Linaro | connect San Francisco 2015

SFO15-407: Performance Overhead of ARM Virtualization

Linaro Connect SFO15
Christoffer Dall, Linaro

Virtualization Use Cases

- Resource Sharing
- Isolation
- High Availability
- Provisioning
- Load balancing

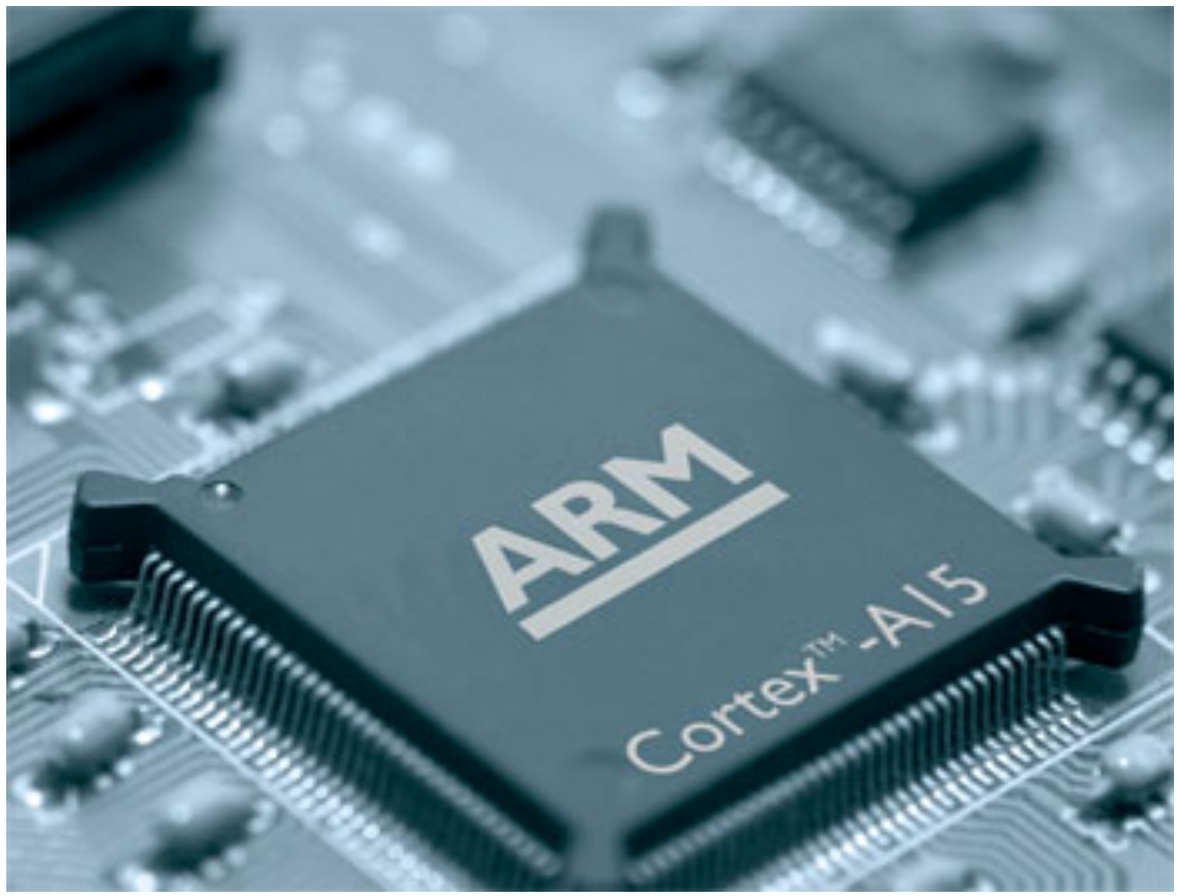
No Free Lunches

There's a cost: **Performance**

Virtualization on ARM

ARM®

Virtualization Extensions



Linaro

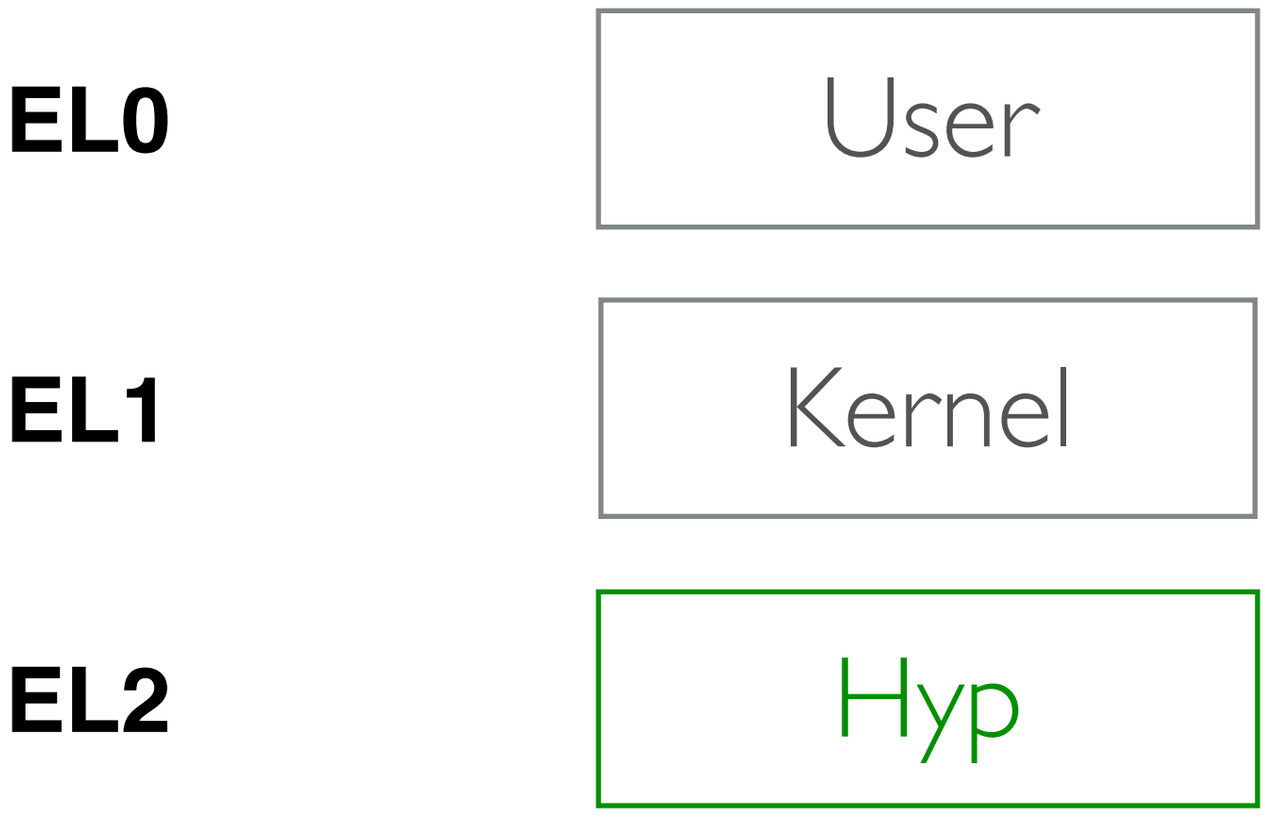
connect
San Francisco 2015

ARM Virtualization Extensions

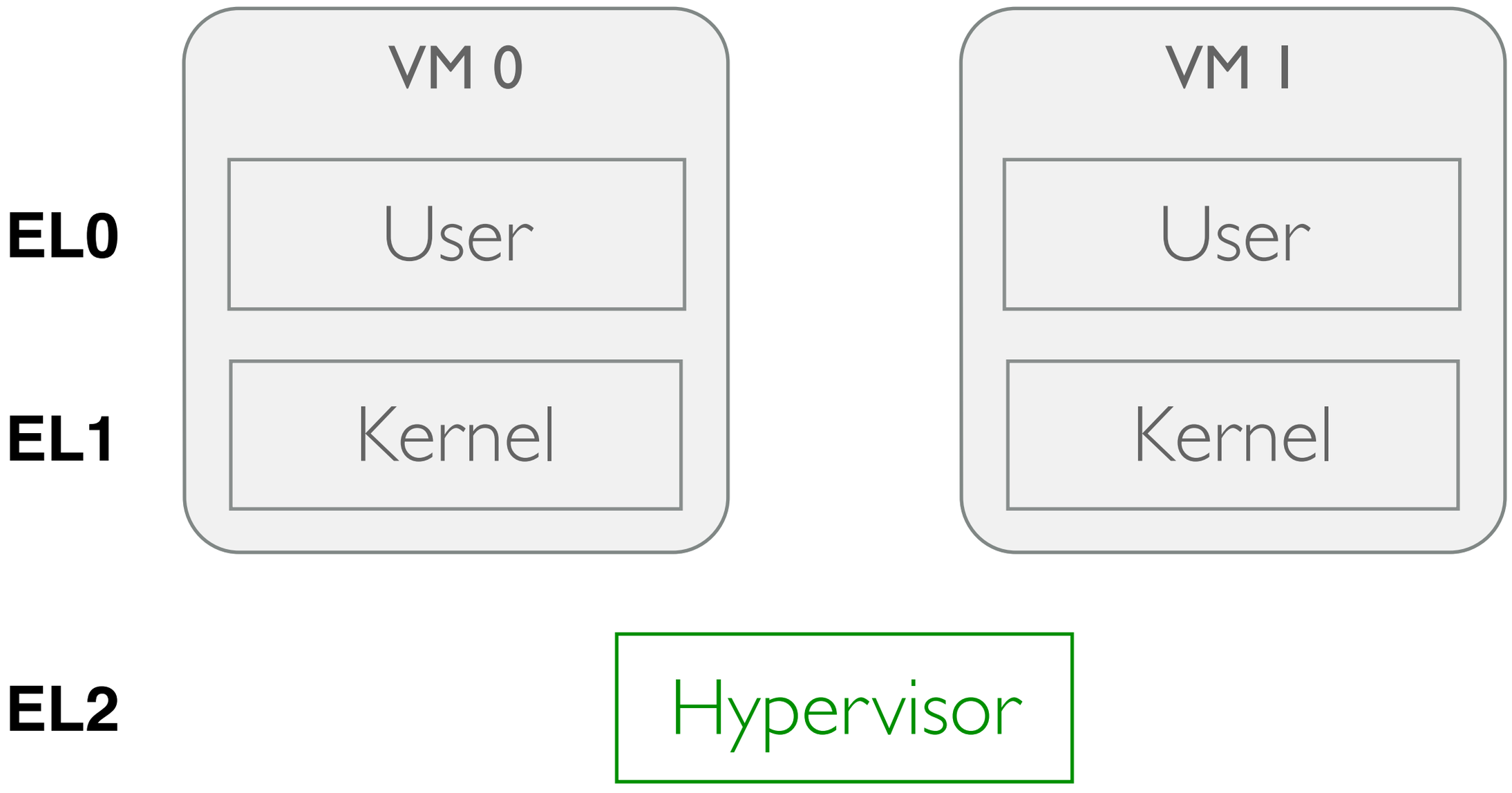
User

Kernel

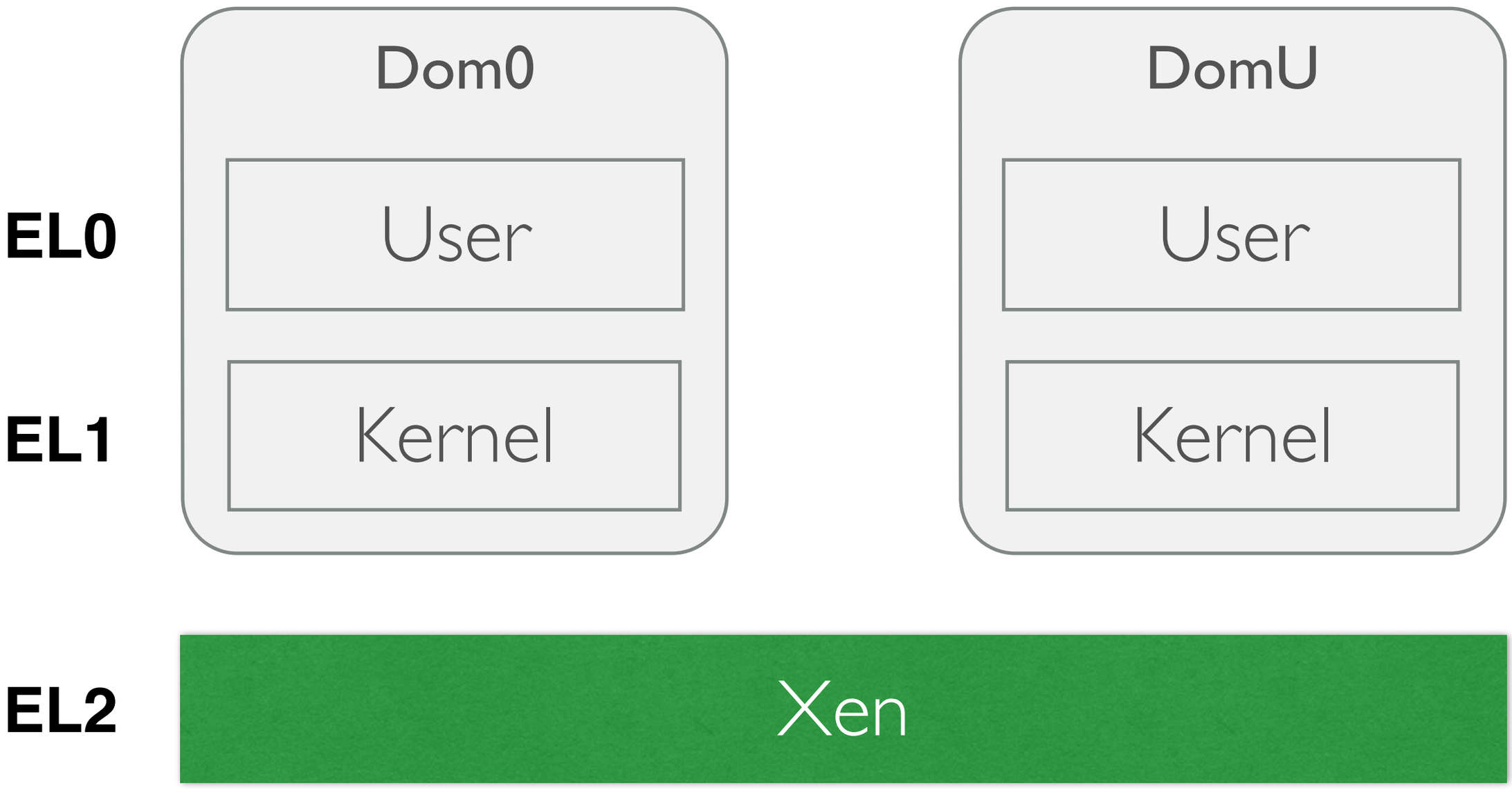
ARM Virtualization Extensions



ARM Virtualization Extensions

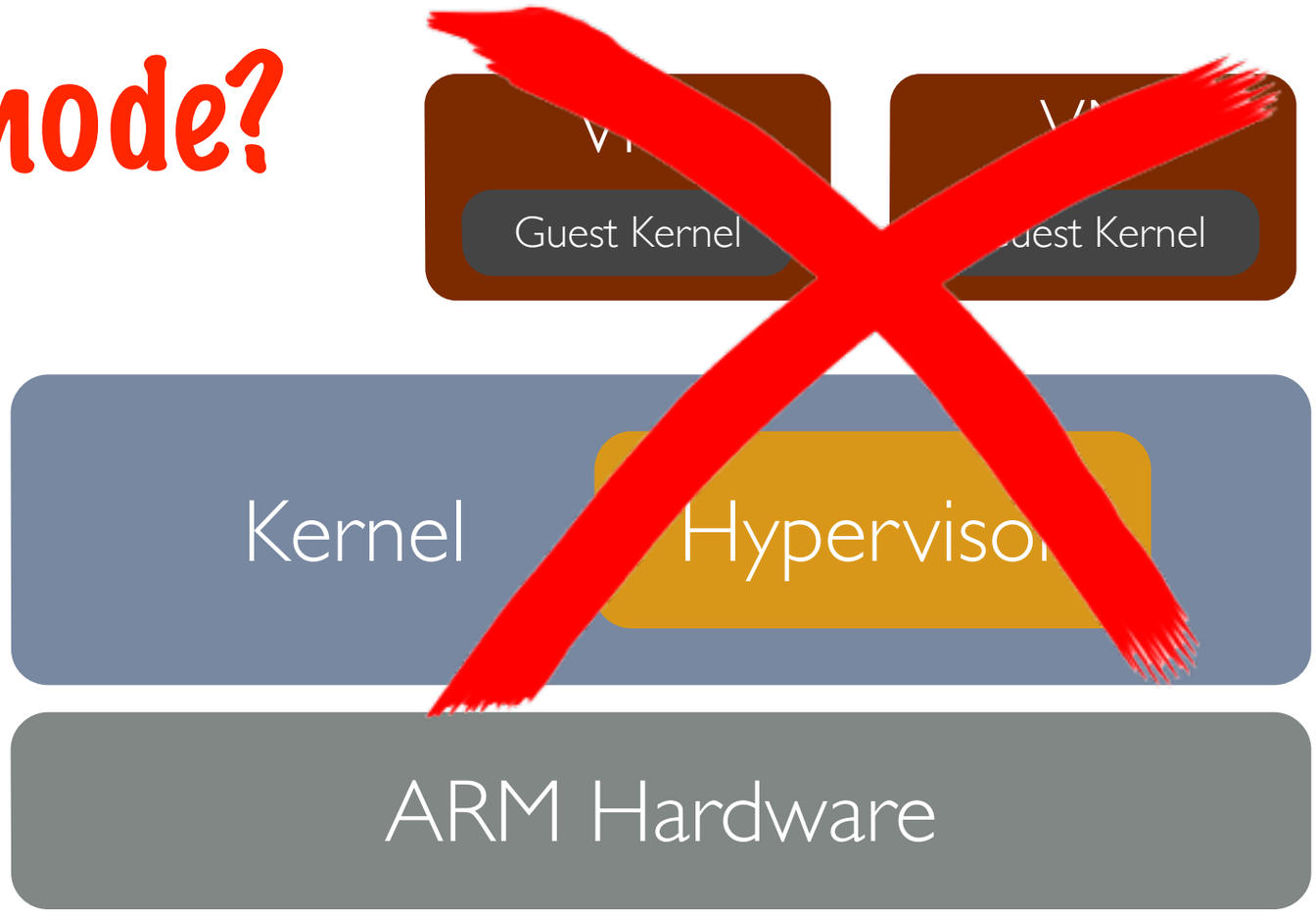


Xen ARM

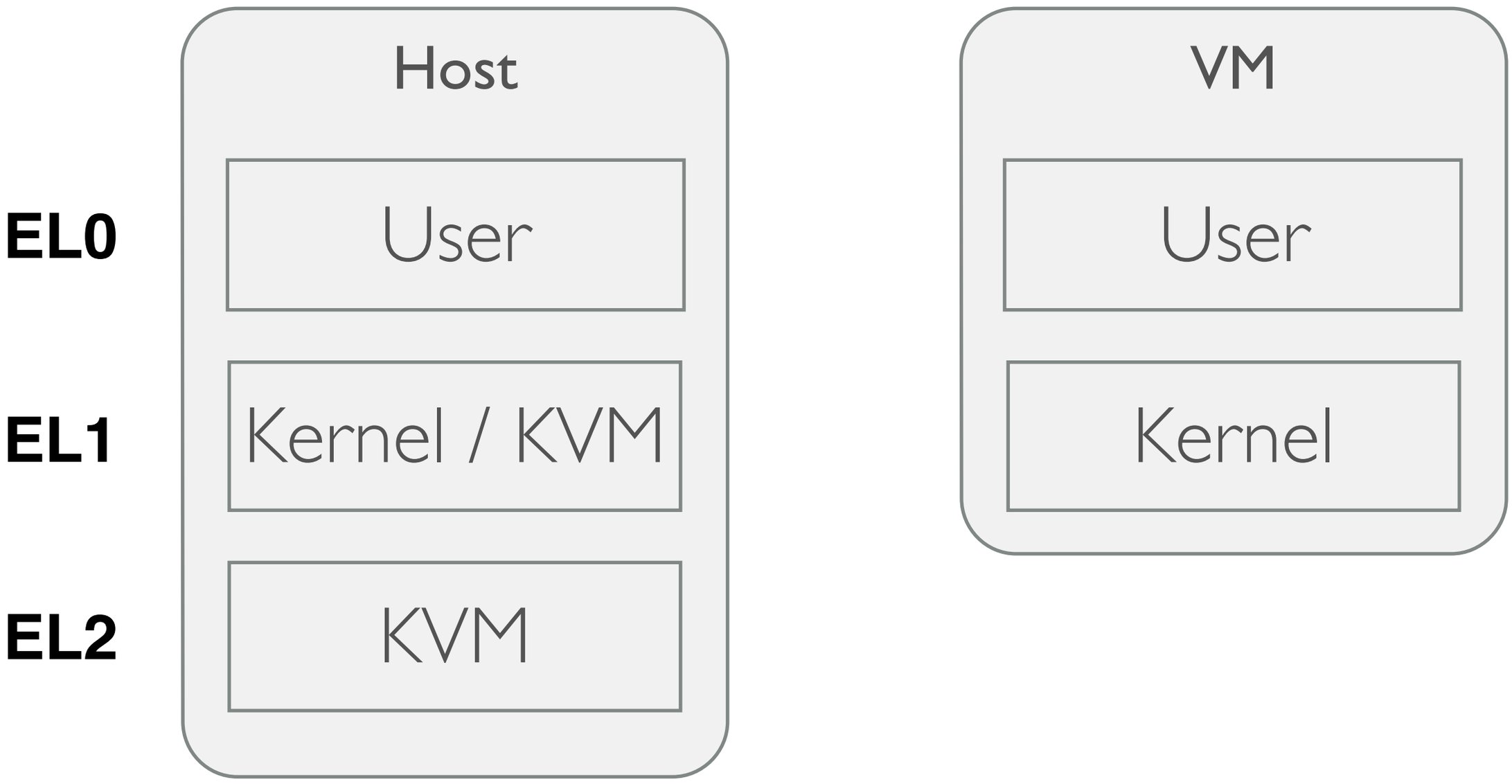


KVM

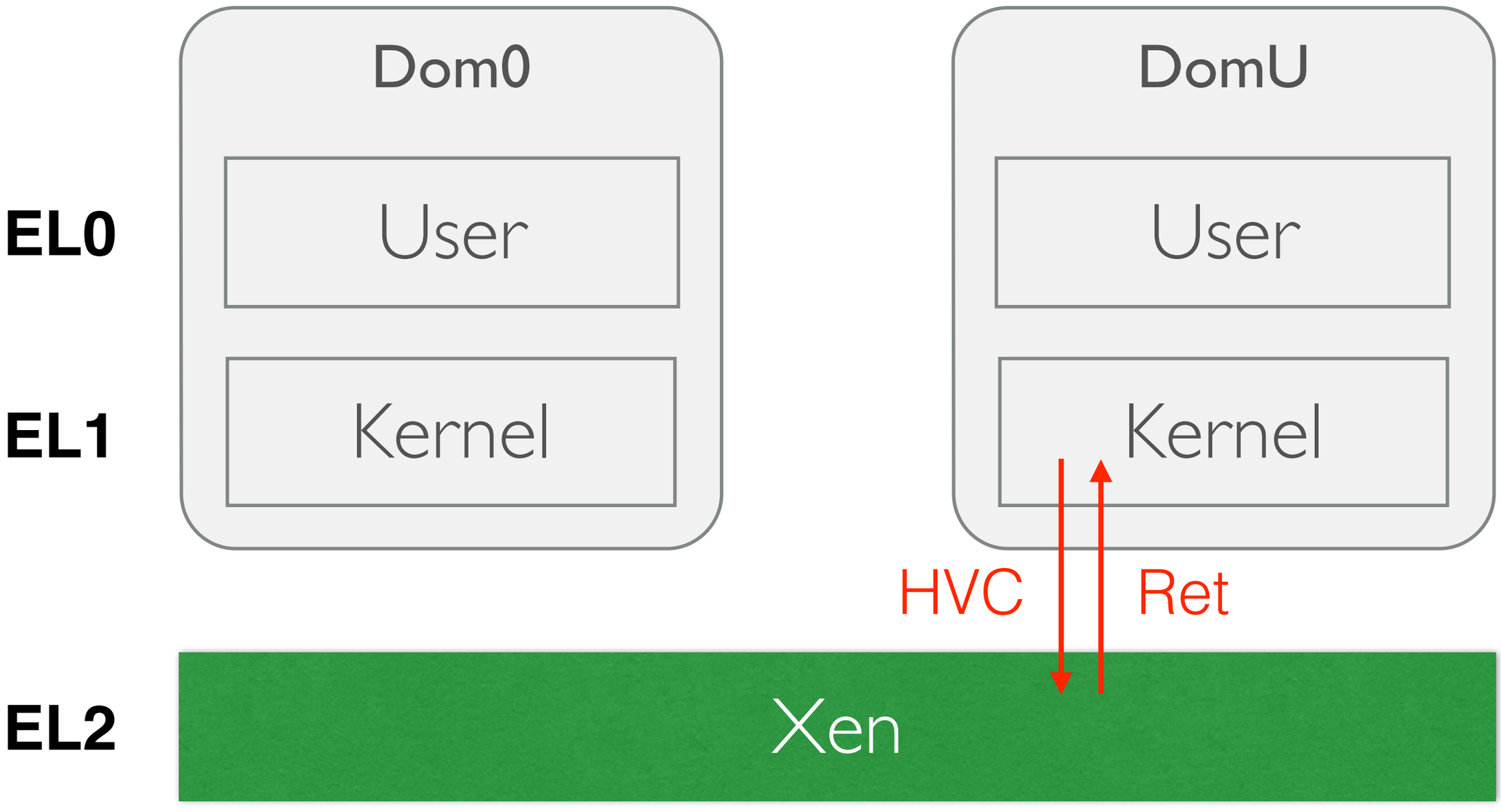
HYP mode?



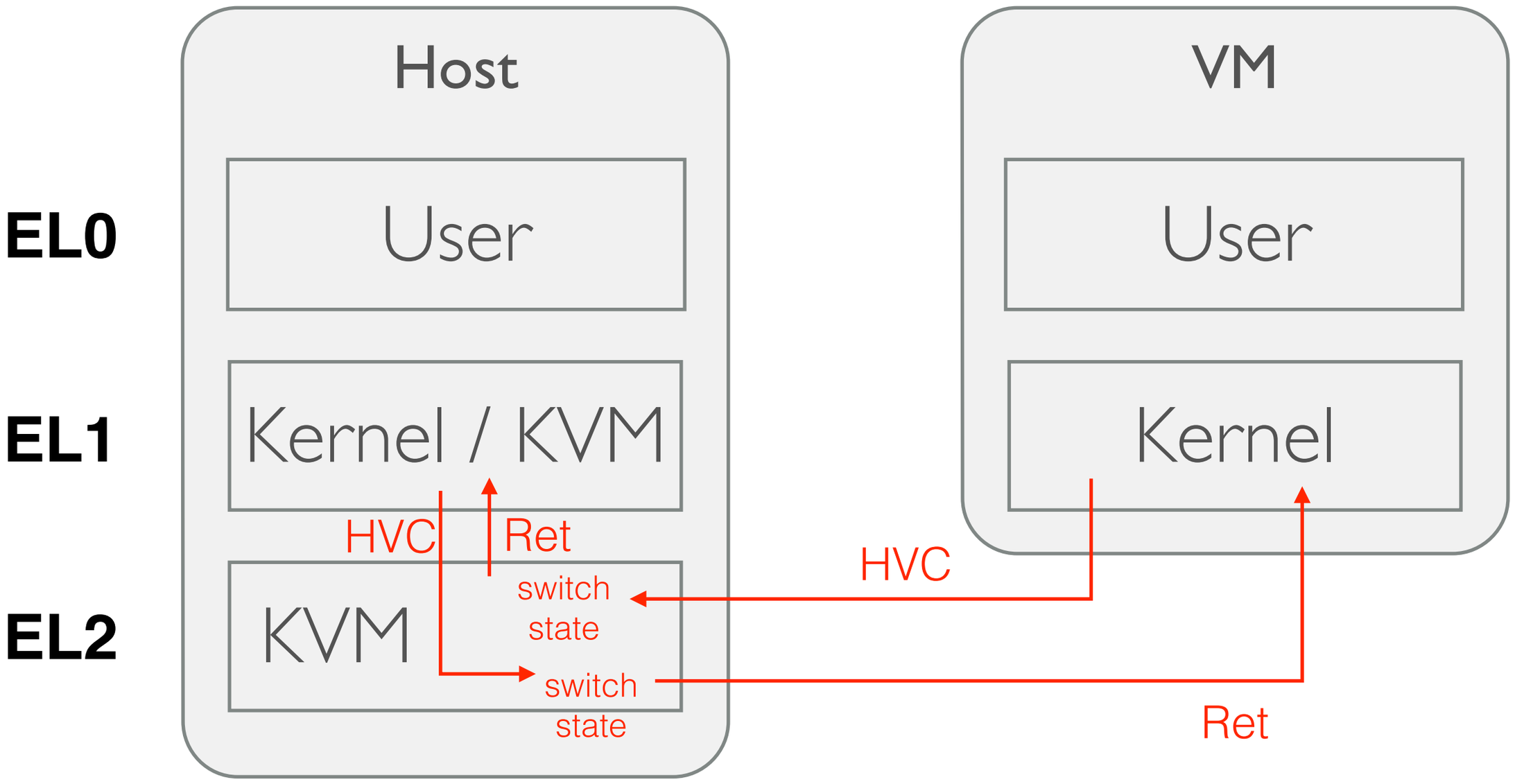
KVM



Hypercall on Xen



Hypercall on KVM



Hypercall Comparison

Xen

1. HVC Instruction
2. Xen handler
3. Return to VM

KVM

1. HVC Instruction
2. Switch EL1 state in KVM EL2
3. Return to host kernel
4. KVM handler
5. HVC Instruction
6. Switch EL1 state in KVM EL2
7. Return to VM


©Warren Photographic

Measurement Methodology

- Compare virtual to native

- CloudLab cluster with both ARM64 servers and x86 servers

Hardware Used

	ARM Server	x86 Server
Type	HP Moonshot m400	Dell r320
CPU	2.4 GHz APM Atlas	2.1 GHz Xeon ES-2450
SMP	8-way	8-way
Memory	64 GB	16 GB
Disk	SATA SSD	7200 RPM SATA HDD
Network	Mellanox ConnectX-3 10GbE	Mellanox MX354A 10GbE

Configuration

- Same configuration across hardware
- Max 12GB of RAM
- Max 4 CPUs
- Hyperthreading disabled on x86

VM Configuration

- 4 vCPUs per VM/host, 8 physical CPUs
- Pin all VCPUs to dedicated PCPUs
- VHOST enabled for KVM

Software Configurations

- Same software version
  - Linux v4.1-rc2+
  - Ubuntu Trusty
  - Same kernel config, manually tweaked x86 and arm64 options

Micro Numbers

Microbenchmark	ARM 64-bit		x86 64-bit	
	KVM	Xen	KVM	Xen
Hypercall	NA	NA	NA	NA
Interrupt Trap	NA	NA	NA	NA
IPI	NA	NA	NA	NA
EOI+ACK	NA	NA	NA	NA
VM Switch	NA	NA	-	-
I/O Latency Out	NA	NA	-	-
I/O Latency In	NA	NA	-	-

All numbers shown in cycles

Hypercall Breakdown

State	KVM ARM	
	Save	Restore
GP Regs	NA	NA
System Regs	NA	NA
FP Regs	NA	NA
VGIC Regs	NA	NA
Timer Regs	NA	NA
EL2 Config Regs	NA	NA
Stage-2 MMU Regs	NA	NA

Save = Save state to Memory

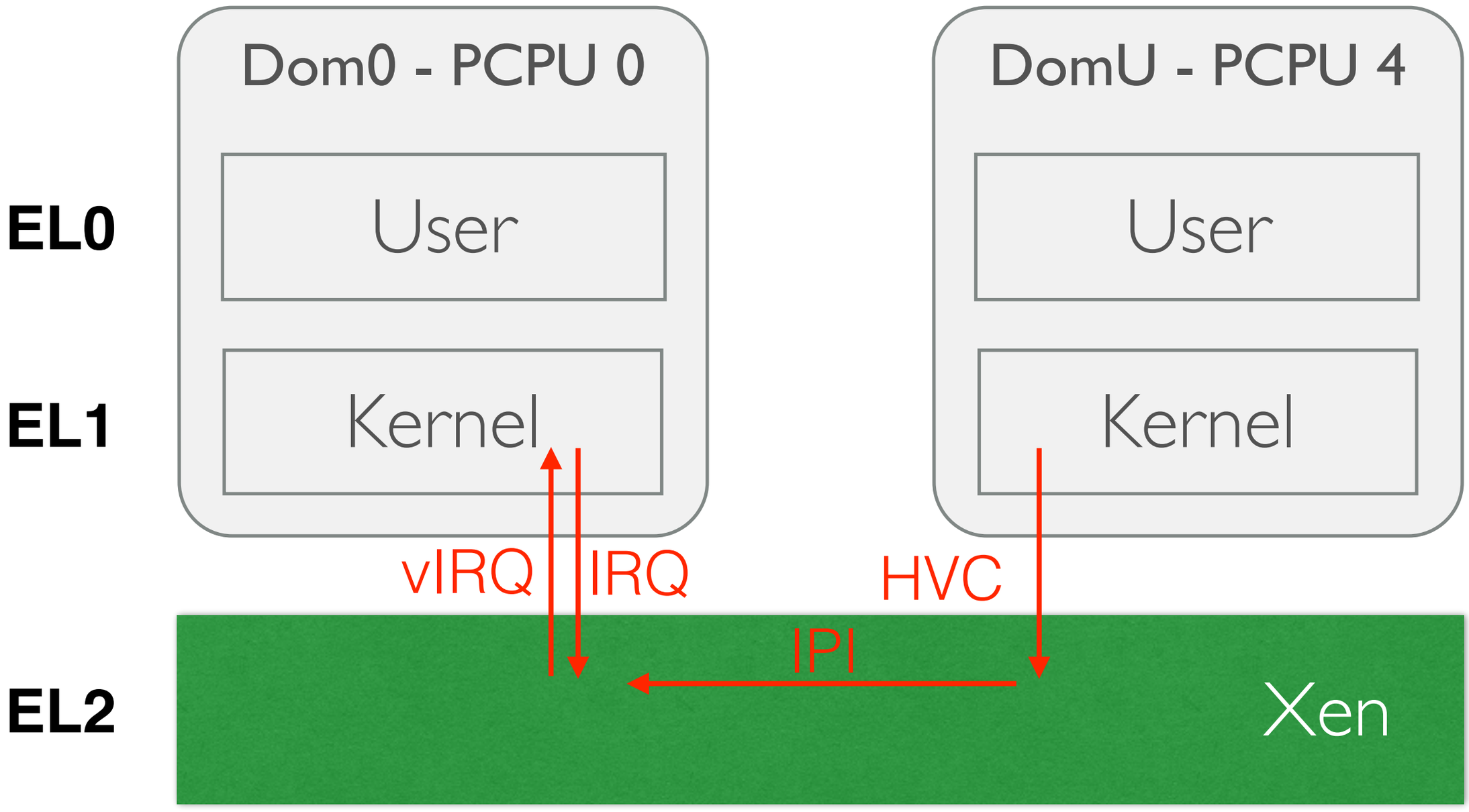
Restore = Restore state From Memory

Micro Numbers

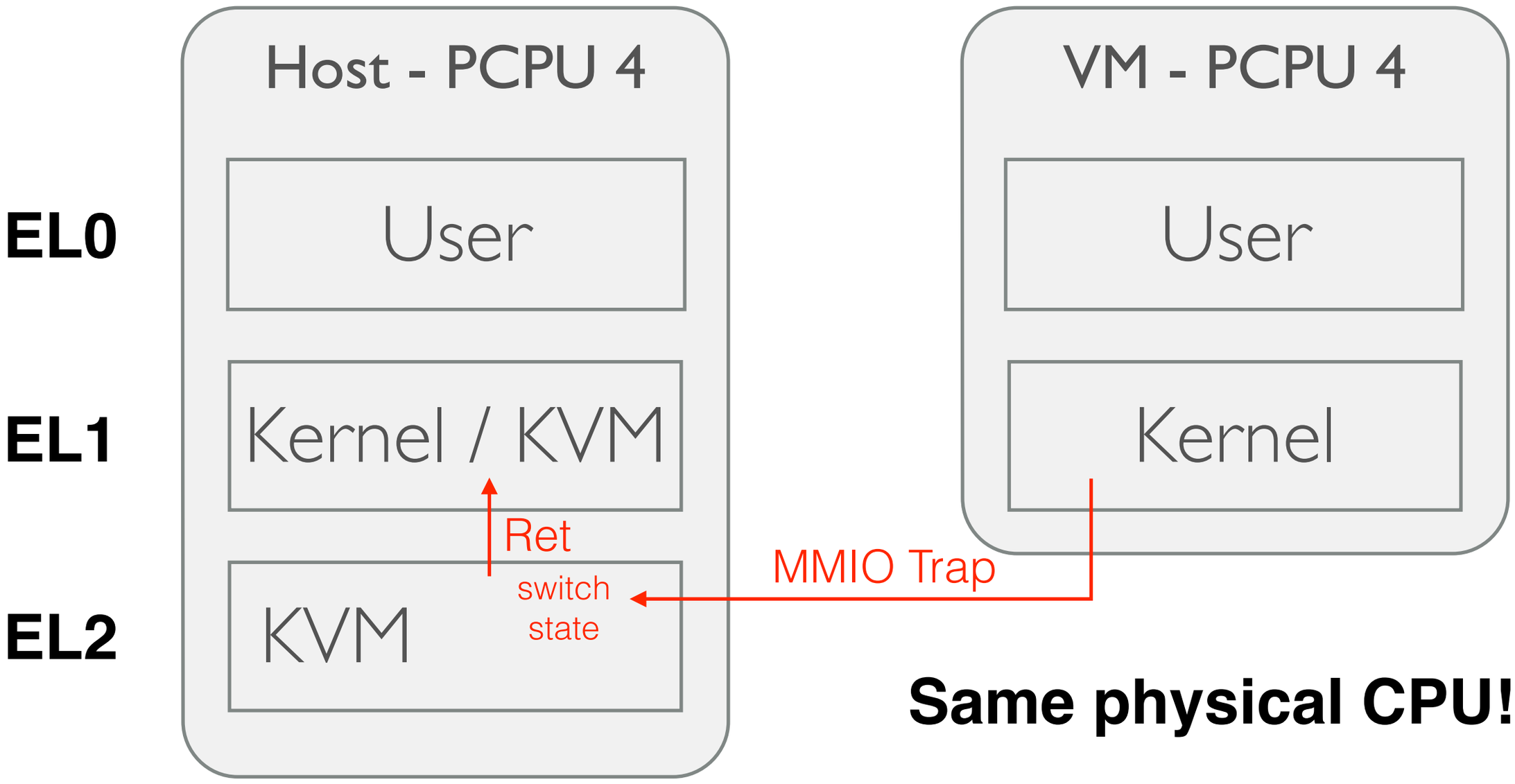
Microbenchmark	ARM 64-bit		x86 64-bit	
	KVM	Xen	KVM	Xen
Hypercall	NA	NA	NA	NA
Interrupt Trap	NA	NA	NA	NA
IPI	NA	NA	NA	NA
EOI+ACK	NA	NA	NA	NA
VM Switch	NA	NA	-	-
I/O Latency Out	NA	NA	-	-
I/O Latency In	NA	NA	-	-

All numbers shown in cycles

I/O Latency Out Xen



I/O Latency Out KVM



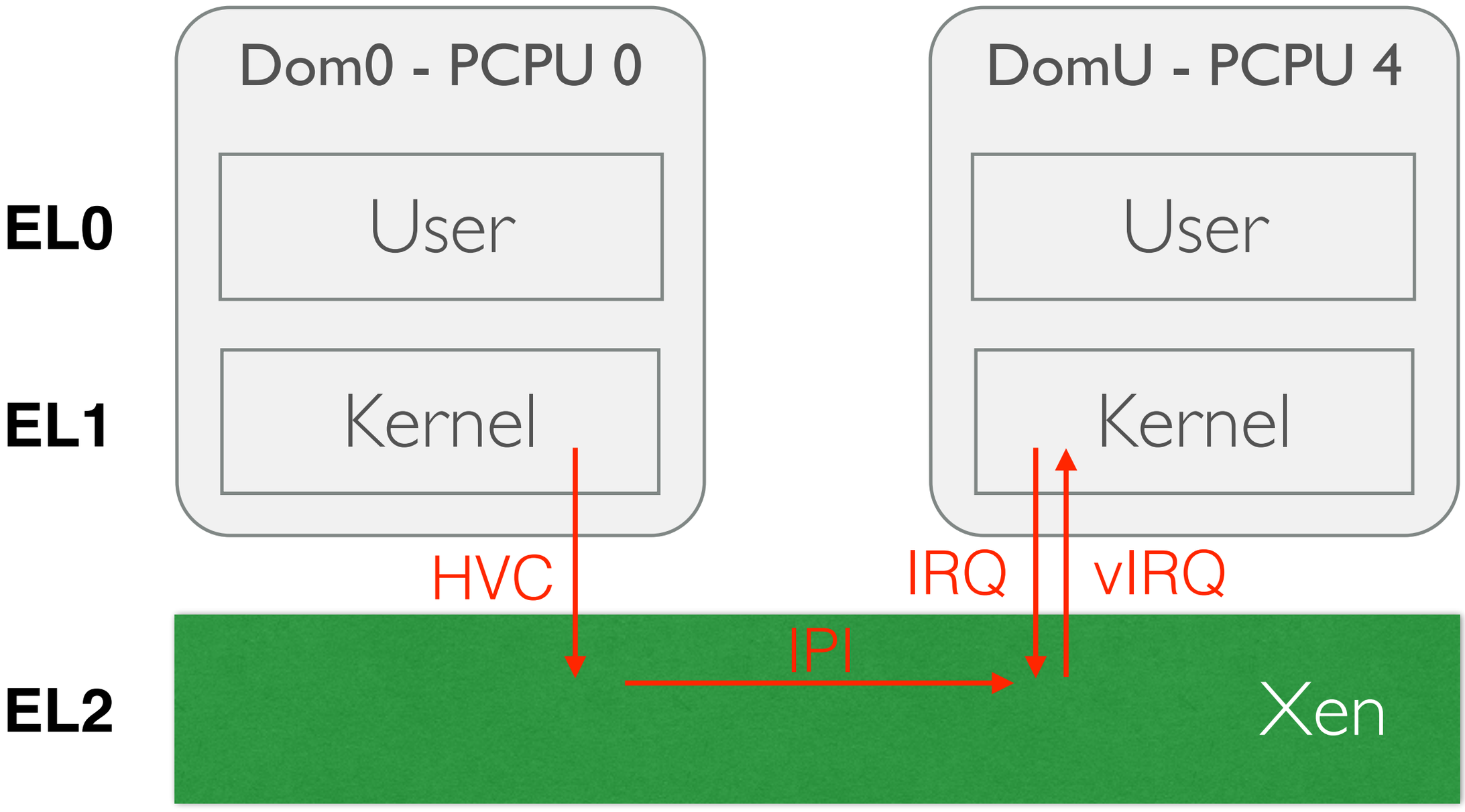
Same physical CPU!

Micro Numbers

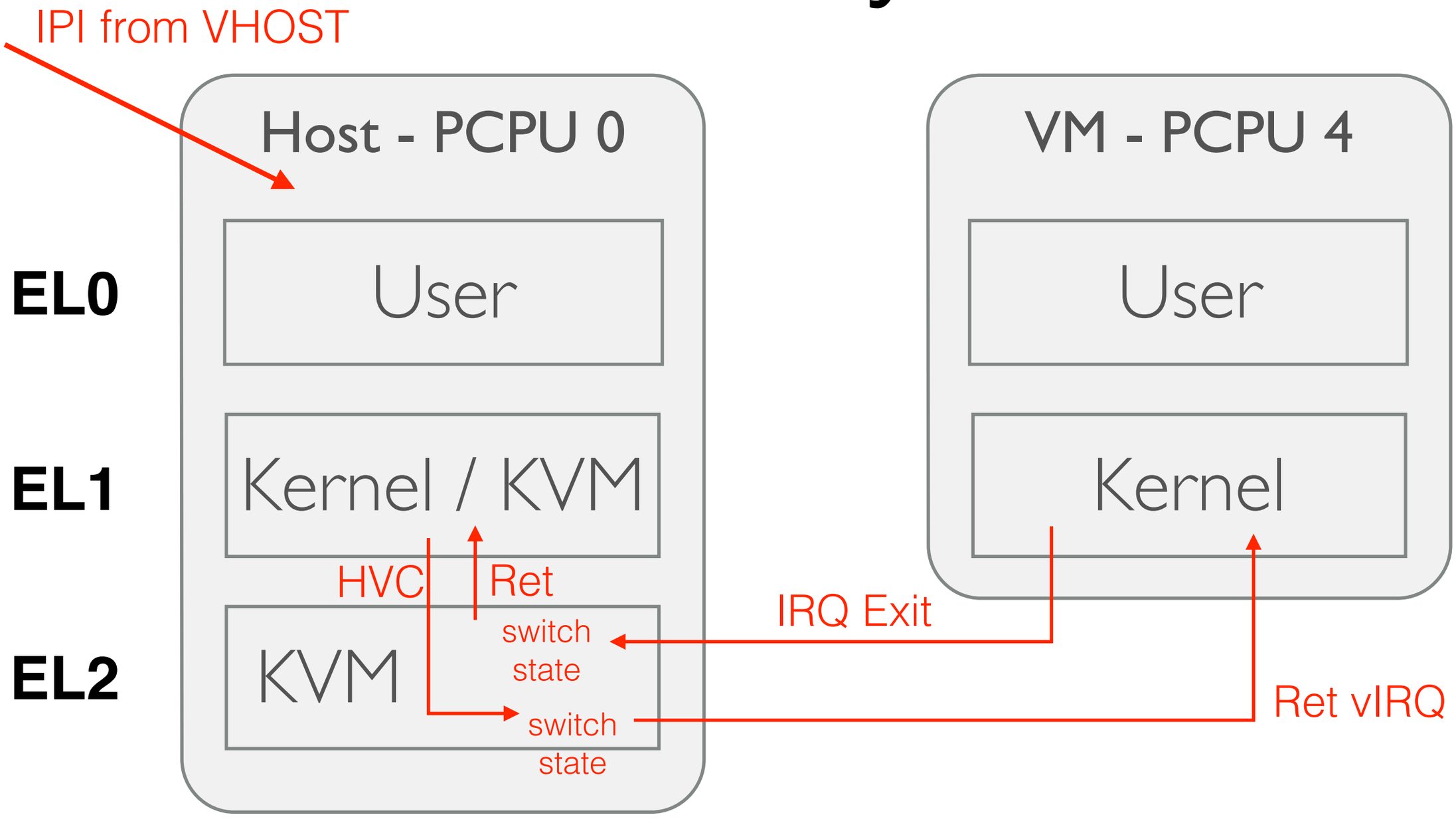
Microbenchmark	ARM 64-bit		x86 64-bit	
	KVM	Xen	KVM	Xen
Hypercall	NA	NA	NA	NA
Interrupt Trap	NA	NA	NA	NA
IPI	NA	NA	NA	NA
EOI+ACK	NA	NA	NA	NA
VM Switch	NA	NA	-	-
I/O Latency Out	NA	NA	-	-
I/O Latency In	NA	NA	-	-

All numbers shown in cycles

I/O Latency In Xen



I/O Latency In KVM



Micro Numbers

Microbenchmark	ARM 64-bit		x86 64-bit	
	KVM	Xen	KVM	Xen
Hypercall	NA	NA	NA	NA
Interrupt Trap	NA	NA	NA	NA
IPI	NA	NA	NA	NA
EOI+ACK	NA	NA	NA	NA
VM Switch	NA	NA	-	-
I/O Latency Out	NA	NA	-	-
I/O Latency In	NA	NA	-	-

All numbers shown in cycles

CPU Intensive Benchmarks

- Kernbench
- Hackbench
- SpecJVM2088

Results are NA

Normalized overhead (lower is better)

Netperf

- TCP_STREAM
- TCP_MAERTS
- TCP_RR

Results are NA

Normalized overhead (lower is better)

Netperf Study

- TCP_STREAM sends bulk data from client to VM
- Xen does not support zero-copy

Netperf Study

- TCP_MAERTS sends bulk data from VM to client

- Xen performance is regression in Linux v4.0 from patch to fight buffer bloat

- Can be reduced to XX% by tuning sysfs

Netperf Study

- TCP_RR sends byte-by-byte on open connection

	Native	KVM	Xen
Trans/sec	NA	NA	NA
Time/trans	NA	NA	NA
Overhead	-	NA	NA
recv to send	NA	NA	NA
VM recv to VM send		NA	NA
recv to VM recv	-	NA	NA
VM send to send		NA	NA

Numbers in µseconds

Application Benchmarks

- Apache
- memcached
- MySQL 20 Threads

Results are NA

Normalized overhead (lower is better)

Conclusions

- Despite better hypercall performance, Xen does not necessarily outperform KVM on ARM.

- ARM servers do not exhibit worse overhead than x86 and is a viable choice.

- Latency is significant with paravirtualized I/O

Future Work

- Further application benchmark analysis
- Device Assignment
- Upstream support for micro-benchmarks
- Automation and regression monitoring