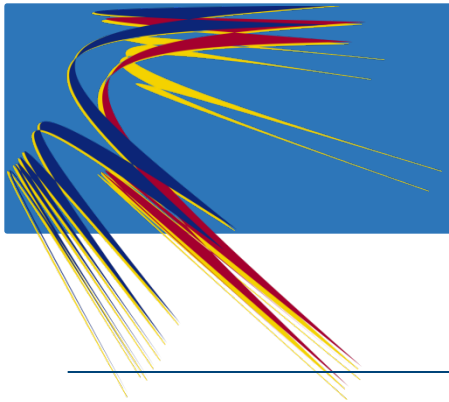


horizon
DIGITAL ECONOMY RESEARCH

Unikernels: Extreme Specialization of Virtual Appliances

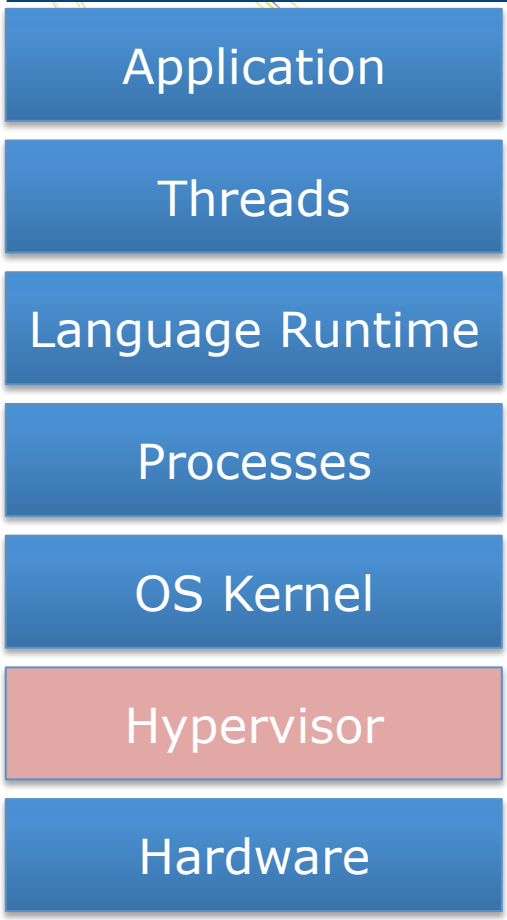
Richard Mortier, University of Nottingham

with Anil Madhavapeddy, Haris Rotsos,
Balraj Singh, Andrew Moore,
Jon Crowcroft, Steve Hand
(University of Cambridge)

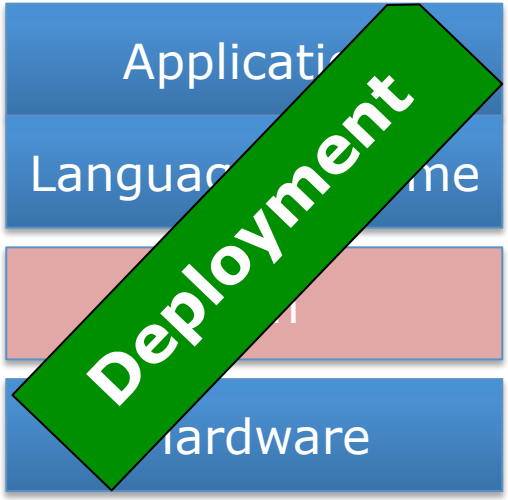
Thomas Gazagnaire (OCamlPro)
Dave Scott (Citrix R&D)

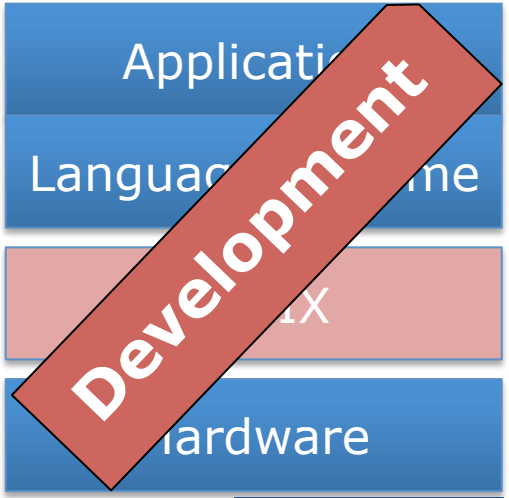
The University of Nottingham

Modern Stacks are Too Large

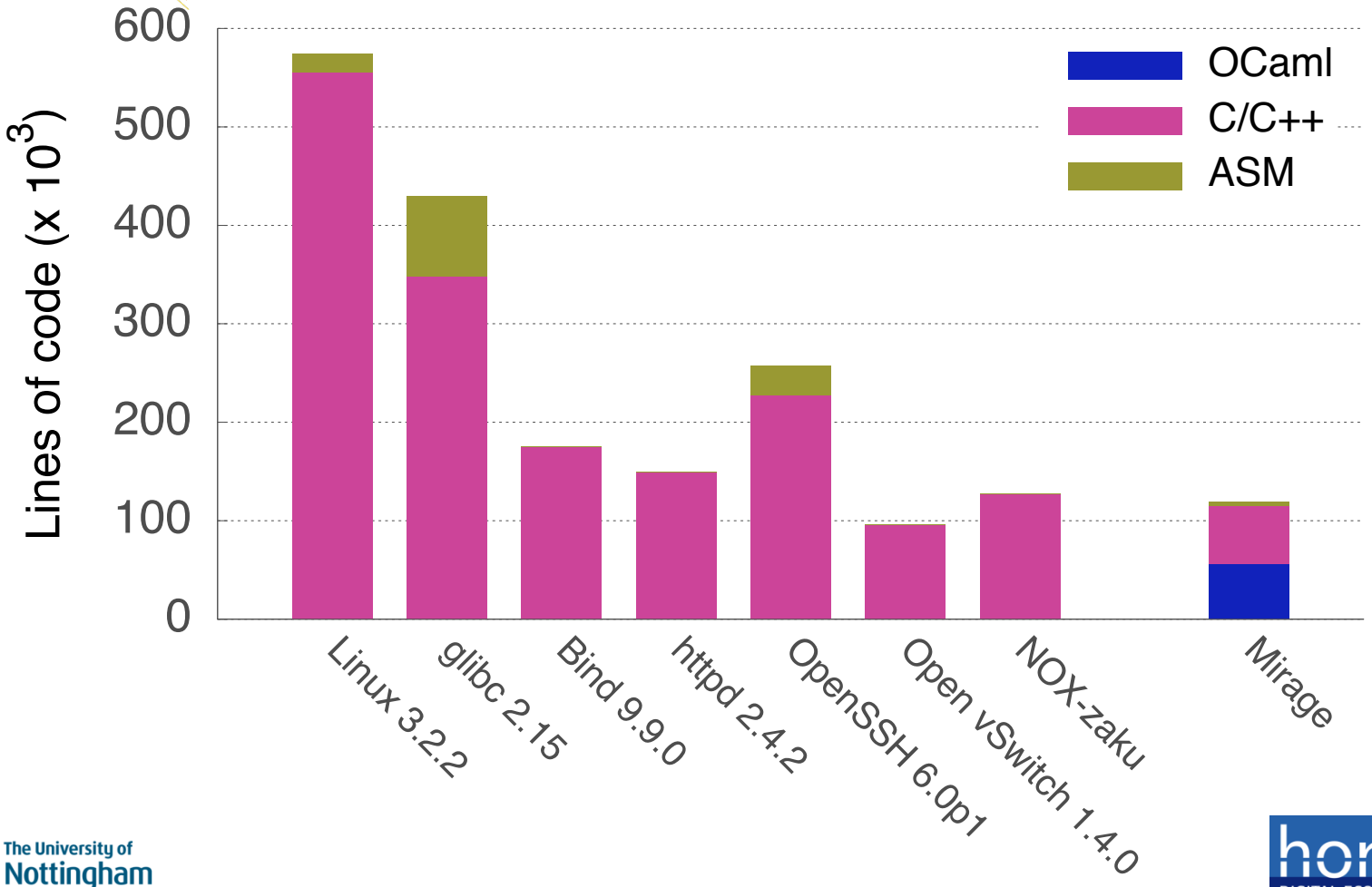


- Millions of lines of code & configuration
- Why build for clouds as for desktops?
- We can simplify!





How Large is Large?



Why Do We Care?

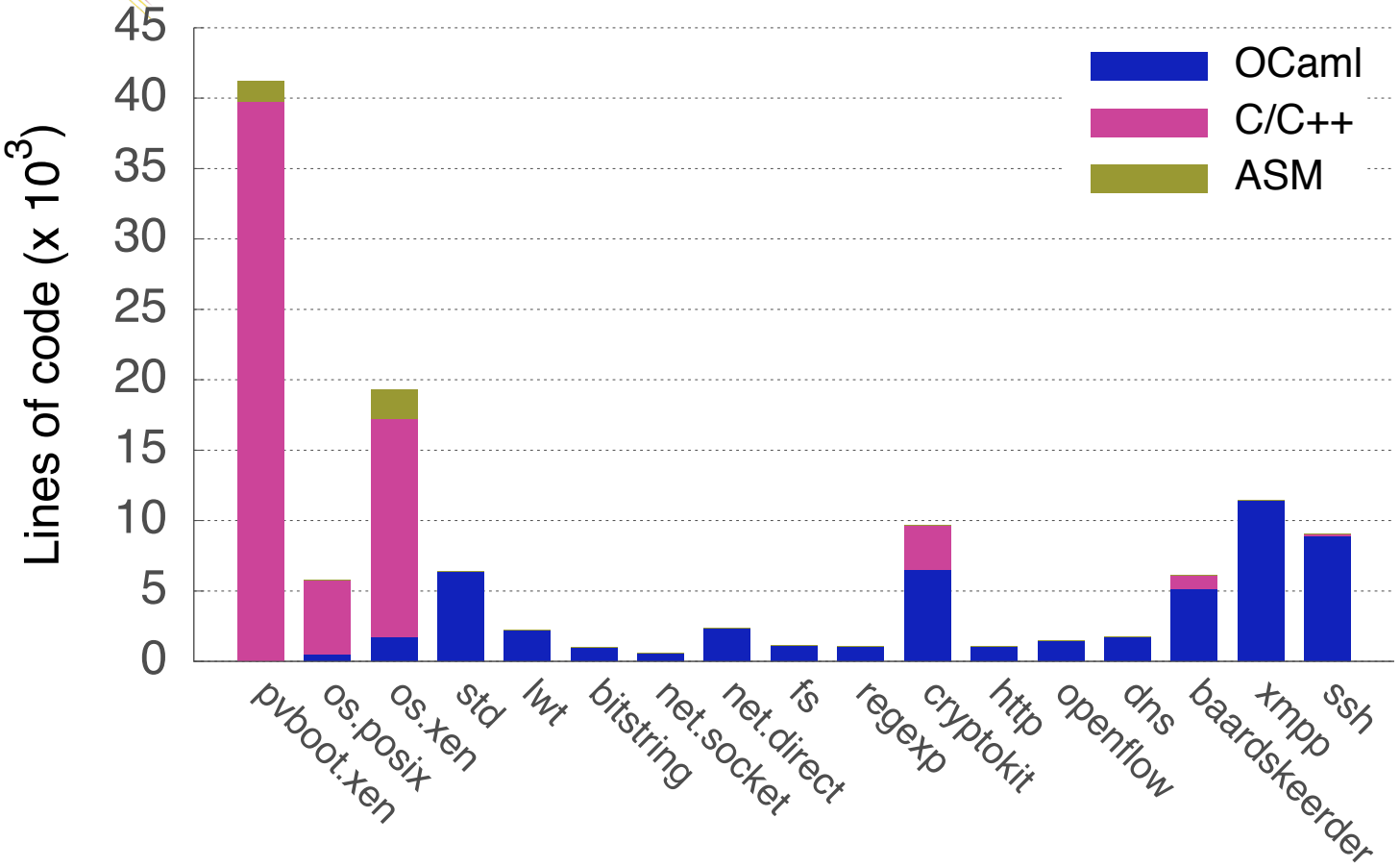
- Critical memory safety bugs still occur! E.g., In March 2012:
- **CVE-2012-1182 – Samba**
  - RPC code generator overflow
  - Variable containing buffer length checked independently of variable used to allocate memory for buffer
  - *Leads to root exploit*
- **CVE-2012-2110 – OpenSSL**
  - Combination of integer conversion bug with realloc wrappers
  - Unsigned treated as signed, but realloc'd buffer size not clamped
  - *Leads to heap corruption*

Threat Model

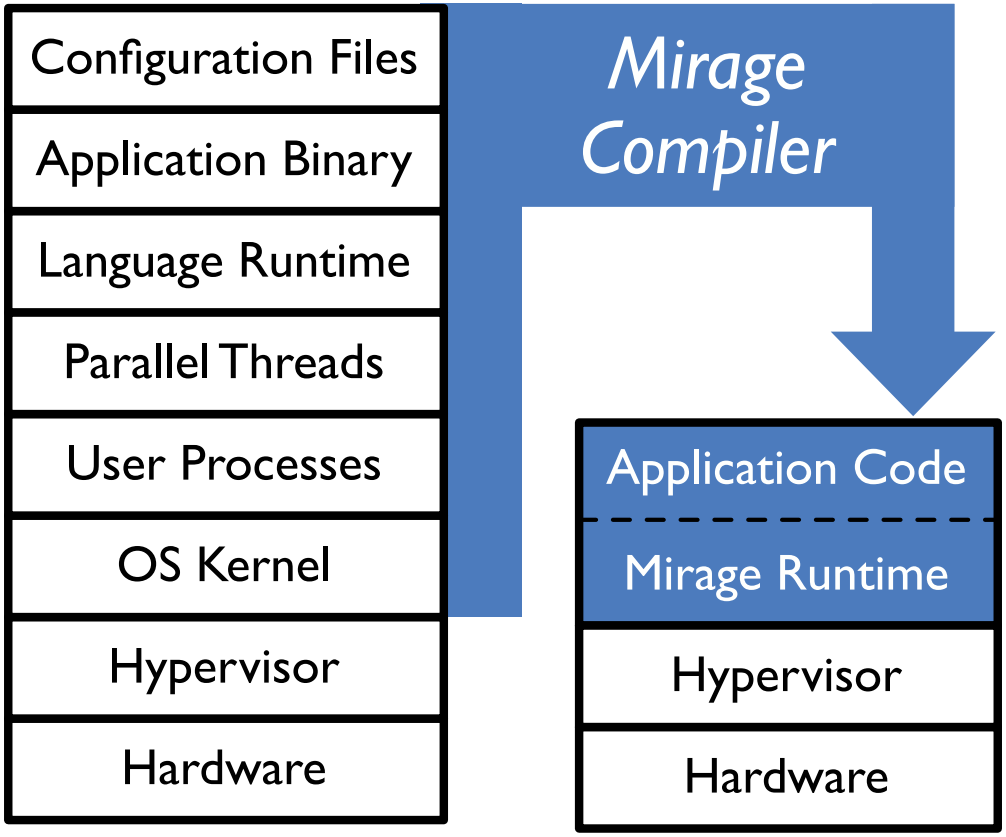
- Attack from without, not within
- Your VMs are not multi-user
- …but they are in a *multi-tenant datacenter*
- …and they are *always network connected*

So, What About Mirage Then?

…starting by recalling that all of Mirage is 5x fewer lines of code than Linux alone!



It's Only Code – Just Build It!



- Cloud appliances, hosted in multi-tenant datacenters are under constant attack
- Build the whole system into a single type-safe virtual machine image
- Reconfigure by ***rebuild*** + ***deploy***

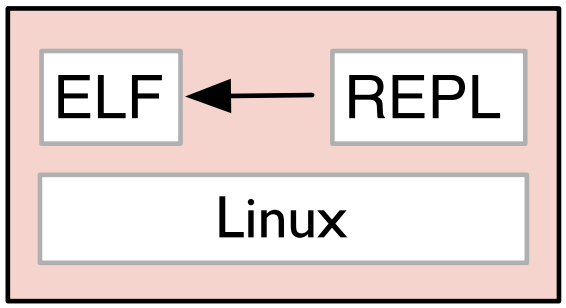
Key Features of Mirage

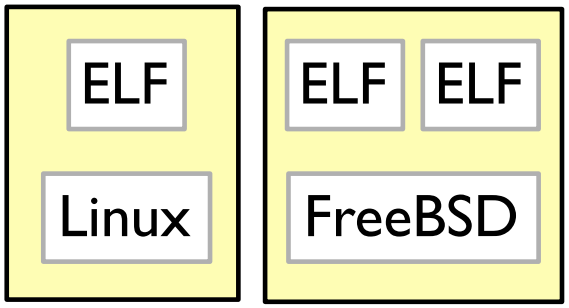
- **Static typing**
  - Eliminates classes of bugs
  - Large set of libraries provided
- **Cooperative concurrency**
  - Wrapped up in *Lwt* syntax extensions
  - Threads encapsulated and hidden within typed modules
- **Fully re-entrant**
  - Initialization via configuration record
- **No dynamic loading**
  - Configuration evaluated at compile-time, and *sealed*
  - Recompile and redeploy to reconfigure

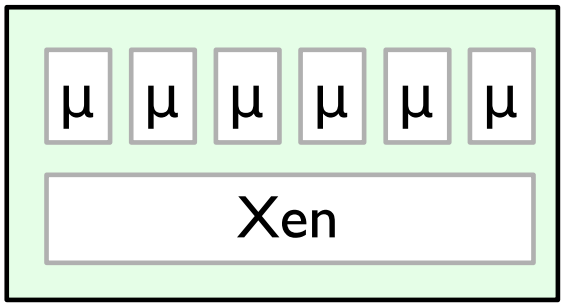
Progressive Specialization

Develop → *Test* → *Deploy*

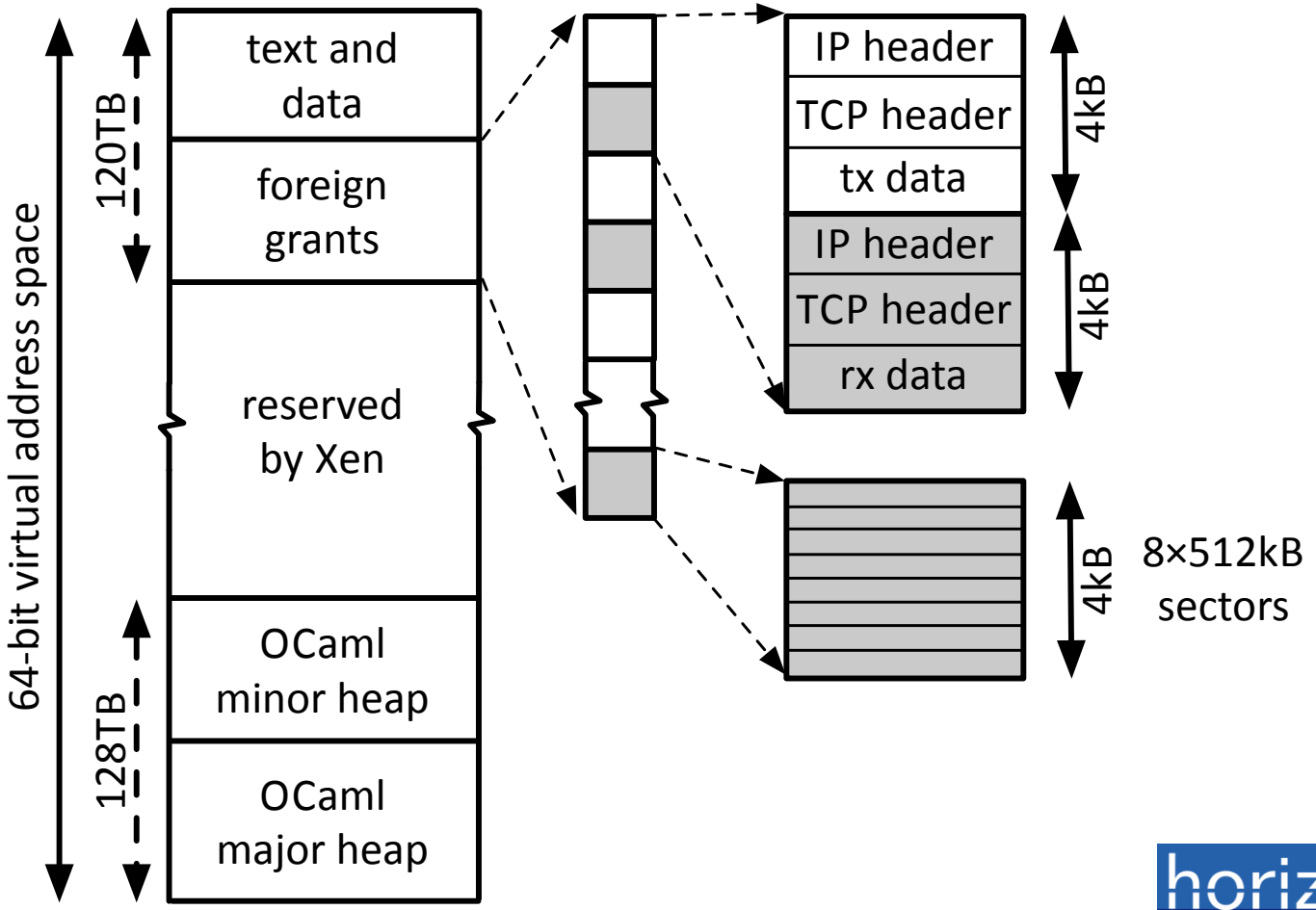
`ubuild posix-socket`	`ubuild posix-direct`	`ubuild xen-direct`
kernel sockets	tuntap+safe I/O stack	safe device drivers
bytecode VM	x86_64 native code	link time optimisation



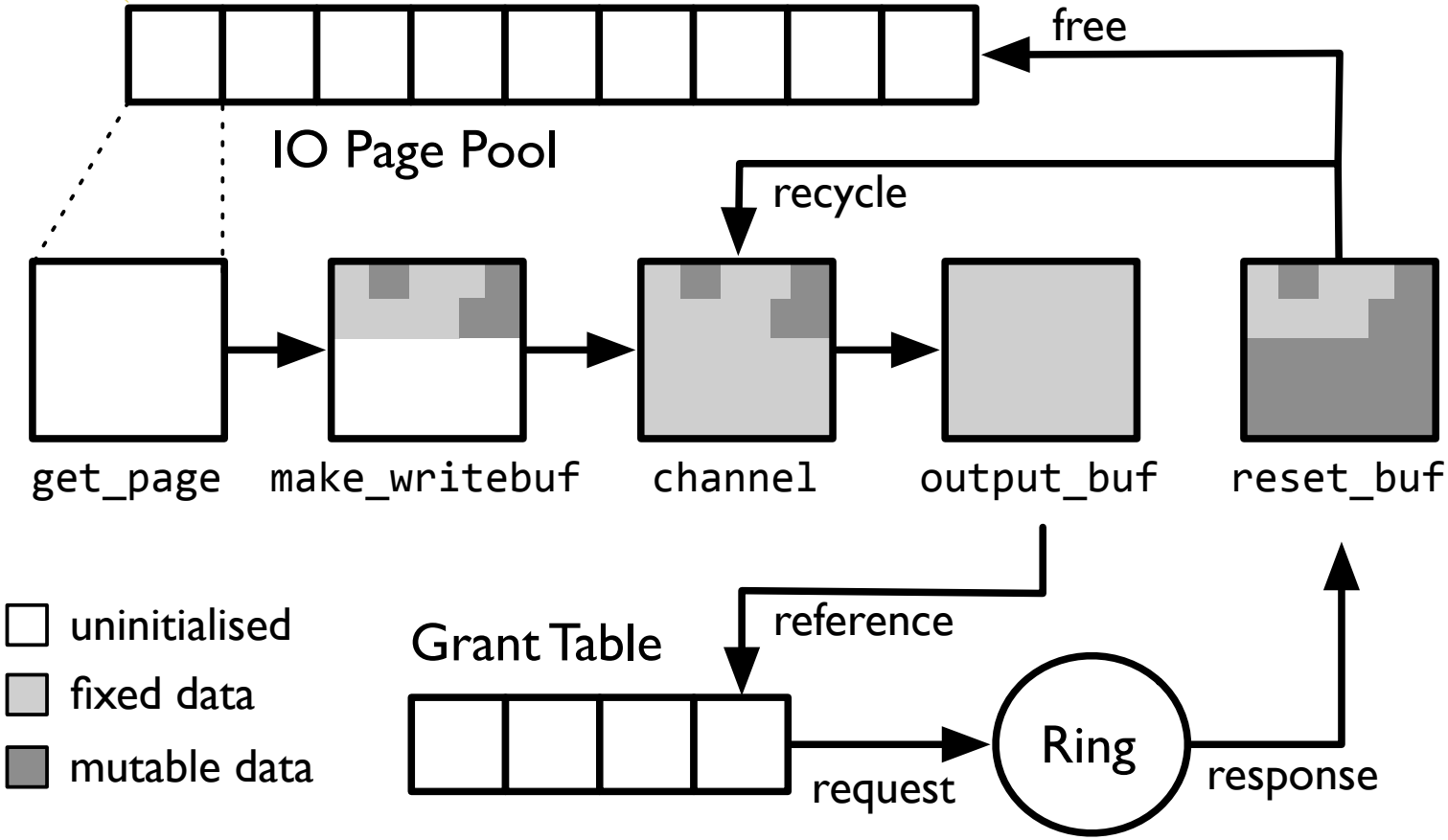




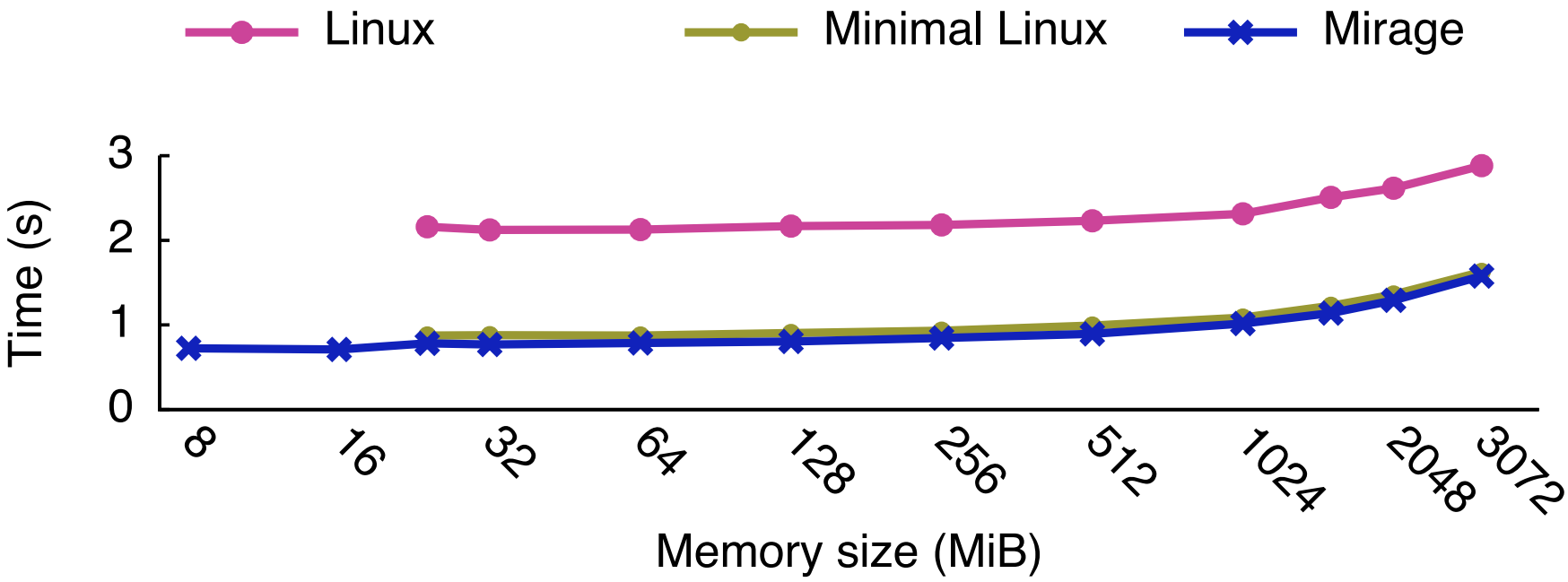
Simplified Memory Management



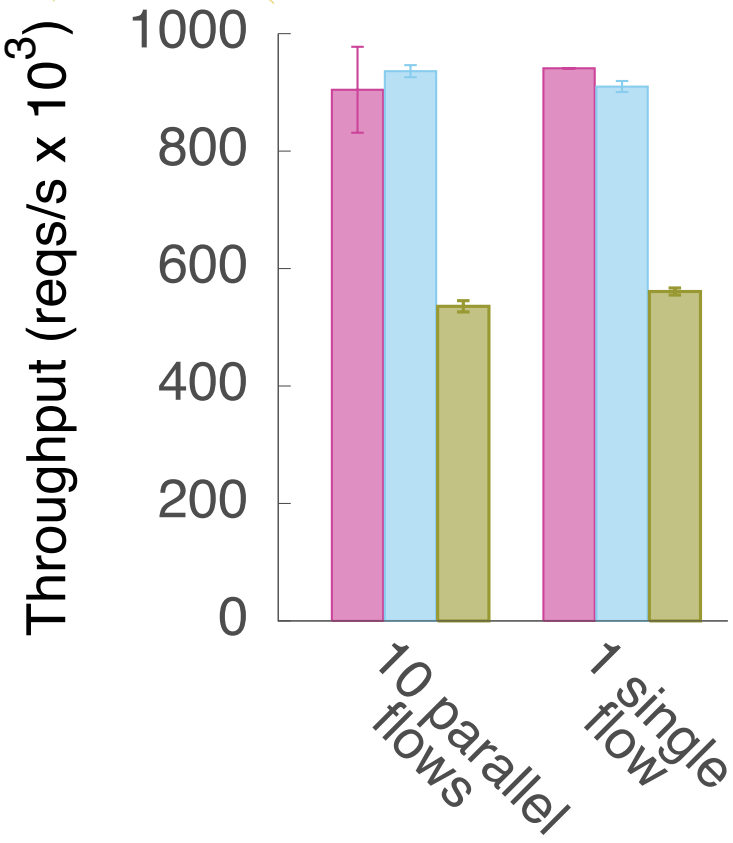
Careful IO Buffer Management



Microbenchmarks: Boot Time



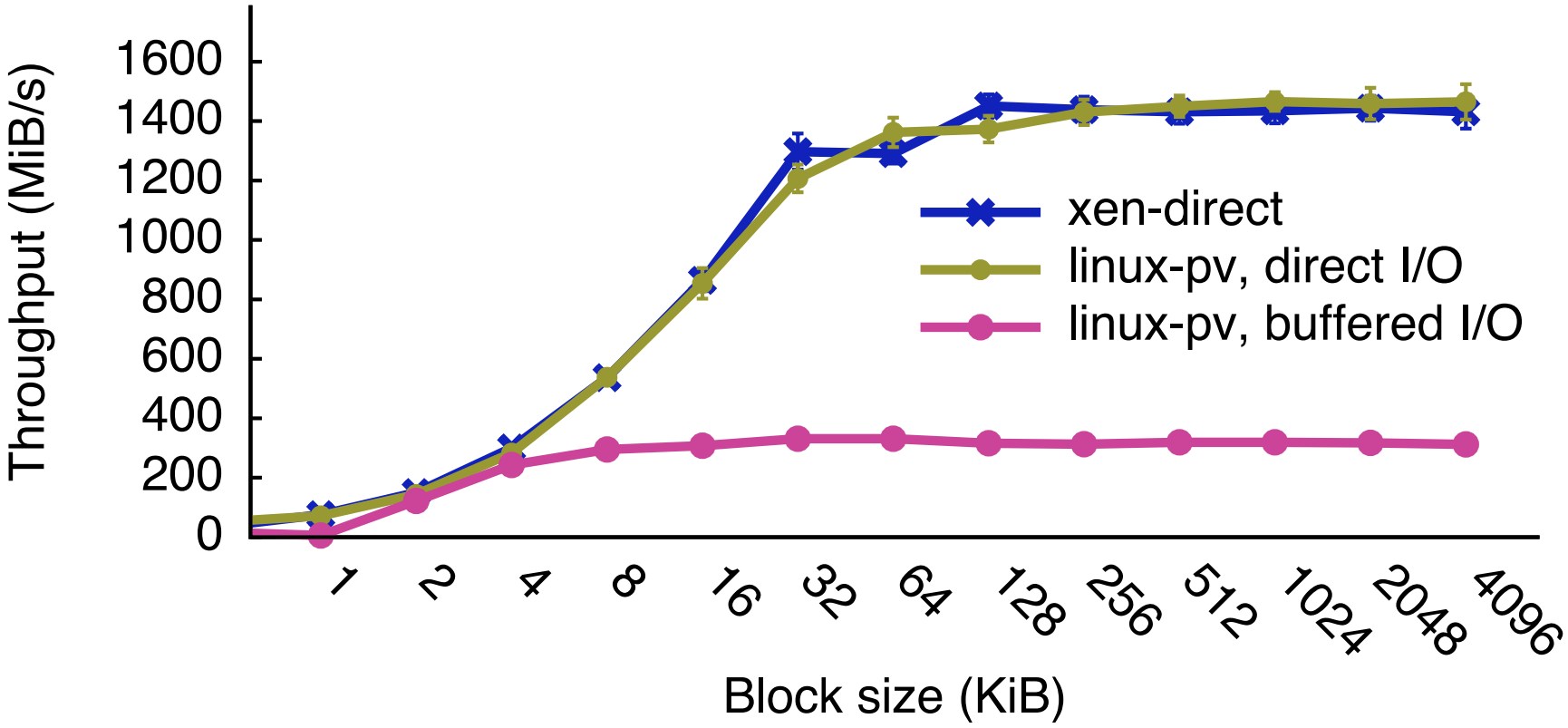
Microbenchmarks: TCP



- linux-pv, tx & rx
- linux-pv tx, xen-direct rx
- xen-direct tx, linux-pv rx

- Simple throughput test
- Performance bug in TX path
  - …being fixed

Microbenchmarks: Block Storage

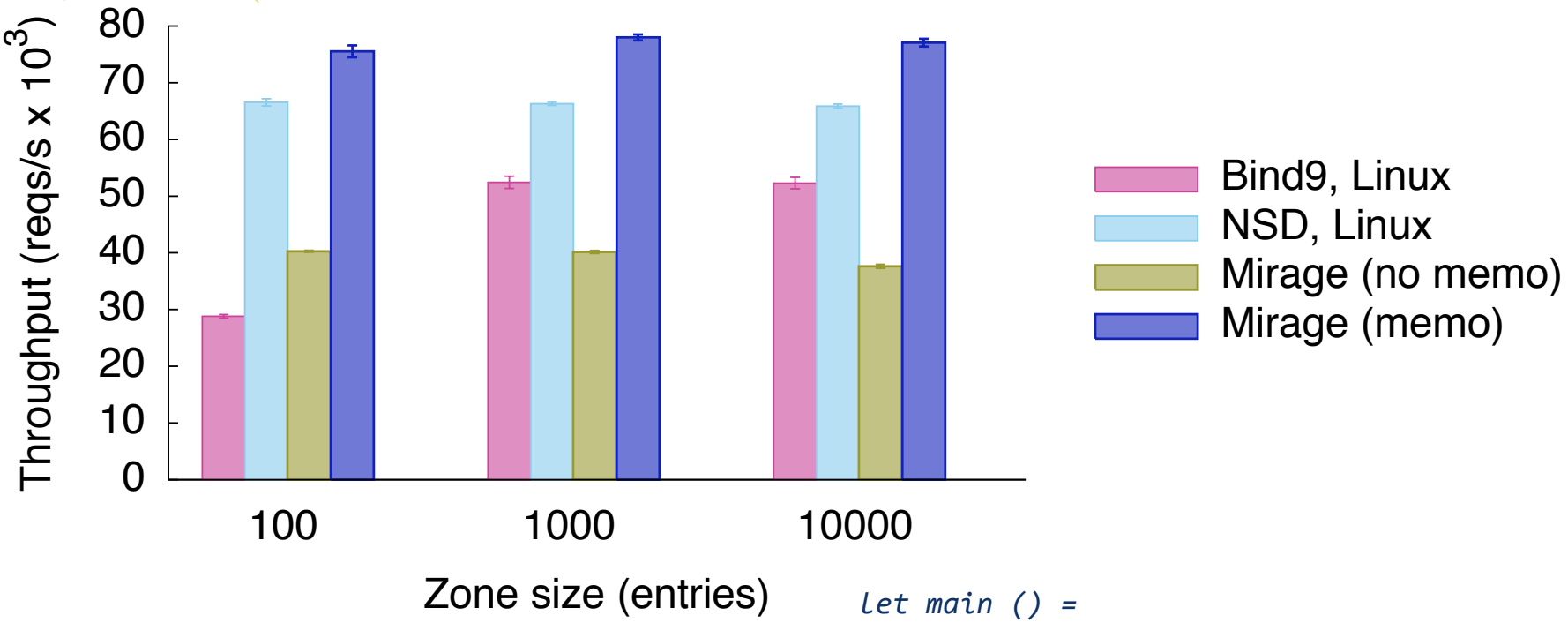


Microbenchmarks: Summary

- Mirage is comparable to or exceeds Linux PV performance
  - I.e., **there is no significant overhead from use of a functional programming language**

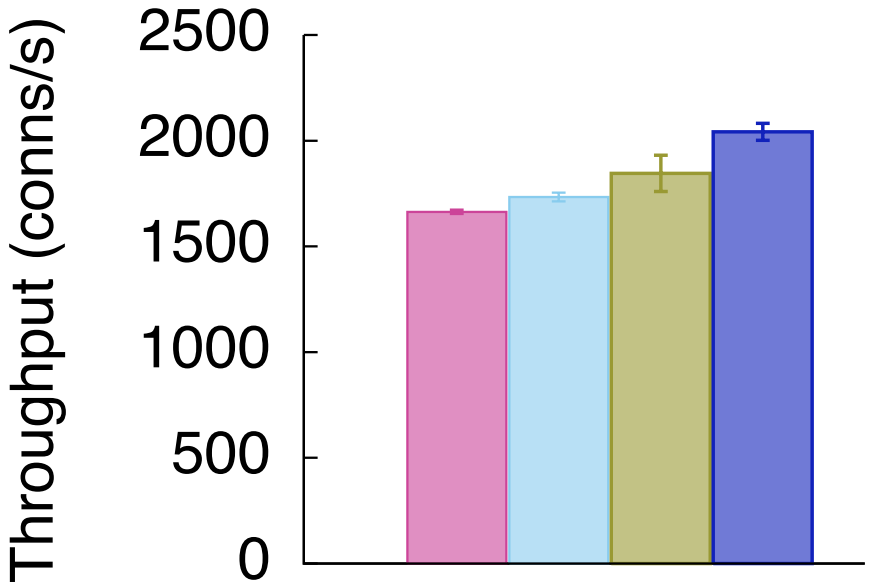
- But what about in some more "realistic" scenarios?
- Will present DNS and Web servers
  - We also have OpenFlow Controller and Switch implementations
  - Switch performance matches Open vSwitch
  - Controller performance is between NOX and NOX-fast

DNS Server Performance




```
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)
```

Scaling via Multiple Instances



- Linux (1 host, 6 vcpus)
- Linux (2 hosts, 3 vcpus)
- Linux (6 hosts, 1 vcpu)
- Mirage (6 unikernels)

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

Roadmap

- Ongoing work
  - FreeBSD kernel module
  - Hardware 64 bit version for rPI and KVM
  - XCP integration
- Plans
  - OpenFlow extensions and appliances
  - Self-scaling appliances
  - Signposts
- Community building
  - http://openmirage.org/wiki/install – always happy to receive patches and advise on building appliances!
  - August 2012 – XenSummit talk @ UCSD
  - September 2012 – OUD 2012 talk @ Copenhagen

http://www.horizon.ac.uk/

Questions?

richard.mortier@
nottingham.ac.uk

https://lists.cam.ac.uk
/mailman/listinfo/cl-mirage

http://openmirage.org
/wiki/install