A CASE FOR TRANSFORMING PARALLEL RUNTIMES **INTO OS KERNELS**

Kyle Hale *Peter Dinda*

halek.co
v3vee.org
presciencelab.org
xstack.sandia.gov/hobbes

NORTHWESTERN
UNIVERSITY

HOBBES

xstack.sandia.gov/hobbes

V3VEE

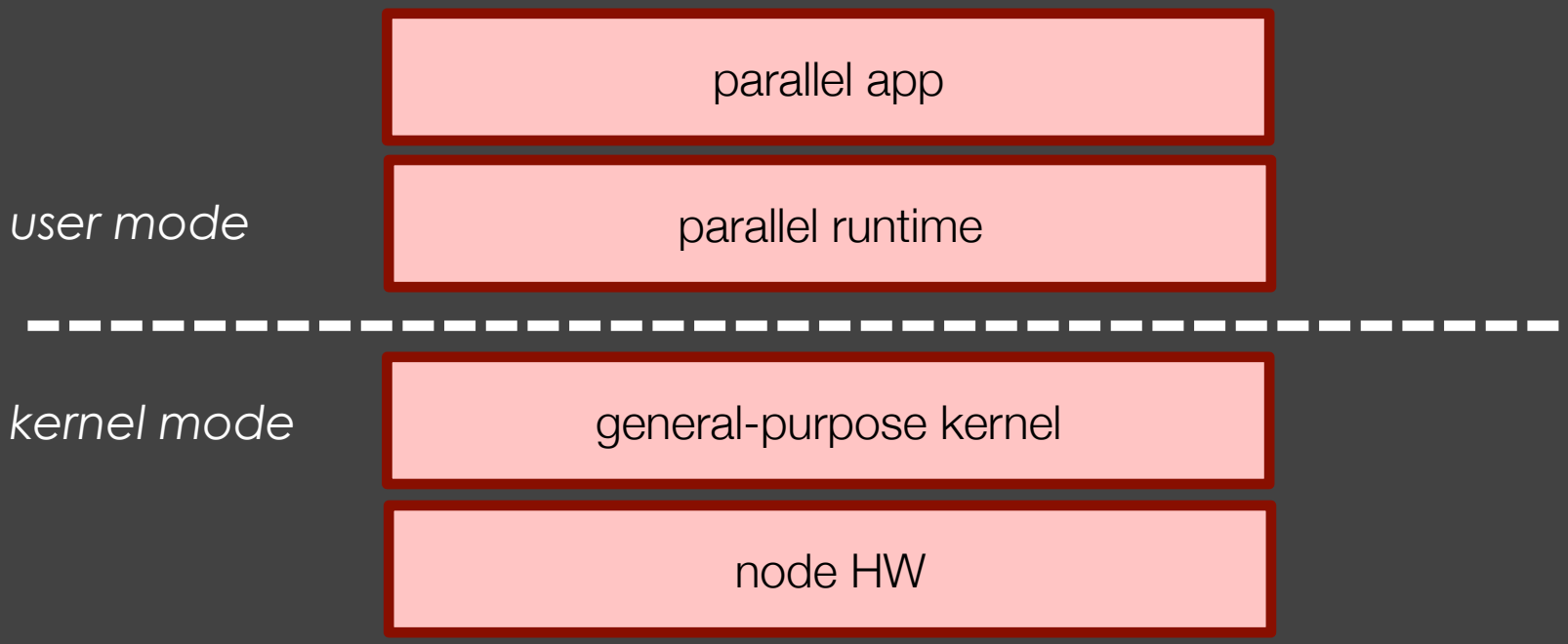
v3vee.org

Palacios

An OS Independent Embeddable VMM

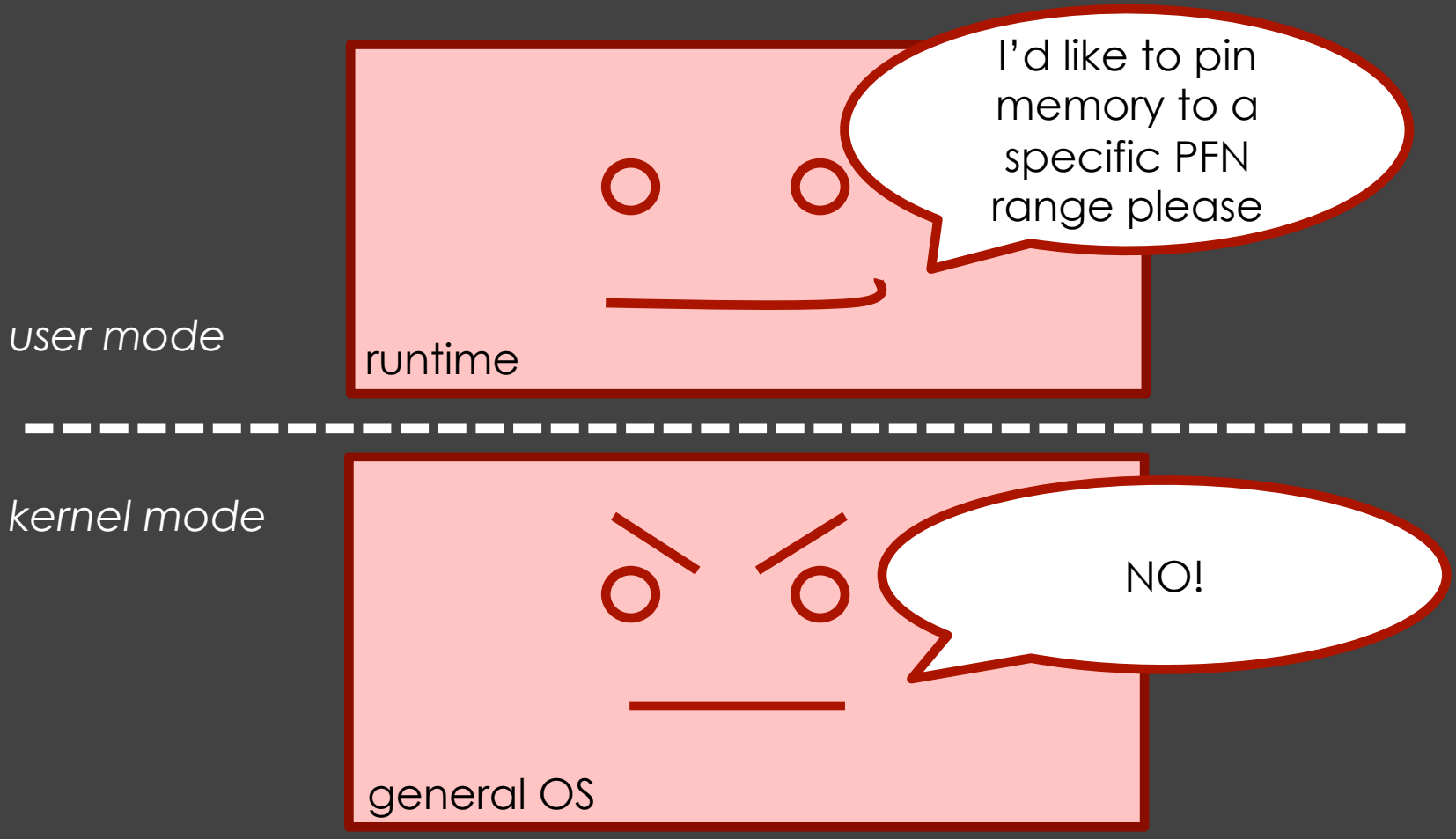
v3vee.org/palacios

THE CURRENT OS/RUNTIME MODEL



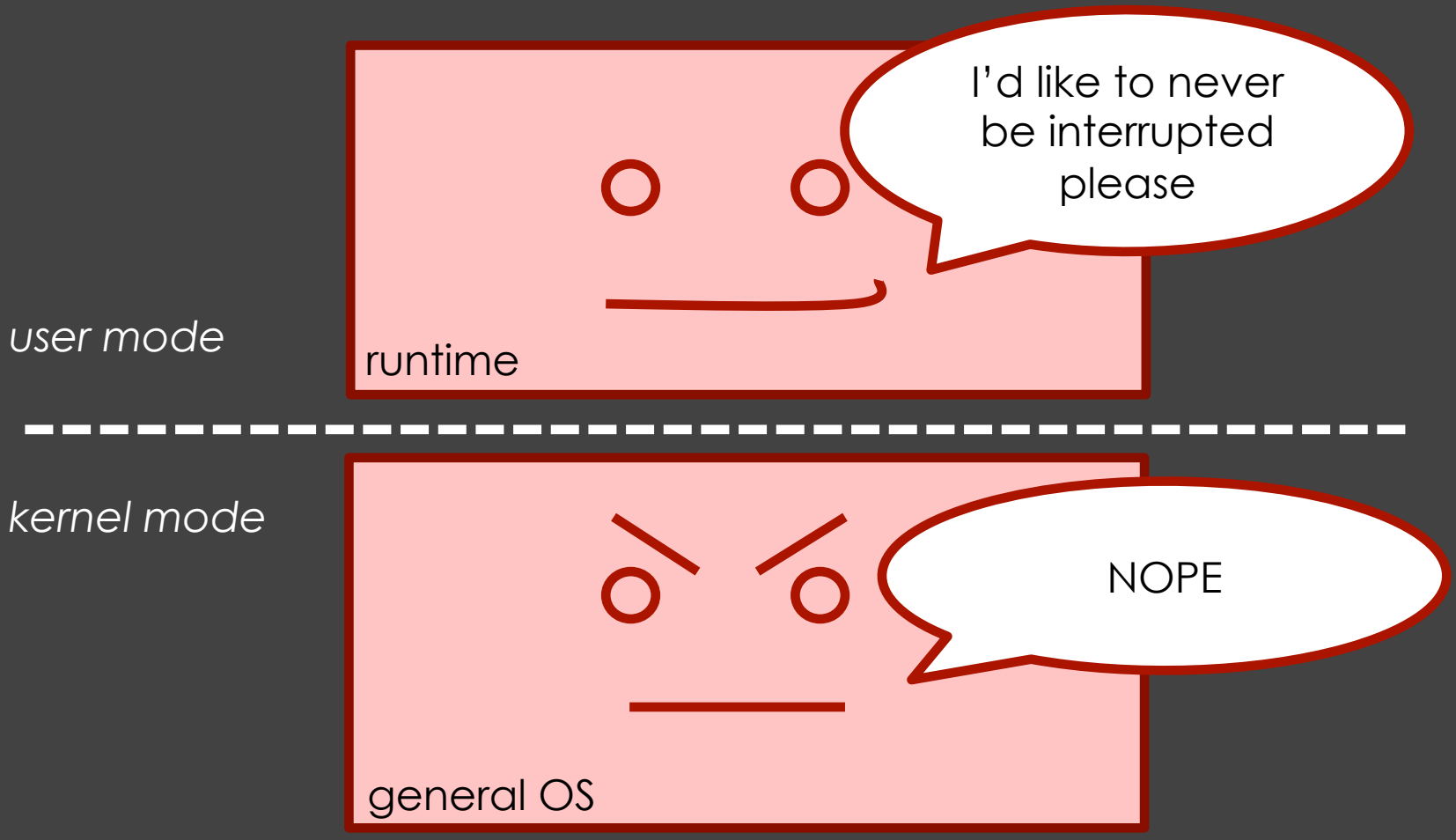
THIS MODEL HAS SOME ISSUES

ARE PROVIDED KERNEL ABSTRACTIONS THE RIGHT ONES?



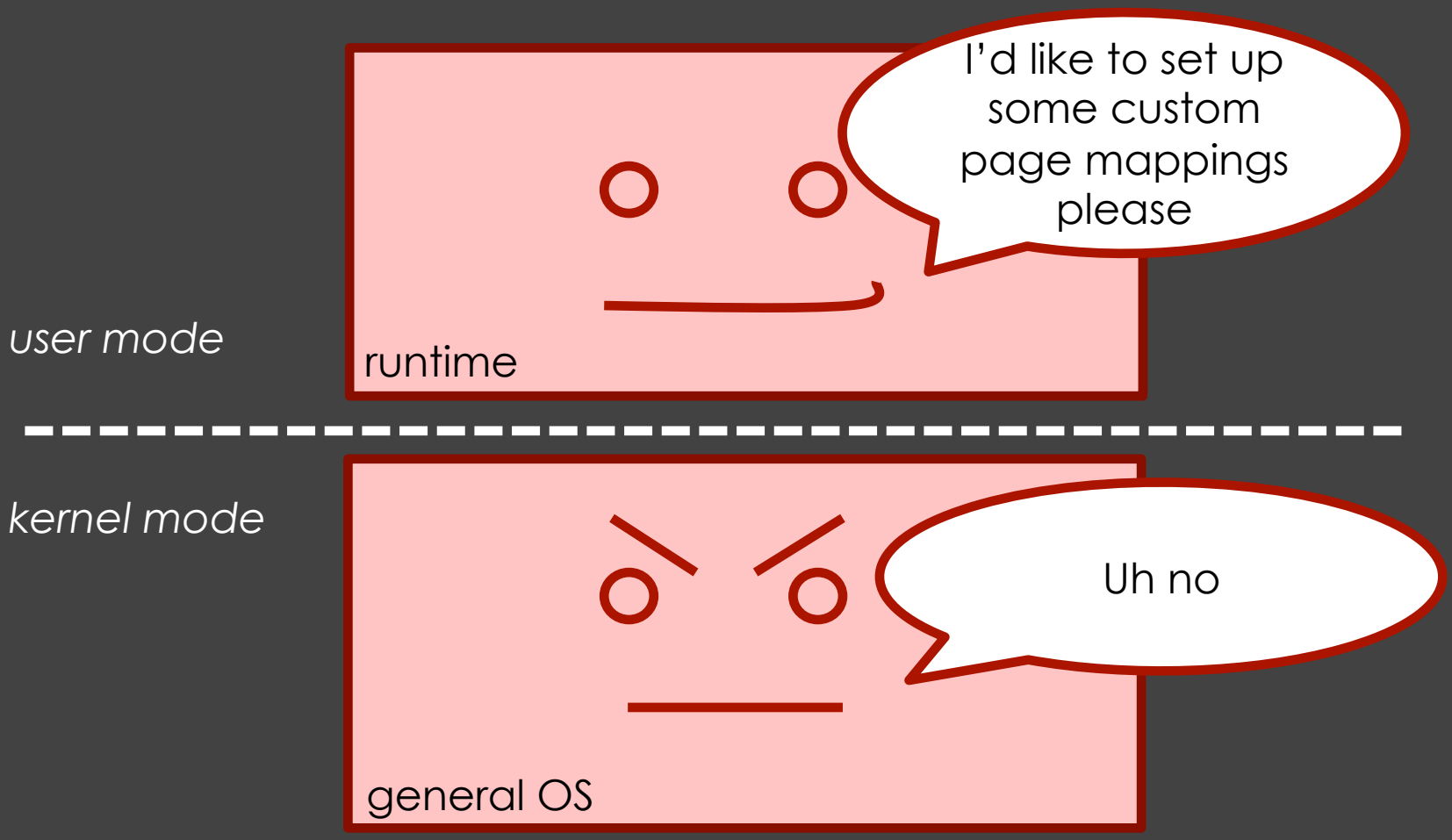
NOT ALWAYS

ARE PROVIDED KERNEL ABSTRACTIONS THE RIGHT ONES?

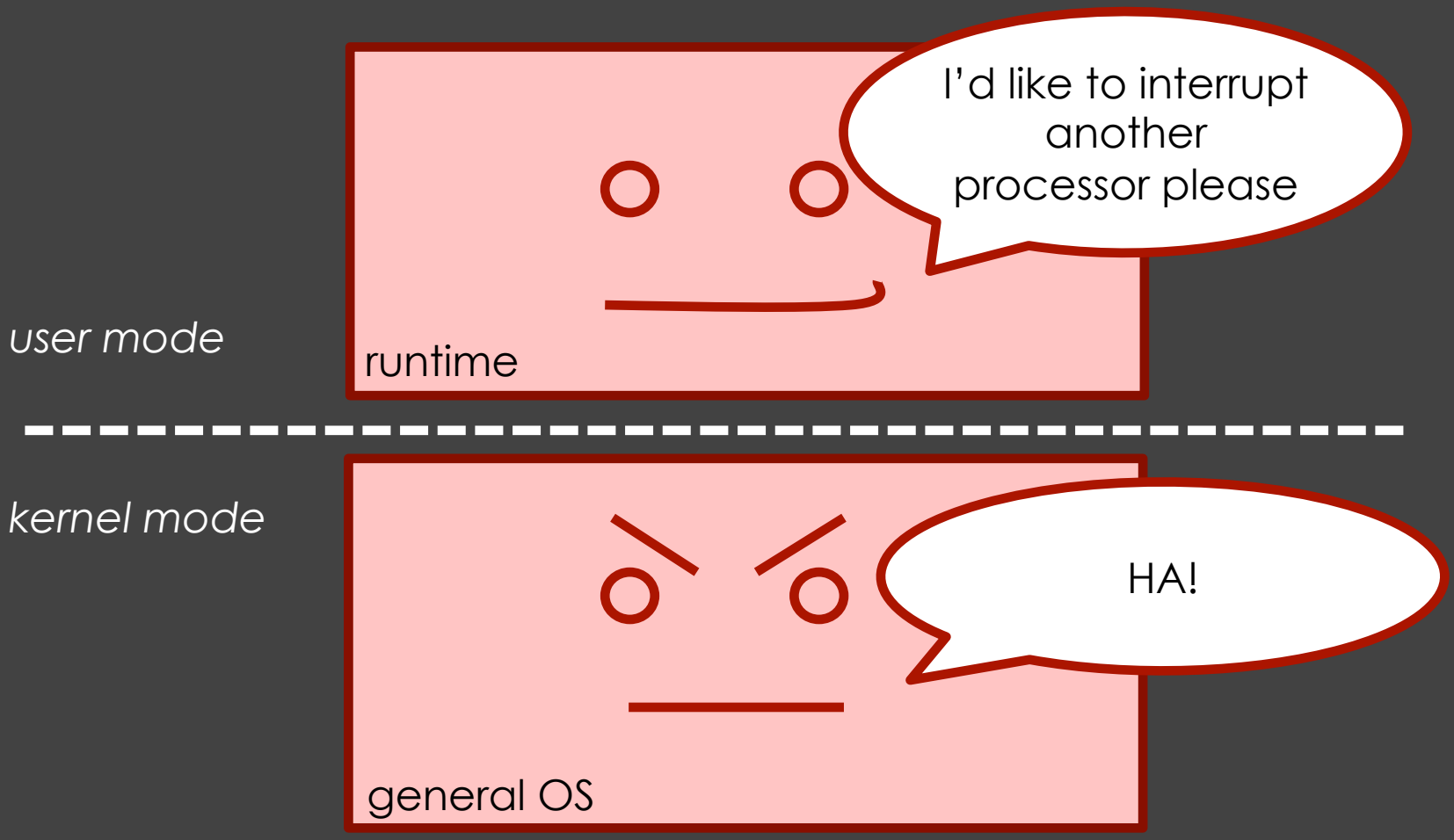


NOT ALWAYS

RESTRICTED ACCESS TO HARDWARE



RESTRICTED ACCESS TO HARDWARE



What are the consequences?

What are the consequences?

WORKAROUNDS & COMPROMISES

What are the consequences?

WORKAROUNDS & COMPROMISES

DUPLICATED FUNCTIONALITY

If runtime had

we could mitigate these issues

If runtime had

FULL HARDWARE ACCESS

we could mitigate these issues

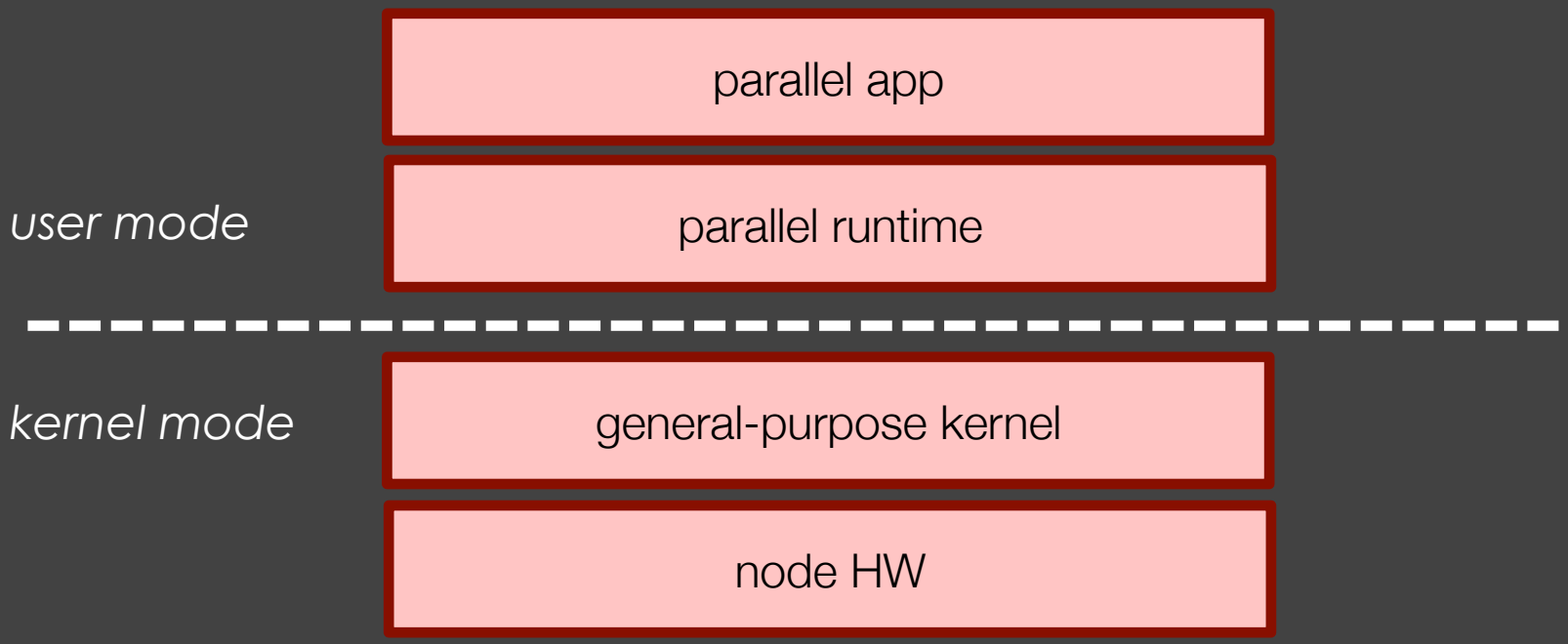
If runtime had

FULL HARDWARE ACCESS

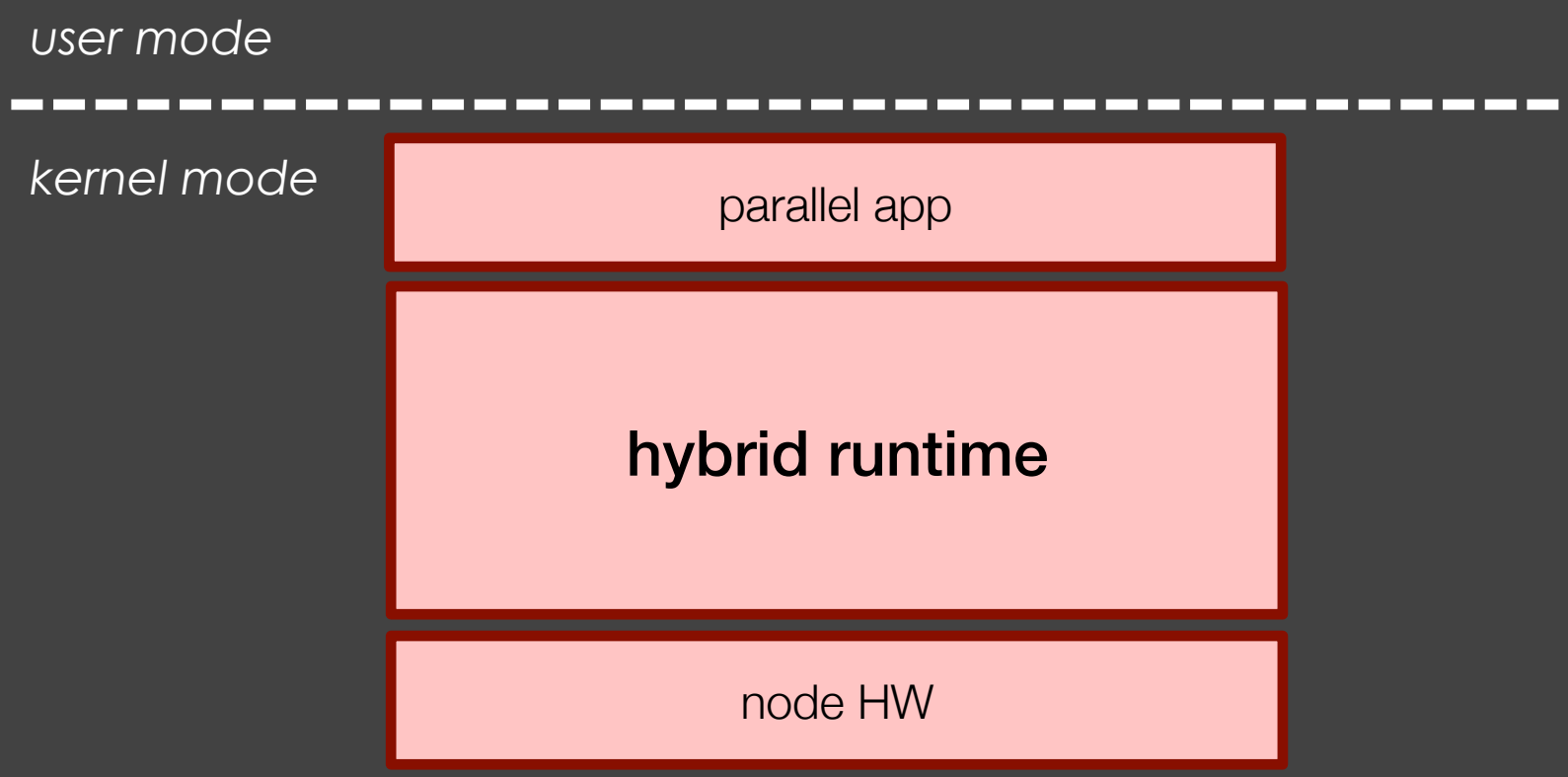
CONTROL OVER KERNEL ABSTRACTIONS

we could mitigate these issues

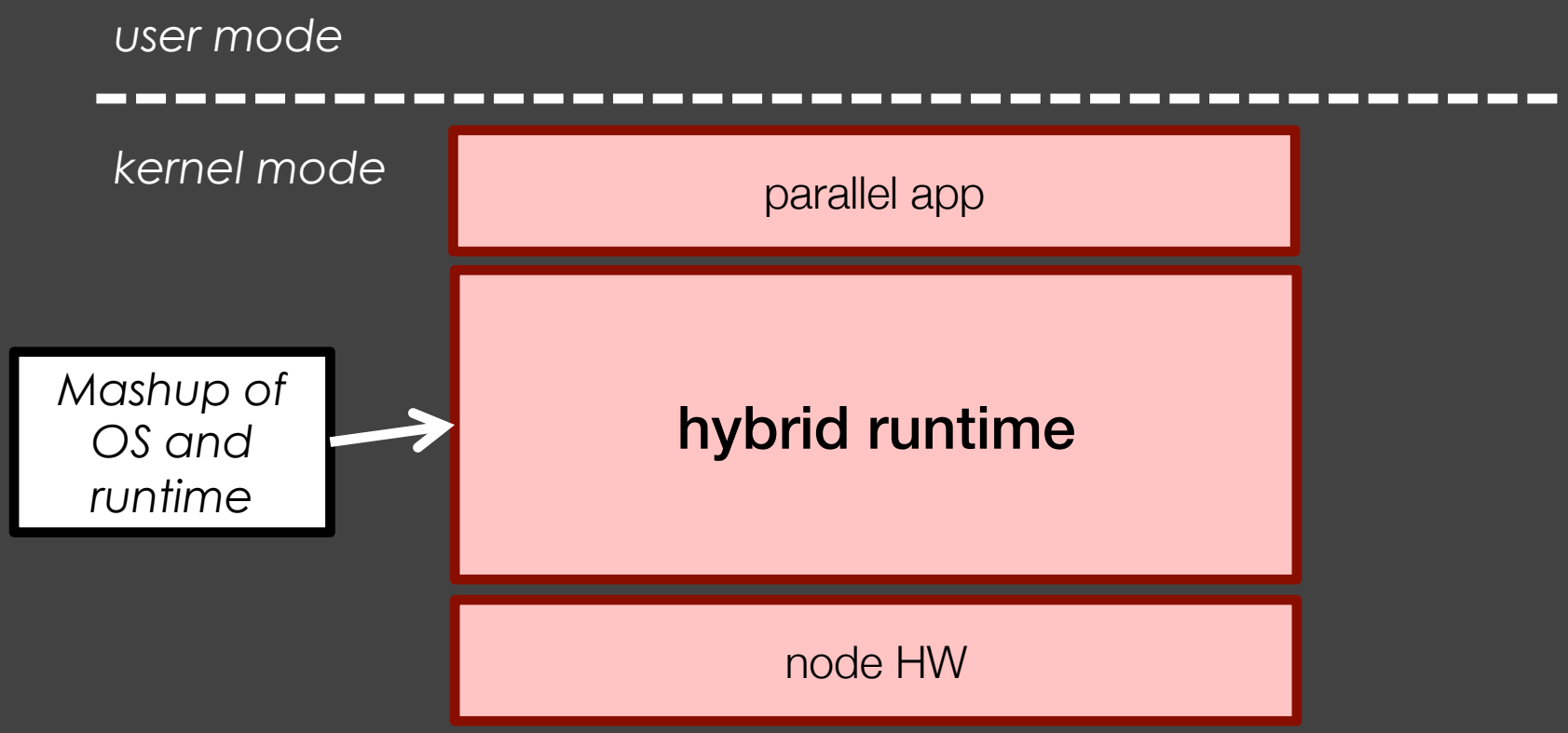
THE CURRENT OS/RUNTIME MODEL



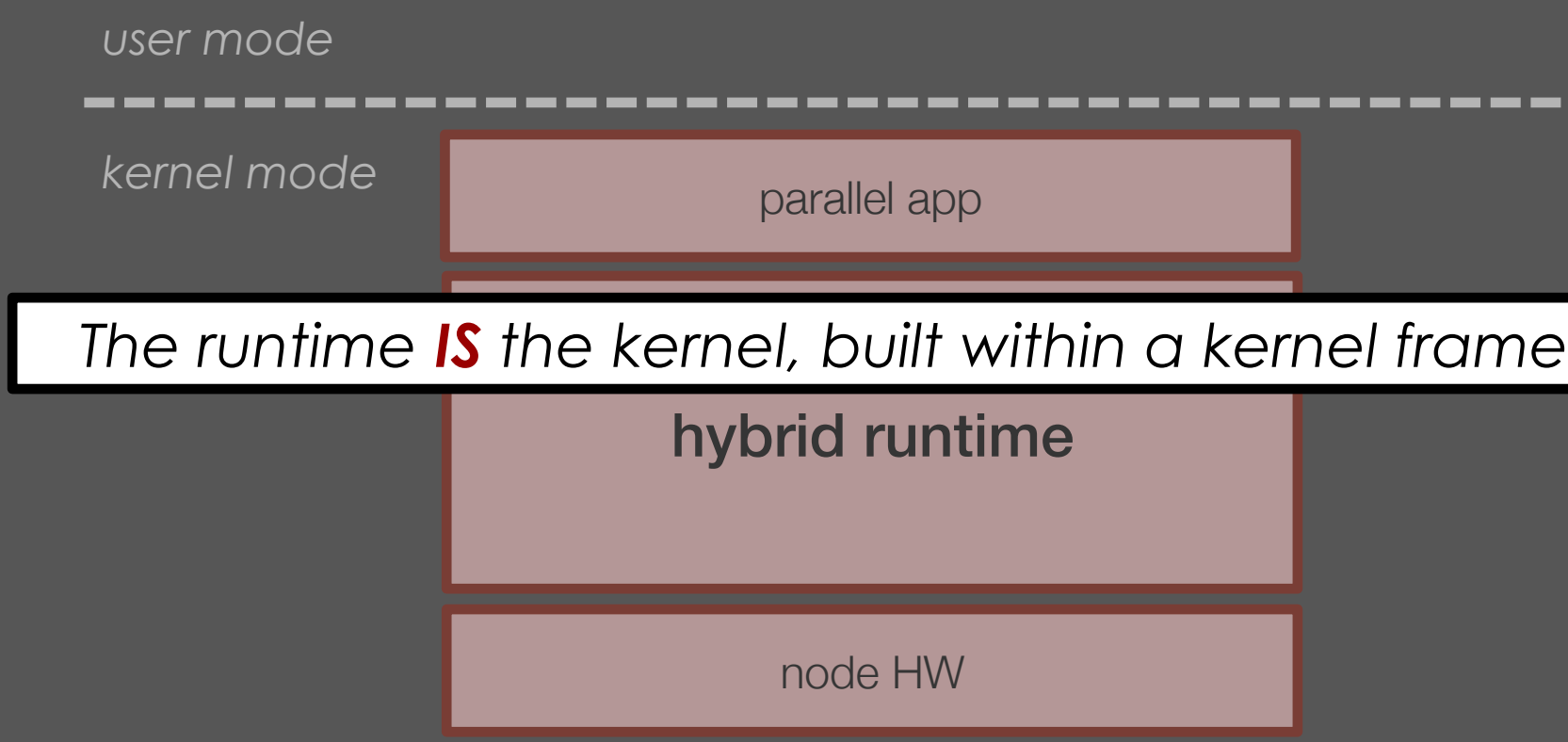
OUR PROPOSED MODEL:
THE HYBRID RUNTIME (HRT)



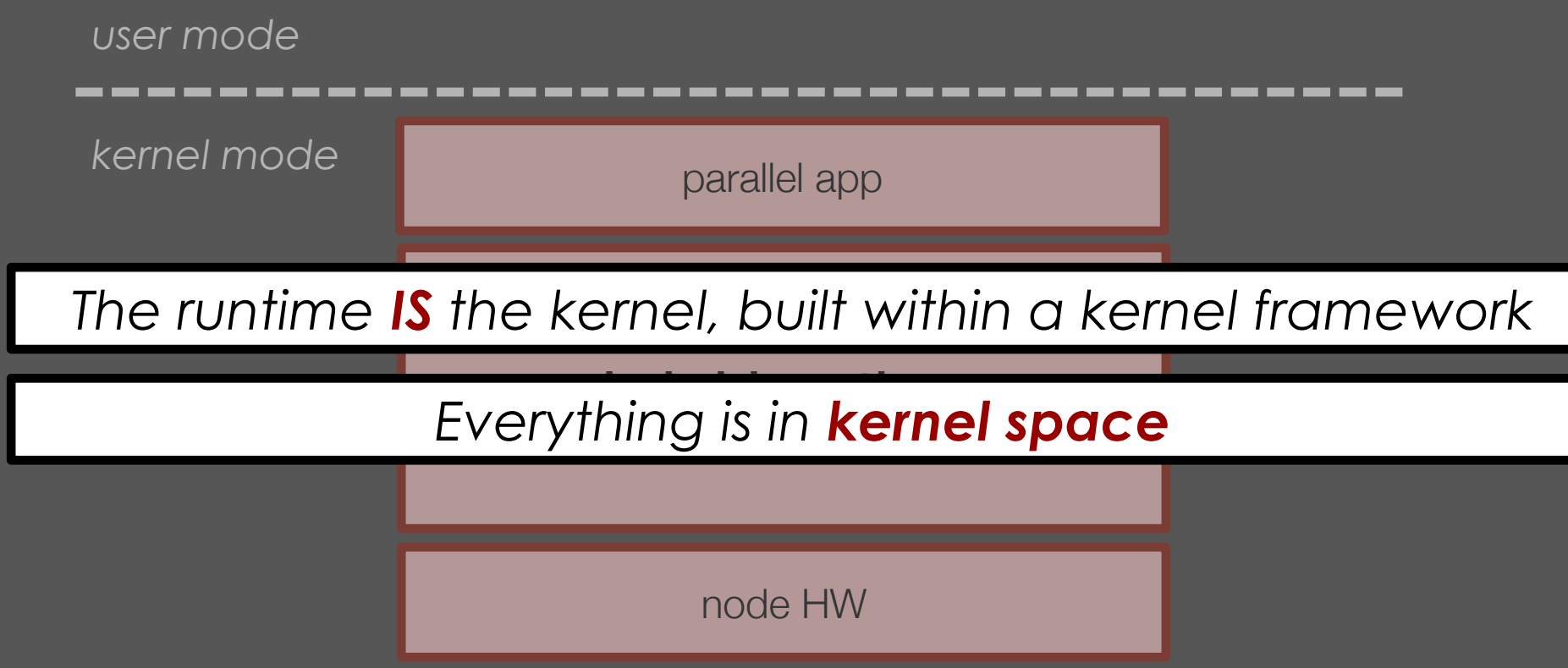
OUR PROPOSED MODEL:
THE HYBRID RUNTIME (HRT)



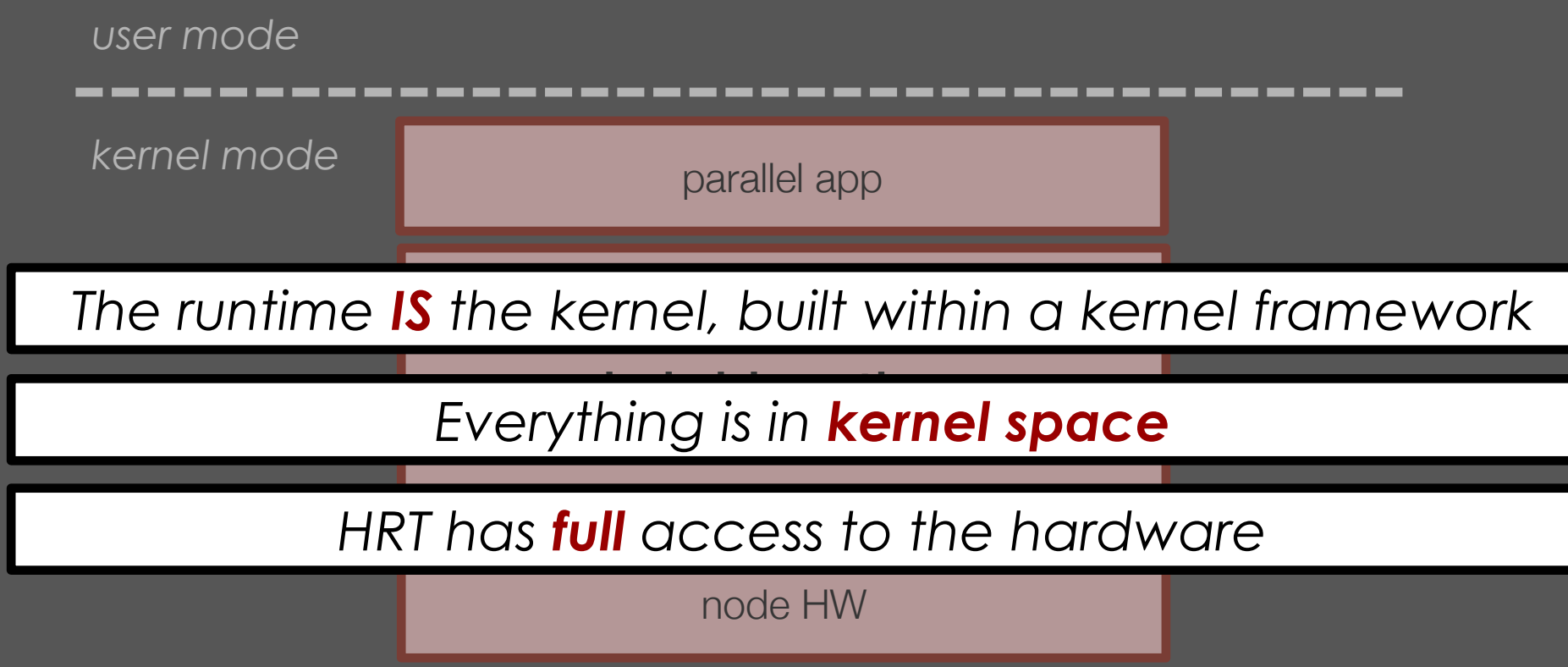
OUR PROPOSED MODEL:
THE HYBRID RUNTIME (HRT)



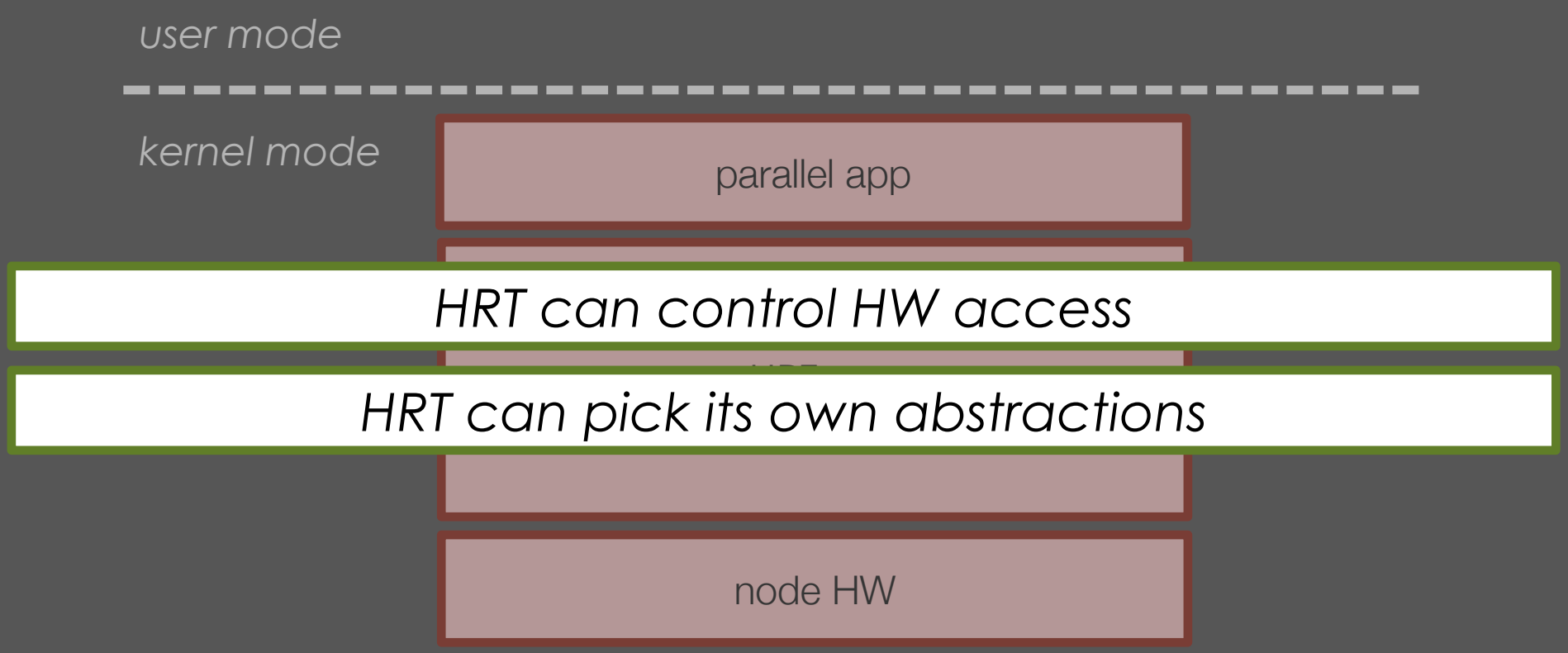
OUR PROPOSED MODEL:
THE HYBRID RUNTIME (HRT)



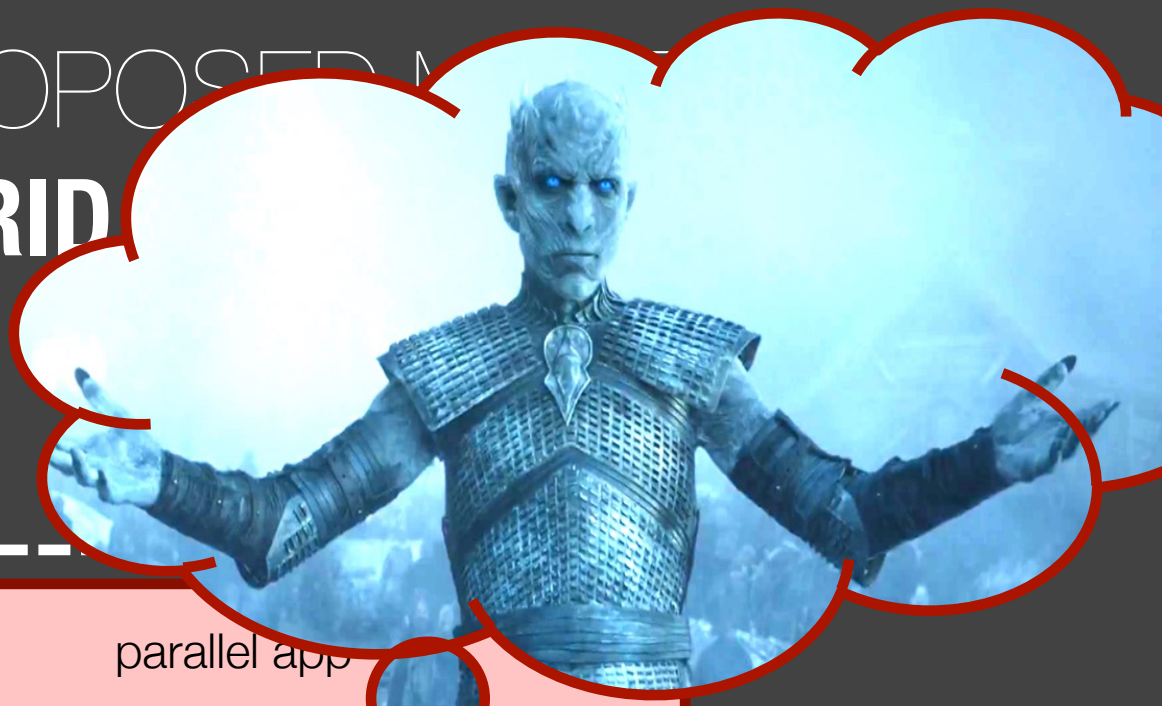
OUR PROPOSED MODEL:
THE HYBRID RUNTIME (HRT)



OUR PROPOSED MODEL:
THE HYBRID RUNTIME (HRT)



OUR PROPOSED M
THE HYBRID



user mode

kernel mode

parallel app

HRT

MORE POWER!

We built a kernel framework to support HRTs

We built a kernel framework to support HRTs

NAUTILUS

We built a kernel framework to support HRTs

NAUTILUS

We ported an existing, complex parallel runtime

We built a kernel framework to support HRTs

NAUTILUS

We ported an existing, complex parallel runtime

LEGION
legion.stanford.edu

We built a kernel framework to support HRTs

NAUTILUS

We ported an existing, complex parallel runtime

LEGION
legion.stanford.edu

We ported our framework to cutting-edge many-core hardware

We built a kernel framework to support HRTs

NAUTILUS

We ported an existing, complex parallel runtime

LEGION
legion.stanford.edu

We ported our framework to cutting-edge many-core hardware

XEON PHI

We built a kernel framework to support HRTs

NAUTILUS

We ported an existing, complex parallel runtime

LEGION
legion.stanford.edu

We ported our framework to cutting-edge many-core hardware

XEON PHI

We evaluated our port on a standard HPC benchmark

We built a kernel framework to support HRTs

NAUTILUS

We ported an existing, complex parallel runtime

LEGION
legion.stanford.edu

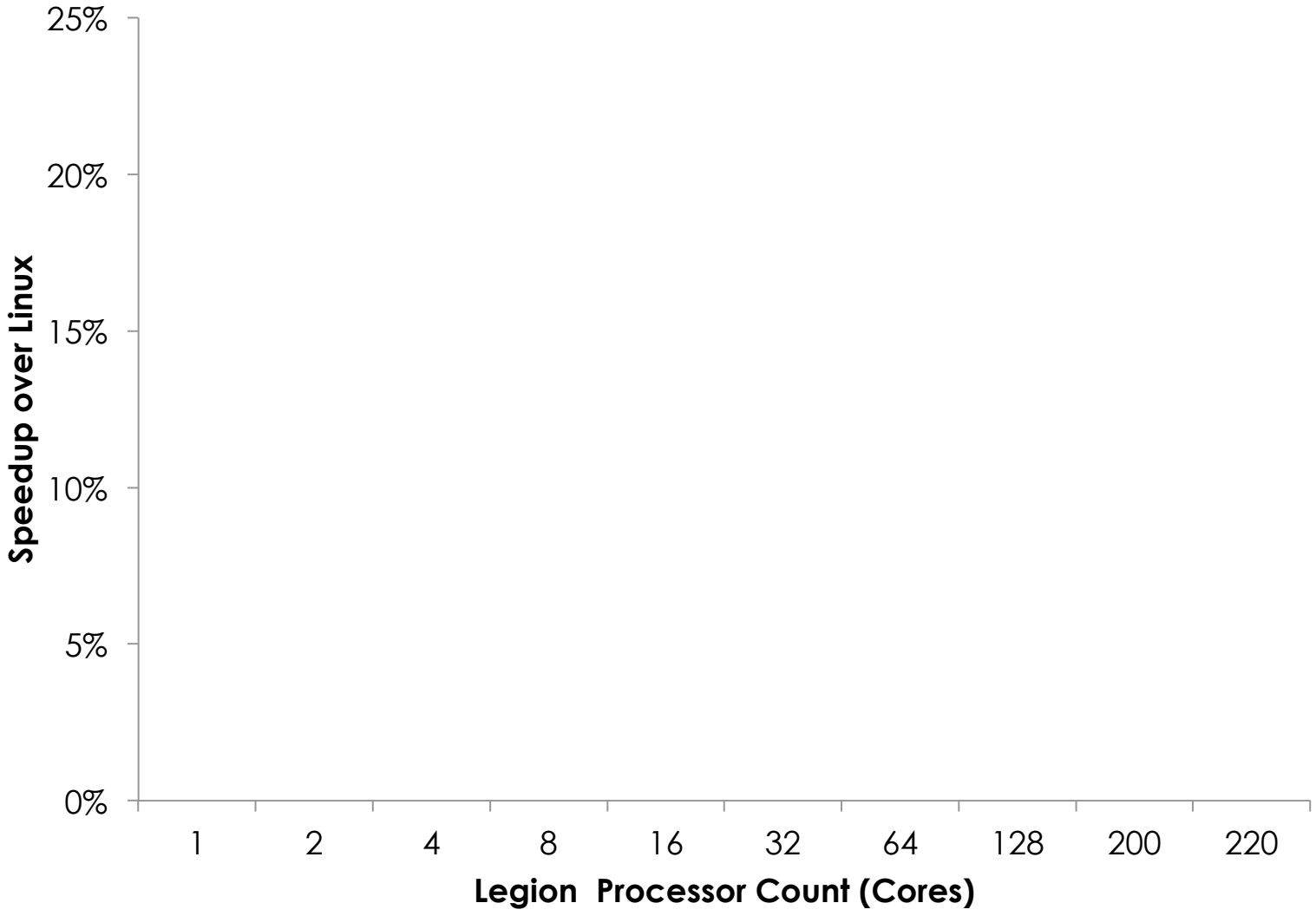
We ported our framework to cutting-edge many-core hardware

XEON PHI

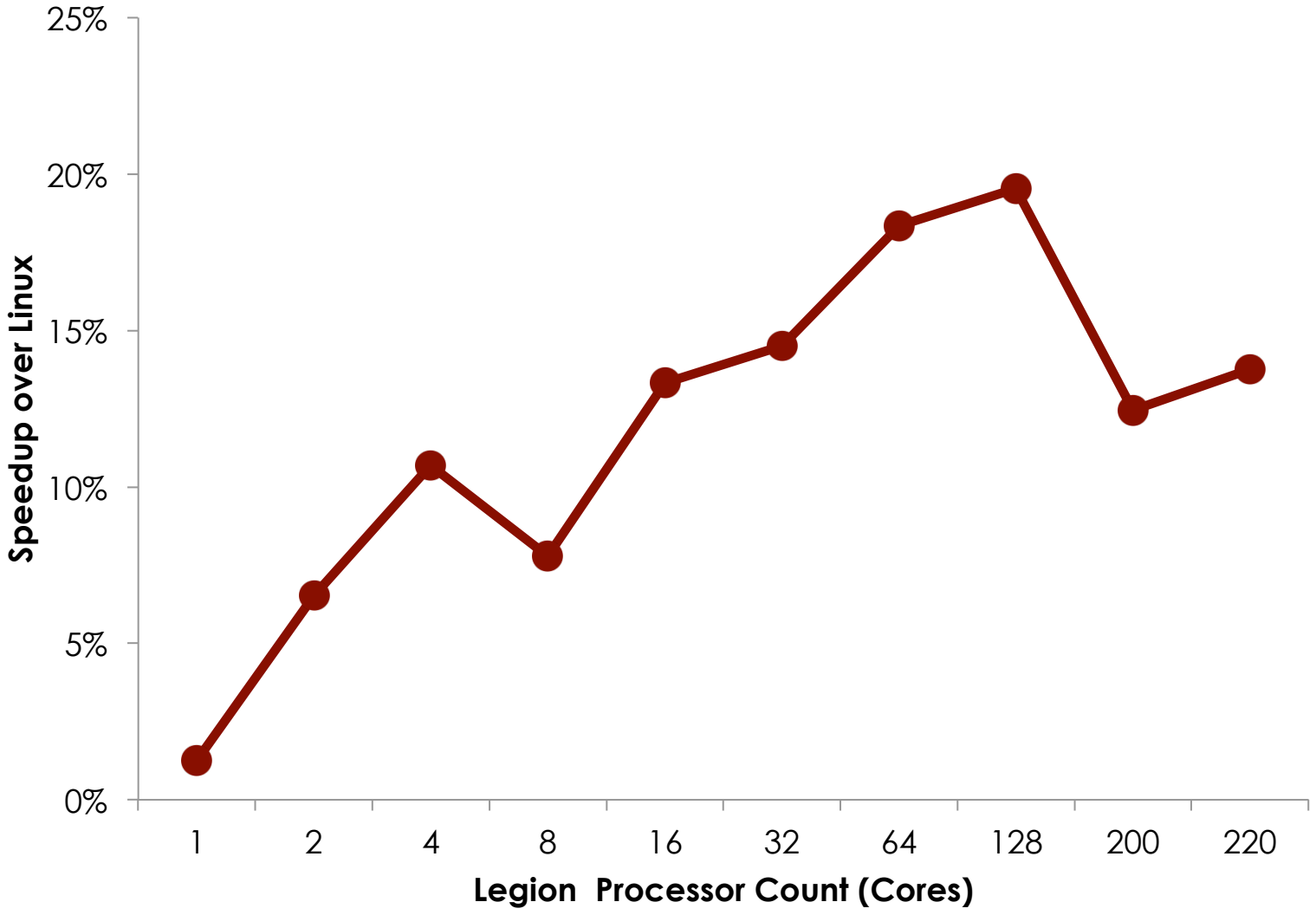
We evaluated our port on a standard HPC benchmark

HPCG

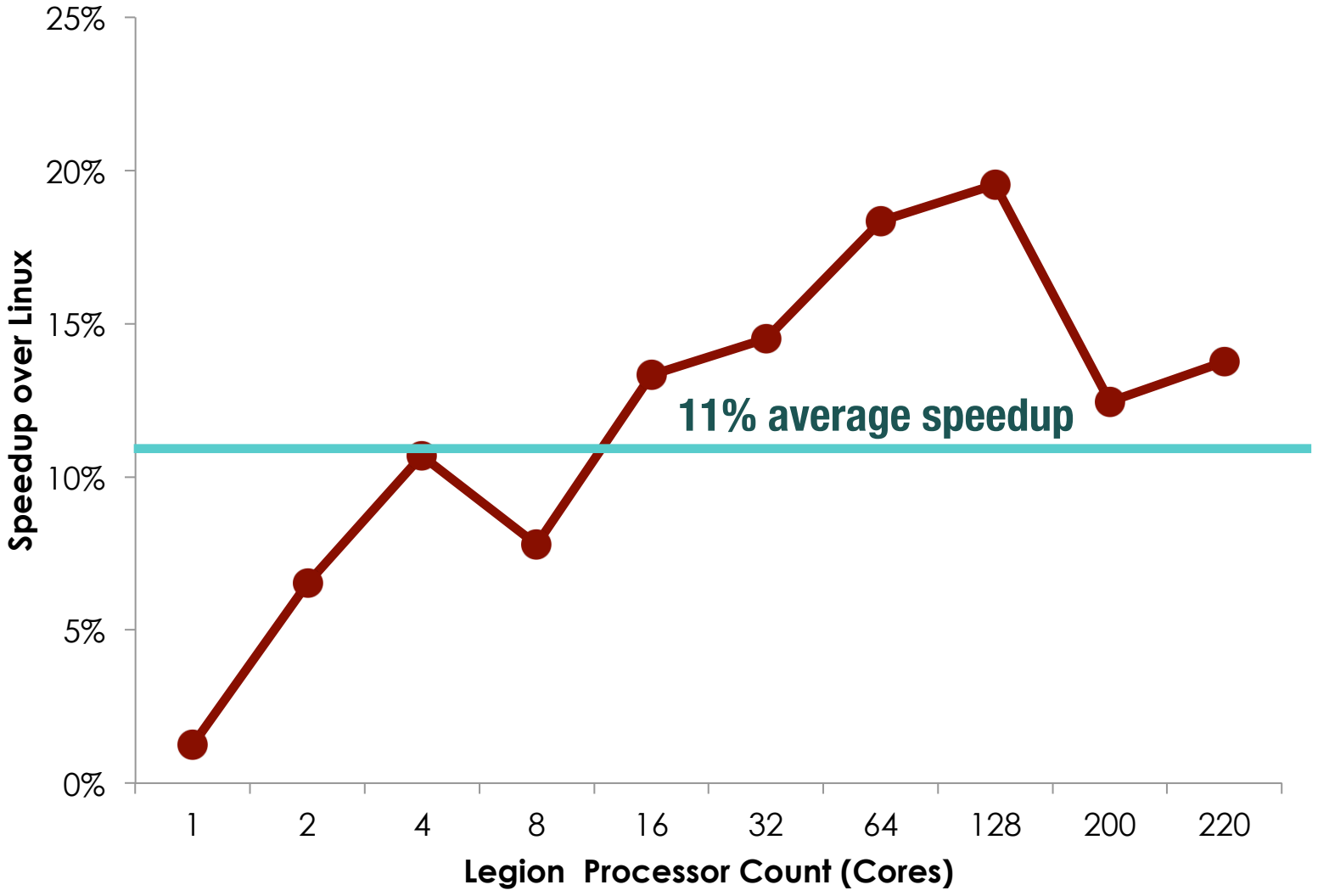
XEON PHI + NAUTILUS + LEGION + HPCG



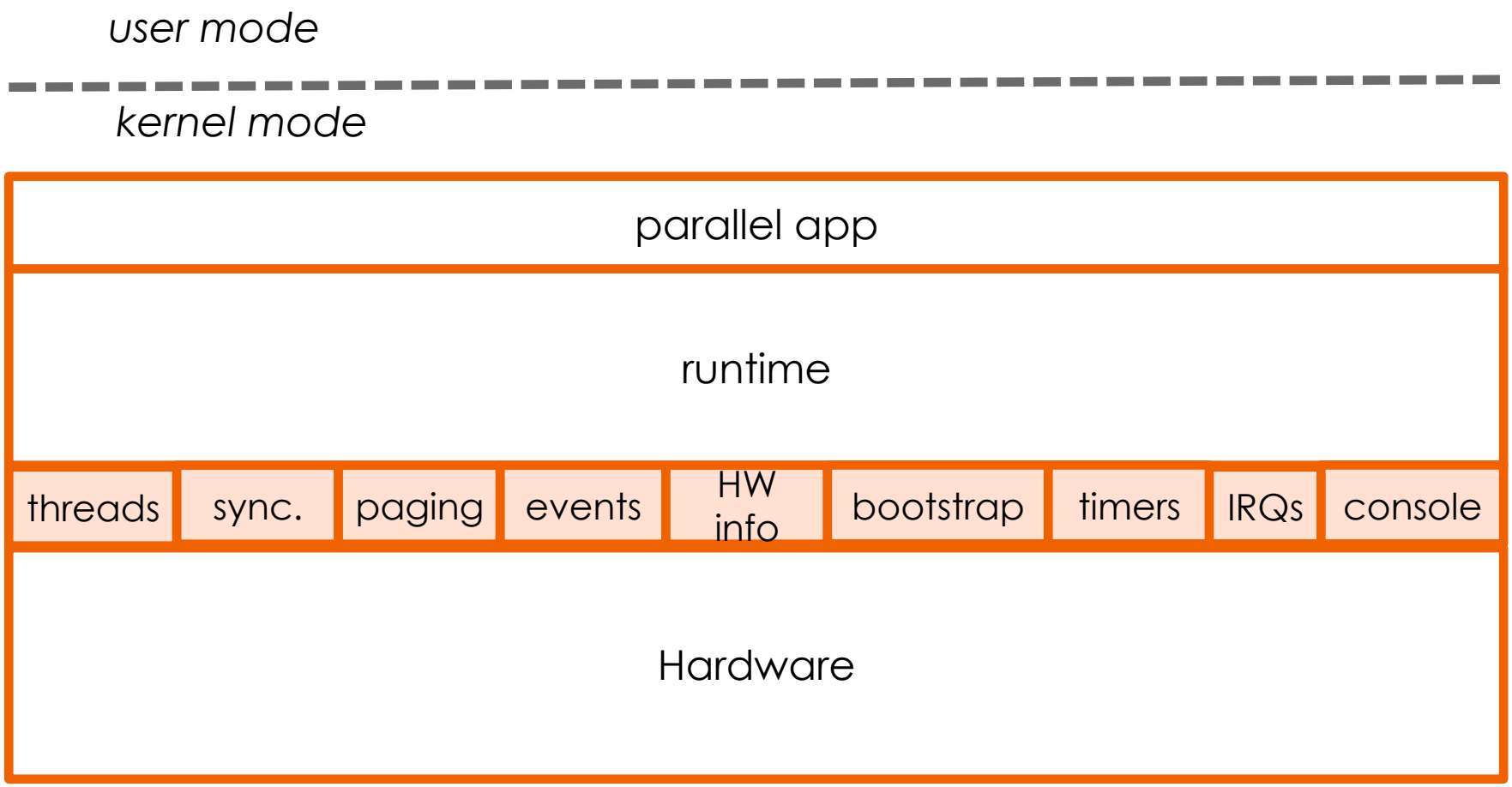
XEON PHI + NAUTILUS + LEGION + HPCG



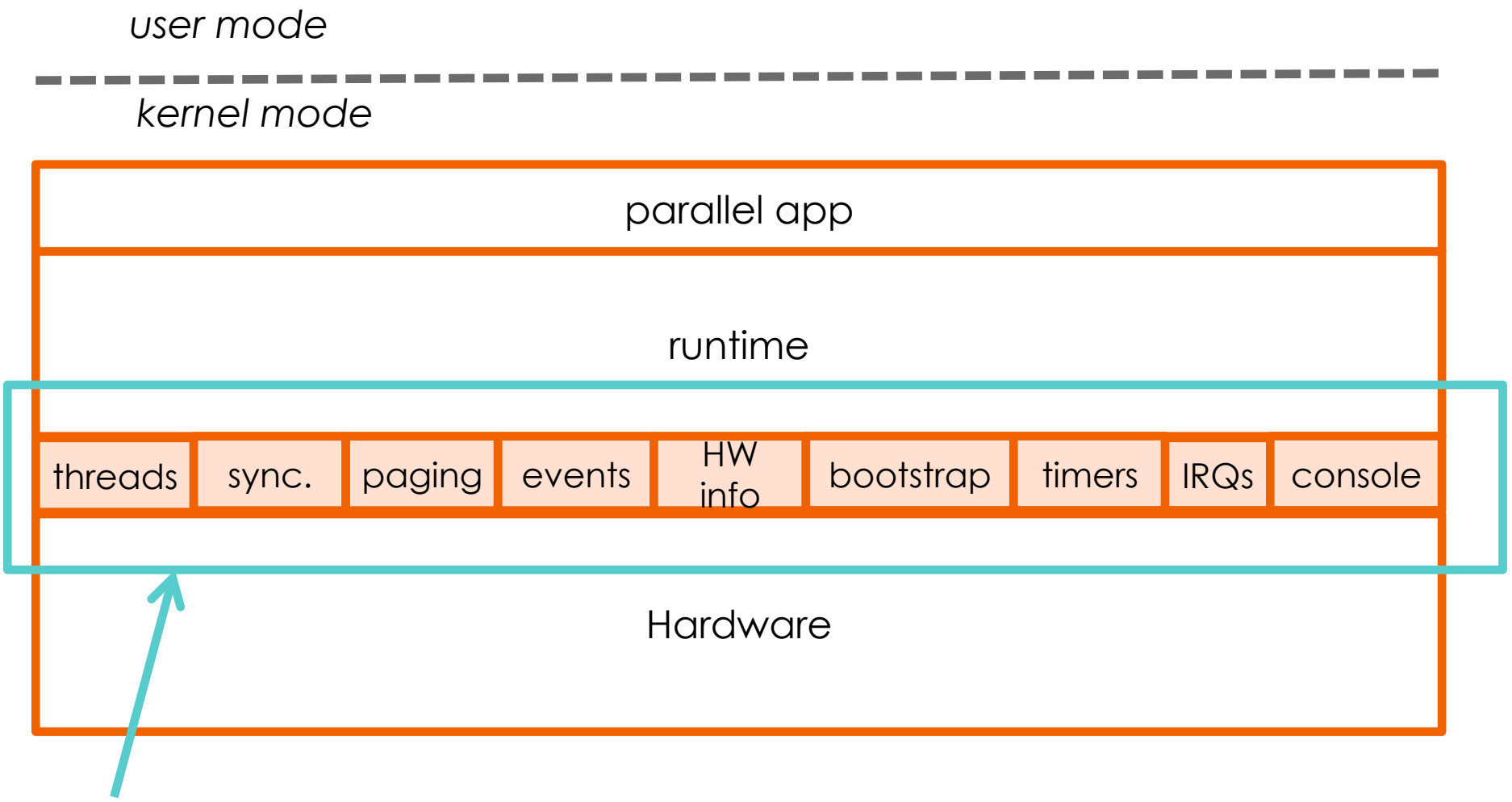
XEON PHI + NAUTILUS + LEGION + HPCG

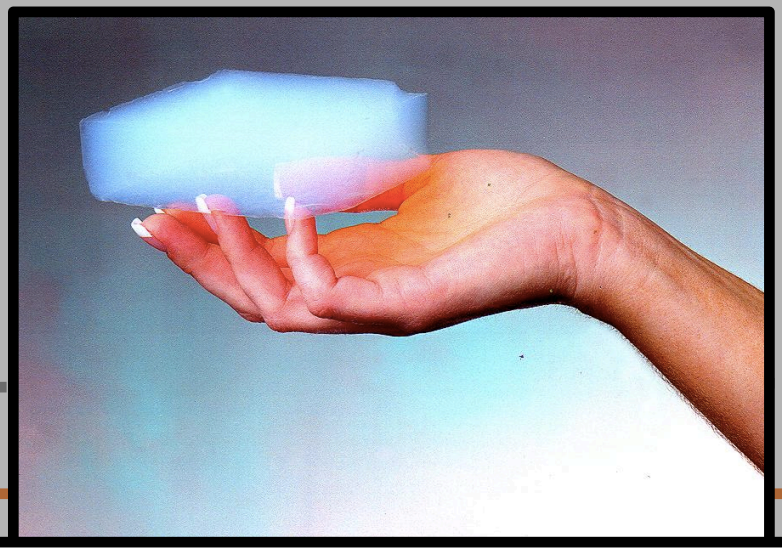


NAUTILUS


user mode
kernel mode
parallel app
runtime
threads
sync.
paging
events
HW info
bootstrap
timers
IRQs
console
Hardware

NAUTILUS


user mode
kernel mode
parallel app
runtime
threads
sync.
paging
events
HW info
bootstrap
timers
IRQs
console
Hardware
Nautilus primitives & utilities (HRT can use or not use any of them)



user mode

kernel mode

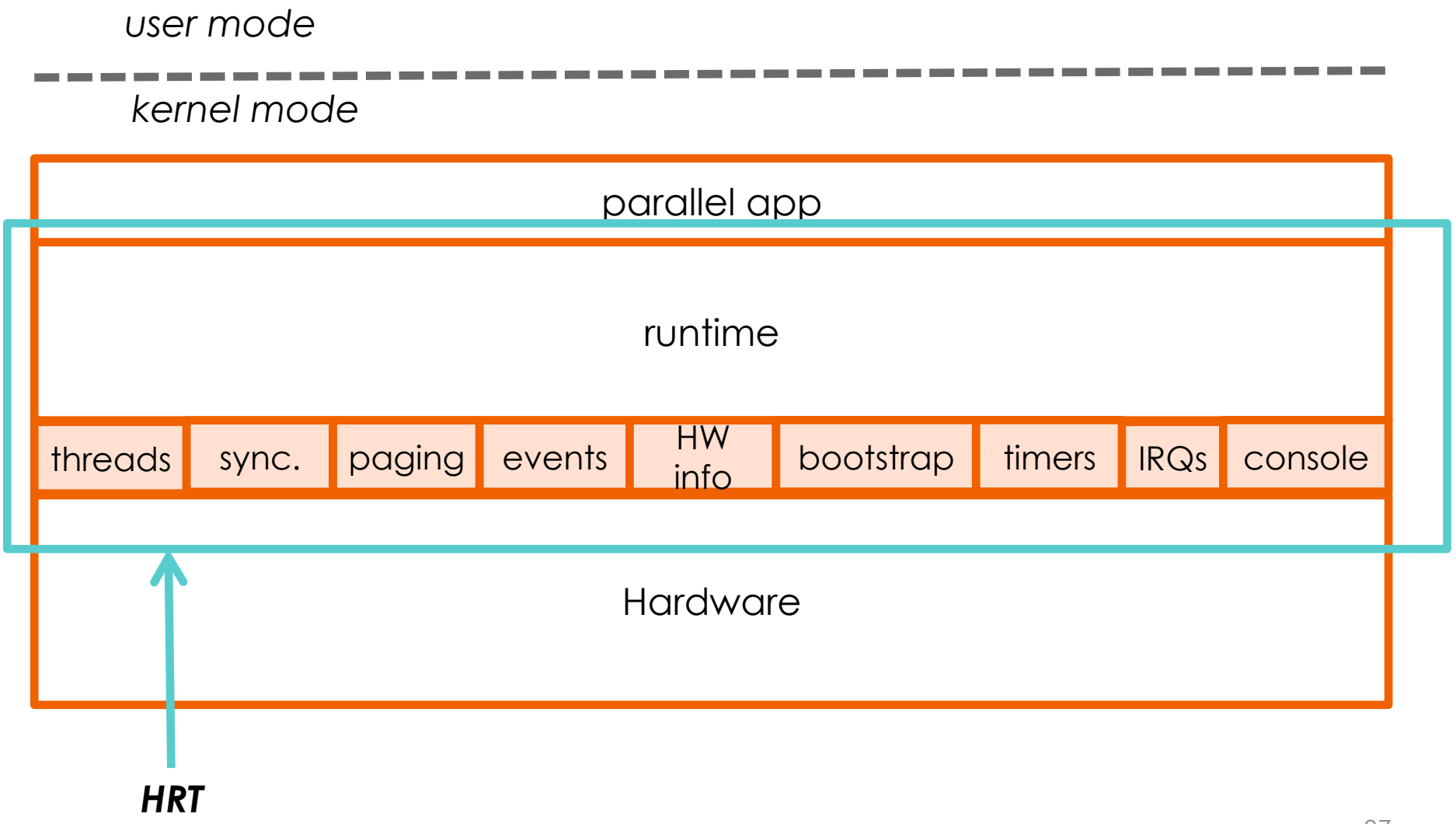
aero**kernel**

threads | sync. | paging | events | HW info | bootstrap | timers | IRQs | console

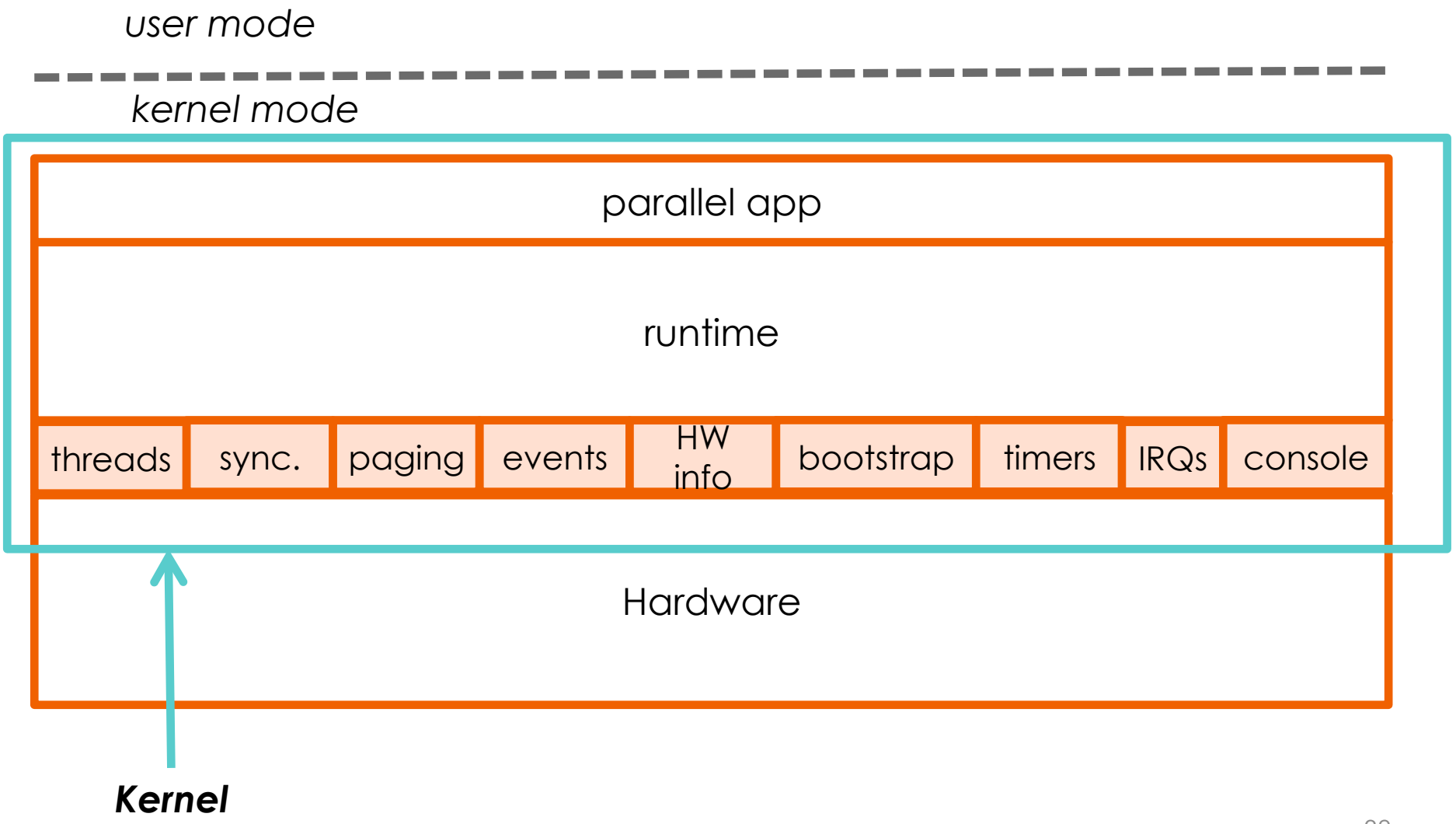
Hardware

Nautilus primitives & utilities (HRT can use or not use any of them)

NAUTILUS


user mode
kernel mode
parallel app
runtime
threads
sync.
paging
events
HW info
bootstrap
timers
IRQs
console
Hardware
HRT

NAUTILUS


user mode
kernel mode
parallel app
runtime
threads
sync.
paging
events
HW info
bootstrap
timers
IRQs
console
Hardware
Kernel

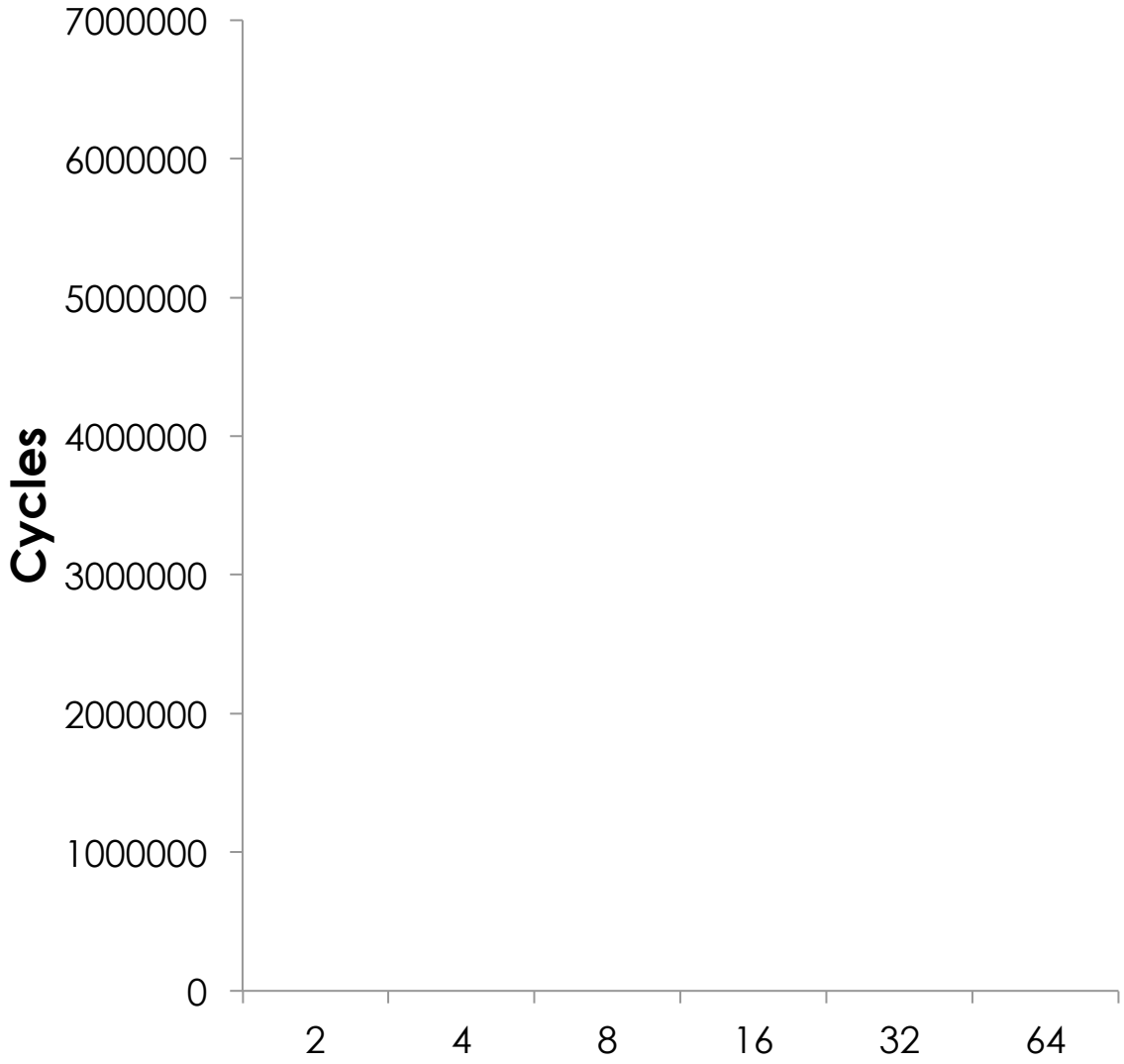
MINIMAL LIGHTWEIGHT PRIMITIVES

FULL HARDWARE ACCESS

VERY FAST BOOT TIMES

LIGHTWEIGHT PRIMITIVES

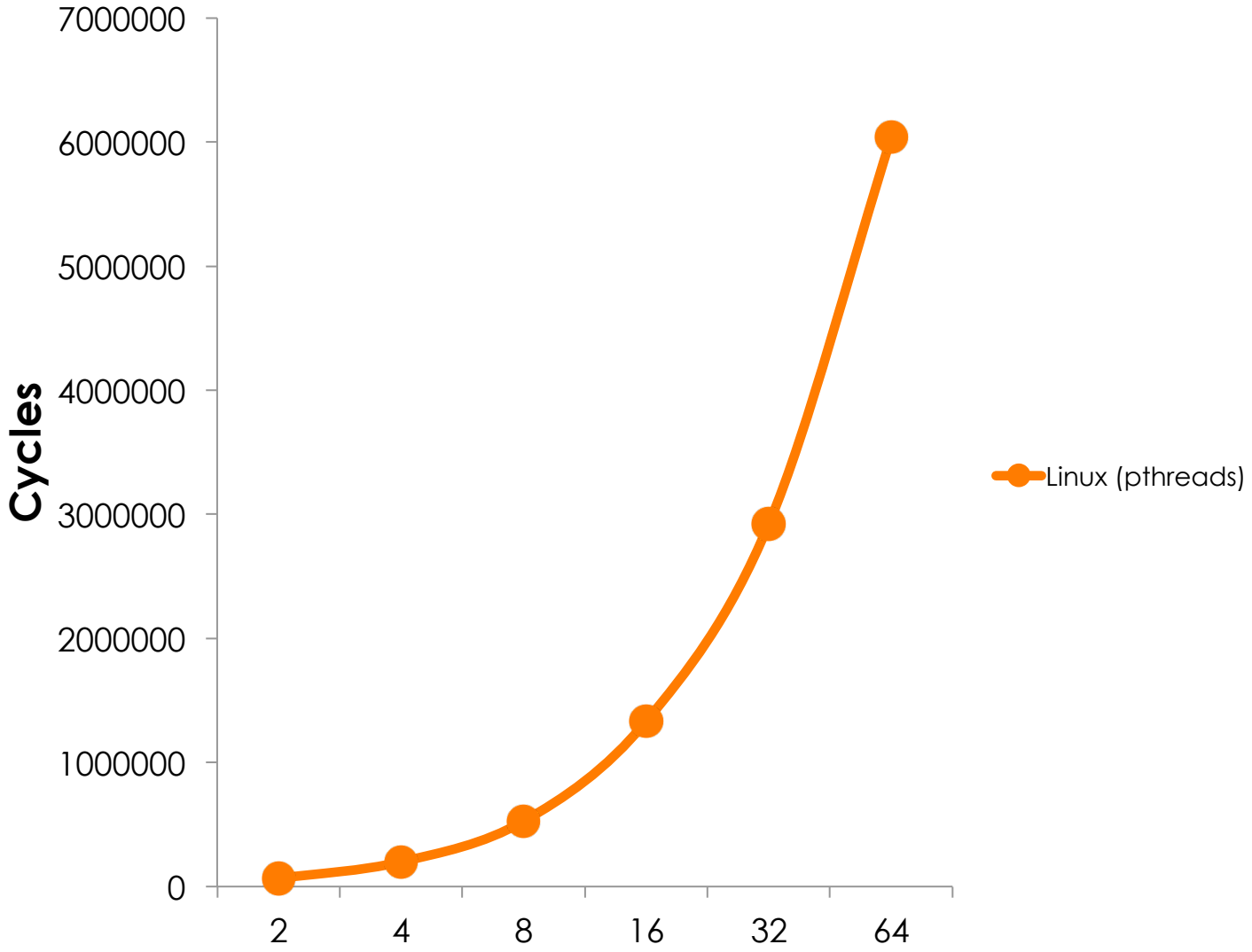
EXAMPLE: THREADS



x86_64 Opteron: 64 cores, 4 sockets, 8 numa zones, 128GB RAM

LIGHTWEIGHT PRIMITIVES

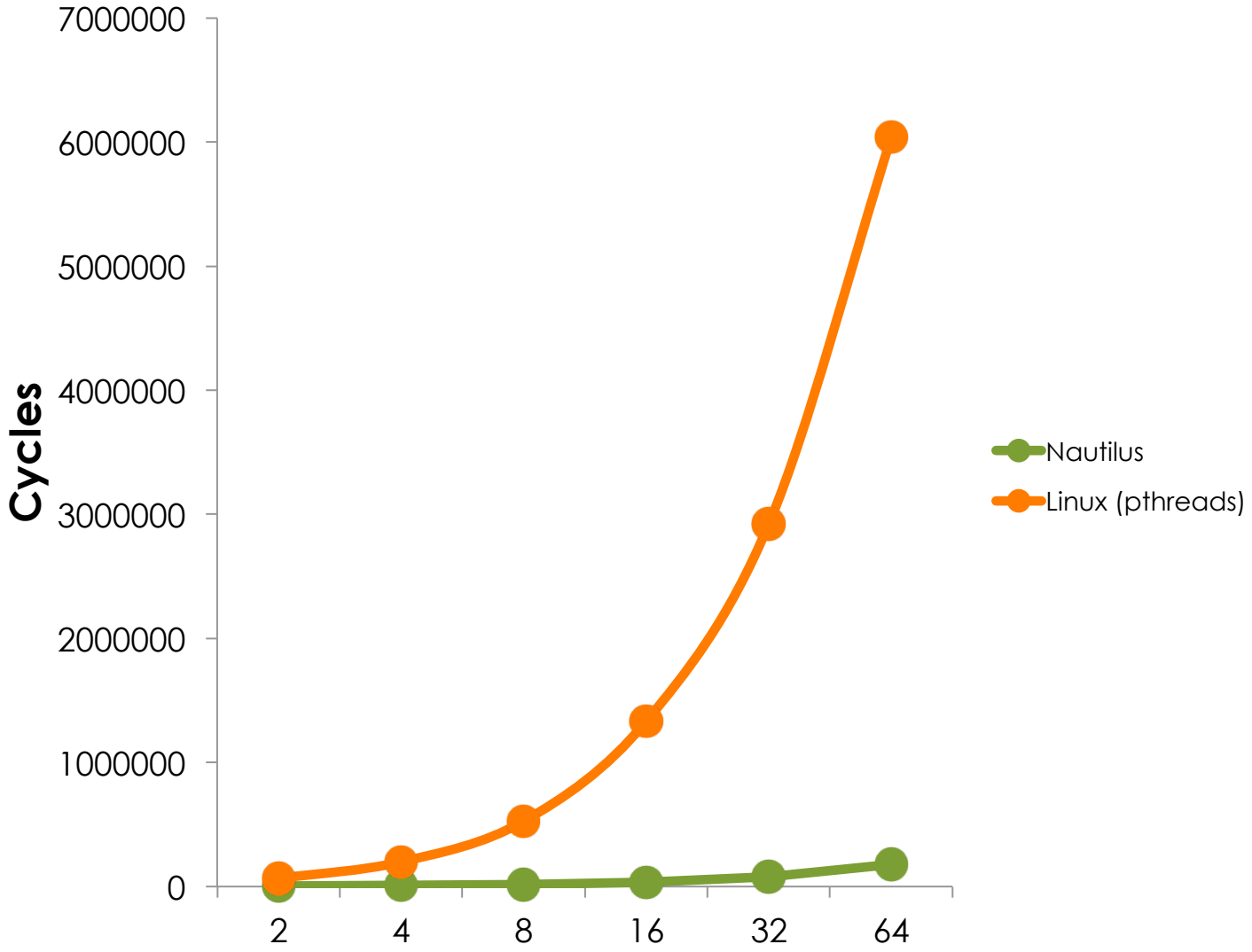
EXAMPLE: THREADS



x86_64 Opteron: 64 cores, 4 sockets, 8 numa zones, 128GB RAM

LIGHTWEIGHT PRIMITIVES

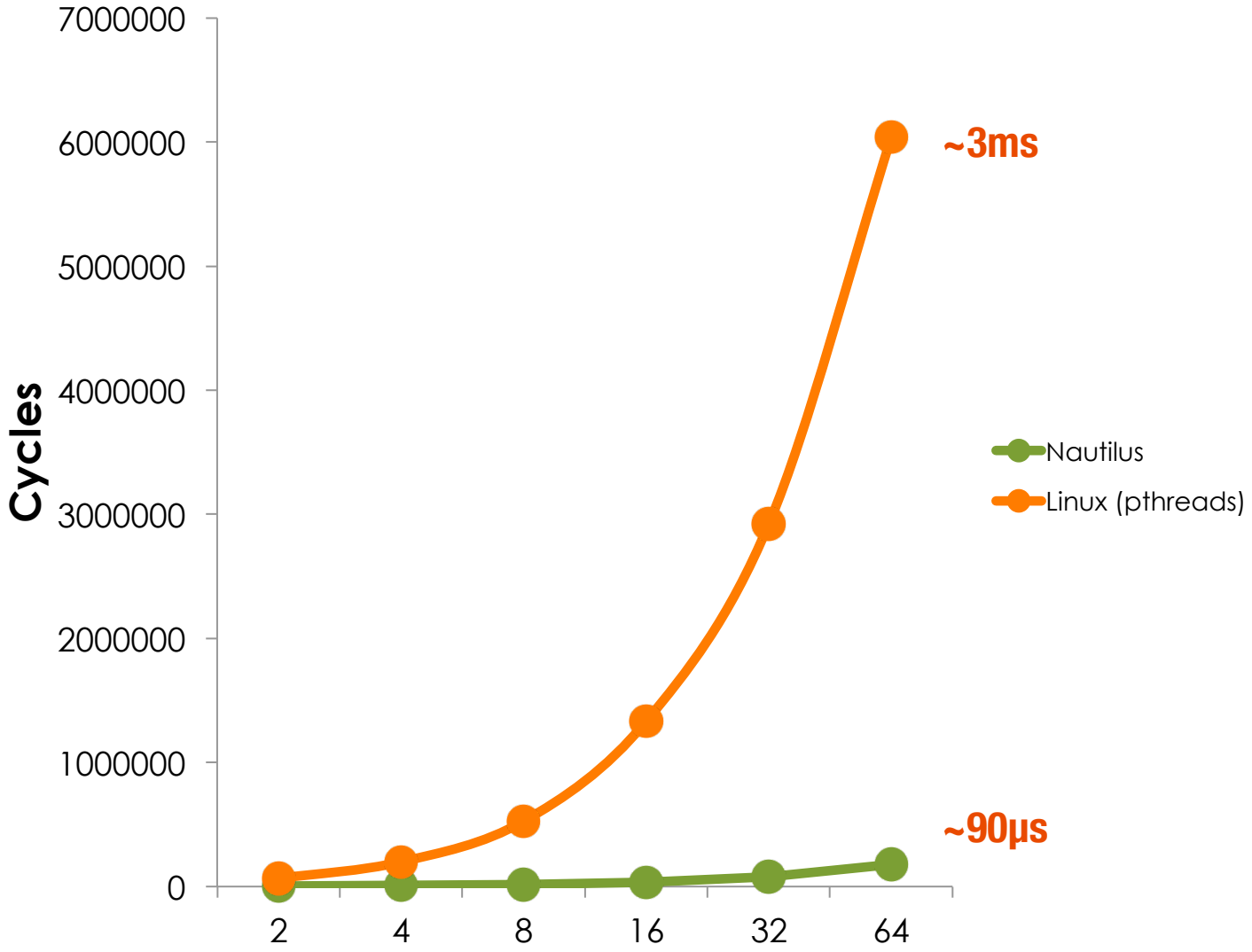
EXAMPLE: THREADS



x86_64 Opteron: 64 cores, 4 sockets, 8 numa zones, 128GB RAM

LIGHTWEIGHT PRIMITIVES

EXAMPLE: THREADS



x86_64 Opteron: 64 cores, 4 sockets, 8 numa zones, 128GB RAM

FULL HARDWARE CONTROL

EXAMPLE: INTERRUPT CONTROL

FULL HARDWARE CONTROL

EXAMPLE: INTERRUPT CONTROL

very simple modification: give runtime control over interrupts in its task scheduler

FULL HARDWARE CONTROL

EXAMPLE: INTERRUPT CONTROL

very simple modification: give runtime control over interrupts in its task scheduler

→ **modest speedups**

FULL HARDWARE CONTROL

EXAMPLE: INTERRUPT CONTROL

very simple modification: give runtime control over interrupts in its task scheduler

→ **modest speedups**

MUCH more to come here

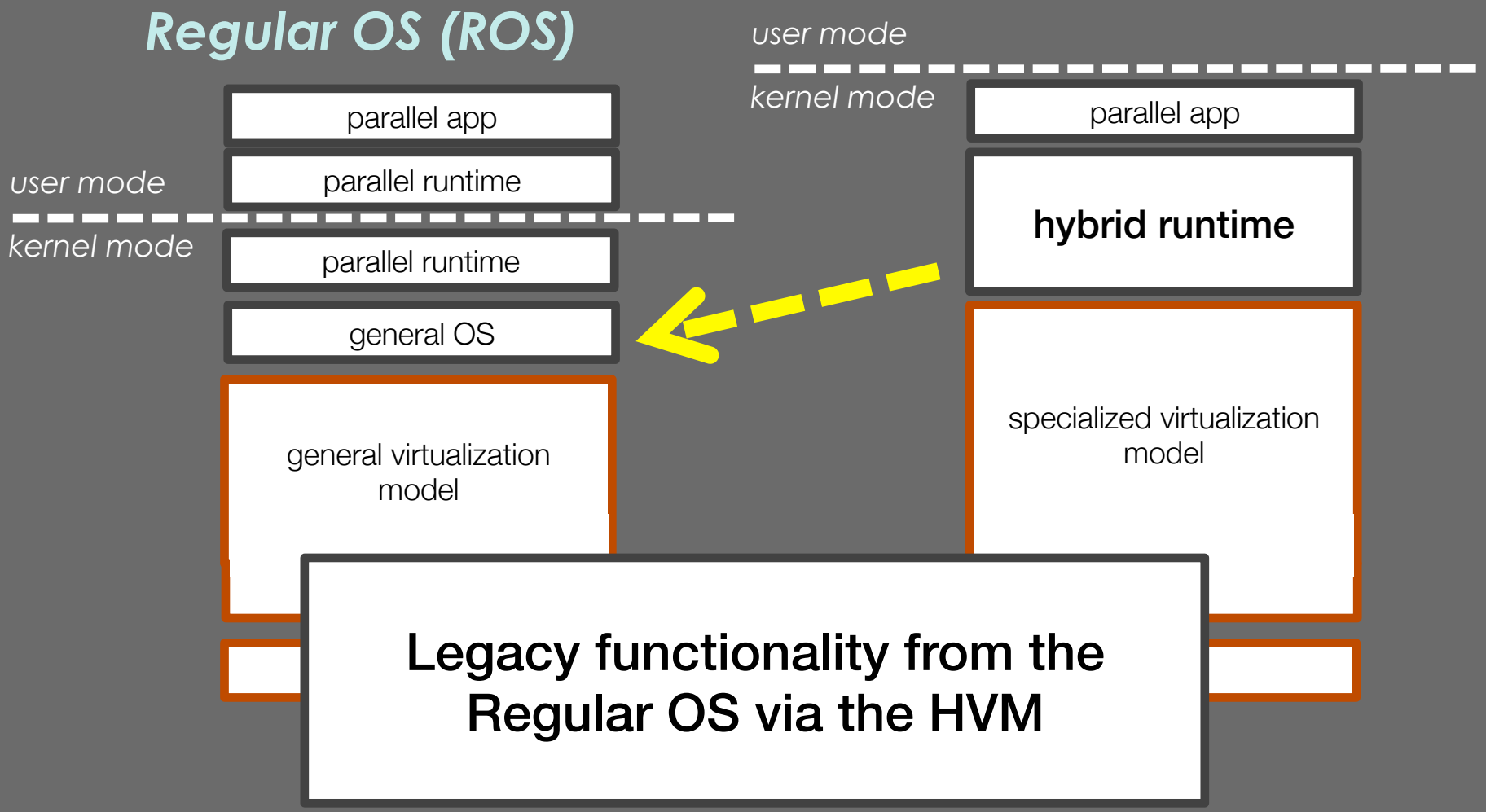
in addition to Legion,
we have 2 other high-level, parallel runtimes running as HRTs

NESL: VCODE interpreter running as HRT

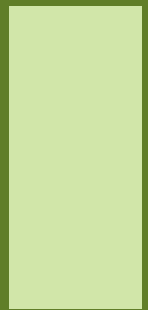
NDPC: home-grown, **co-designed** HRT

INTEGRATING THE HRT WITH A LEGACY OS

THE HYBRID VIRTUAL MACHINE


Regular OS (ROS)
user mode
kernel mode
parallel app
parallel runtime
parallel runtime
general OS
general virtualization model
user mode
kernel mode
parallel app
hybrid runtime
specialized virtualization model
Legacy functionality from the Regular OS via the HVM

LINUX FORK + EXEC ~ 714µs



HVM + HRT CORE BOOT ~ 61µs

LINUX FORK + EXEC ~ 714µs

HRT boot is CHEAP!

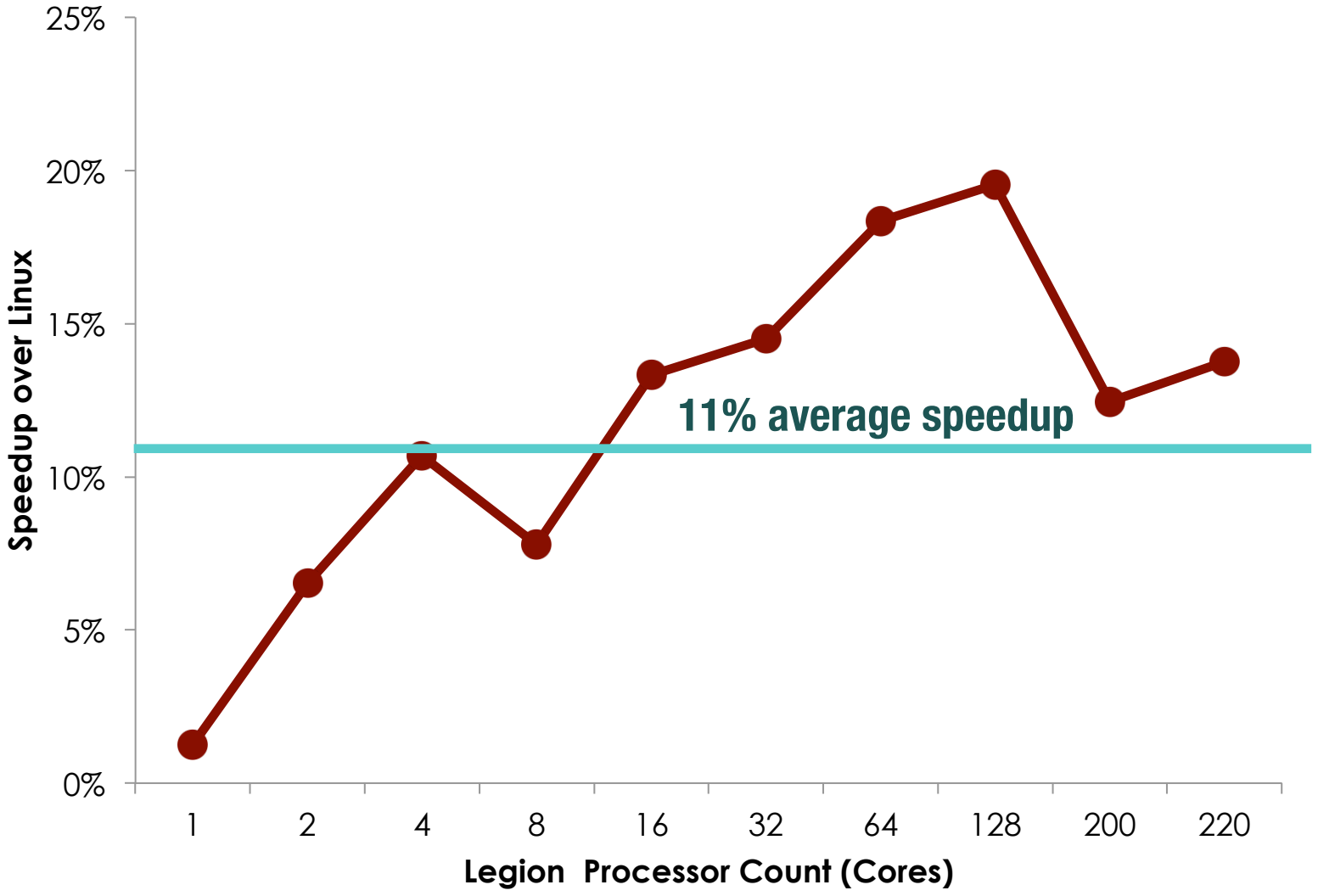
HVM + HRT CORE BOOT ~ 61µs

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
7	Texas Advanced Computing Center/Univ. of Texas United States	**Stampede** - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462,462	5,168.1	8,520.1	4,510
18	DOE/SC/Pacific Northwest National	**cascade** - Atipa Visione IF442 Blade Server,	194,616	2,539.1	3,388.0	1,384
53	Purdue University United States	**Conte** - Cluster Platform SL250s Gen8, Xeon E5-2670 8C 2.600GHz, Infiniband FDR, Intel Xeon Phi 5110P Hewlett-Packard	77,520	976.8	1,341.1	510
64	Tulip Trading Australia	**C01N** - SuperBlade SBI-7127RG-E, Intel Xeon E5-2695v2 12C 2.4GHz, Infiniband FDR, Intel Xeon Phi 7120P Supermicro	160,600	798.3	3,164.5	619
69	Intel United States	**Endeavor** - Intel Cluster, Intel Xeon E5-2697v2 12C 2.700GHz, Infiniband FDR, Intel	51,392	758.9	933.5	387.2

NAUTILUS + XEON PHI

```
[root@v-test-t620 nautilus]#
[root@v-test-t620 nautilus]# philix -d -b weever -k nautilus.bin
```

XEON PHI + NAUTILUS + LEGION + HPCG



A CASE FOR TRANSFORMING PARALLEL RUNTIMES INTO OS KERNELS

my website
halek.co

our development blog
haltloop.com

our lab
presciencelab.org

the Hobbes project
xstack.sandia.gov/hobbes

follow us here for:

- **experience report on building OS for Phi**
- **philix release (soon)**

Kyle Hale

Peter Dinda

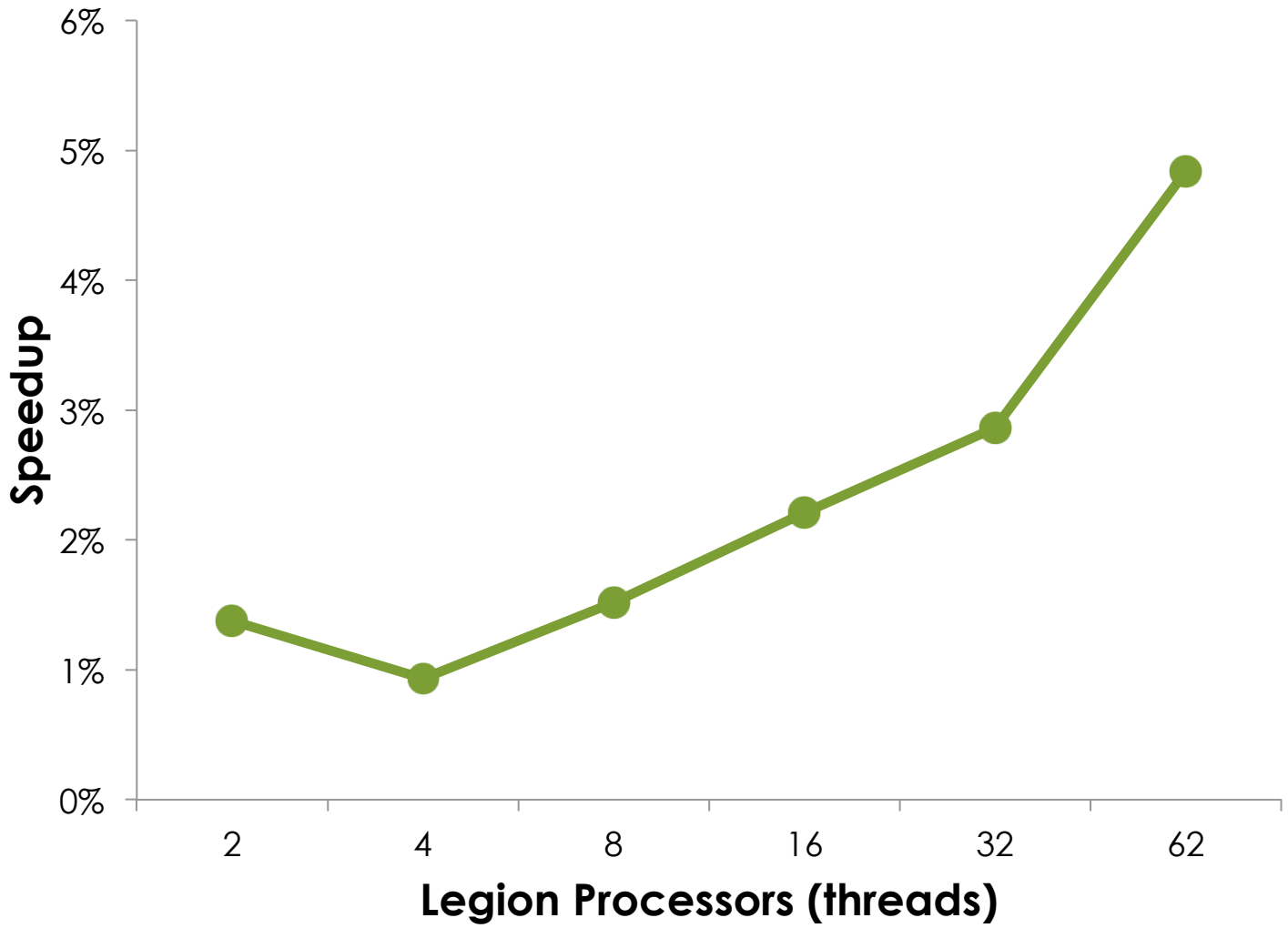
BACKUPS

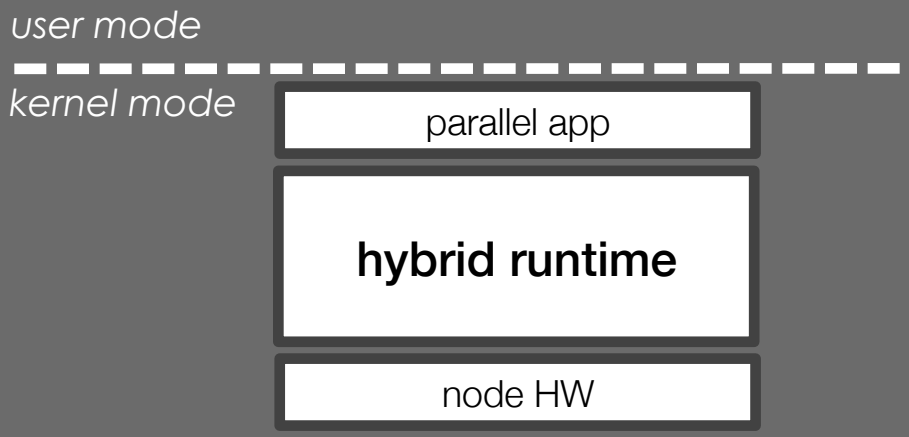
FULL HARDWARE CONTROL

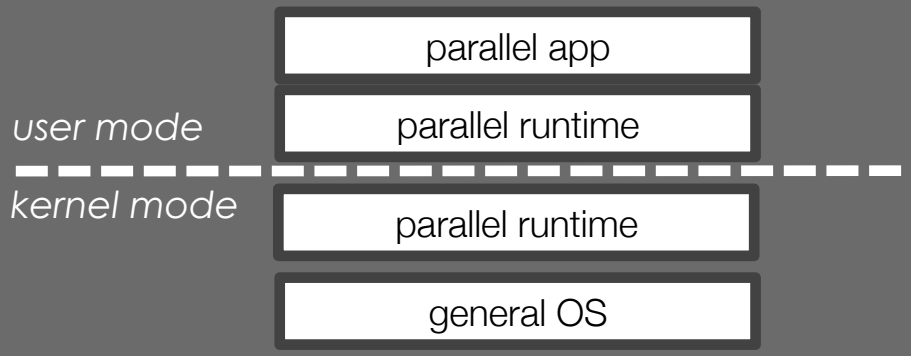
EXAMPLE: INTERRUPT CONTROL

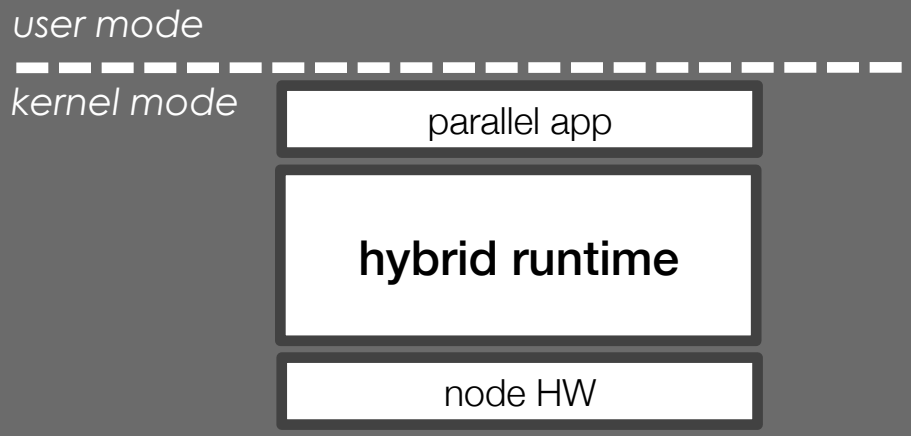
FULL HARDWARE CONTROL

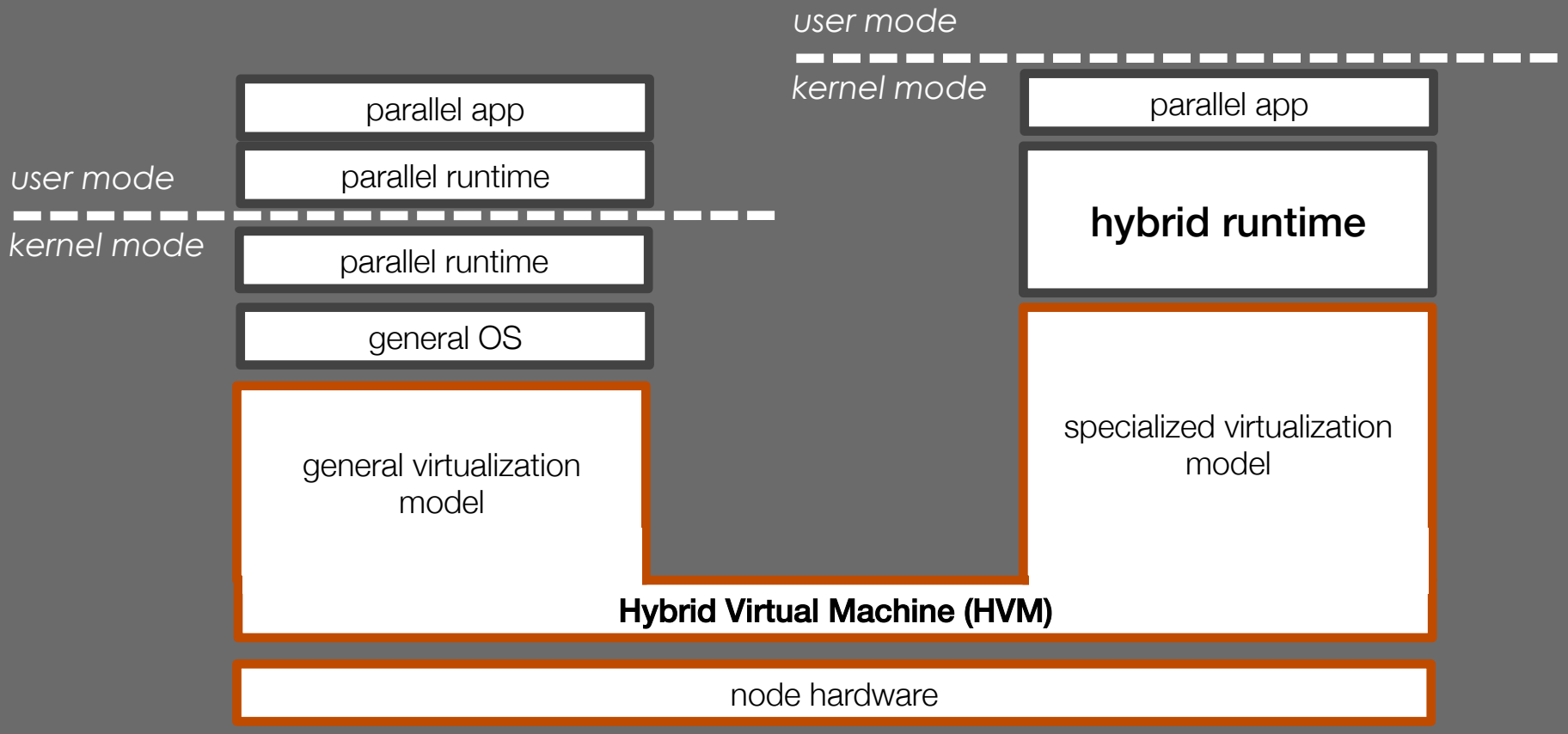
EXAMPLE: INTERRUPT CONTROL

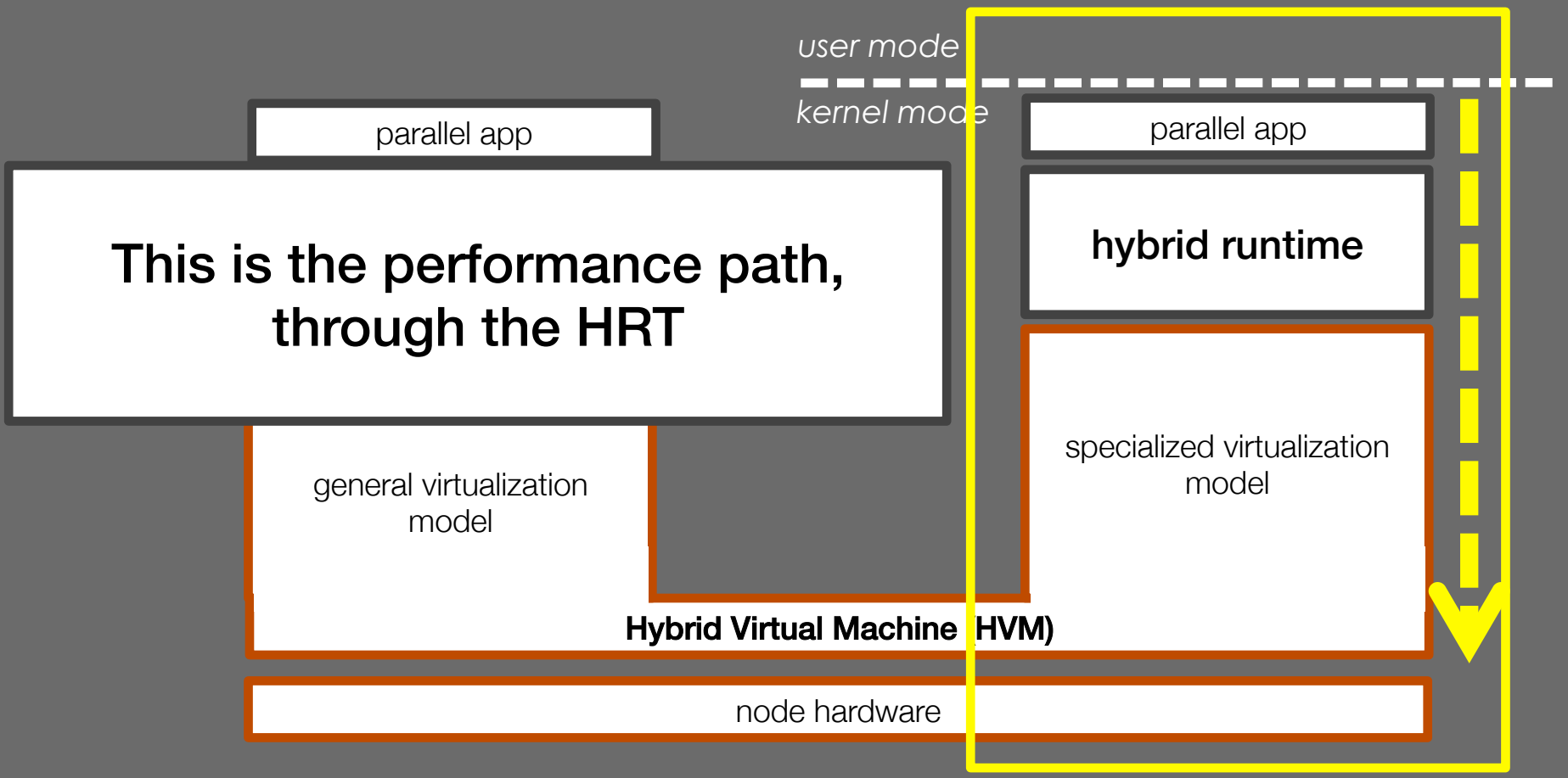


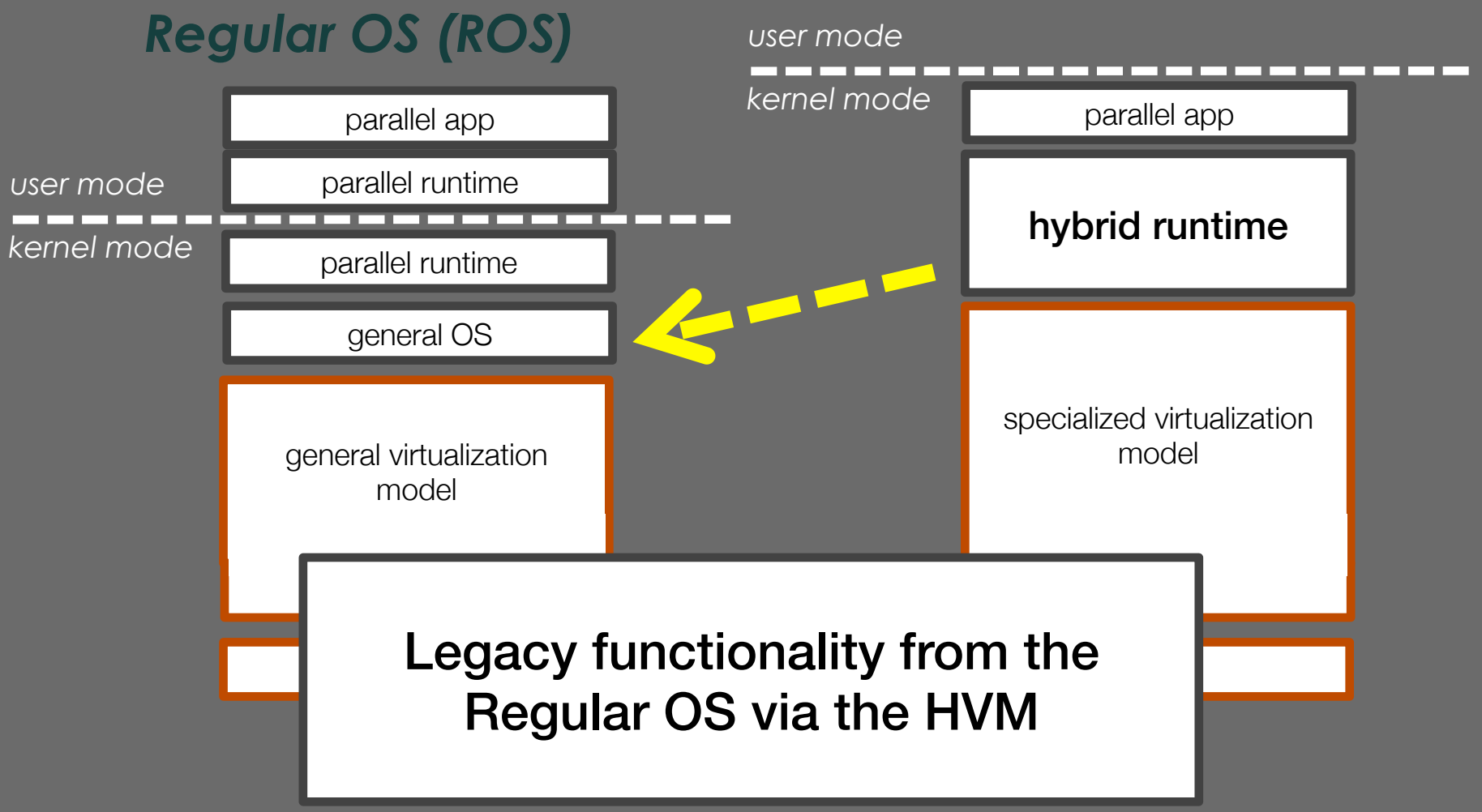

user mode
kernel mode
parallel app
hybrid runtime
node HW

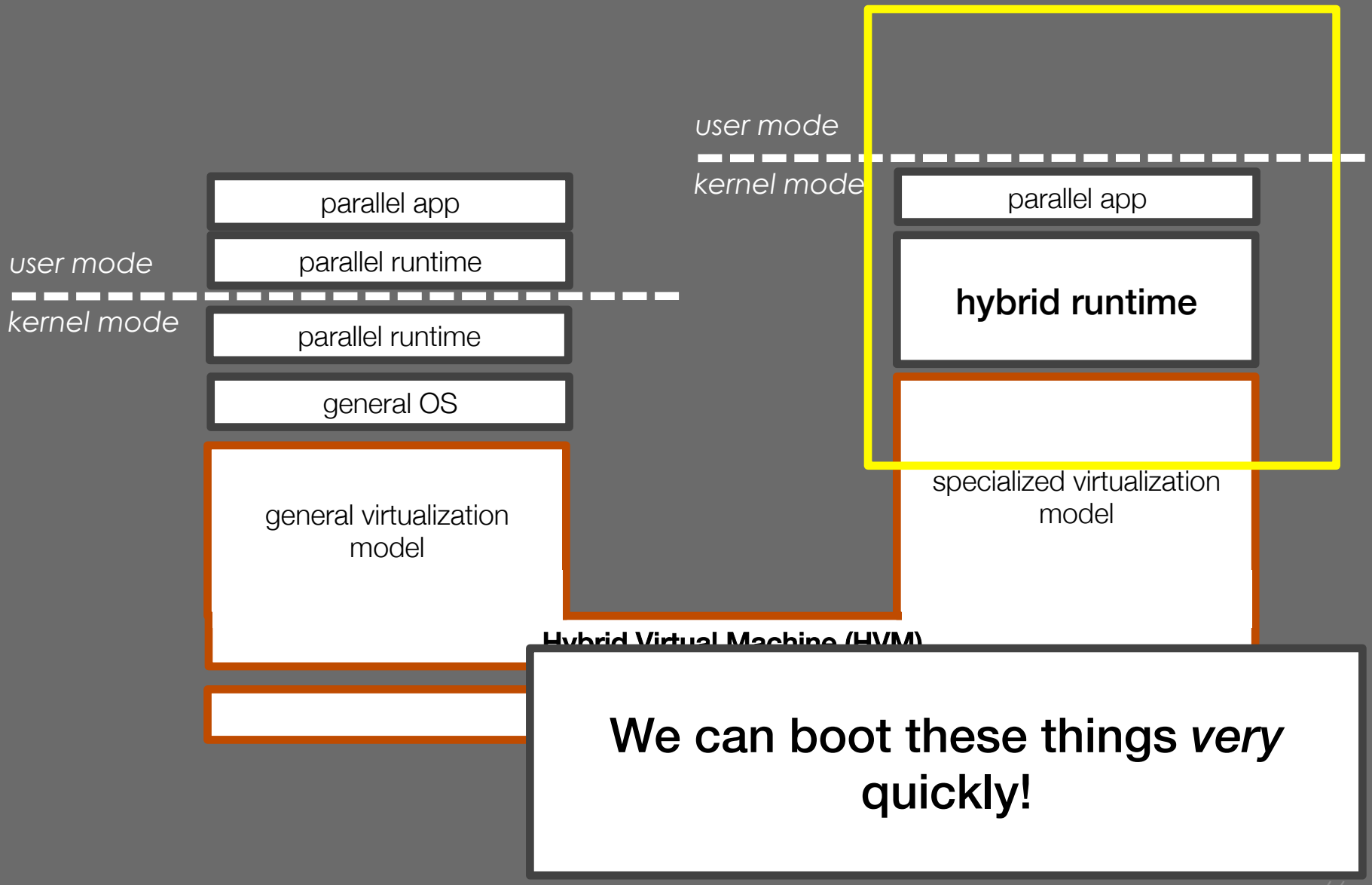

parallel app
parallel runtime
user mode
kernel mode
parallel runtime
general OS


user mode
kernel mode
parallel app
hybrid runtime
node HW

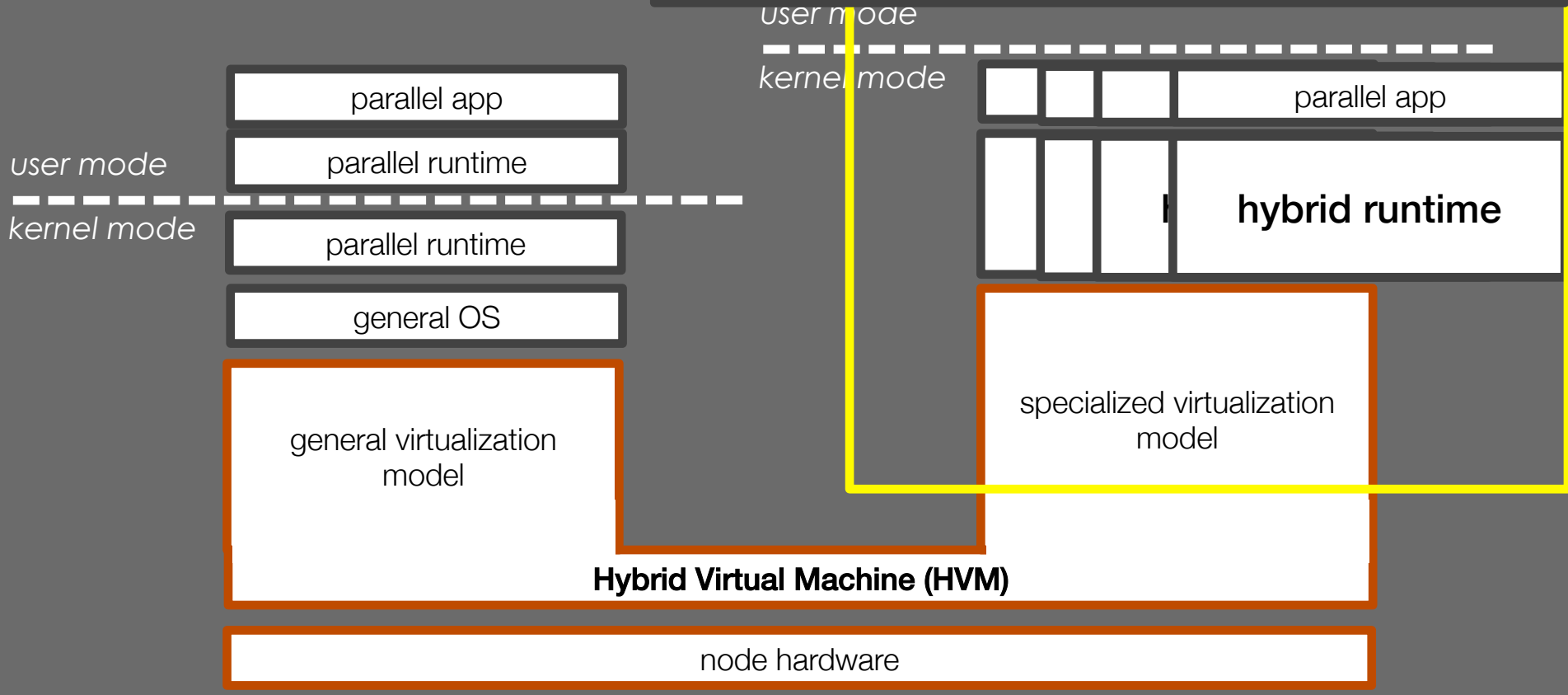

user mode
kernel mode
parallel app
parallel runtime
parallel runtime
general OS
general virtualization model
user mode
kernel mode
parallel app
hybrid runtime
specialized virtualization model
Hybrid Virtual Machine (HVM)
node hardware


user mode
kernel mode
parallel app
This is the performance path, through the HRT
general virtualization model
parallel app
hybrid runtime
specialized virtualization model
Hybrid Virtual Machine (HVM)
node hardware

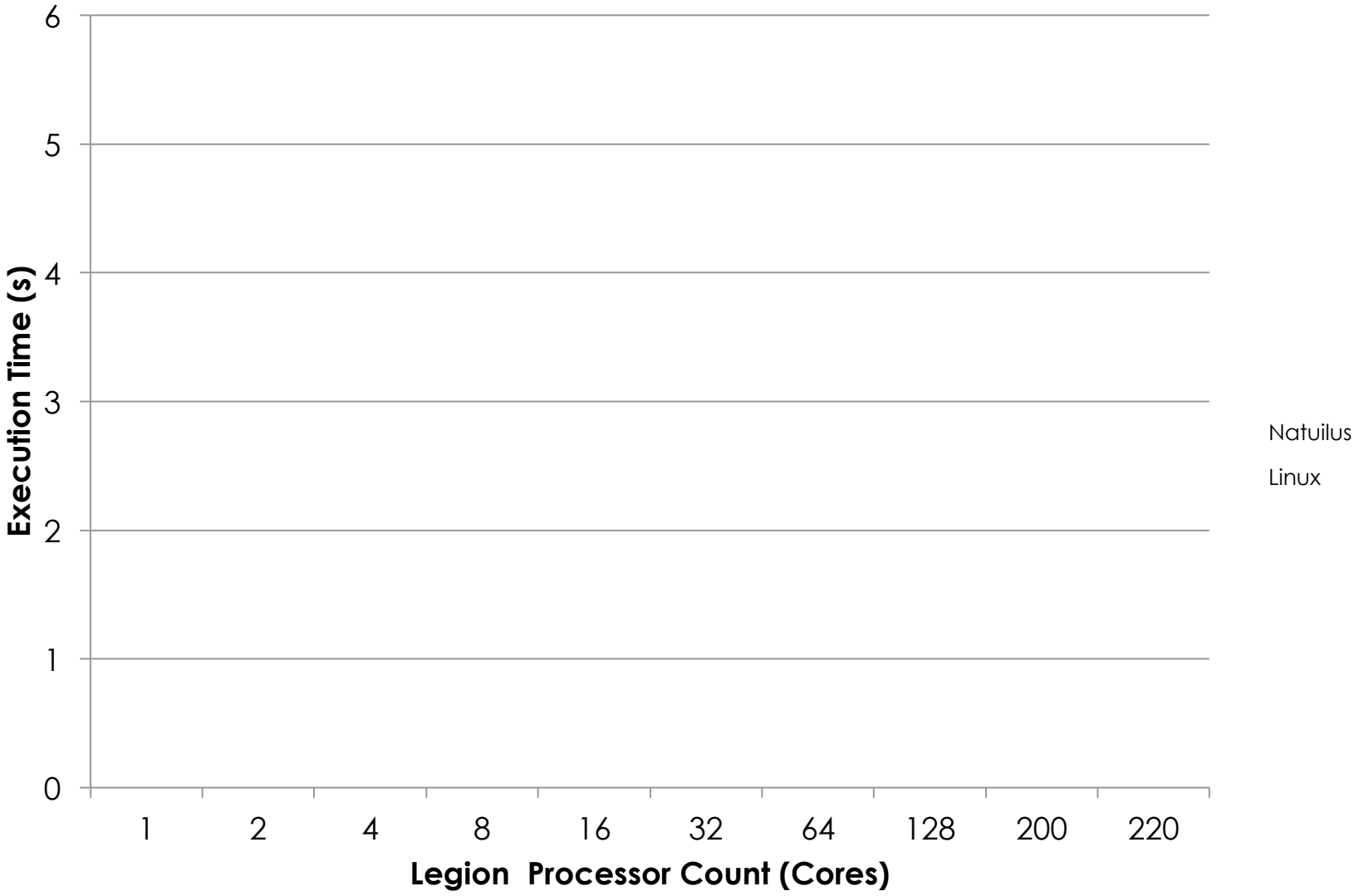

Regular OS (ROS)
user mode
kernel mode
parallel app
parallel runtime
parallel runtime
general OS
general virtualization model
user mode
kernel mode
parallel app
hybrid runtime
specialized virtualization model
Legacy functionality from the Regular OS via the HVM


user mode
kernel mode
parallel app
parallel runtime
parallel runtime
general OS
general virtualization model
user mode
kernel mode
parallel app
hybrid runtime
specialized virtualization model
Hybrid Virtual Machine (HVM)
We can boot these things *very* quickly!

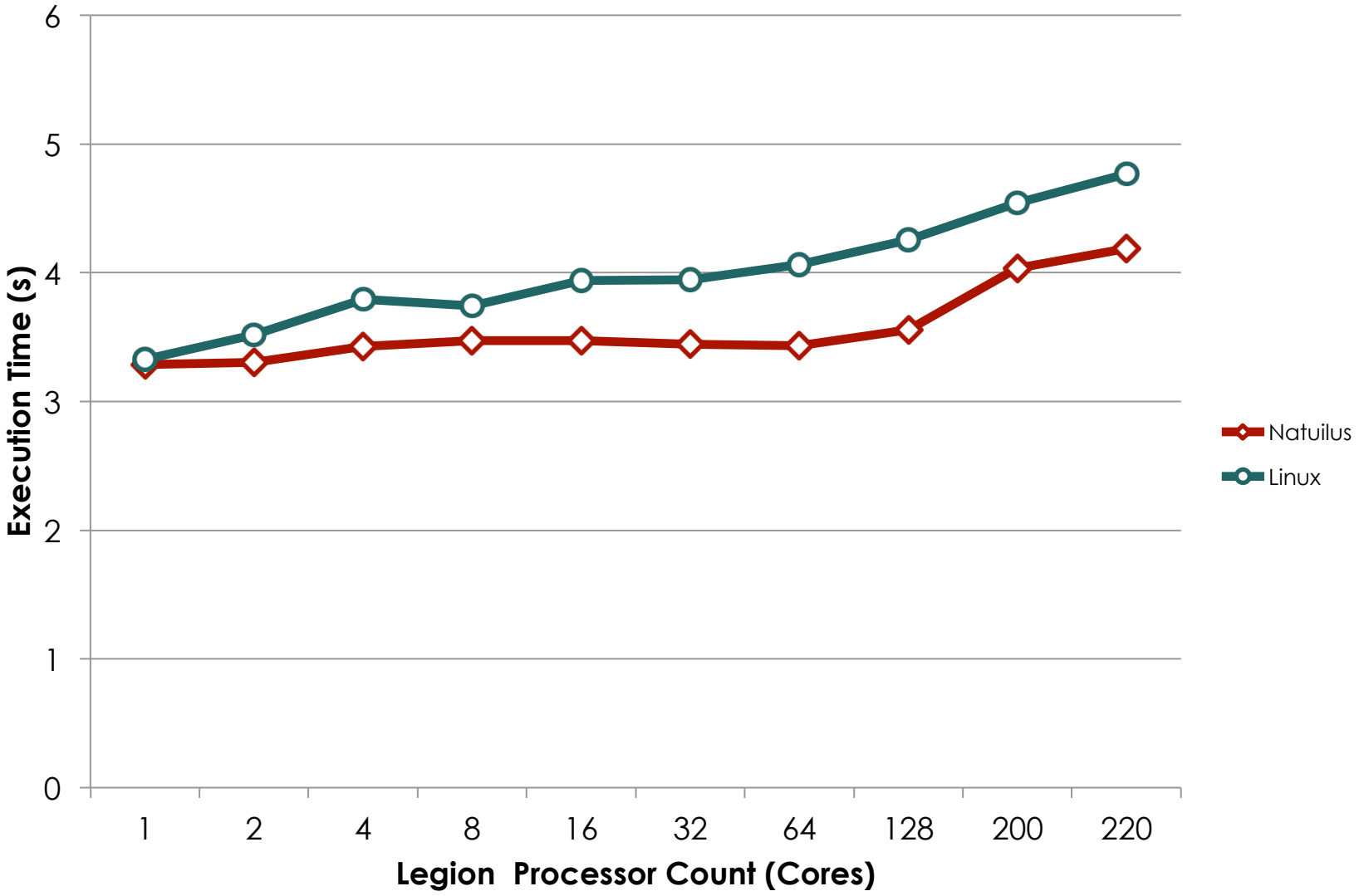
several auxiliary HRTs spawned in *less than a millisecond*



HPCG IN LEGION ON XEON PHI



HPCG IN LEGION ON XEON PHI



port of NESL

- nested data parallel language aimed at vector machines

port of NESL

- nested data parallel language aimed at vector machines

- we can run unmodified NESL programs in our kernel-mode VCODE interpreter

the first co-designed HRT: NDPC

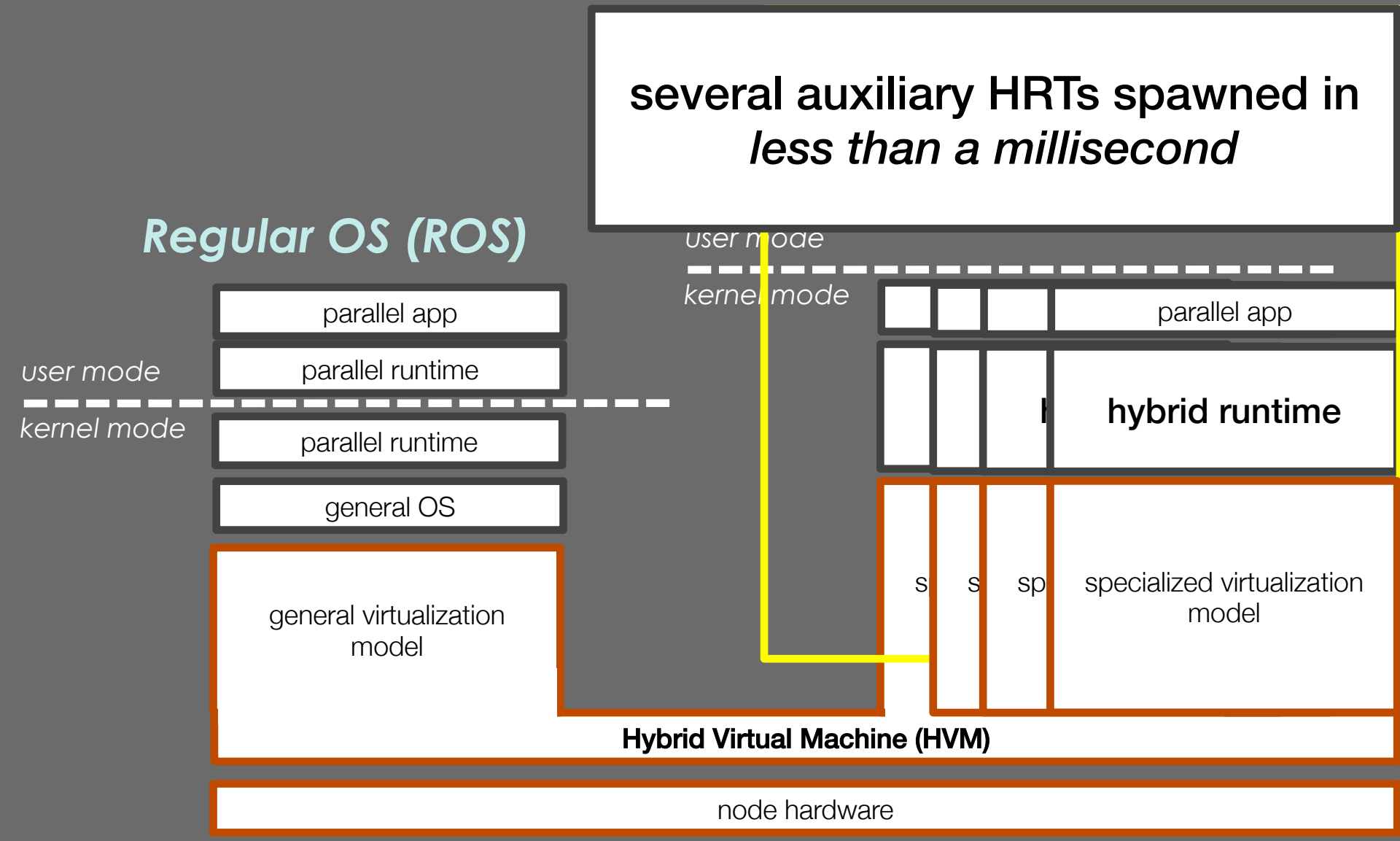
- Nested Data Parallelism in C/C++
- subset of NESL

the first co-designed HRT: NDPC

- Nested Data Parallelism in C/C++
- subset of NESL
- fork/join parallelism over flattened vector processing

the first co-designed HRT: NDPC

- Nested Data Parallelism in C/C++
- subset of NESL
- fork/join parallelism over flattened vector processing
- allows us to explore runtime/kernel co-design
- e.g. smart kernel-mode thread fork


several auxiliary HRTs spawned in
less than a millisecond
Regular OS (ROS)
user mode
kernel mode
parallel app
parallel runtime
parallel runtime
general OS
general virtualization model
user mode
kernel mode
parallel app
hybrid runtime
specialized virtualization model
Hybrid Virtual Machine (HVM)
node hardware

to get started with your own Xeon Phi prototype kernel:

to get started with your own Xeon Phi prototype kernel:

- follow our blog
- use our tool (philix) to boot it and leverage MPSS stack

to get started with your own Xeon Phi prototype kernel:

- follow our blog
- use our tool (philix) to boot it and leverage MPSS stack

to get started with your own Xeon Phi prototype kernel:

- follow our blog
- use our tool (philix) to boot it and leverage MPSS stack

find out more @ haltloop.com