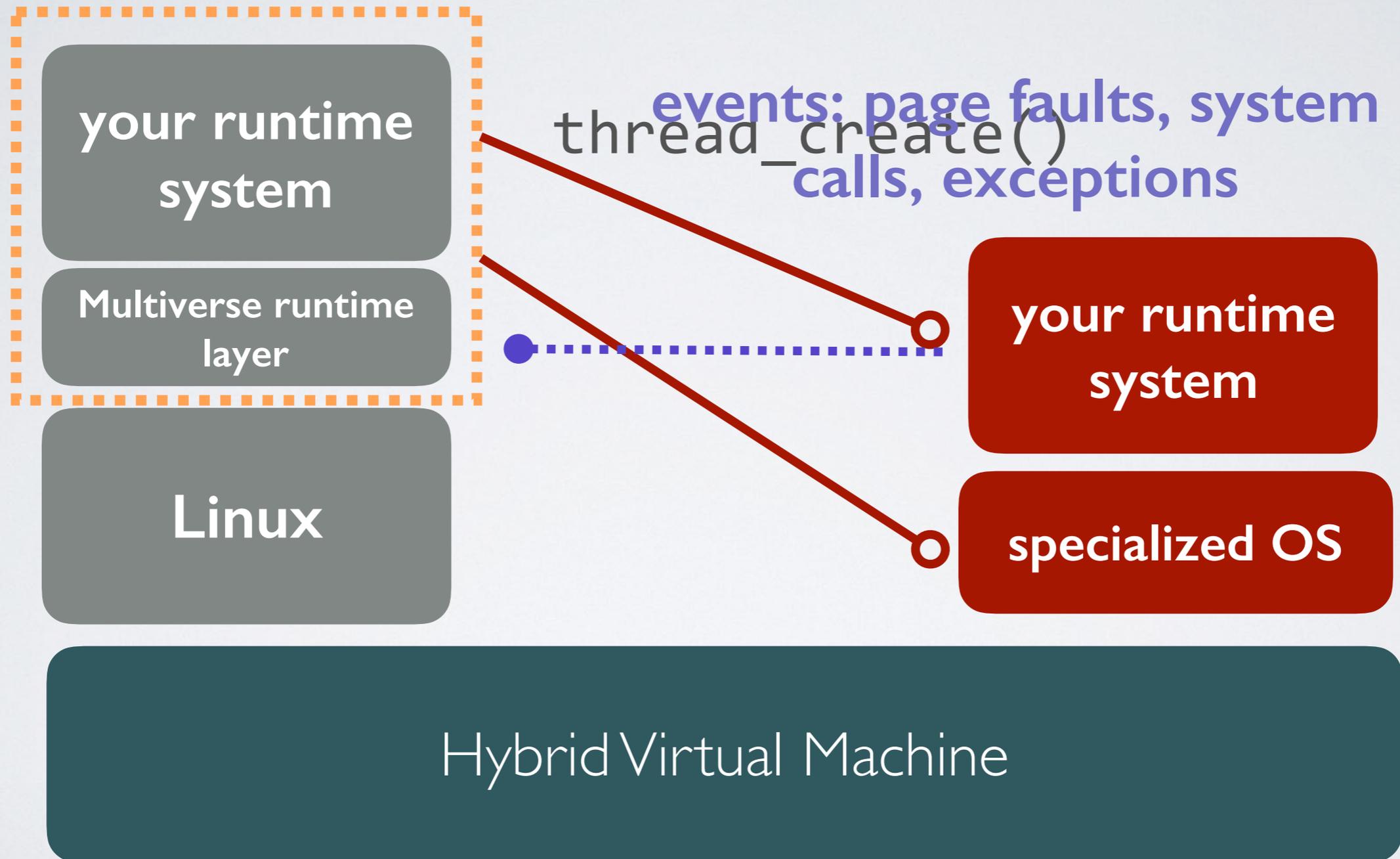
leverage general-purpose OS for legacy functionality



MULTIVERSE IMPLEMENTS AUTOMATIC HYBRIDIZATION

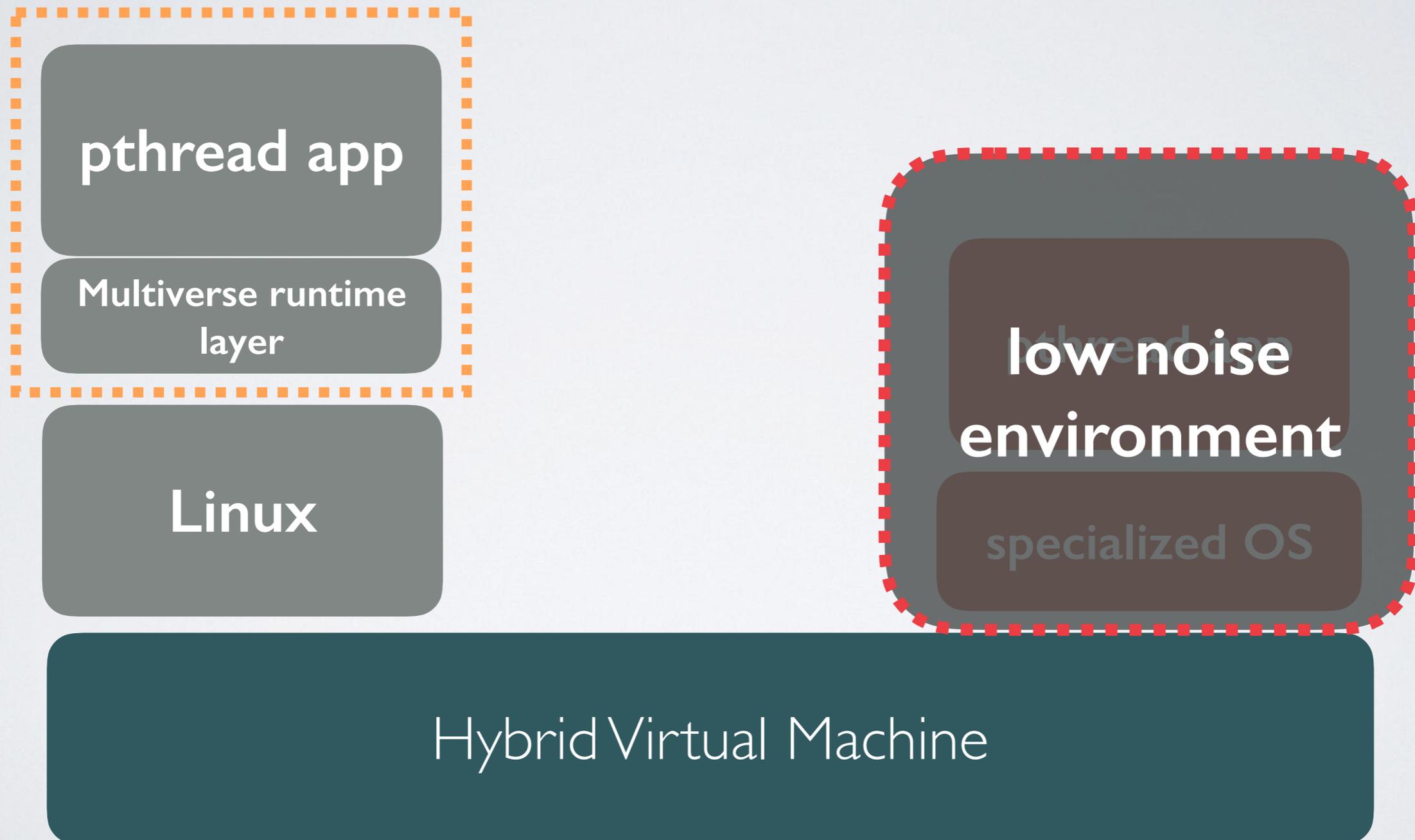


ON TOP OF VIRTUALIZATION



FREEBIE: WORKS WITH OUT
OF THE BOX PTHREADS APPS

FREEBIE: WORKS WITH OUT OF THE BOX PTHREADS APPS



RACKET



most widely used Scheme implementation

runtime system: **800K lines of code**

complex, JIT-based runtime, managed
memory, garbage collection, etc.

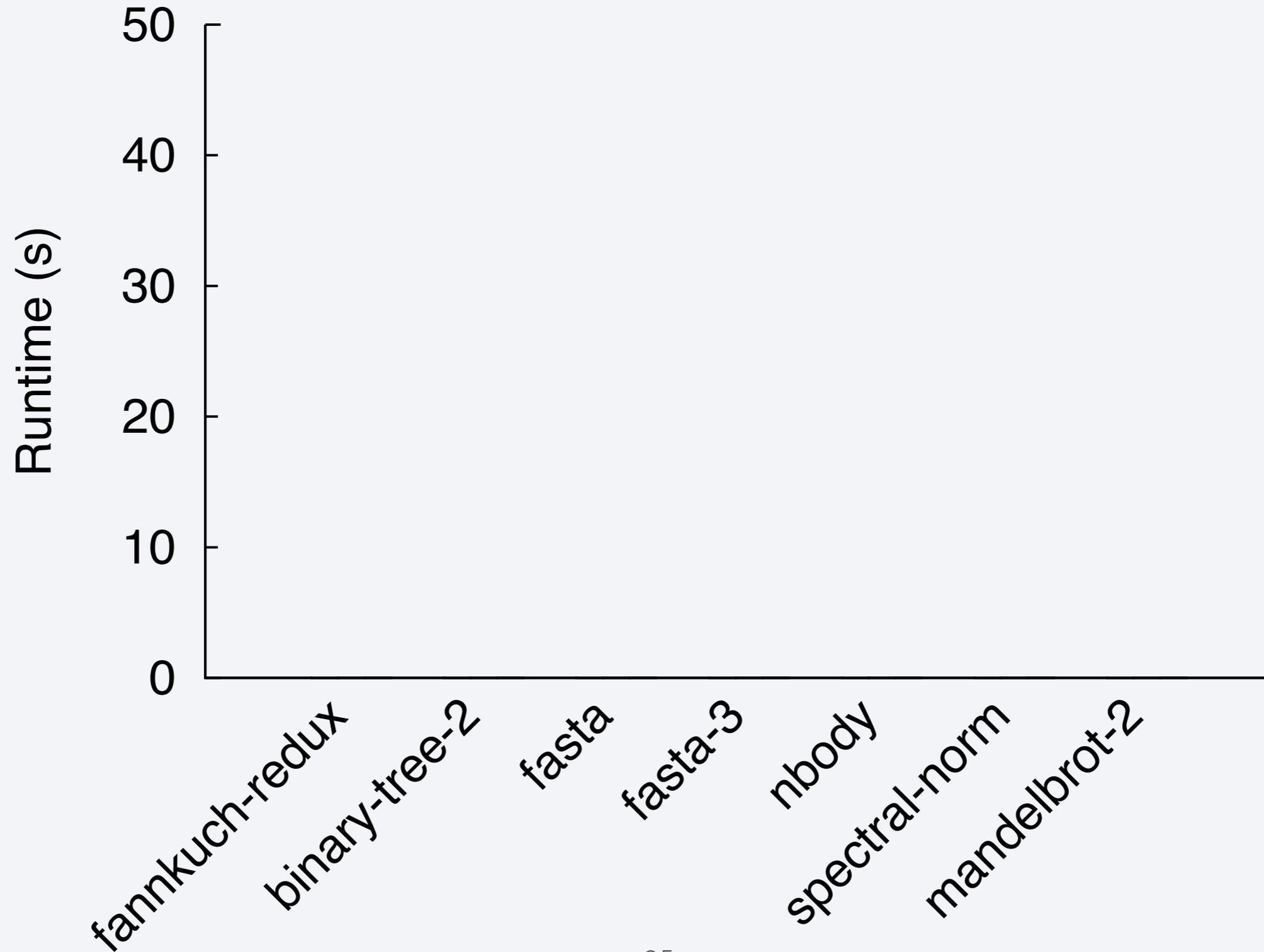
**complexity makes it good option for testing
Multiverse**

```
# 1S
bench-write.out      go          mracket-GOLD
binary-tree-2.rkt   intsum-native
bytes                ism
collects            isn
doall.sh             lgn-hpcg
doruns.sh            lgo
fannkuch-redux.rkt lost+found
fasta-3.rkt          lpm
fasta.rkt            lpn
g                    mandelbrot-2.rkt
#
```

}

```
#  
# ls  
bench-write.out      go      mracket-GOLD  
binary-tree-2.rkt   intsum-native  multiverse-racket  
bytes               ism     multiverse.log  
collects           isn     nbody.rkt  
doall.sh           lgn-hpcg    racket  
doruns.sh          lgo     results  
fannkuch-redux.rkt lost+found  spectral-norm.rkt  
fasta-3.rkt        lpm     test.out  
fasta.rkt          lpn     test.t  
g                 mandelbrot-2.rkt  
# █
```

LOW OVERHEADS



MULTIVERSE

generate a **specialized OS kernel (HRT)** from a Linux user-space app/runtime system by rebuilding

hybridized runtime is **bridged with Linux** to provide legacy functionality (e.g. system calls)

very slight overheads (**100s of ns**) for forwarded events, little effect on performance overall

THANKS

my website: <http://halek.co>

Nautilus Aerokernel Framework: <http://nautilus.halek.co>

Prescience Lab: presciencelab.org

Hobbes Project: xstack.sandia.gov/hobbes



HOBBS
xstack.sandia.gov/hobbes



Northwestern
University

ILLINOIS INSTITUTE
OF TECHNOLOGY 

BACKUPS

HOBBES

xstack.sandia.gov/hobbes



nautilus.halek.co



Palacios

An OS Independent Embeddable VMM

v3vee.org/palacios

