Unikernels: Functional Library Operating Systems for the Cloud

xen.org

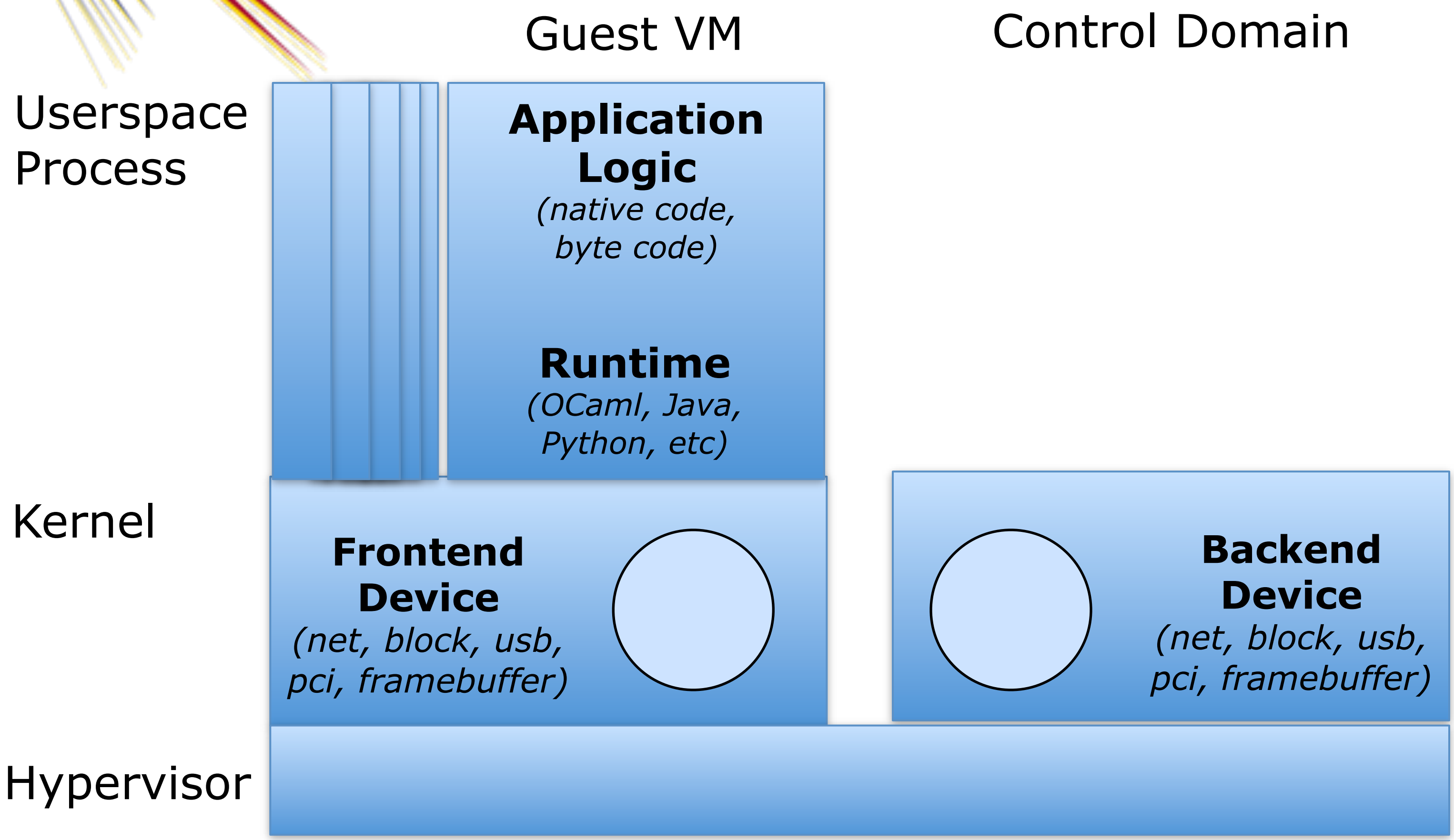
Anil Madhavapeddy, University of Cambridge
with a merry crew:

Haris Rotsos, Balraj Singh, Steven Smith
Jon Crowcroft, Steve Hand
(University of Cambridge)
Richard Mortier *(University of Nottingham)*
Thomas Gazagnaire *(OCamlPro)*
Dave Scott *(Citrix Systems R&D)*

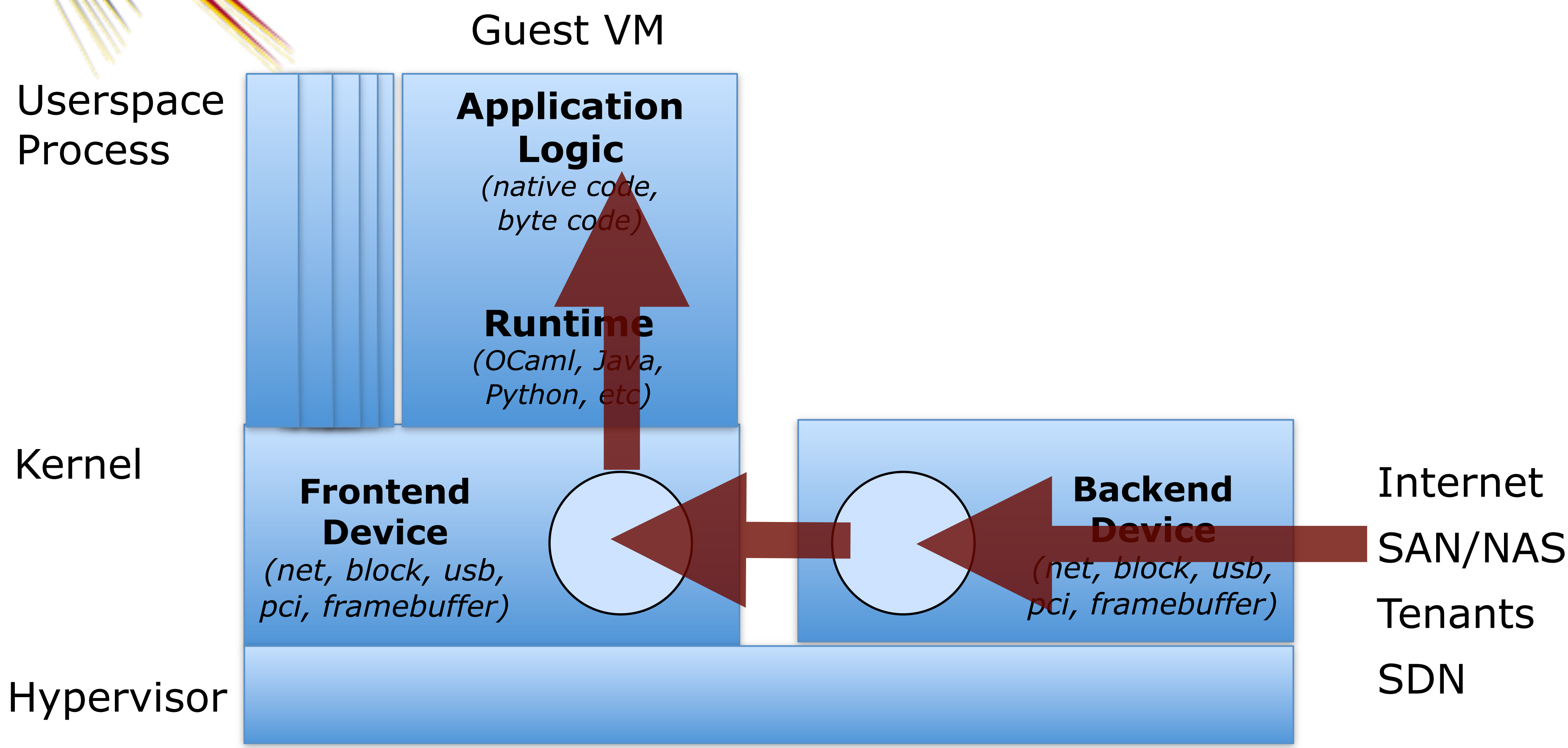
openmirage.org

@avsm

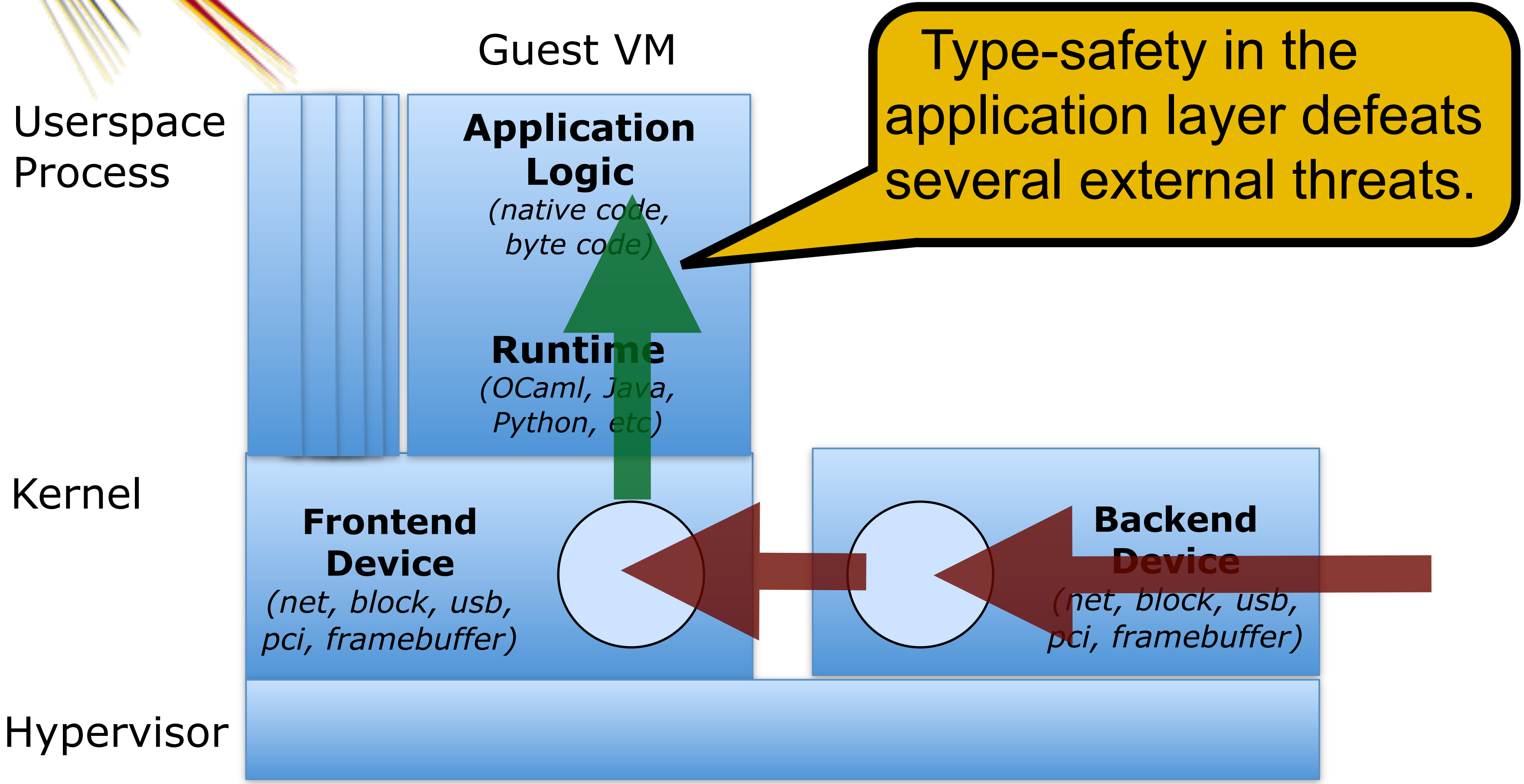
The Cloud Threat Model



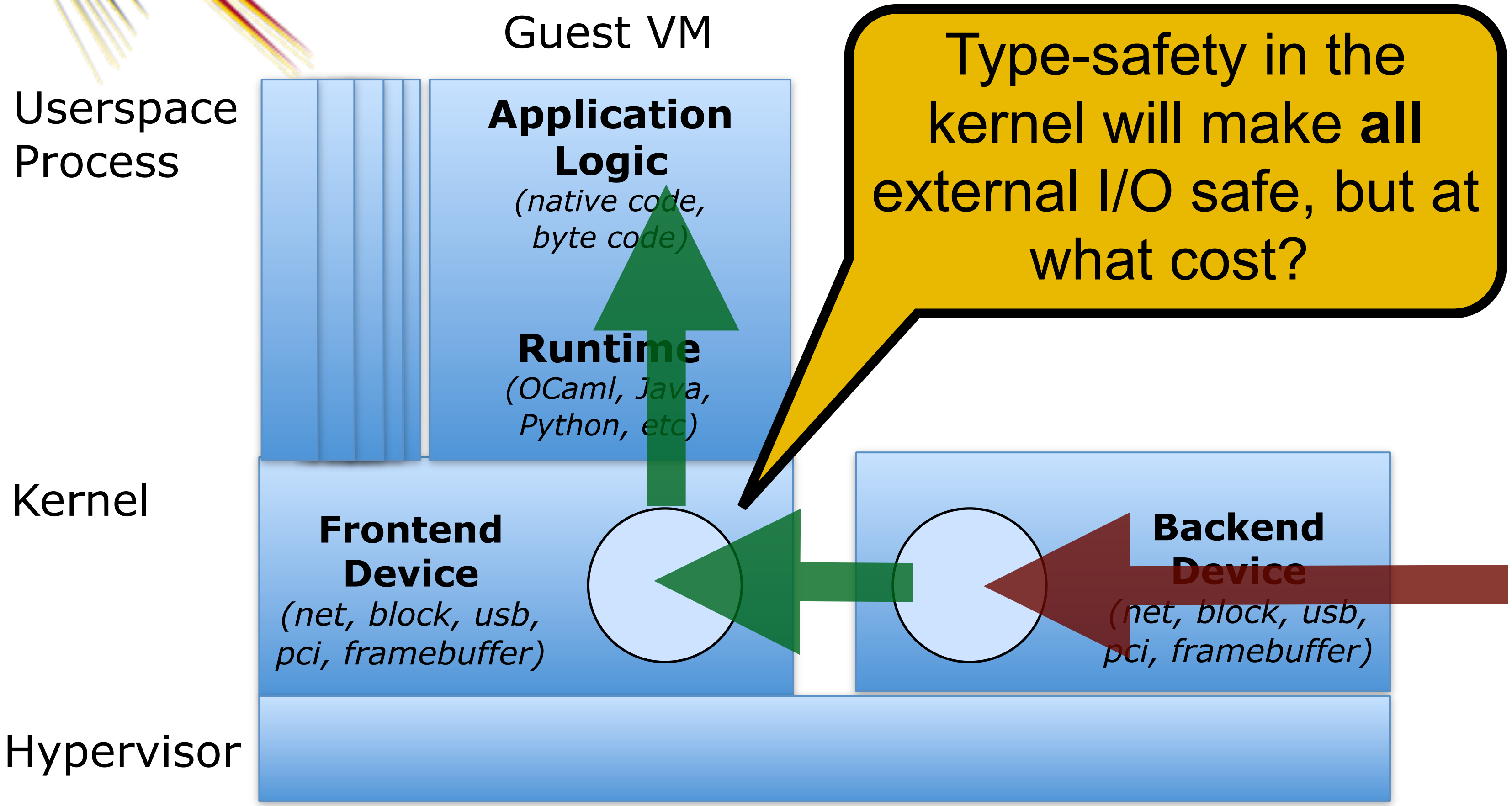
The Cloud Threat Model



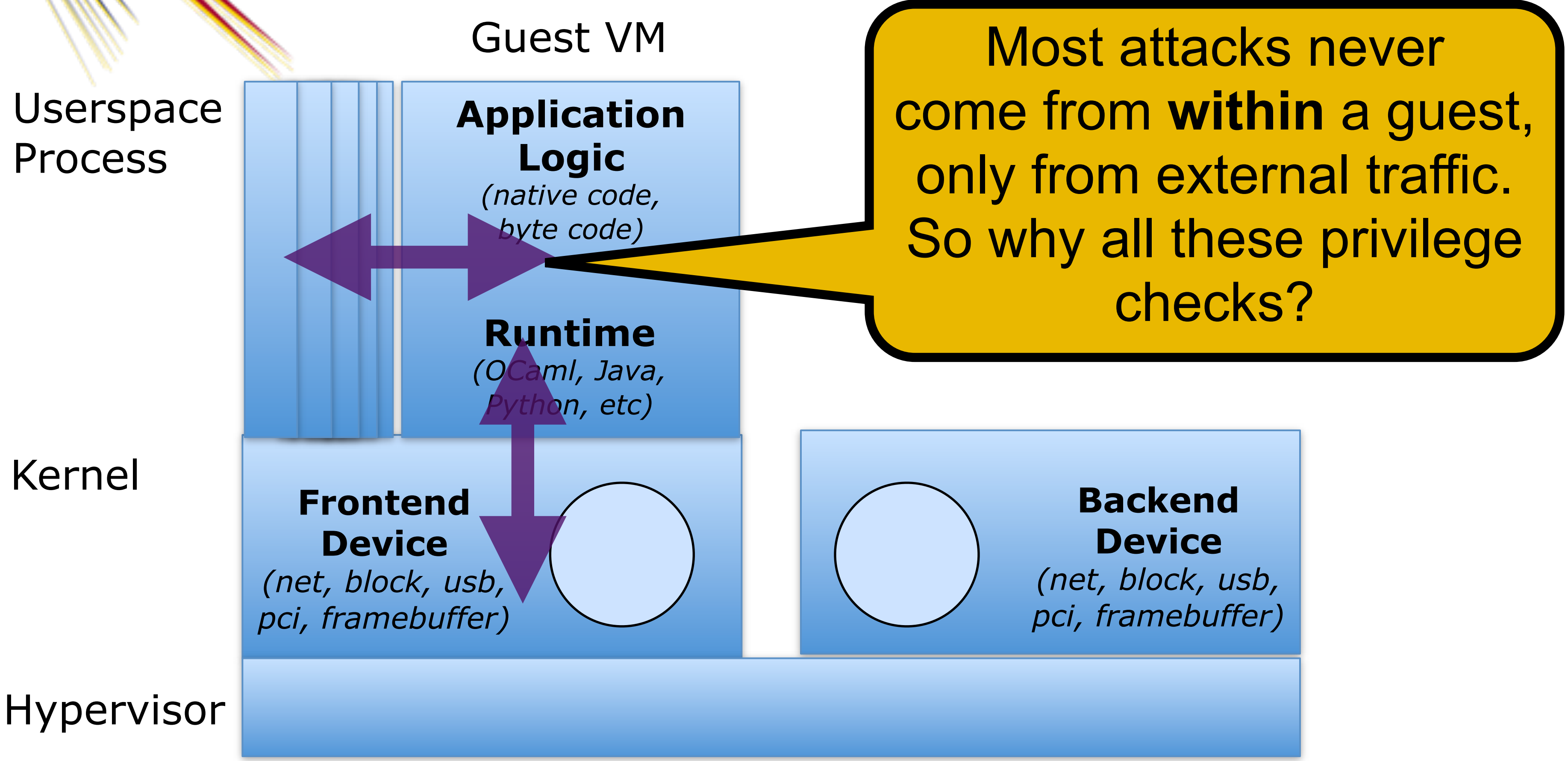
The Cloud Threat Model



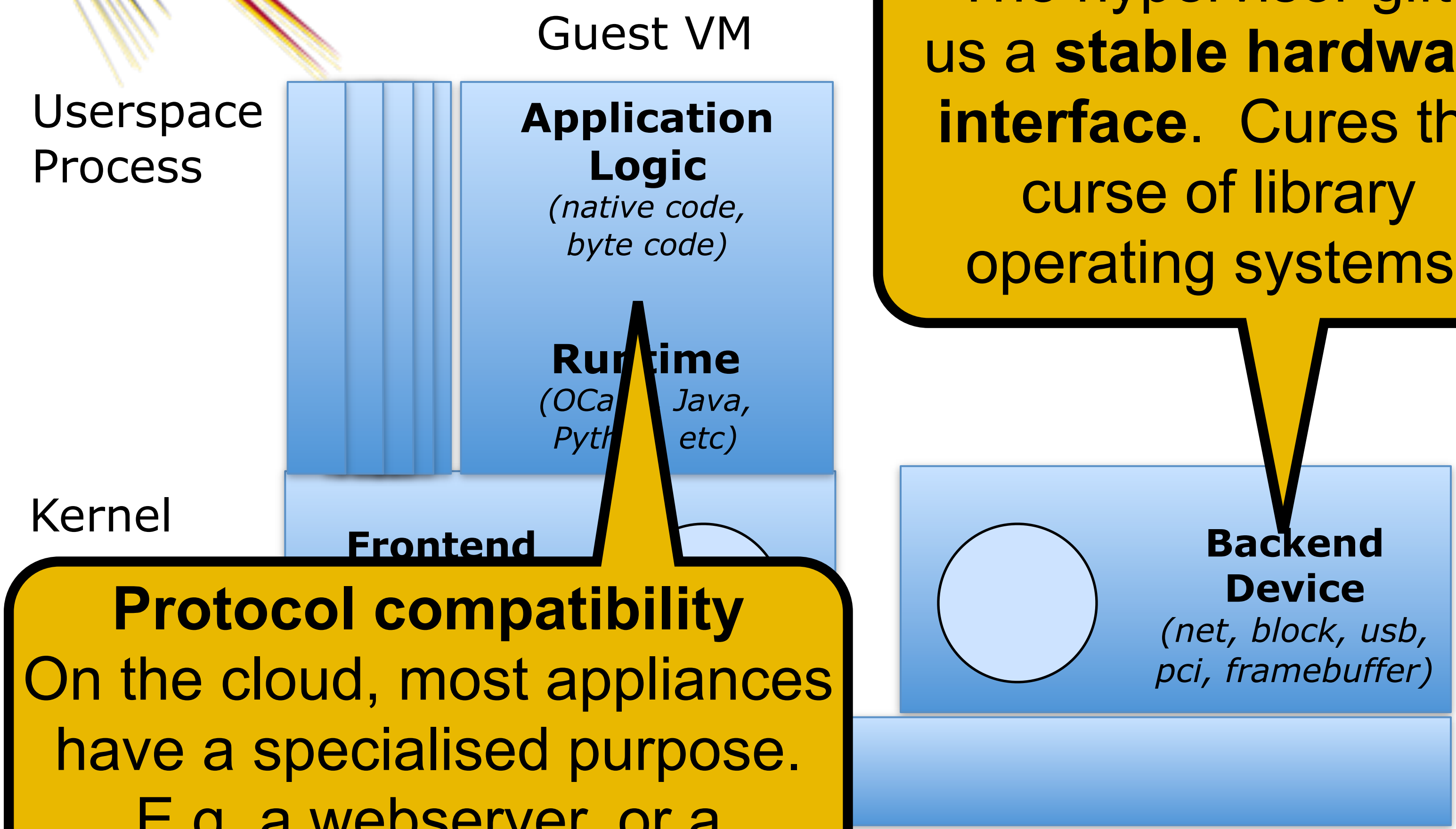
The Cloud Threat Model



The Cloud Threat Model



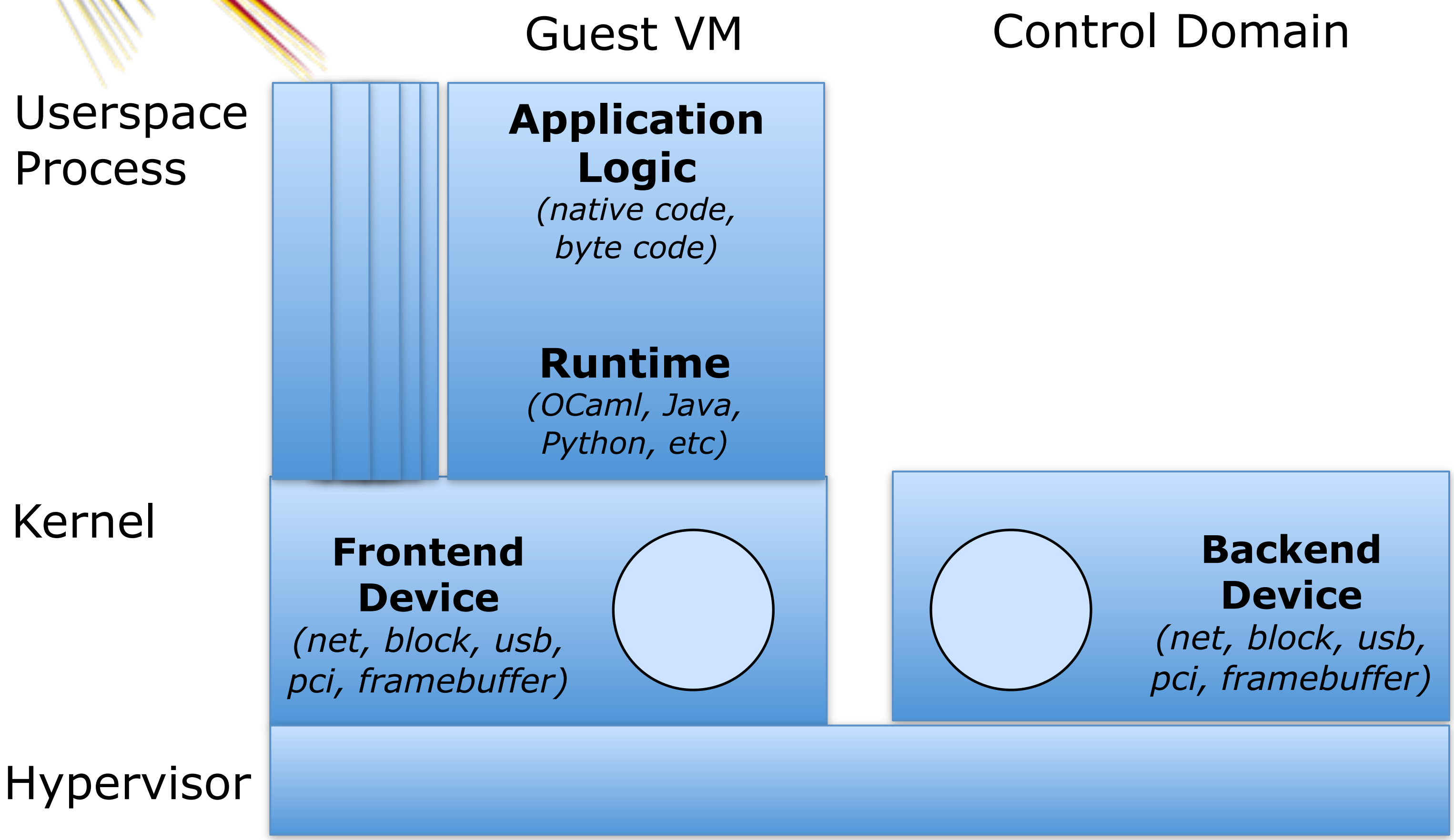
Key Design Insights



The hypervisor gifts us a **stable hardware interface**. Cures the curse of library operating systems!

Protocol compatibility
On the cloud, most appliances have a specialised purpose. E.g. a webserver, or a database VM.

Key Design Insights

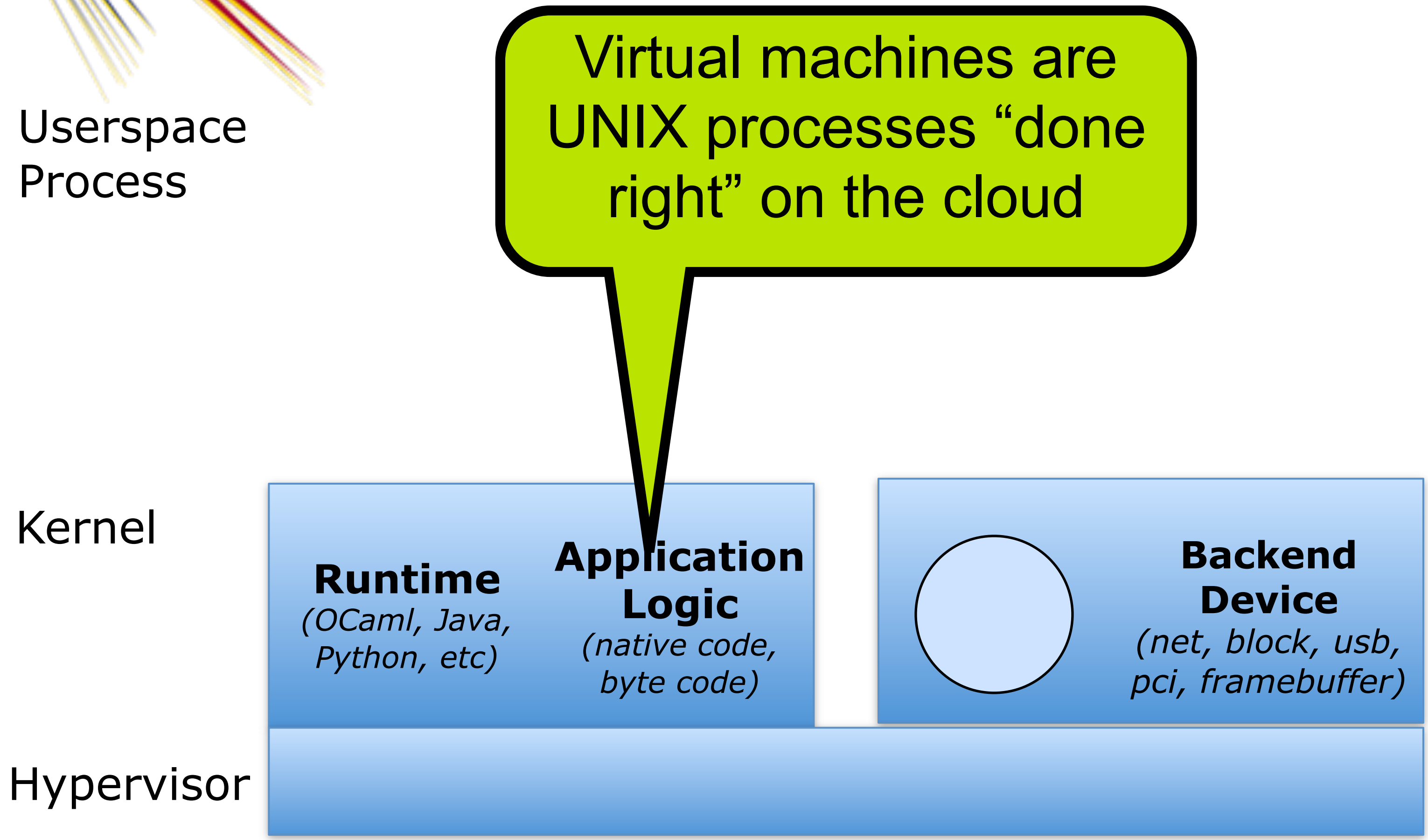


UNIVERSITY OF CAMBRIDGE

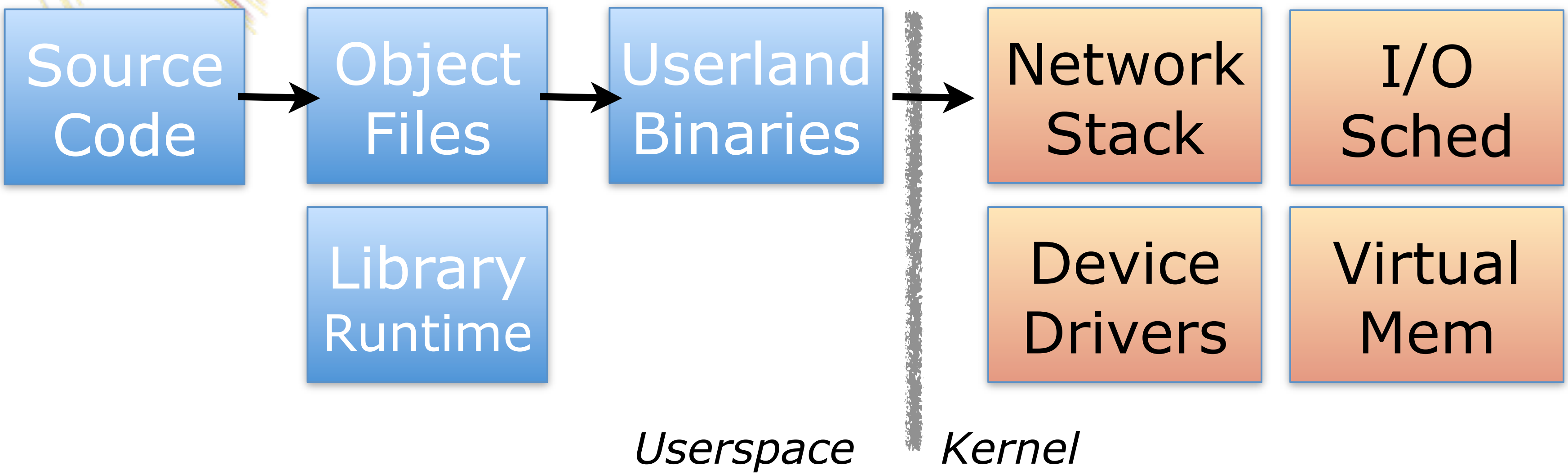
openmirage.org

horizon DIGITAL ECONOMY RESEARCH

Unikernels!

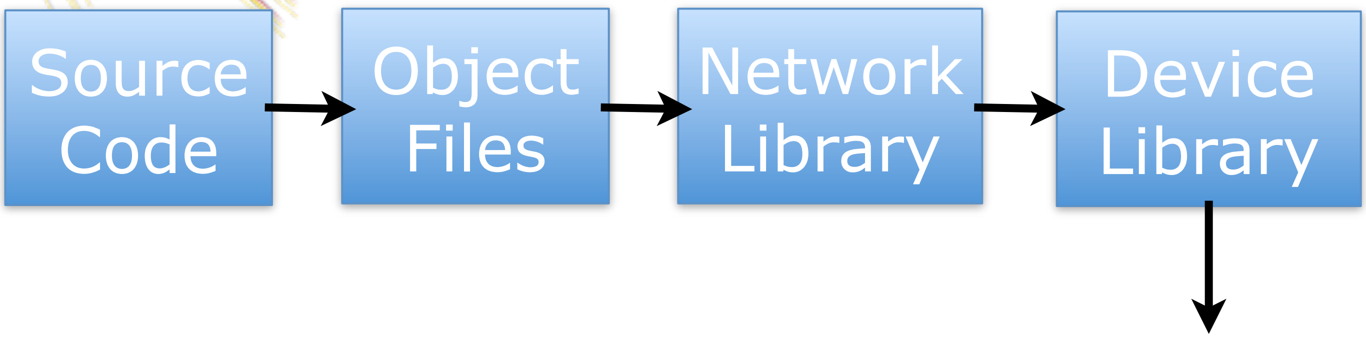

Virtual machines are UNIX processes “done right” on the cloud
Userspace Process
Kernel
Hypervisor
Runtime
(OCaml, Java, Python, etc)
Application Logic
(native code, byte code)
Backend Device
(net, block, usb, pci, framebuffer)

Virtual Appliances: current

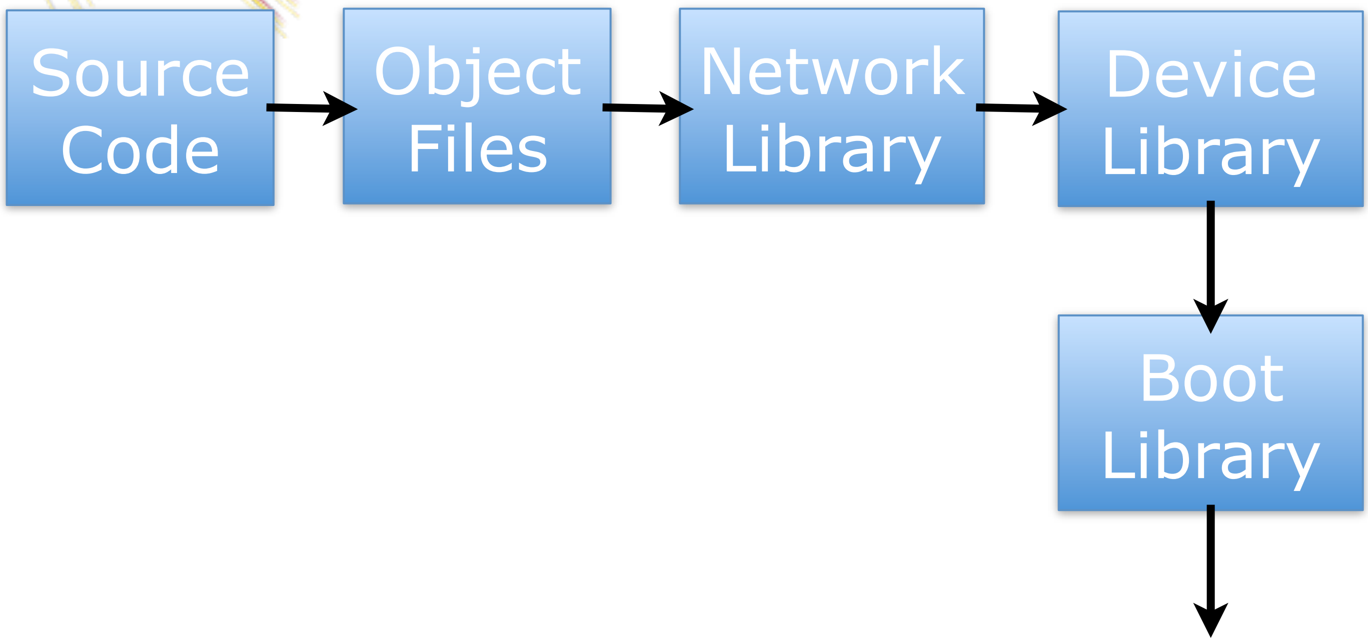


Compiler has to stop at userspace.
Every level has a different API, calling convention, and privilege requirements.

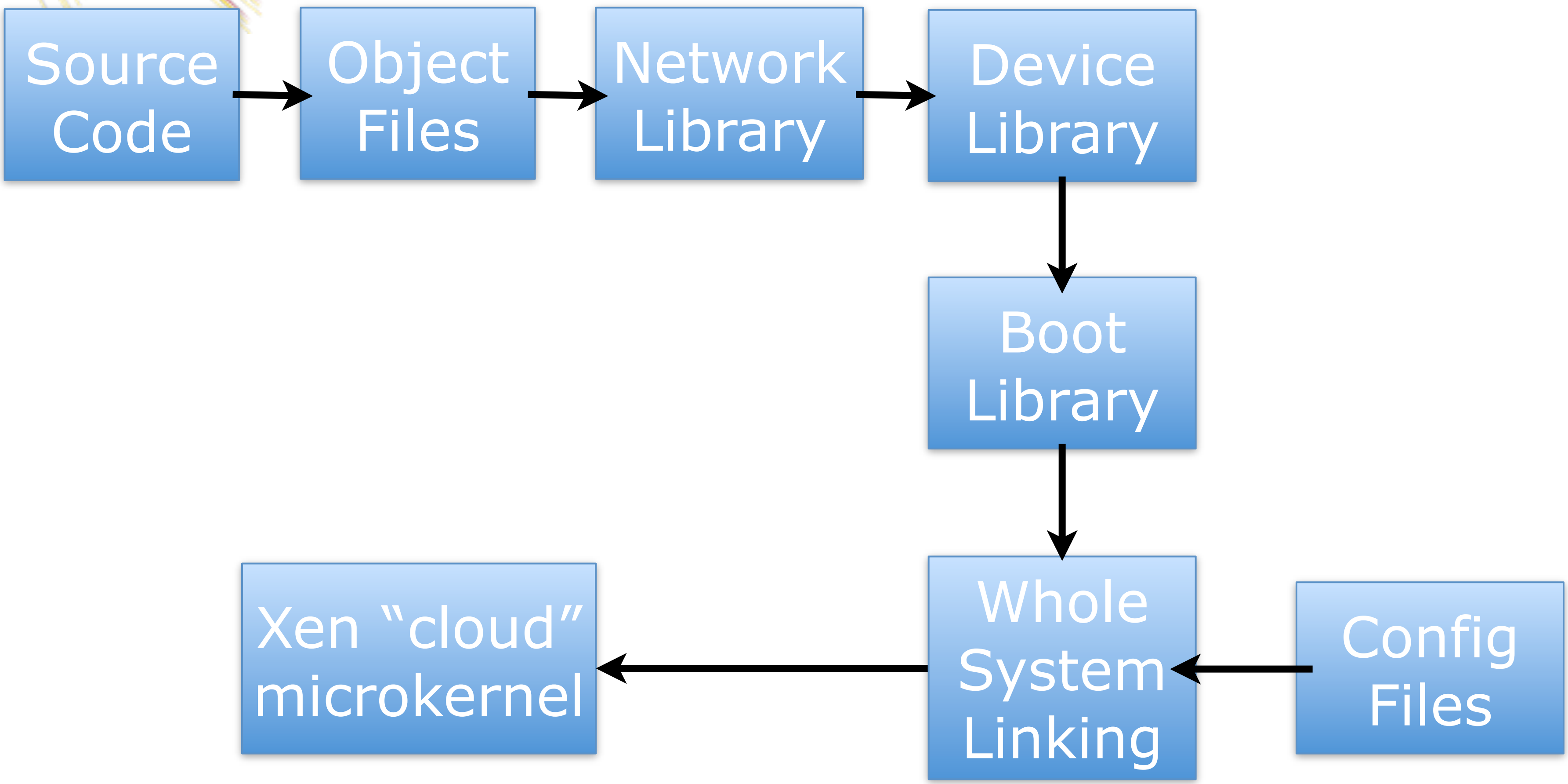
Virtual Appliances: specialised



Virtual Appliances: specialised



Virtual Appliances: specialised



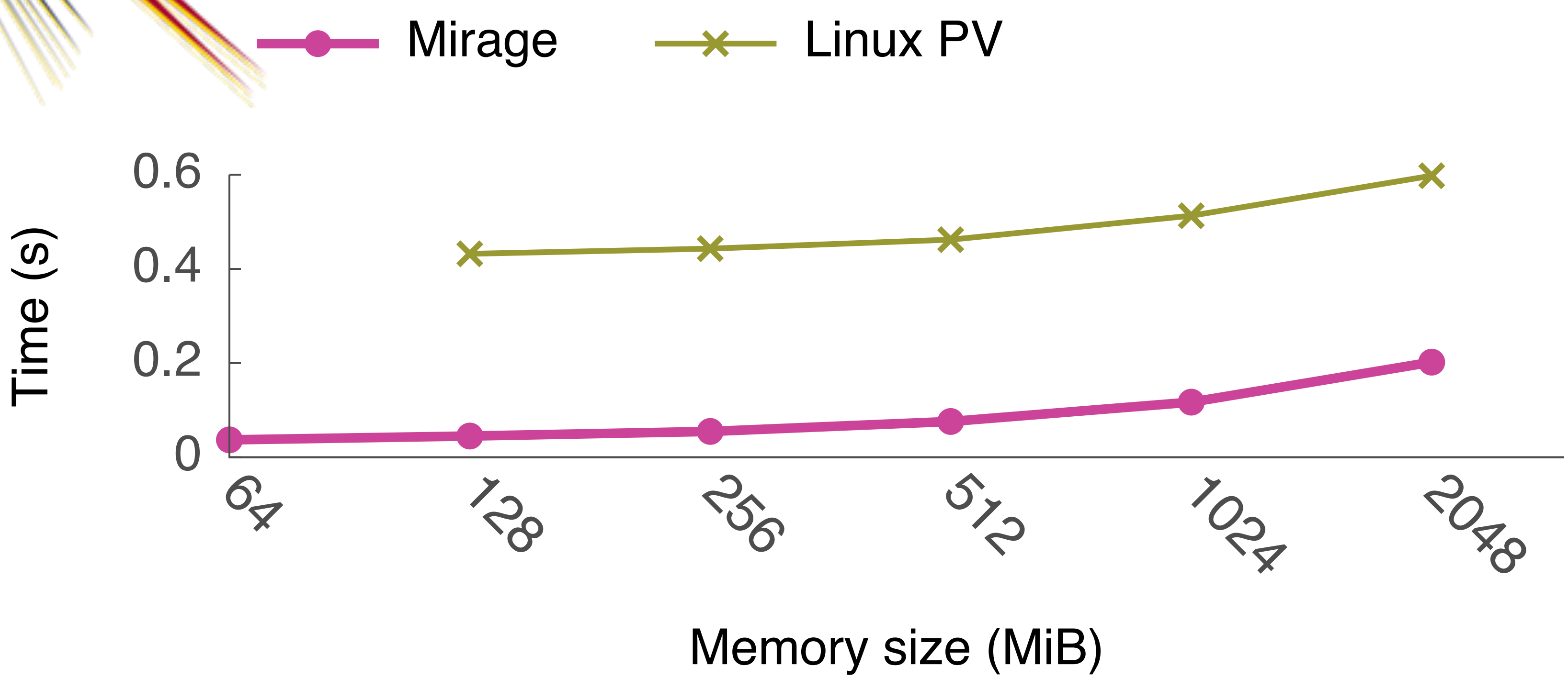
Appliance Image Size

Appliance	Standard Build	Dead Code Elimination
DNS	0.449 MB	0.184 MB
Web Server	0.674 MB	0.172 MB
Openflow learning switch	0.393 MB	0.164 MB
Openflow controller	0.392 MB	0.168 MB

All configuration and data compiled into the image by the toolchain.

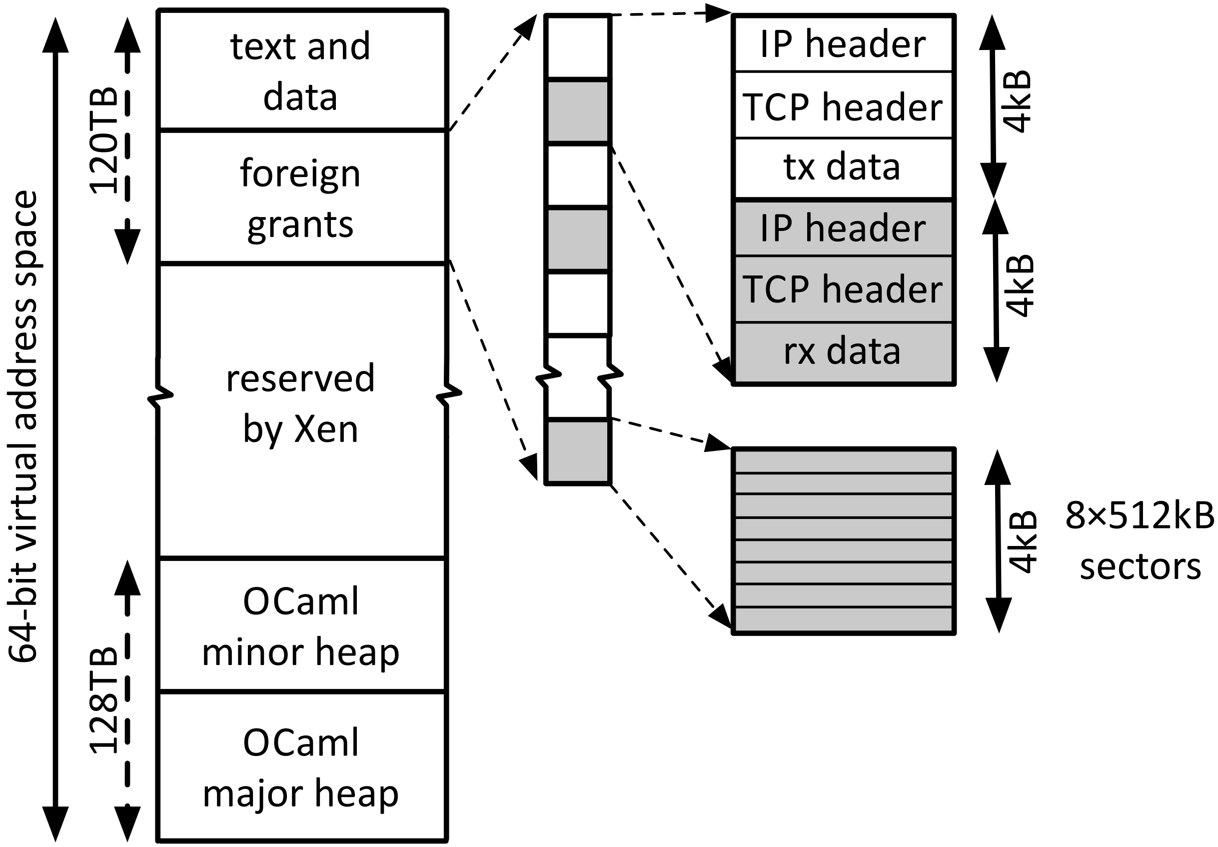
Live migration is easy and fun :-)

Microbenchmarks: Boot Time



Unikernels are compact enough to boot and respond to network traffic in real-time.

Simplified Memory Management



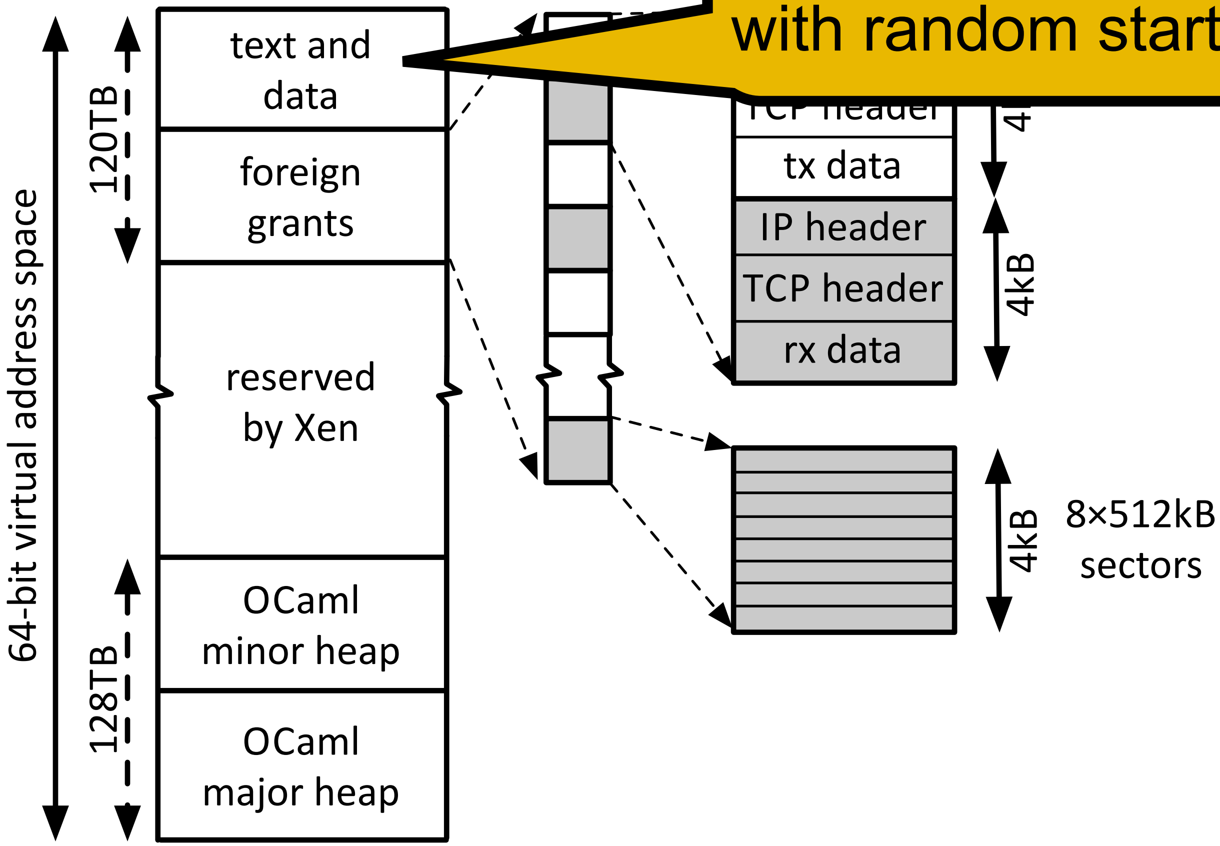
UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH

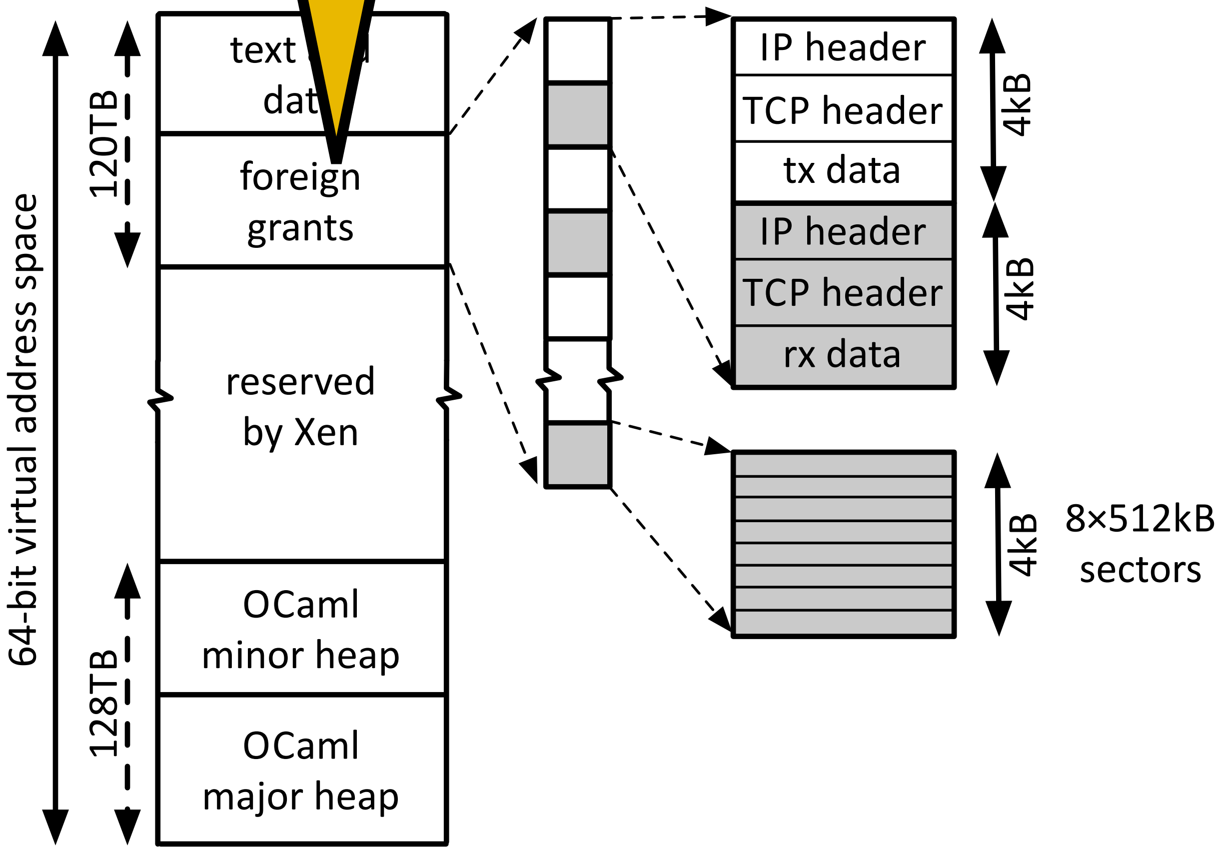
Simplified Memory Management

Compiled native source code and runtime statically linked with random start offset

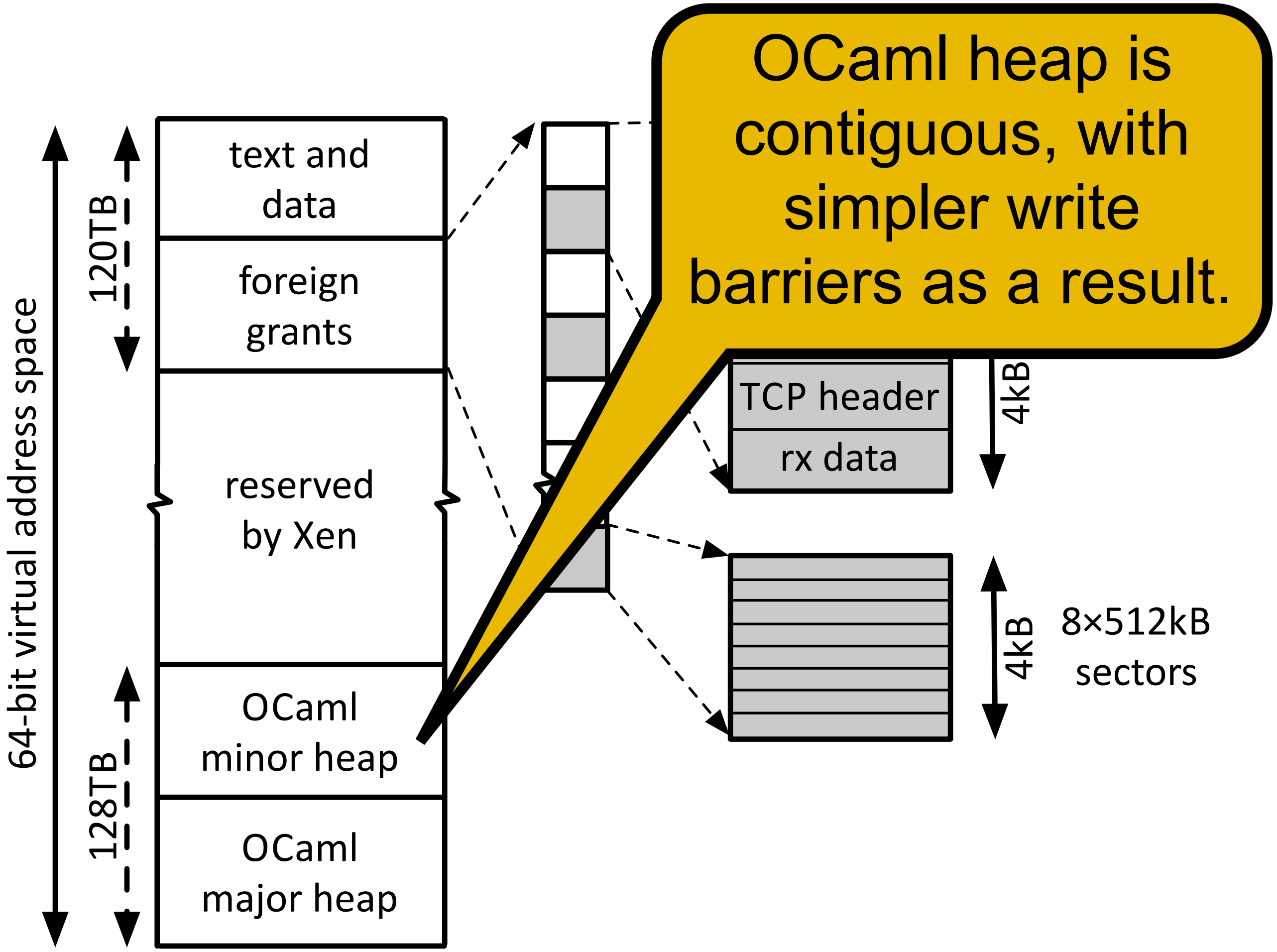


ory Management

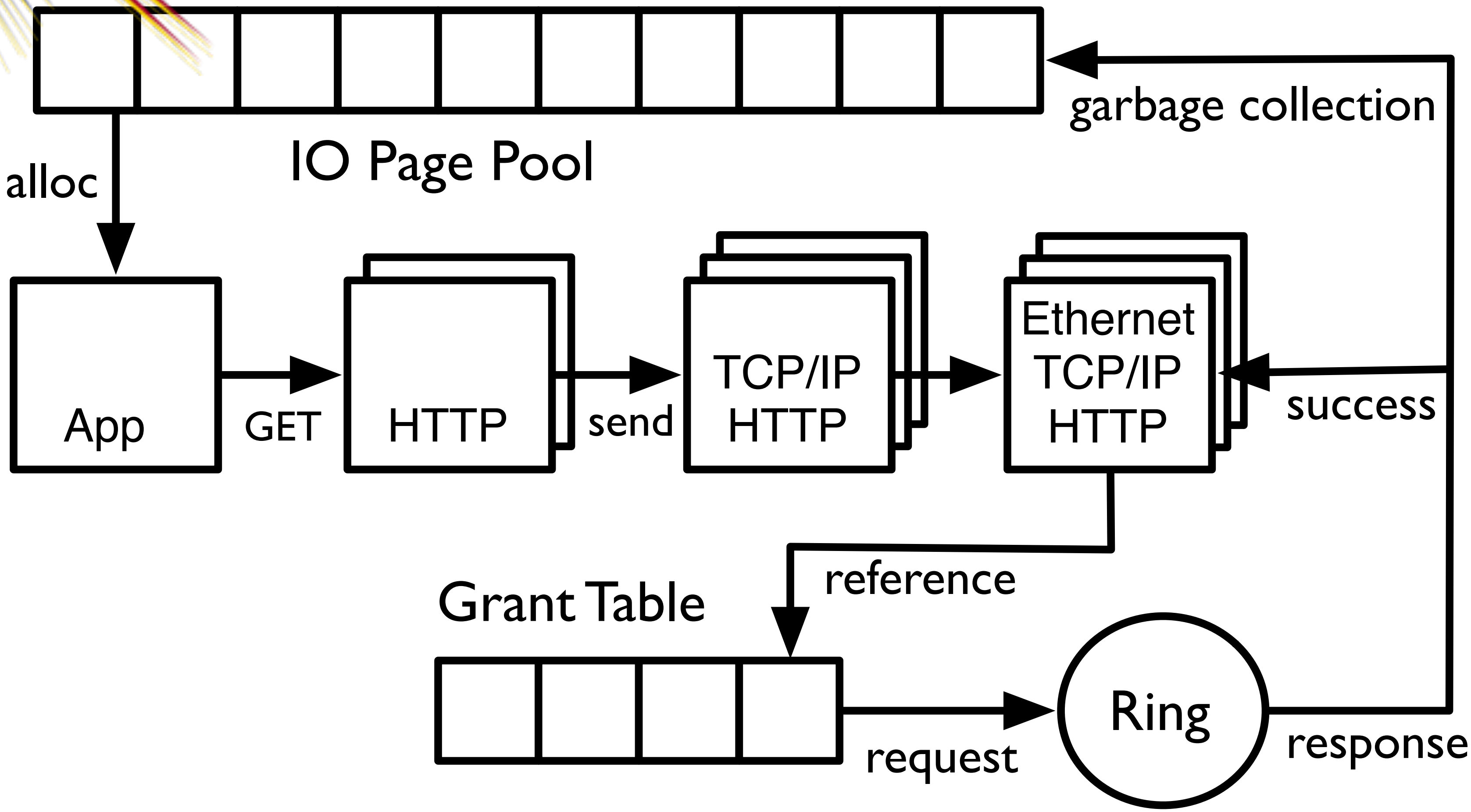
All I/O memory gets mapped into a reserved area and can be distinguished thusly.



Simplified Memory Management



Zero-Copy IO Buffer Management



DNS Server Code

Cooperative threads as functions

Statically evaluated configuration

```
let main () =
  lwt zones = read key "zones" "zone.db" in
  Net.Manager.bind (fun mgr dev →
    let src = 'any_addr, 53 in
    Dns.Server.listen dev src zones
  )
```

Functional callbacks

Libraries directly link to network stack

UNIVERSITY OF CAMBRIDGE

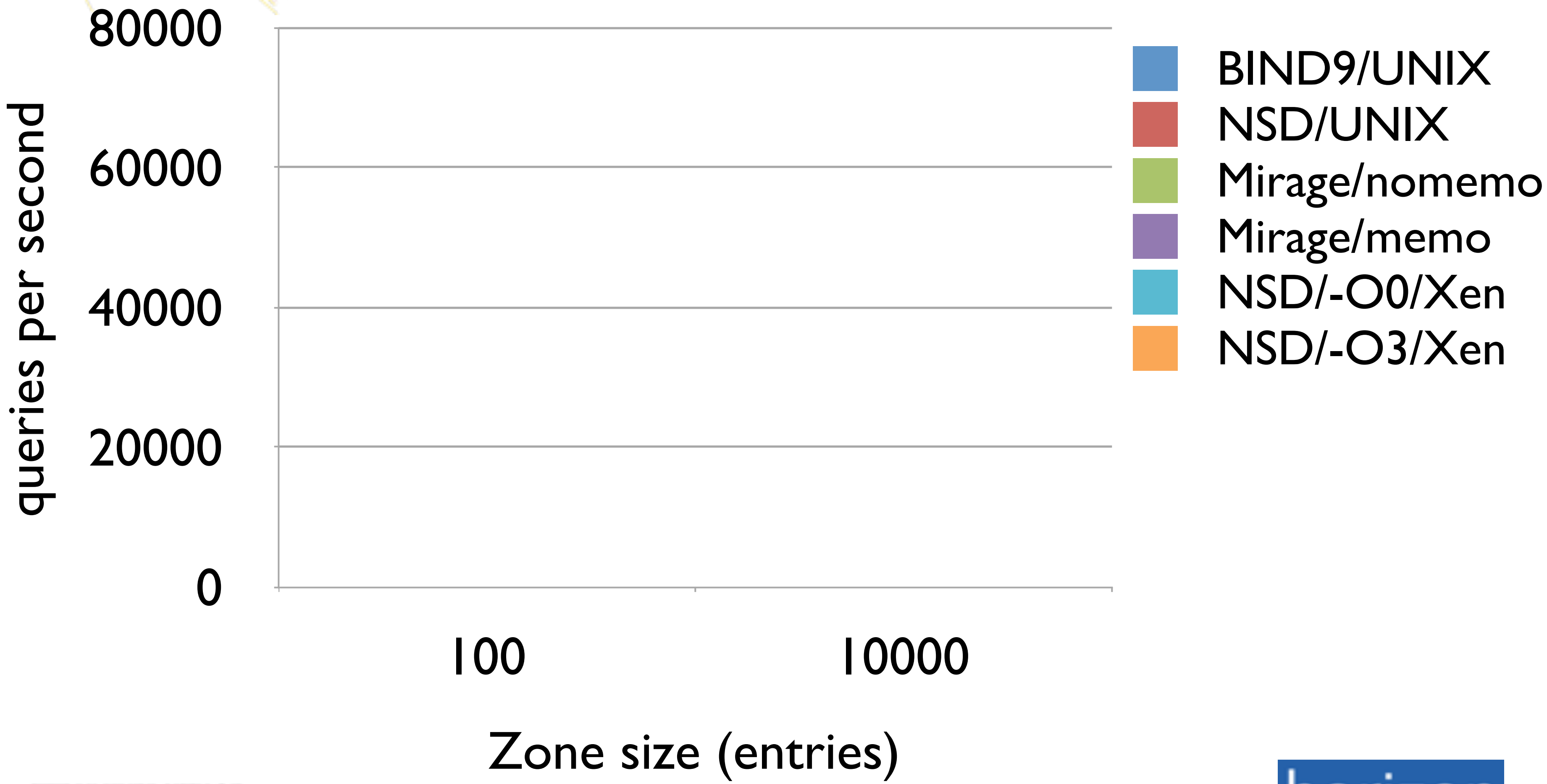
openmirage.org

horizon DIGITAL ECONOMY RESEARCH

DNS Server Performance

- BIND9/UNIX
- NSD/UNIX
- Mirage/nomemo
- Mirage/memo
- NSD/-O0/Xen
- NSD/-O3/Xen

DNS Server Performance

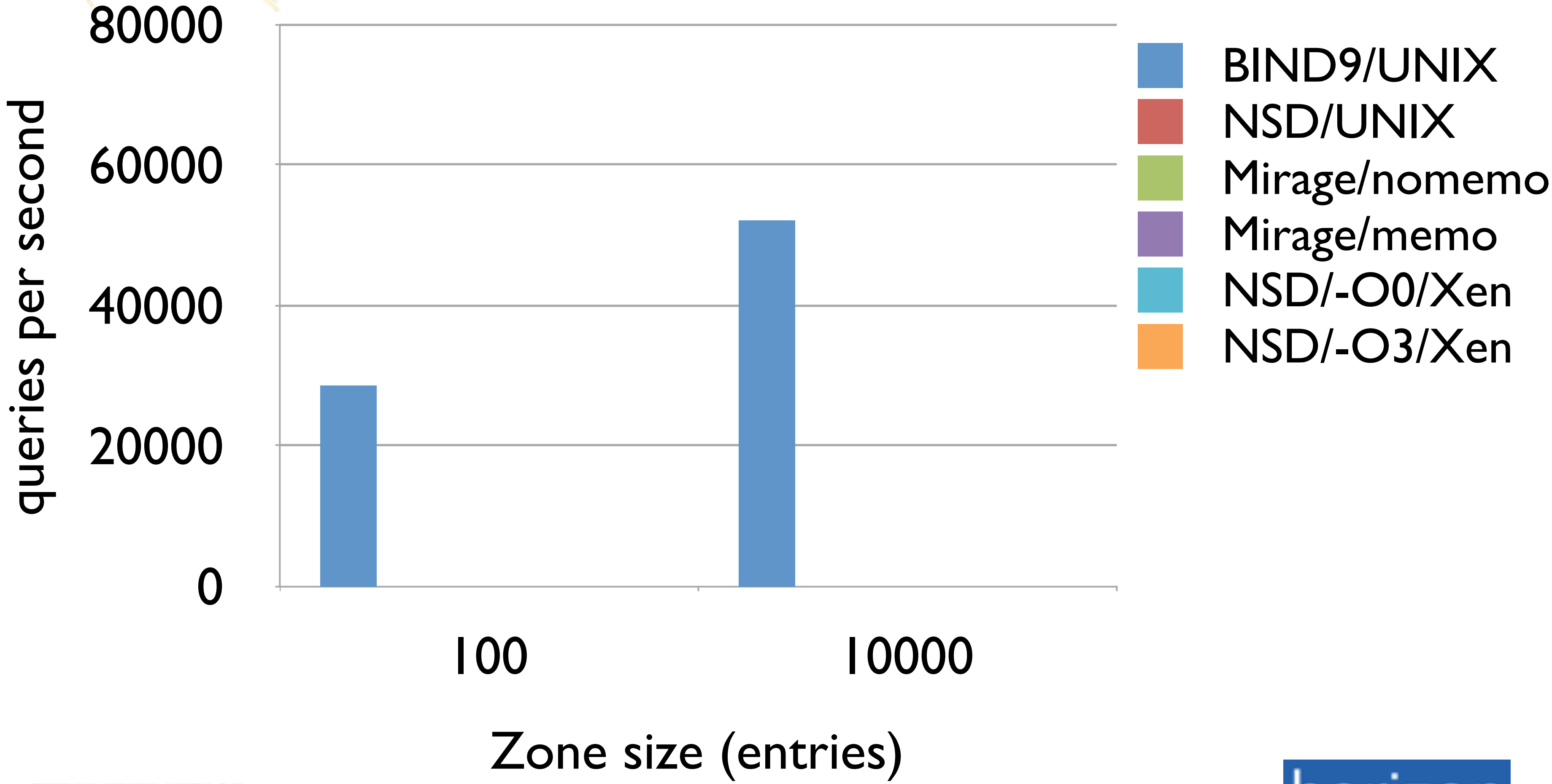


UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH

DNS Server Performance

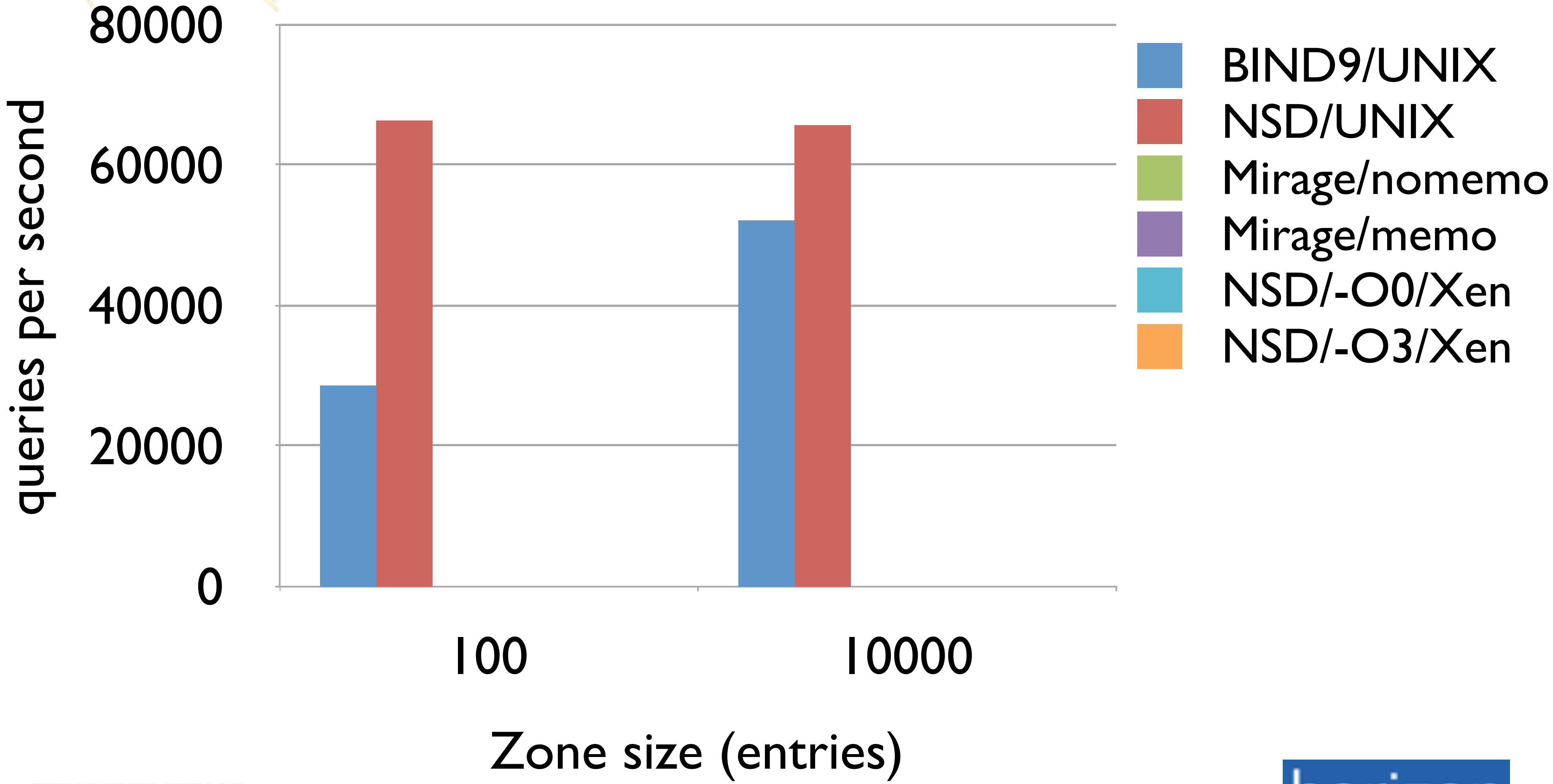


UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH

DNS Server Performance

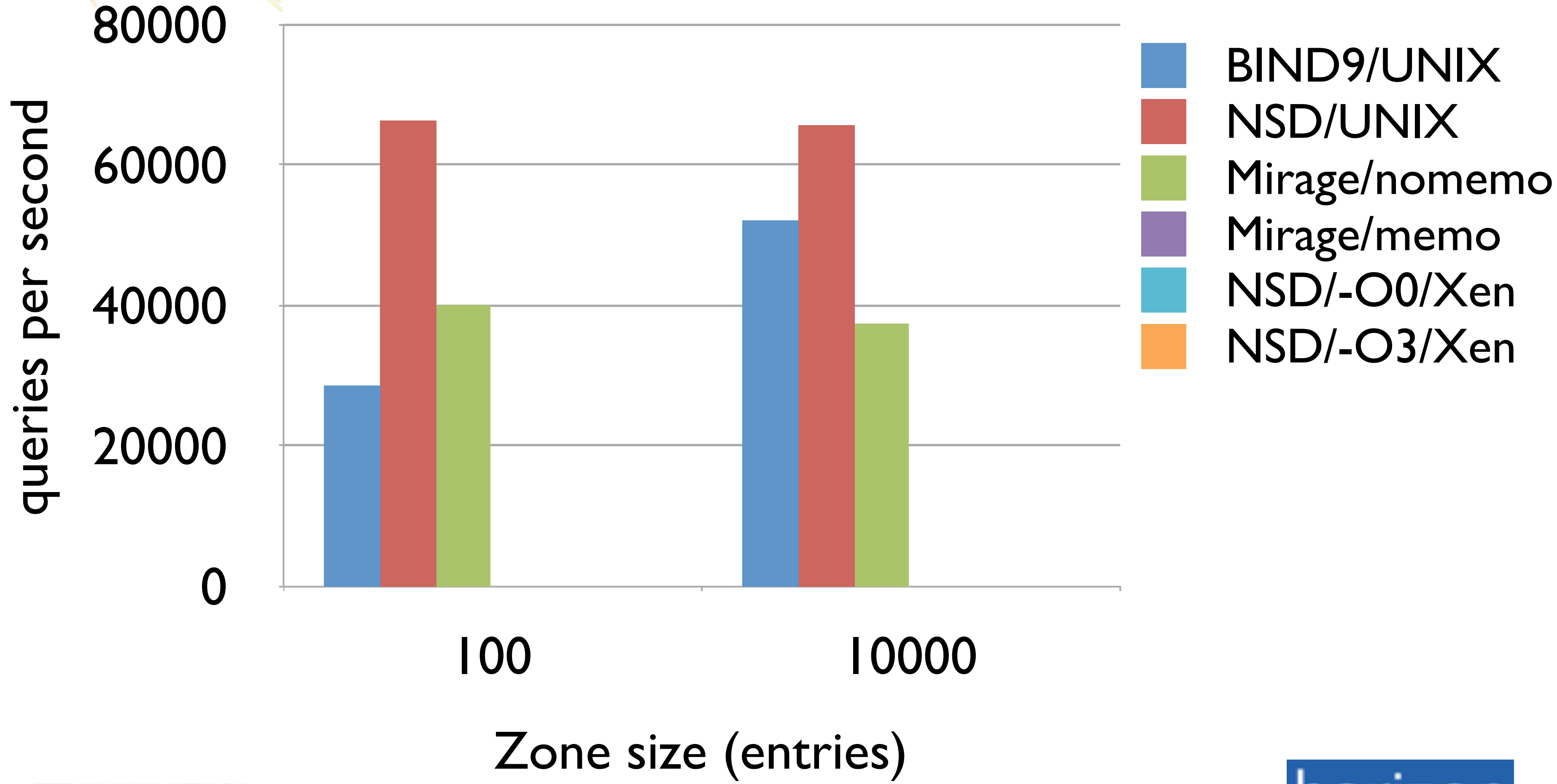


UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH

DNS Server Performance

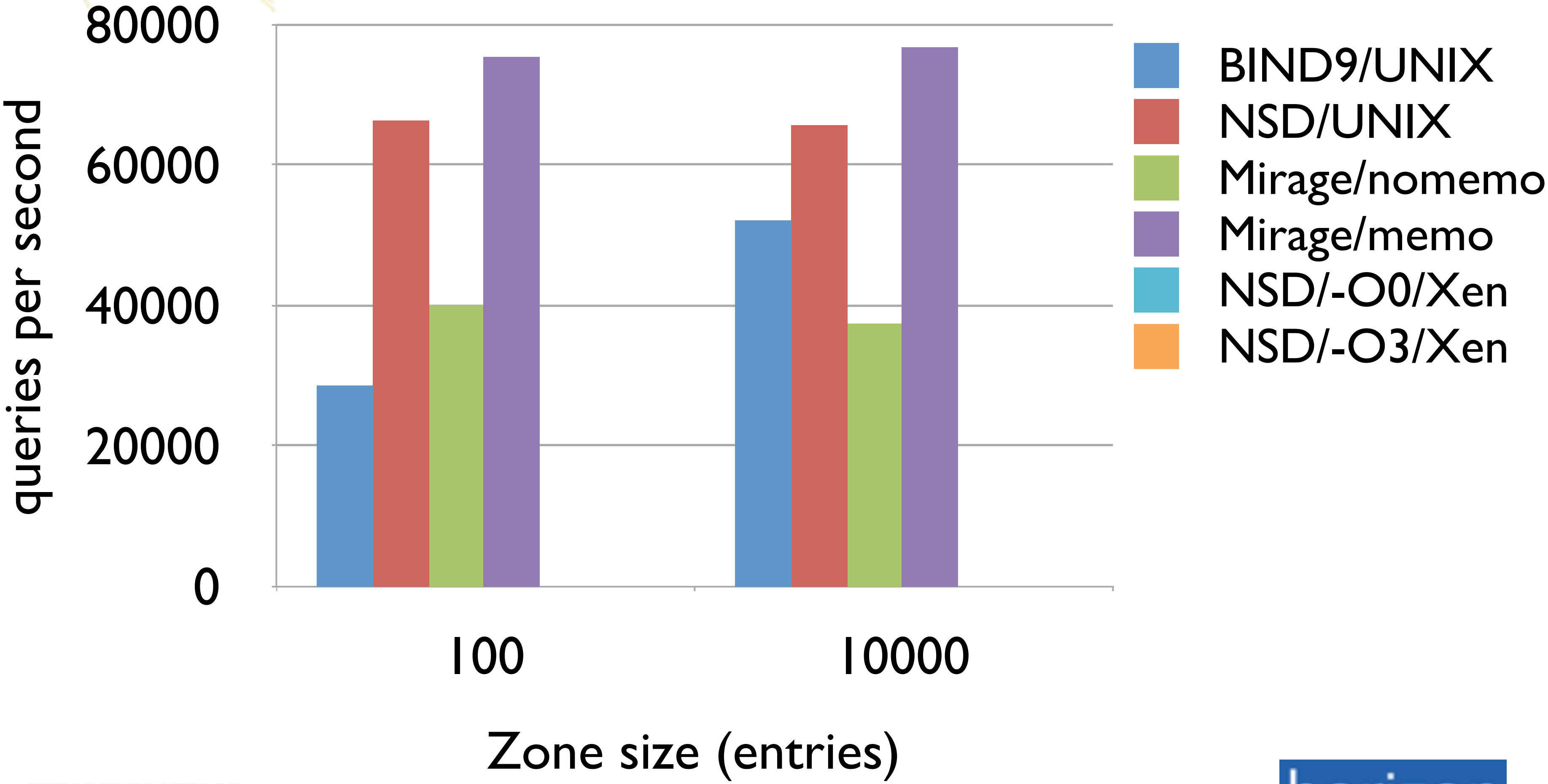


UNIVERSITY OF CAMBRIDGE

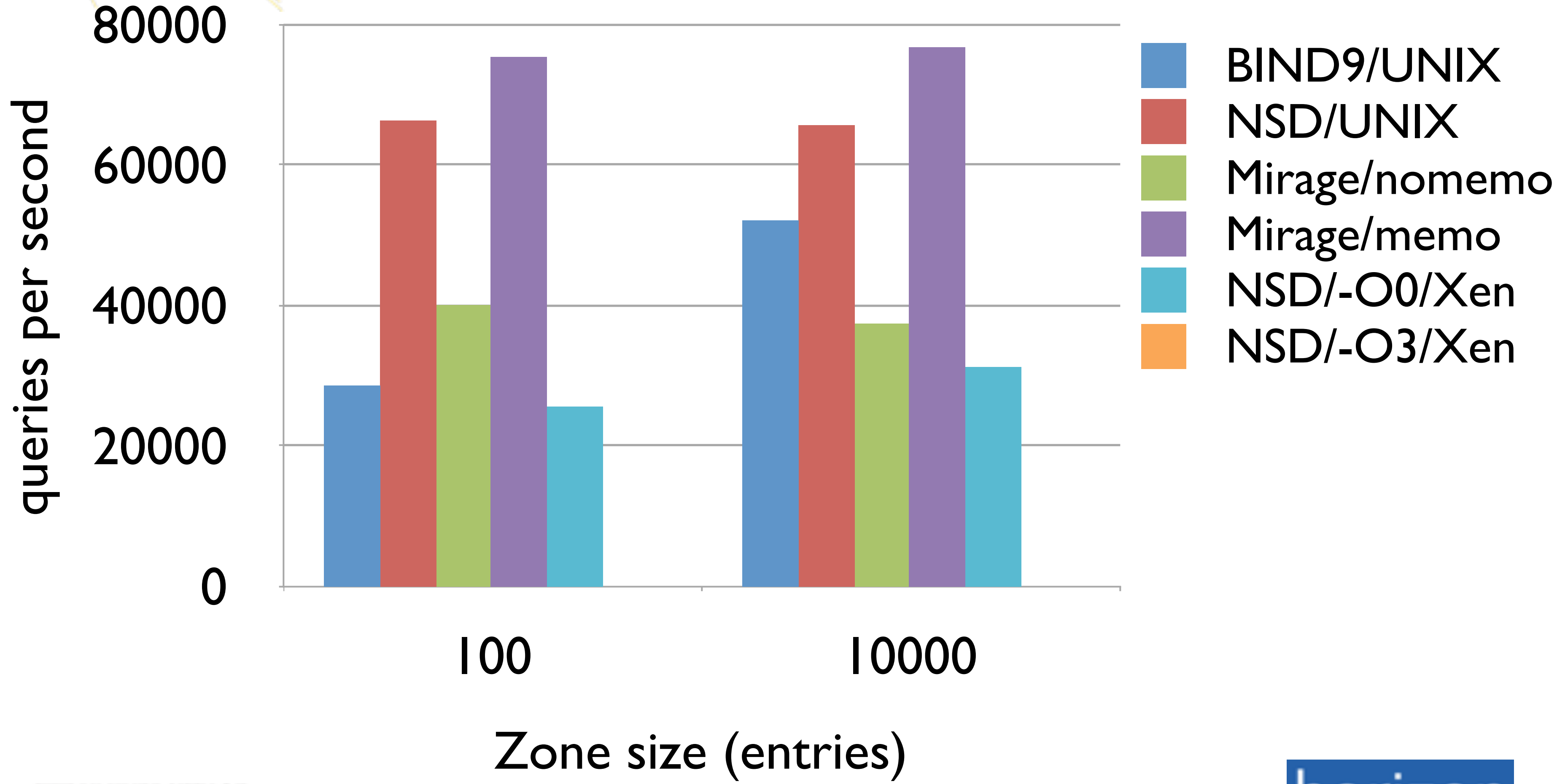
openmirage.org

horizon DIGITAL ECONOMY RESEARCH

DNS Server Performance



DNS Server Performance

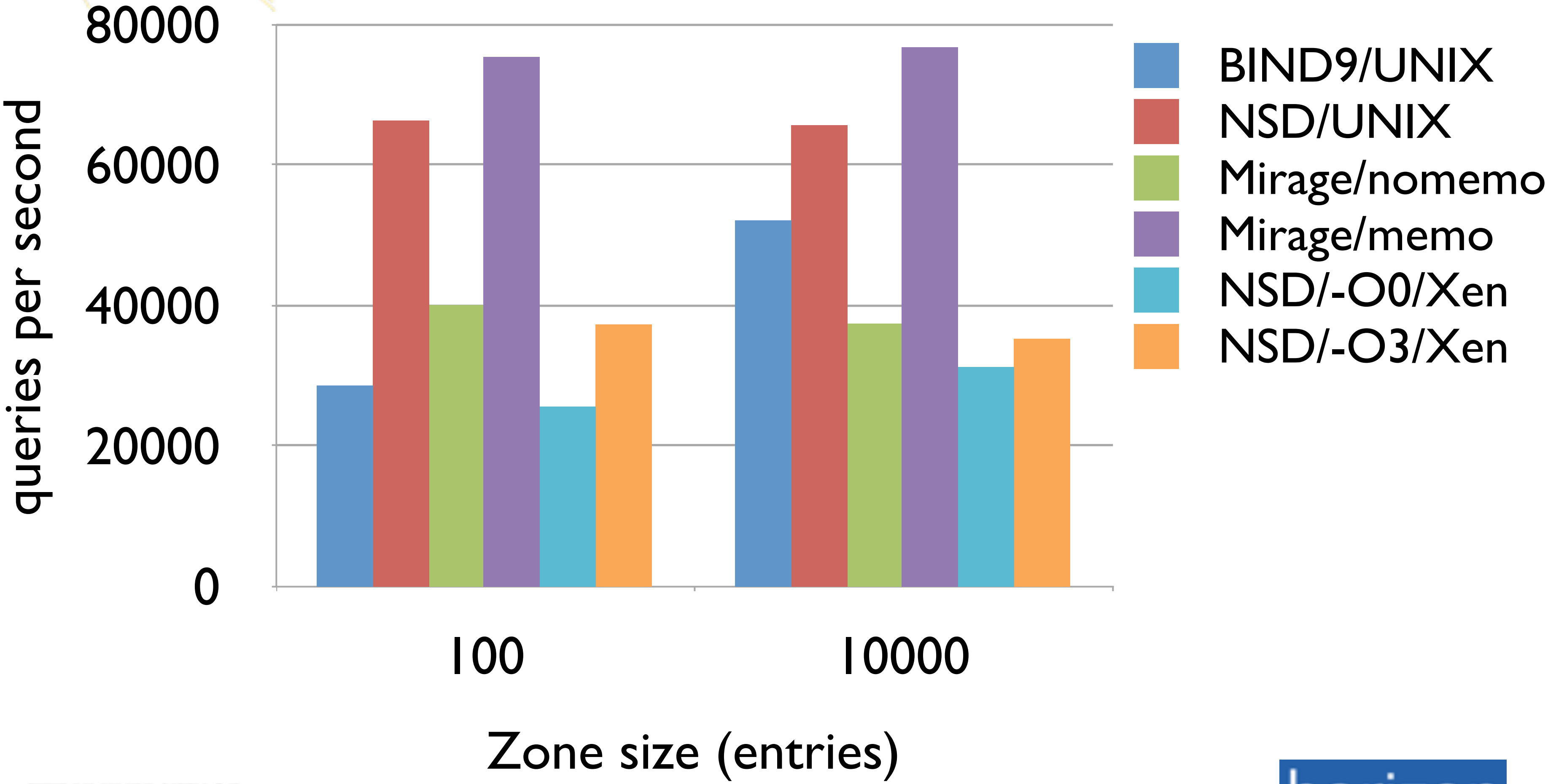


UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH

DNS Server Performance



UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH

Summary

- **OCaml is the baseline language for all new code**
  - C runtime is small, and getting smaller.
  - Is fully event-driven and non-preemptive
- **Rewriting protocols wasn't *that hard***
  - Not necessarily the best research strategy though.
  - But an extremely useful learning experience.
  - Tech transfer is vital.
- **Unikernels fit perfectly on the cloud**
  - Internet protocol building blocks
  - Seamless interop with legacy code through VMs

Mirage Online

- Pure OCaml code at http://github.com/mirage/ for:
  - Device drivers (netfront/blkfront/xenstore)
  - TCP/IPv4 and DHCPv4
  - HTTP
  - DNS(SEC)
  - SSH
  - OpenFlow (controller/switch)
  - vchan IPC
  - 9P :-)
  - NFS
  - FAT32
  - Distributed k/v store: arakoon.org

Mirage Online

- **Online resources:**
  - **http://www.openmirage.org**
  - (Code) http://github.com/mirage
  - (O'Reilly Book) http://realworldocaml.org

- http://www.cl.cam.ac.uk/projects/ocamllabs
  - funded by industry/academia (Jane Street Capital, Citrix, EU FP7, RCUK/Horizon)
  - Focus: real world functional programming with OCaml
  - Need compiler hackers, protocol heads, PL/type theory systems. Must enjoy open source work!

Reserve Slides

Key Research Questions Now

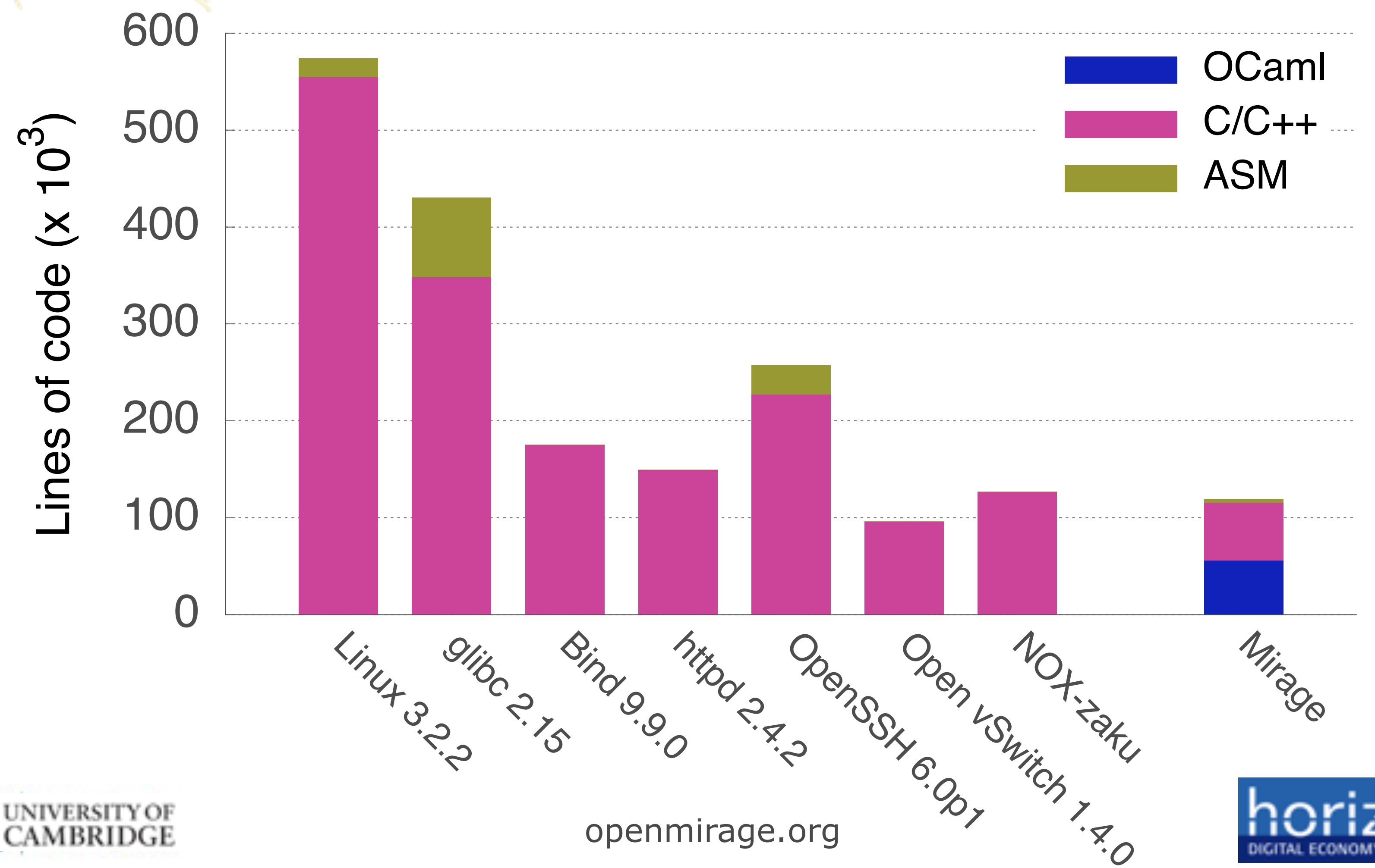
- **Interoperability --- with billions of VMs out there**
  - A unikernel per-language?
  - Interconnect strategies? Heap sharing?
  - Formal method integration easier or harder?
- **Coordination --- planetary scale computers**
  - Resources are highly elastic now.
  - How to coordinate a million microkernels?
  - "Warehouse Scale Computing"
- **Library Applications --- where are they?**
  - Irminsule, a git-like functional distributed database
  - Beanstalk, a self-scaling web server

Optional **VM Sealing**

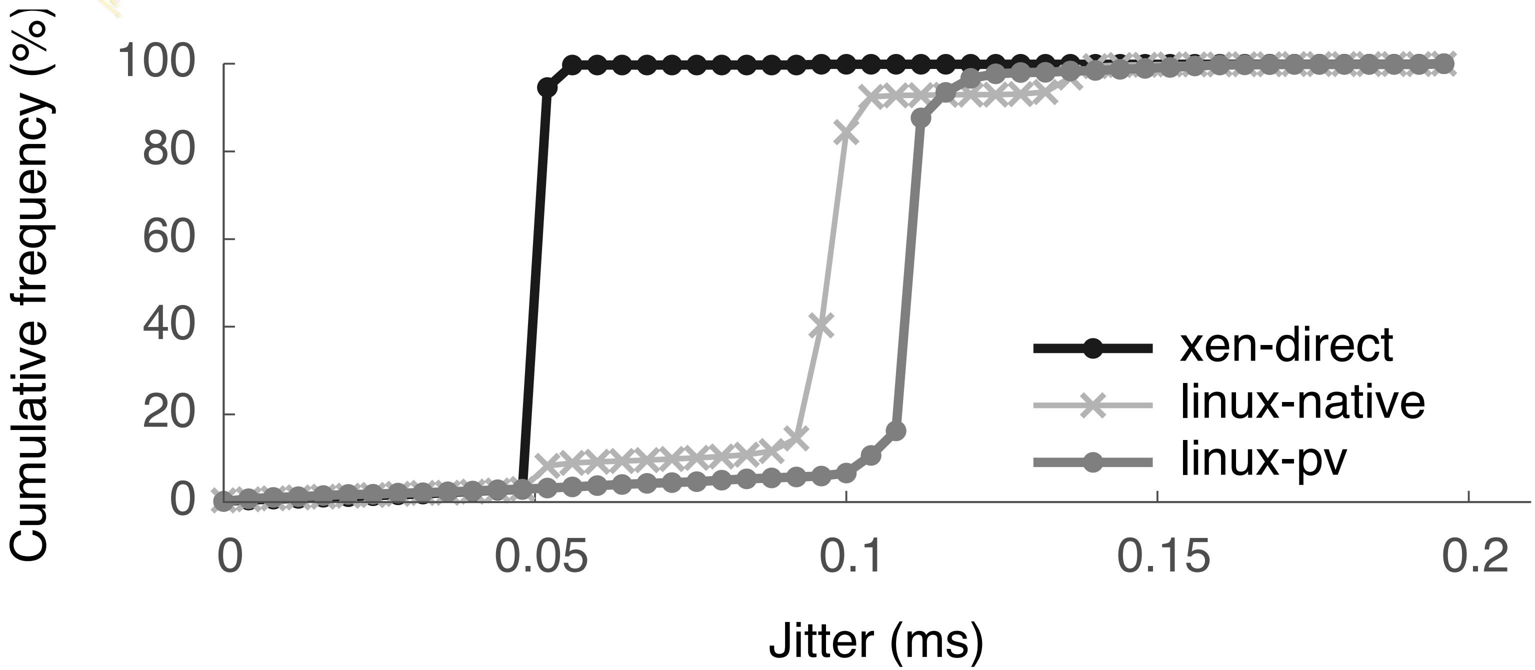
- **Single address-space** and **no dynamic loading**
  - W^X address space
  - Address offsets are randomized at *compile-time*
- **Dropping page table privileges**:
  - Added *freeze* hypercall called just before app starts
  - Subsequent page table updates are rejected by Xen.
  - Exception for I/O mappings if they are non-exec and do not modify any existing mappings.

- Very easy in unikernels due to **focus on compile-time specialisation** instead of run-time complexity.

How Large is Large?

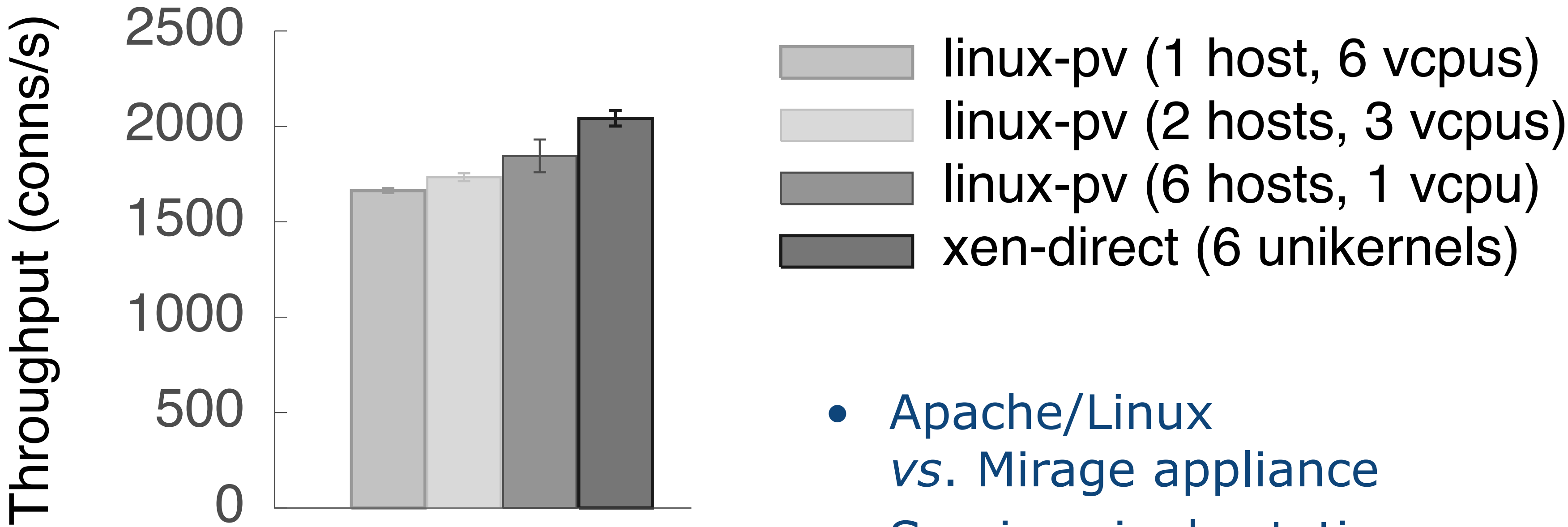


Event Driven co-threads



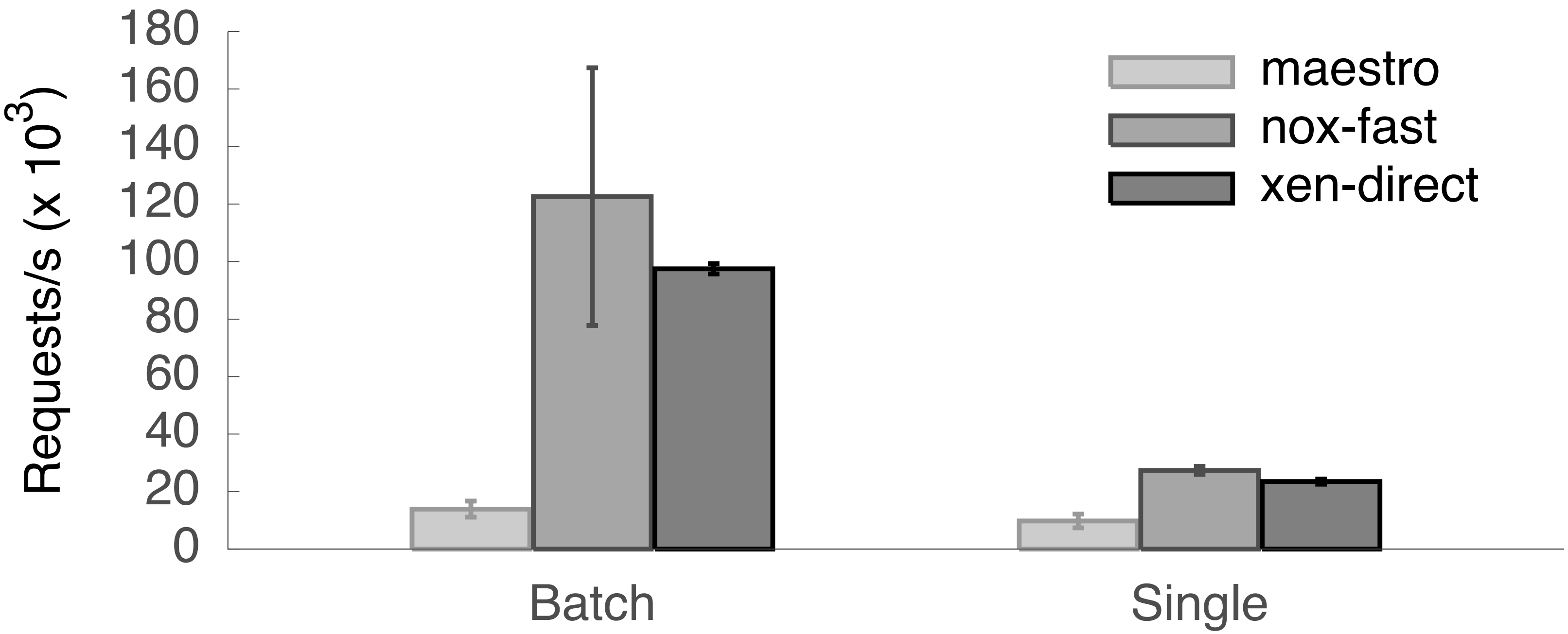
Garbage collected heap management is more efficient in a single address-space environment. Thread latency can be reduced by eliminating multiple levels of scheduling.

Scaling via Parallel Instances



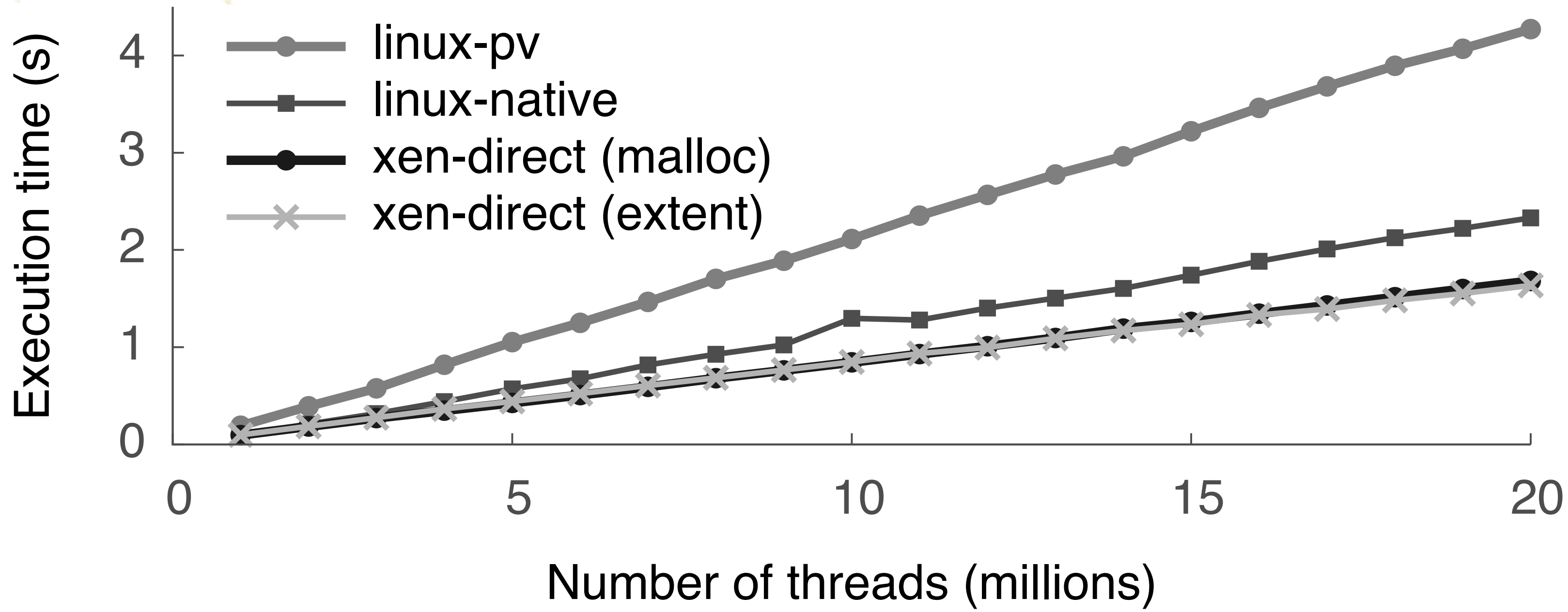
- Apache/Linux *vs*. Mirage appliance
- Serving single static page

Openflow Controller performance



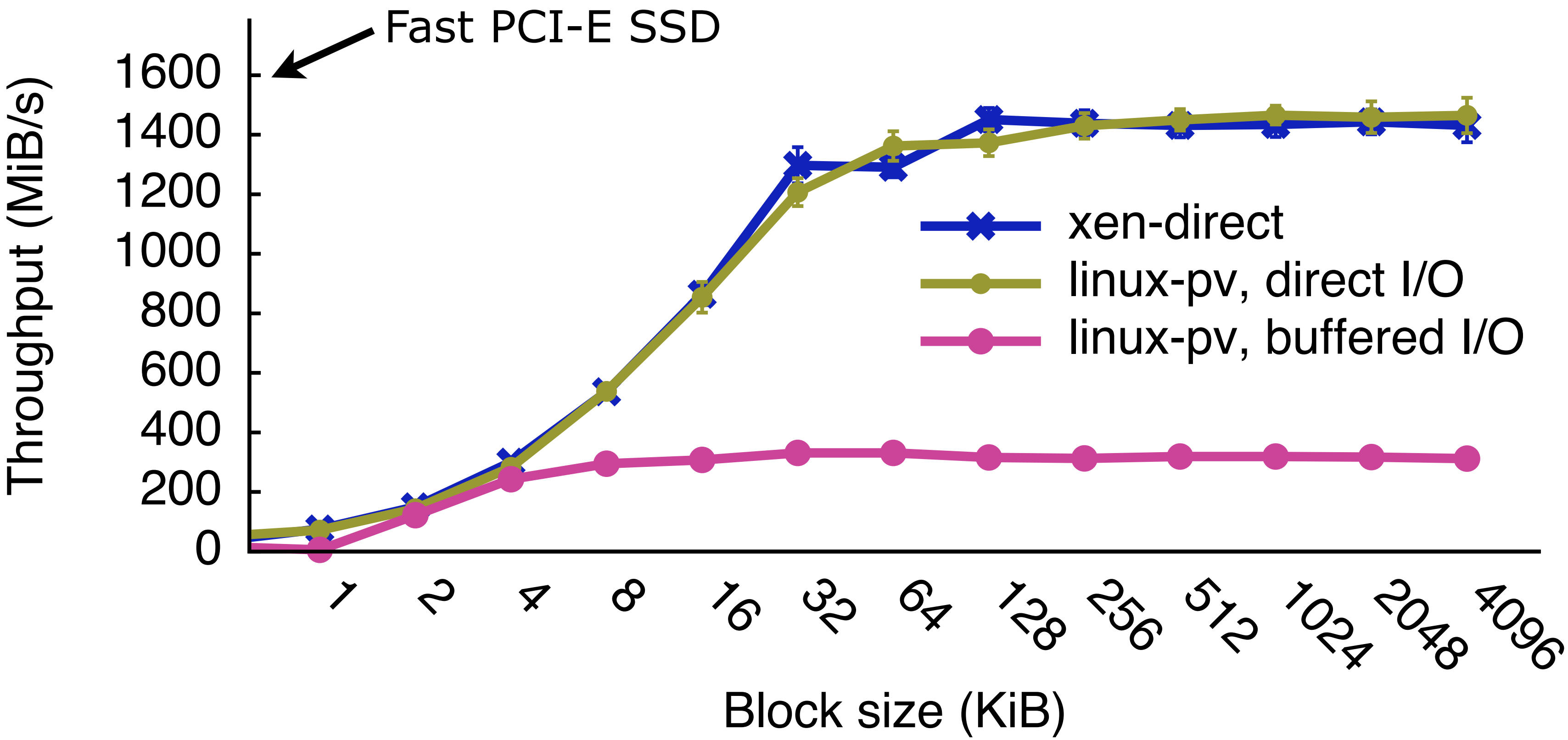
- Openflow controller is competitive with Nox (C++), but much more high-level. Applications can link directly against the switch to route their data.

Single-instance Thread Scaling



Threads are heap allocated values, so benefit from the faster garbage collection cycle in the Mirage Xen version, and the scheduler can be overridden by application-specific needs.

Microbenchmarks: Block Storage



UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH

Progressive Specialization

Develop	*Test*	*Deploy*
`ubuild posix-socket`	`ubuild posix-direct`	`ubuild xen-direct`
kernel sockets	tuntap+safe I/O stack	safe device drivers
bytecode VM	x86_64 native code	link time optimisation
ELF ← REPL / Linux	ELF / Linux; ELF ELF / FreeBSD	μ μ μ μ μ μ / Xen

UNIVERSITY OF CAMBRIDGE

openmirage.org

horizon DIGITAL ECONOMY RESEARCH