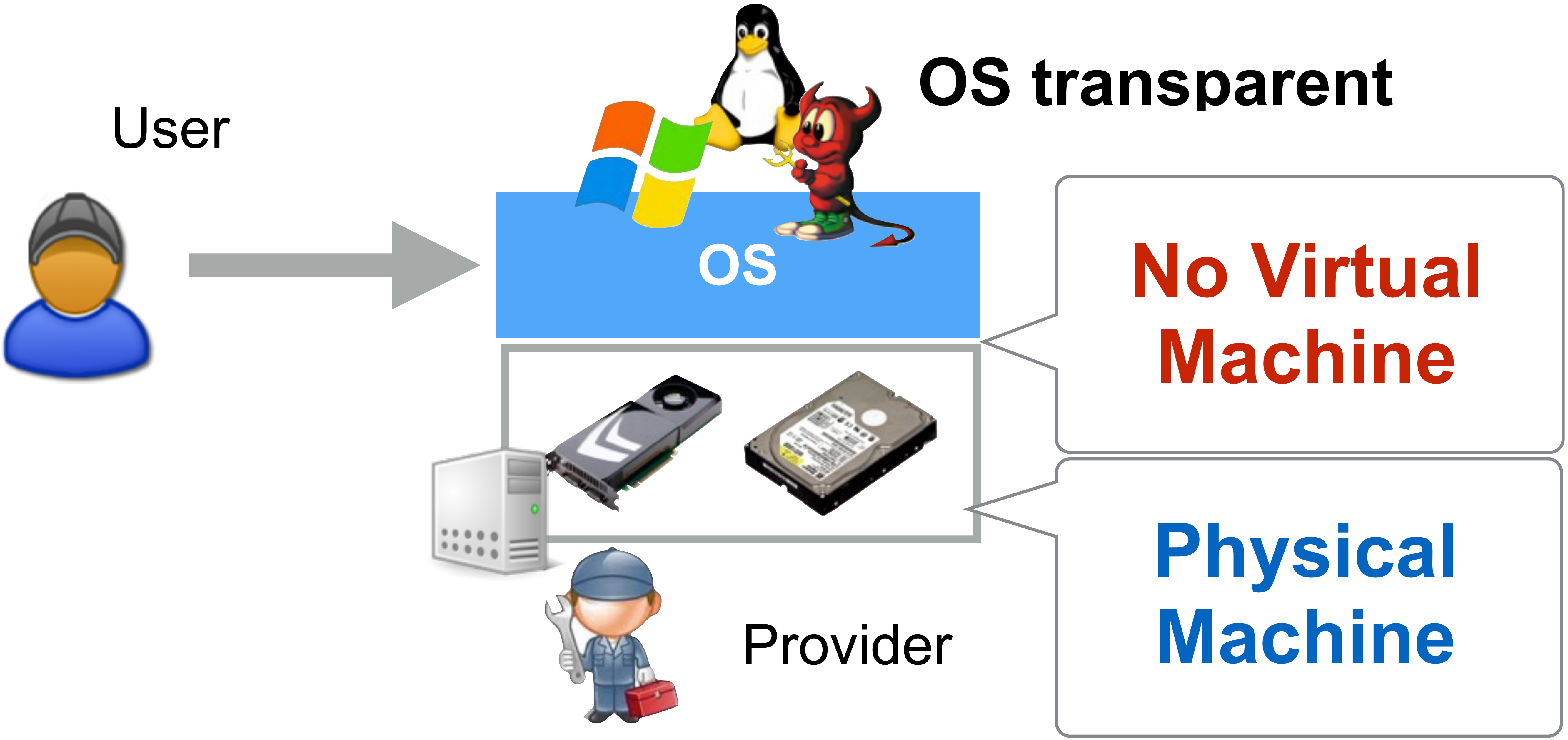
Improving Agility and Elasticity in Bare-metal Clouds

Yushi Omote[†], Takahiro Shinagawa[‡], Kazuhiko Kato[†]

[†]University of Tsukuba, [‡]The University of Tokyo

Bare-metal Clouds

An IaaS for high performance and device functionality



User

OS transparent

OS

No Virtual Machine

Physical Machine

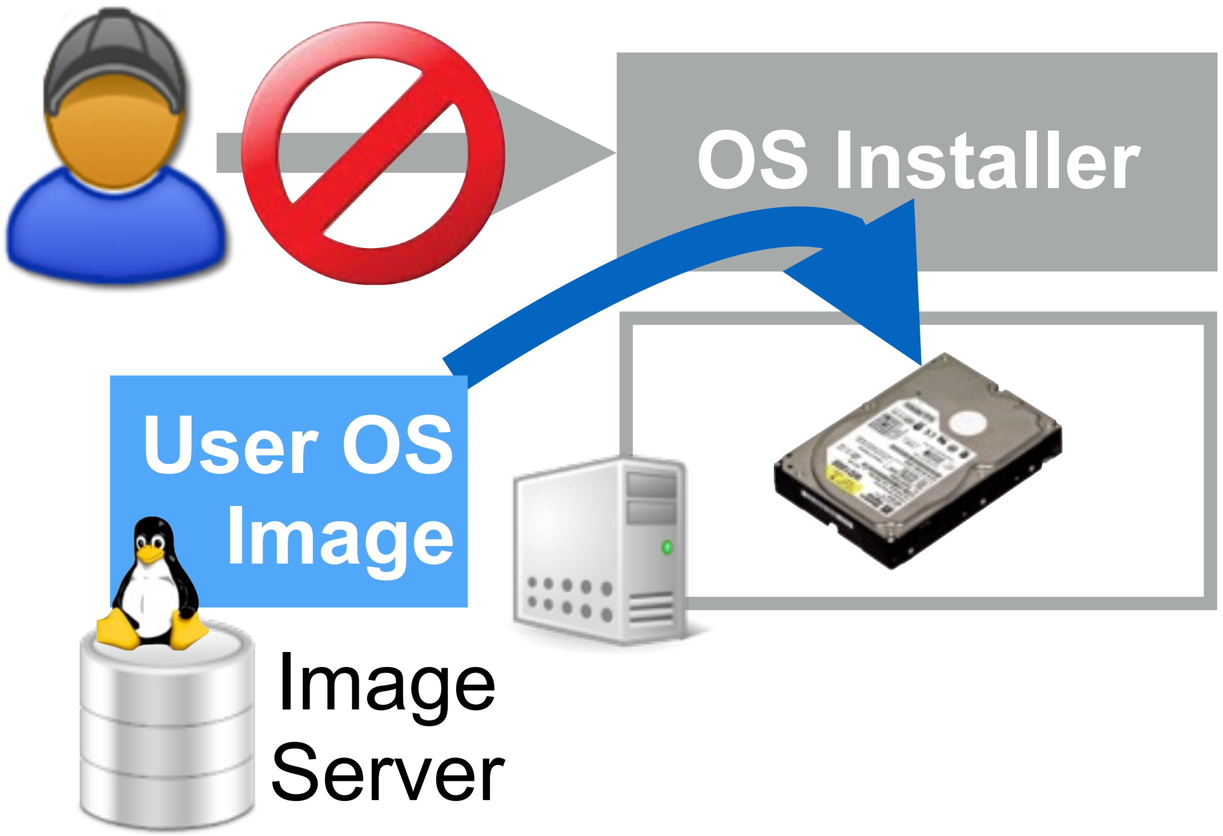
Provider

OS-deployment Problem

Long wait time sacrifices agility and elasticity

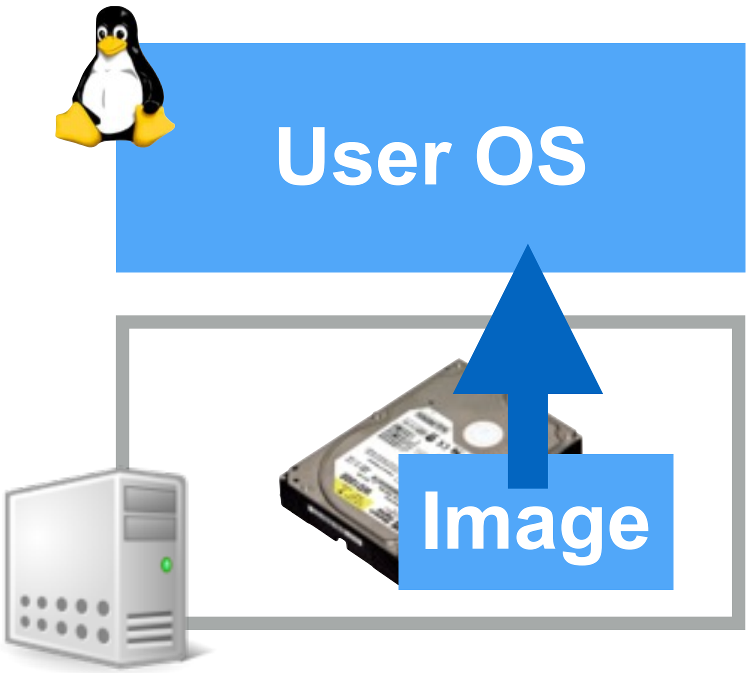
(1) Image Copy

(Tens of minutes)



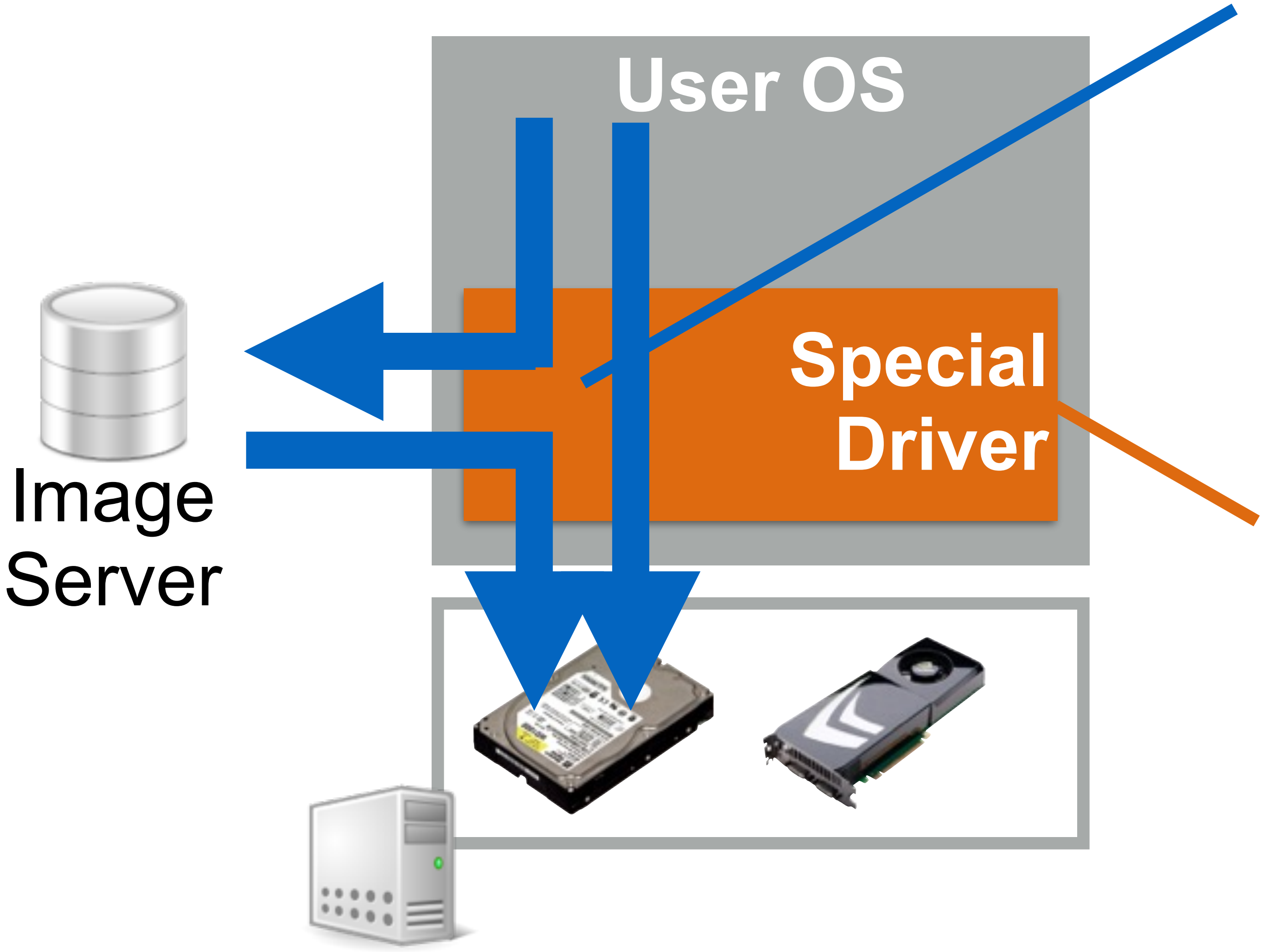
(2) Reboot from Local Disk

(A few minutes)



Existing Approach 1
OS Streaming Deployment

[Clerc et al. IPCCC'10]



Network Boot
+ Background Copy

○ Agility and Elasticity

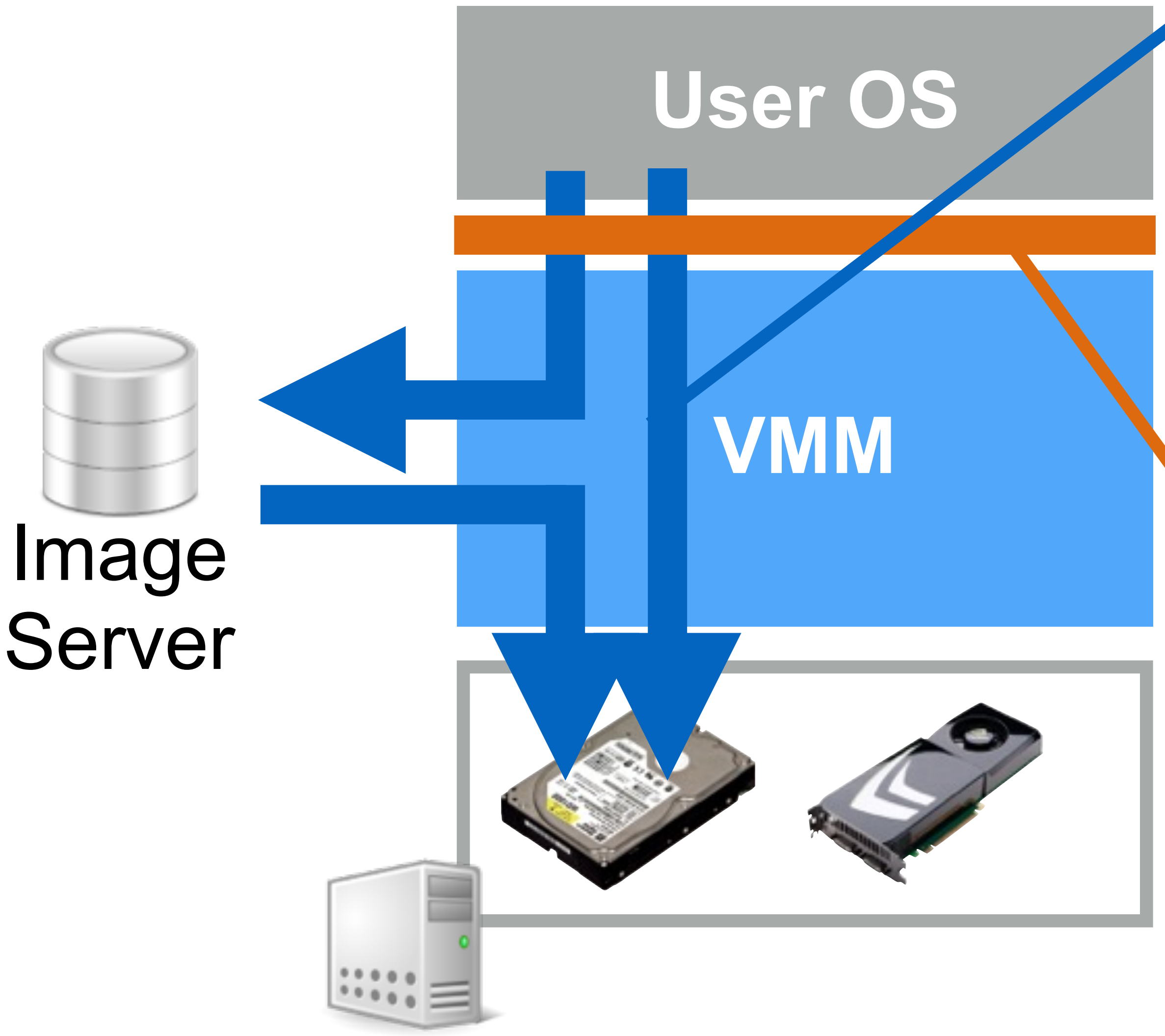
○ Performance

OS-specific drivers are required.

✗ OS transparency

Existing Approach 2
Conventional VMMs

[VMware'01, Xen'03, KVM'07]



Streaming deployment with VMMs

○ Agility and Elasticity

○ OS transparency

Continuous virtualization overhead

✗ Performance

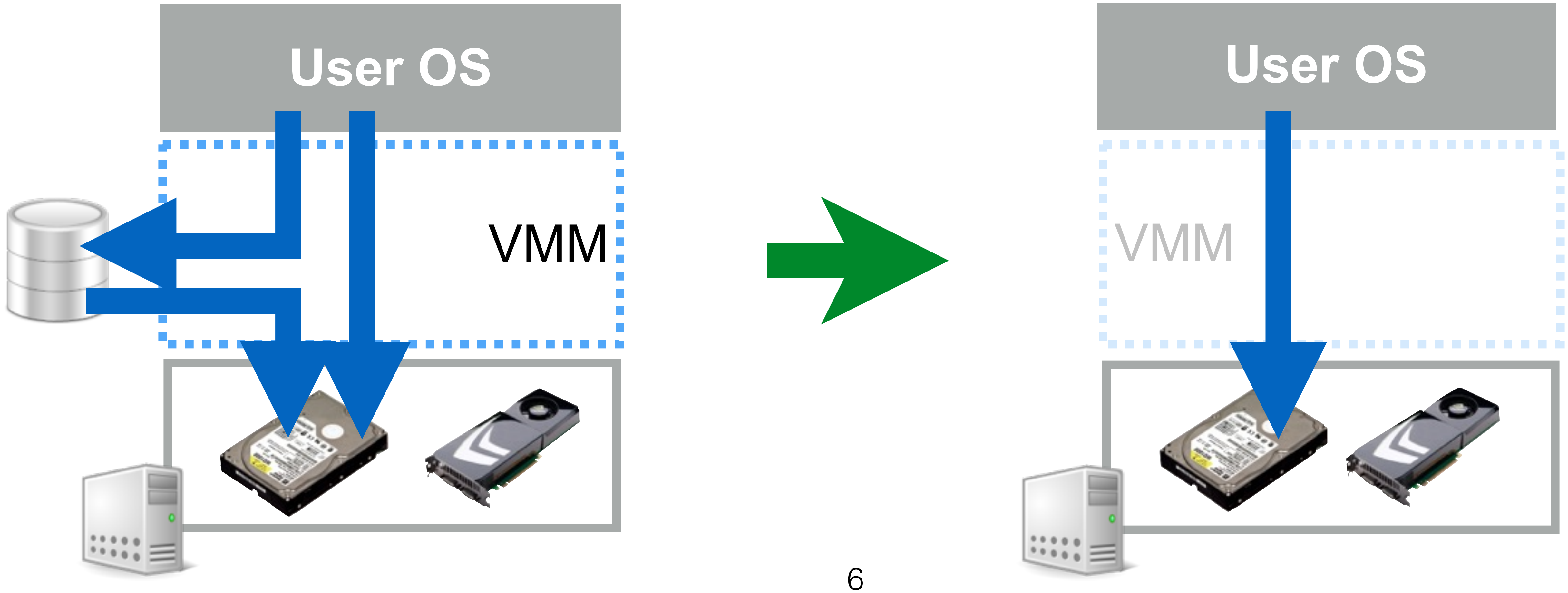
OS Deployment with a Special-purpose VMM

1) Streaming deployment

- ○ Agility and Elasticity
- ○ OS transparency

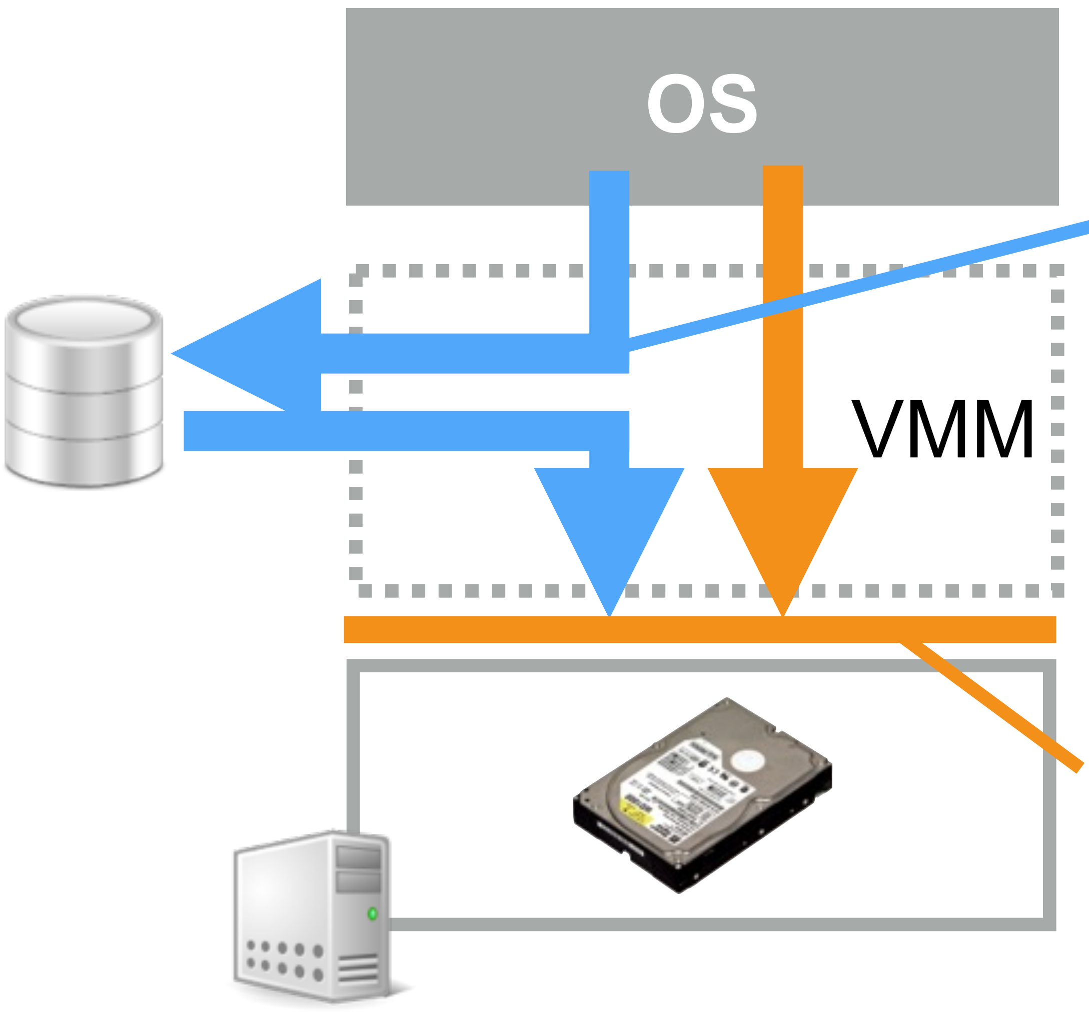
2) Seamless de-virtualization

- ○ Performance



Challenge

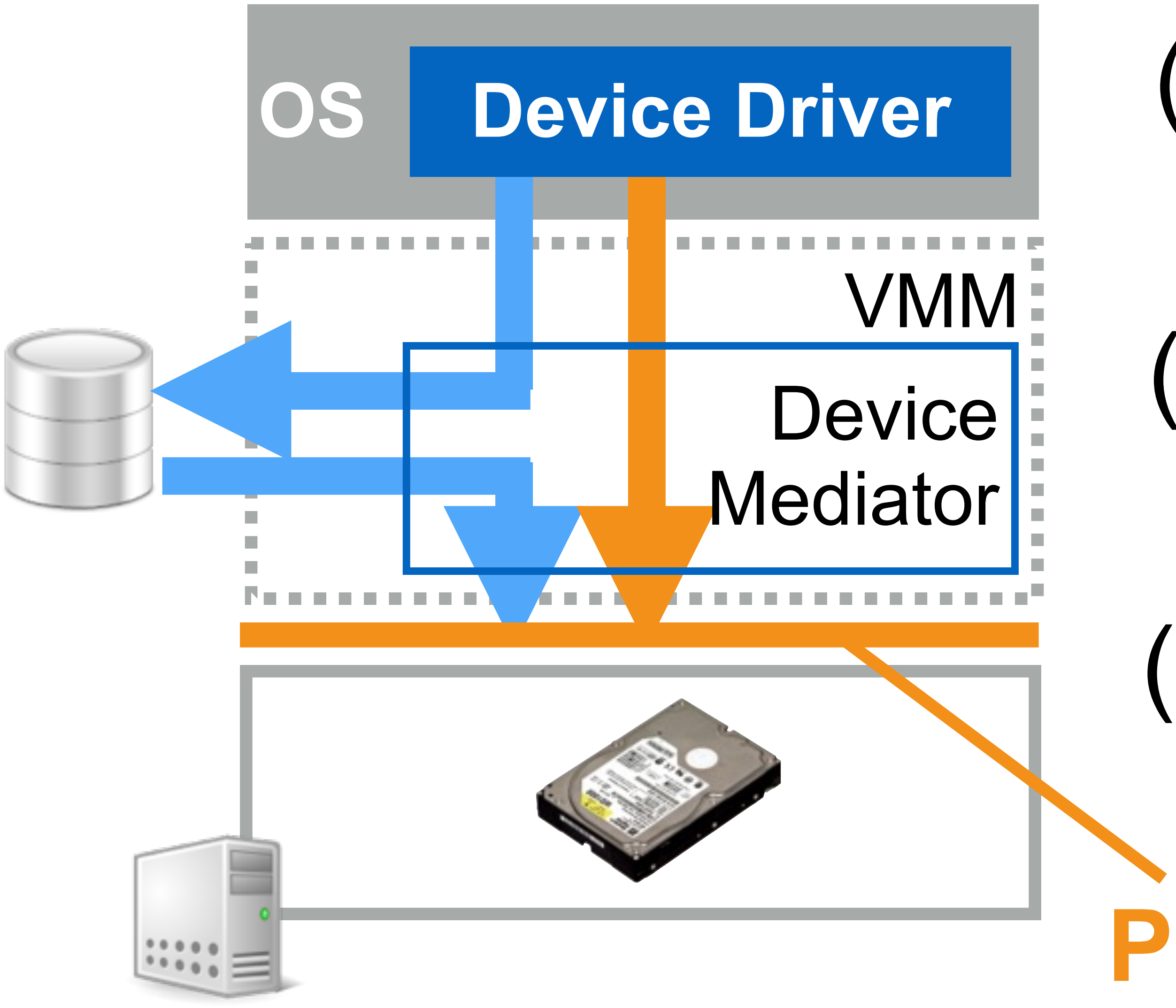
Expose & Control Physical Devices



	Virtual Devices?	Direct I/O?
Control I/Os	○	✗
Expose physical interface	✗	○

Device-interface-level I/O mediation

A device mediator performs:



(1) I/O interpretation
to understand I/O context

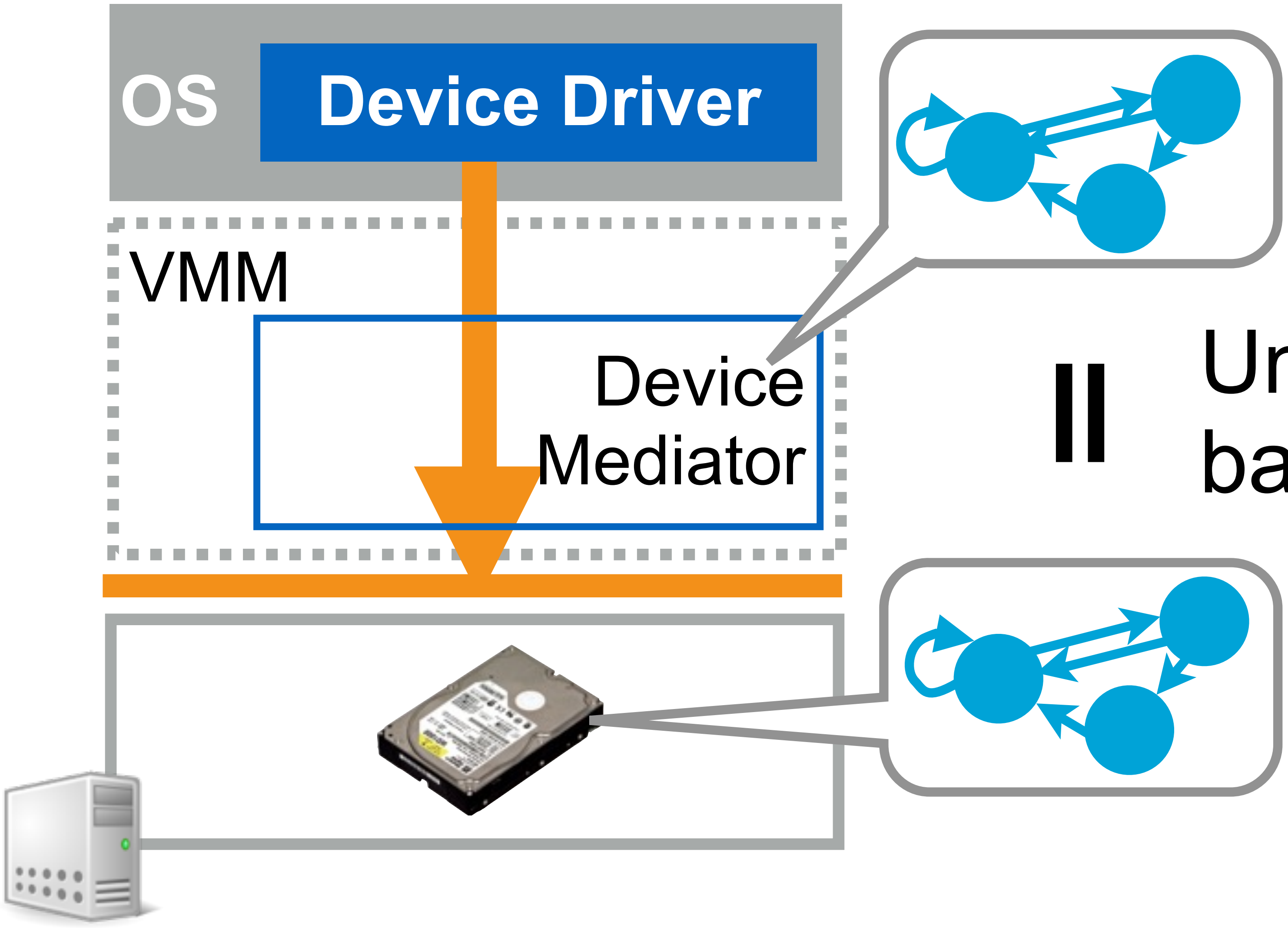
(2) I/O redirection
to perform network booting

(3) I/O multiplexing
to perform background install

Physical device interface

I/O Interpretation

Determine when/how to mediate I/O requests

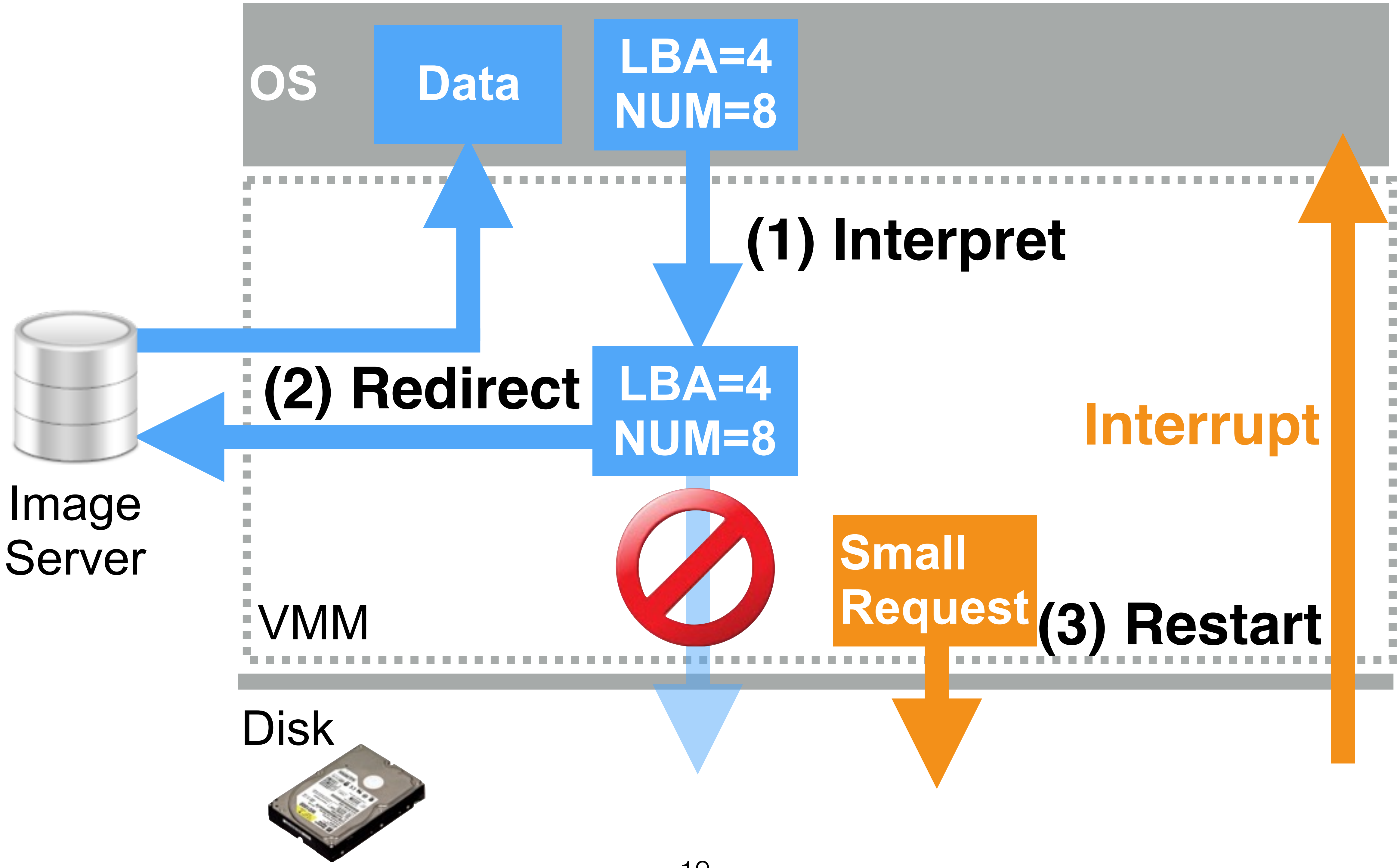


=

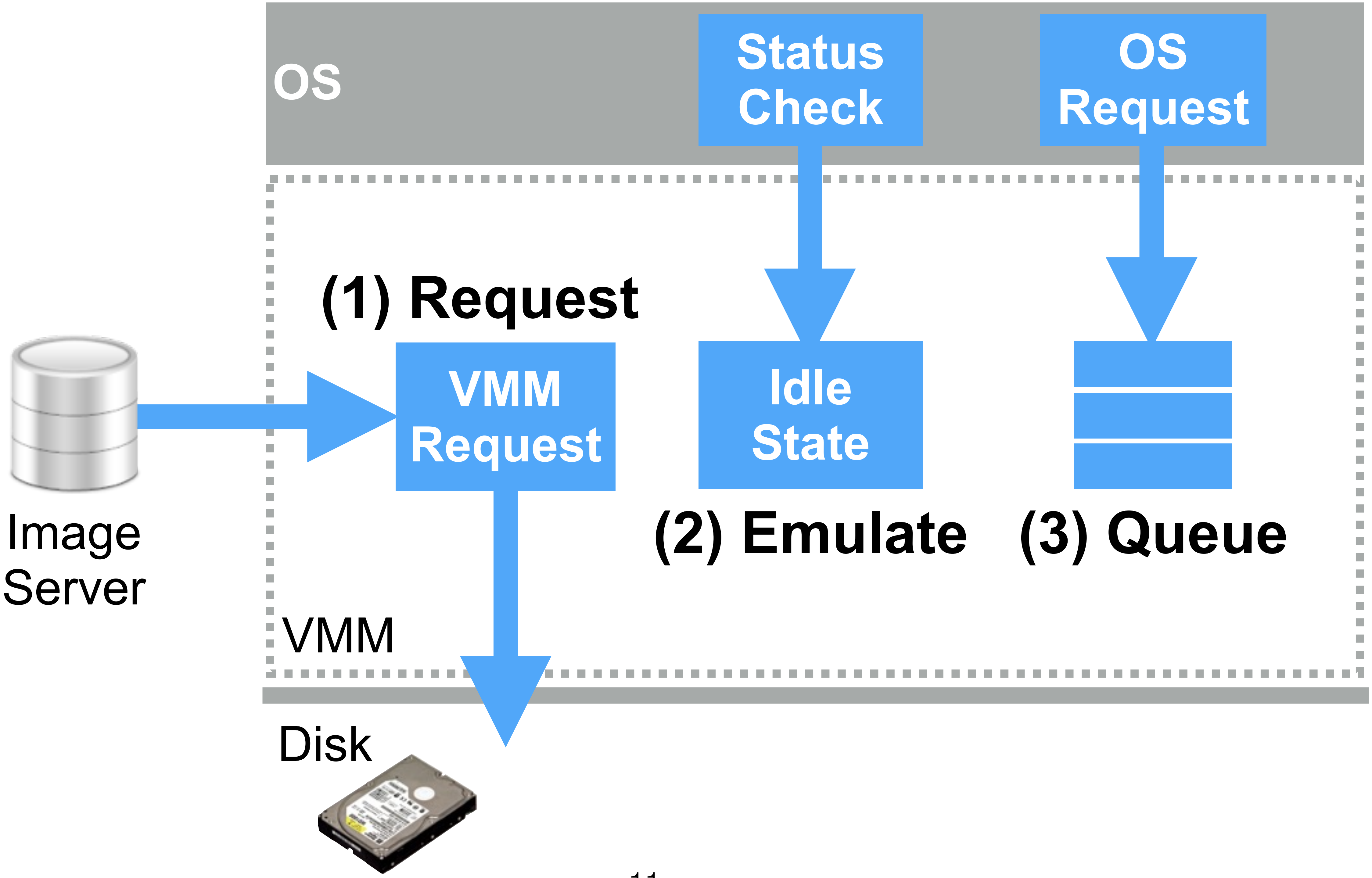
Understand state transitions based on monitoring I/O

Device State Transitions

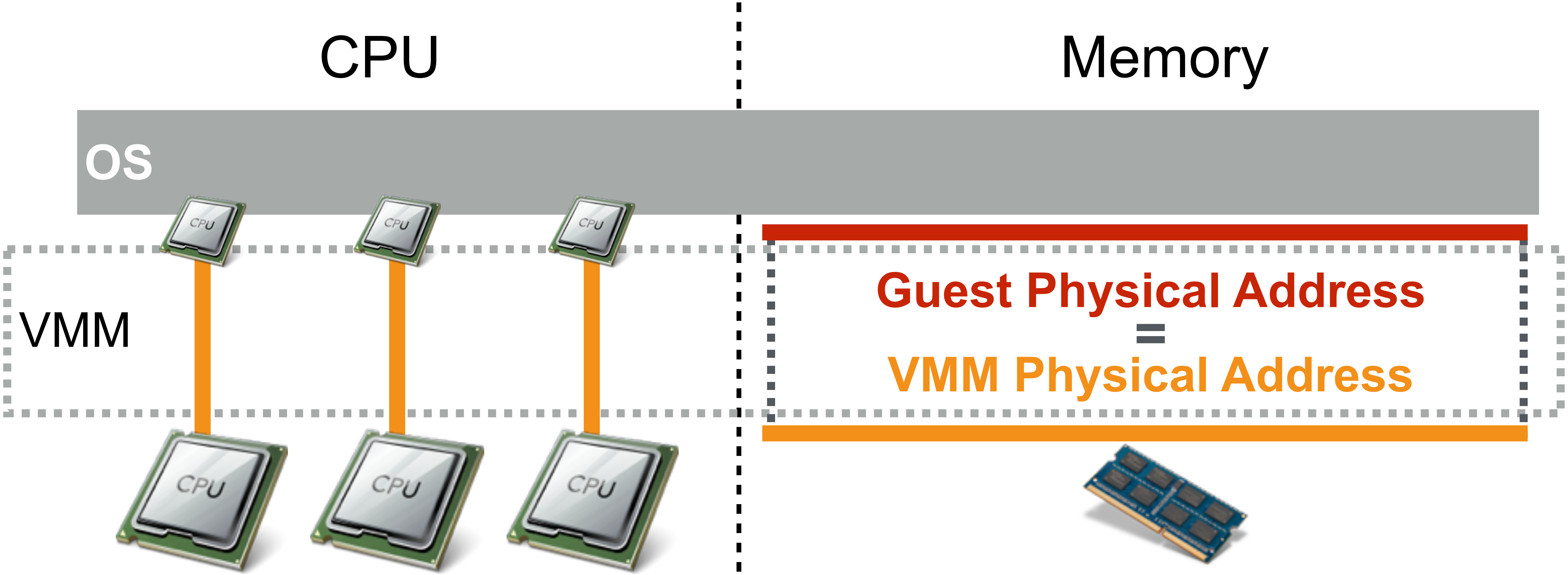
I/O Redirection



I/O Multiplexing



CPU/Memory Virtualization for De-virtualizable VMM



No indirection

VMM runs passively with VMX

No guest scheduling

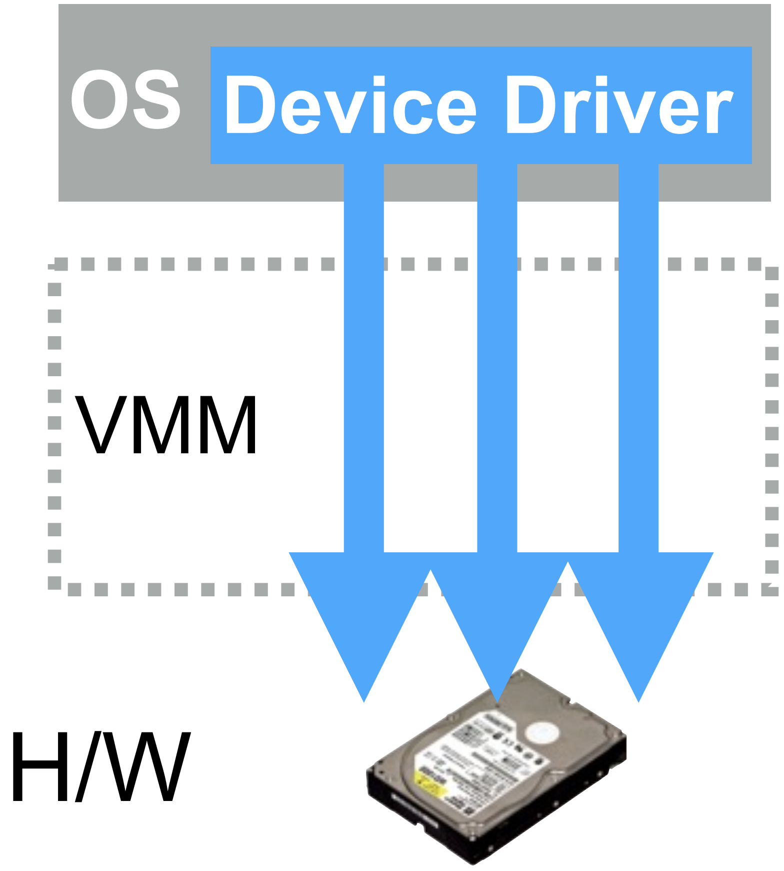
Identity Mapping

VMM exposes physical memory

Mark VMM regions as *reserved* (via BIOS INT15/e802)

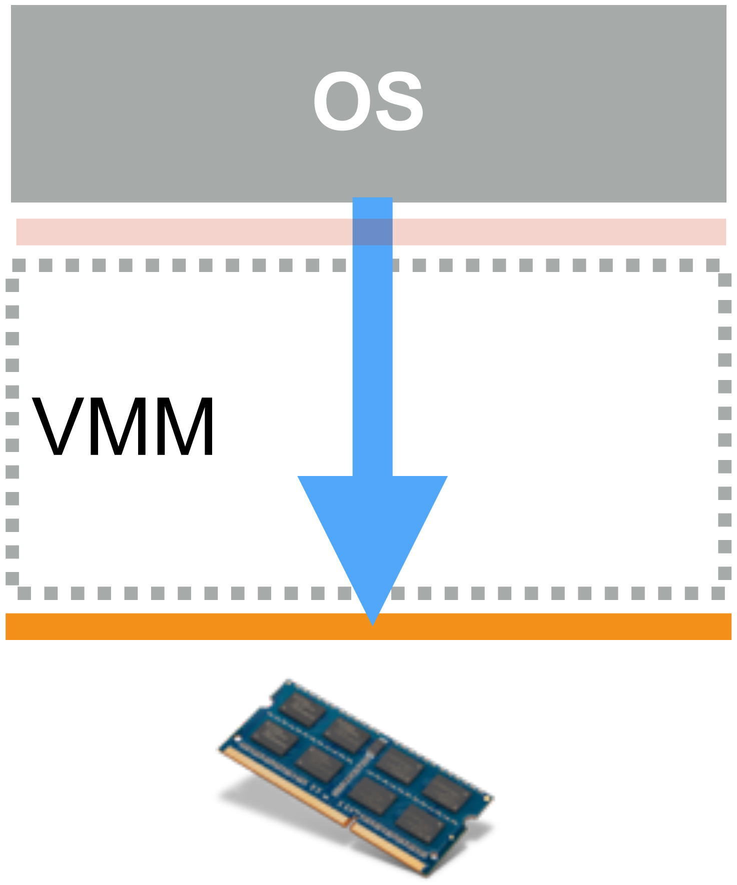
De-virtualization

(1) Turns off IO VM exits



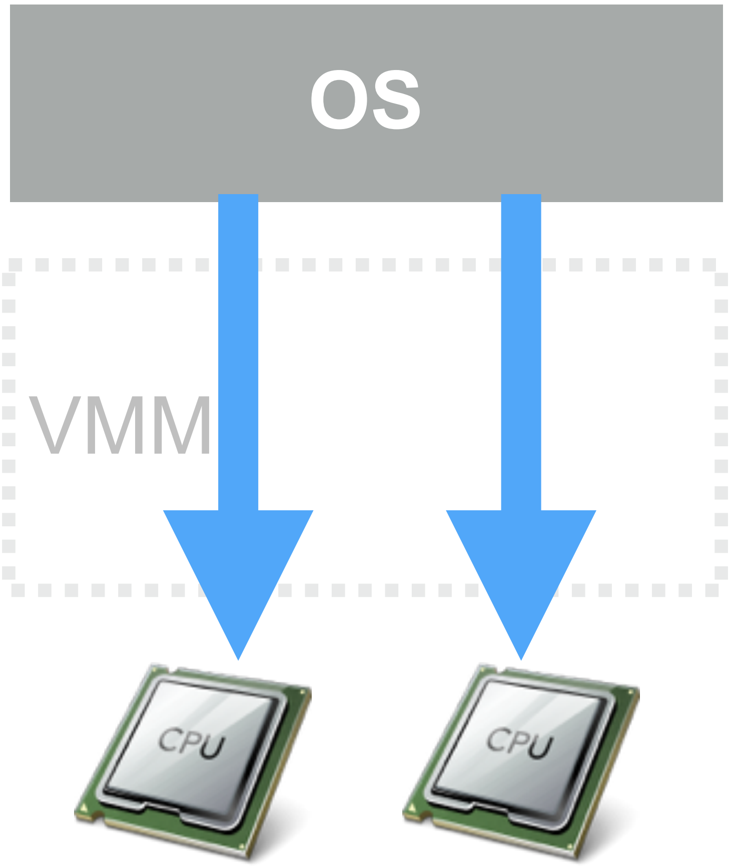
Find safe I/O timing

(2) Turns off nested paging



Unsynchronized TLB flush

(3) Turns off CPU virtualization



Ease VM exits condition (VMXOFF Issue)

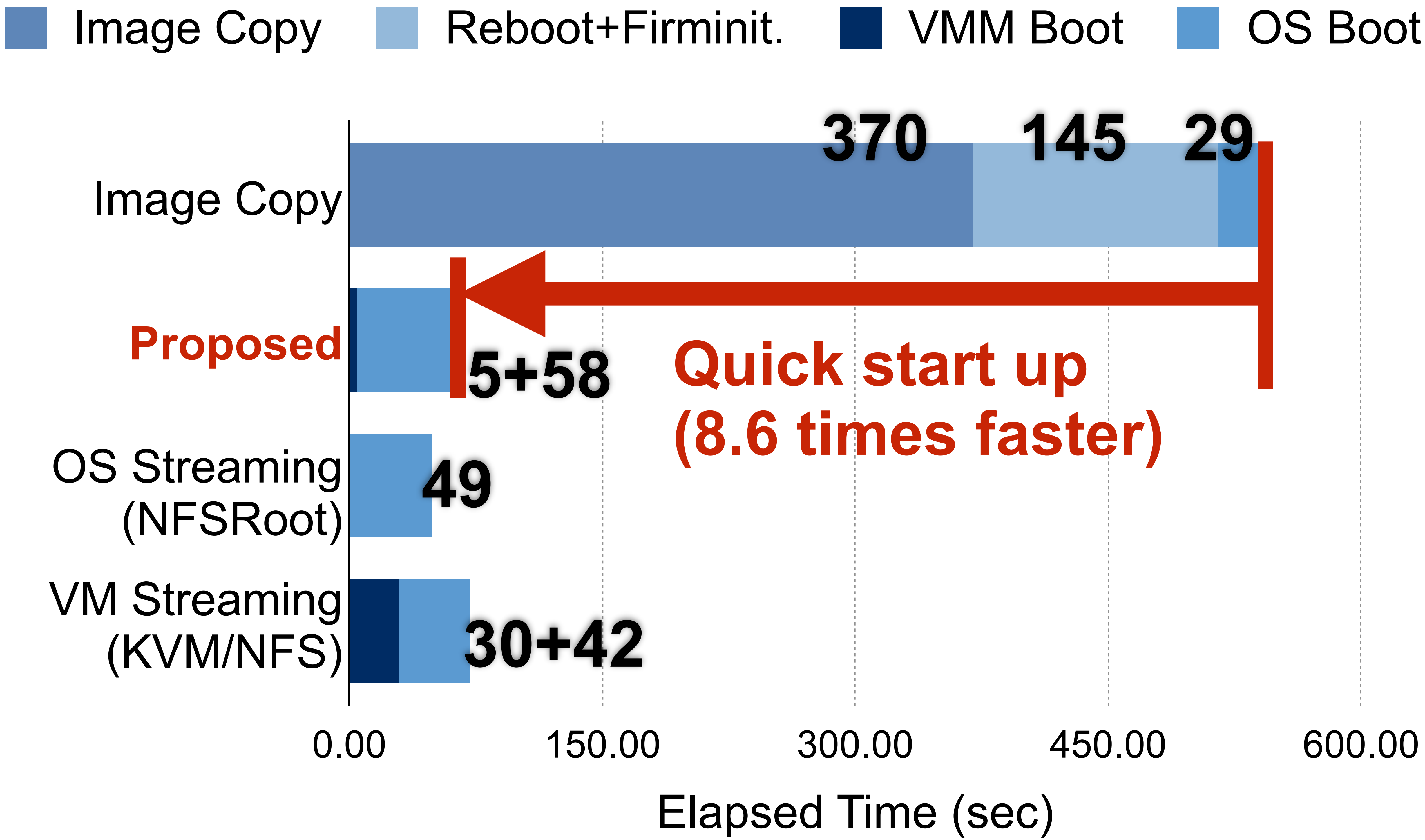
Performance Evaluation

- Deployed 32-GB OS Image (Ubuntu 14.04 64-bit)
  - OS-startup Time
  - Cassandra Throughput
  - Storage Throughput
  - InfiniBand Latency

A HPC Cluster
Intel Xeon X5680 (3.33 GHz) / 96GB RAM
HDD 500GB/7200 RPM SATA
Mellanox InfiniBand (4X QDR)
Intel 82575 EM GbE Network Card

Interconnected by
A Mellanox Grid Director InfiniBand Switch &
A FUJITSU SR- S348TC1 GbE Switch

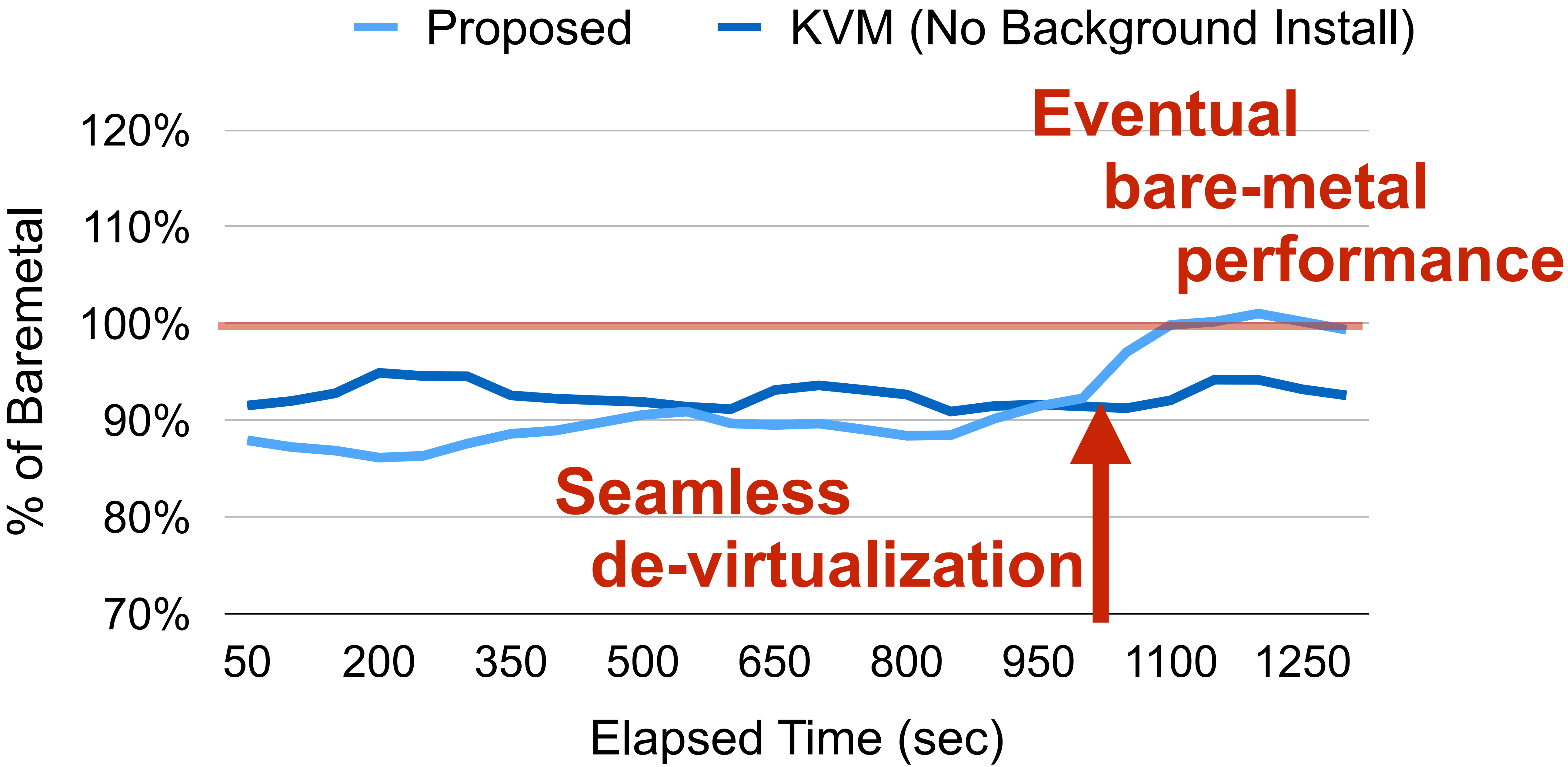
OS-startup Time



Quick start up (8.6 times faster)

Cassandra Throughput

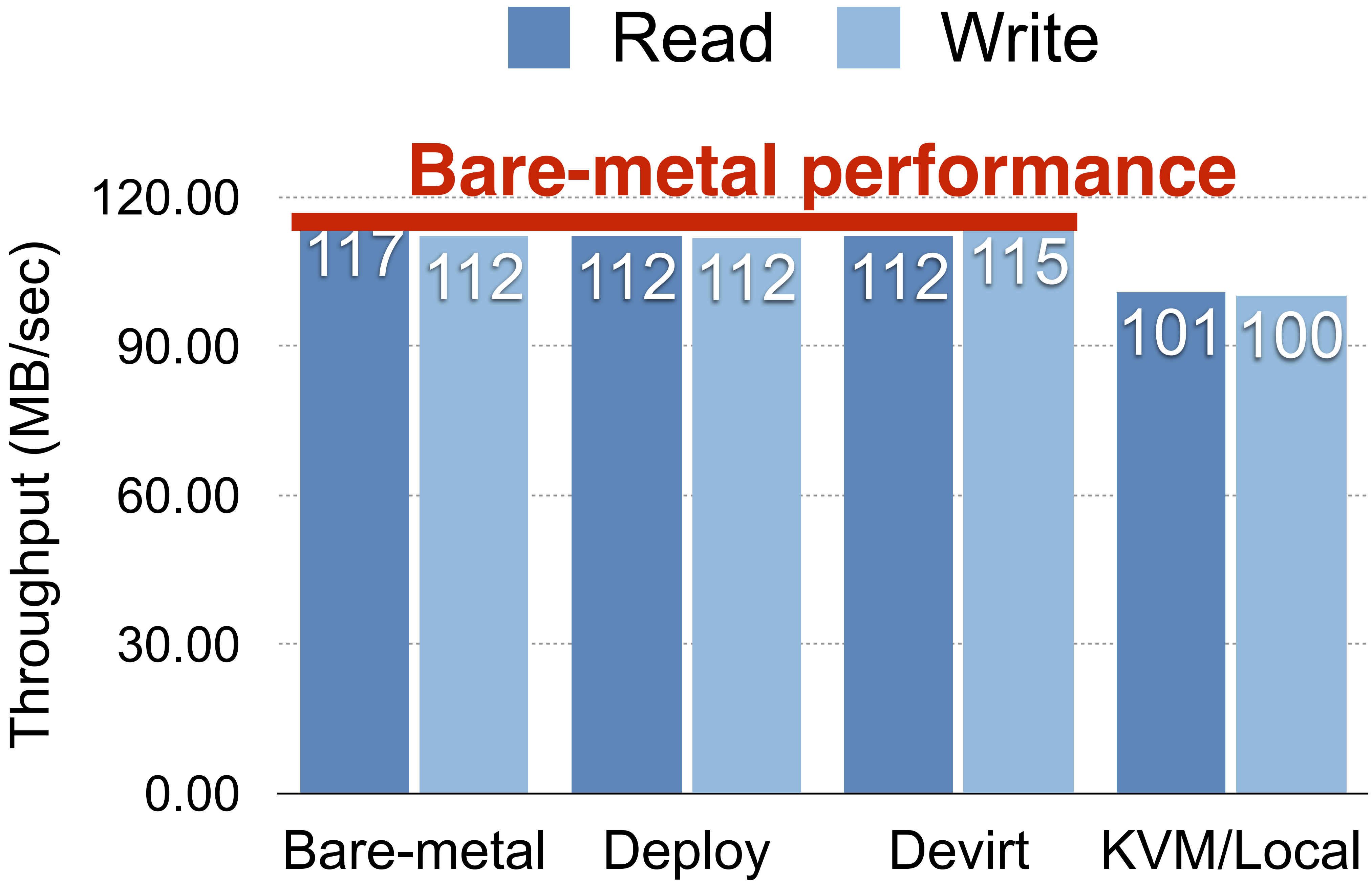
(Throughout Deployment)



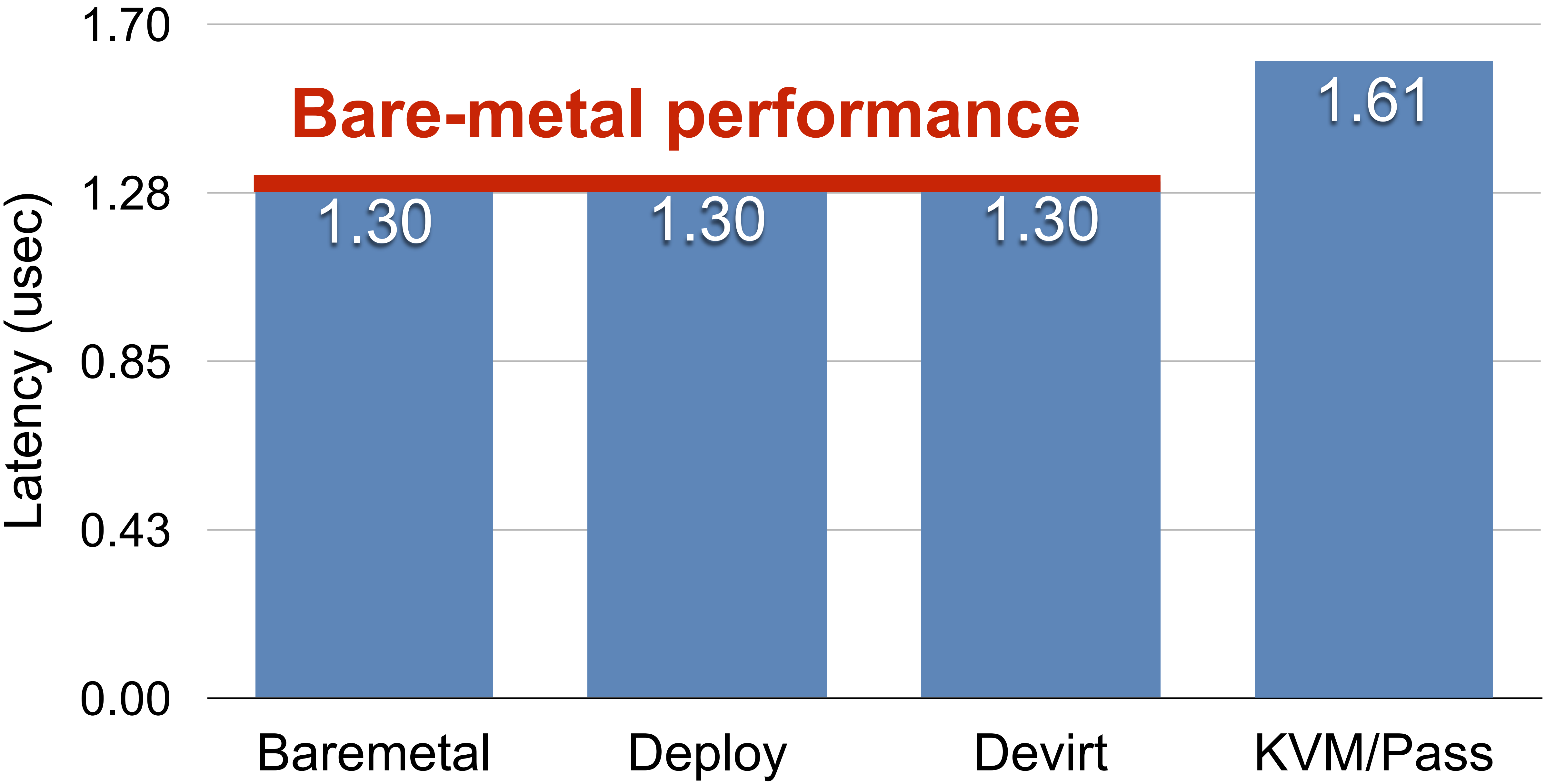
Eventual bare-metal performance

Seamless de-virtualization

Storage Throughput



InfiniBand RDMA latency



Conclusion

- Improved agility and elasticity in bare-metal clouds
  - De-virtualizable VMM with streaming deployment
    - Device-interface-level I/O mediation
  - Achieved quick startup of an OS
    - 8.6 times faster than image copy
    - Preserved high performance & OS-transparency

Future work

- Generating device mediators from specification
    - Reduce development cost of device mediators
- More advanced features of IaaS clouds
    - Live migration and checkpointing

Thank you