Fido: Fast Inter-Virtual-Machine Communication for Enterprise Appliances

Anton Burtsev[†], Kiran Srinivasan,
Prashanth Radhakrishnan, Lakshmi N. Bairavasundaram,
Kaladhar Voruganti, Garth R. Goodson

[†]University of Utah,
School of Computing

NetApp, Inc

Enterprise appliances

Network attached storage, routers, etc.

- High performance
- Scalable and highly-available access

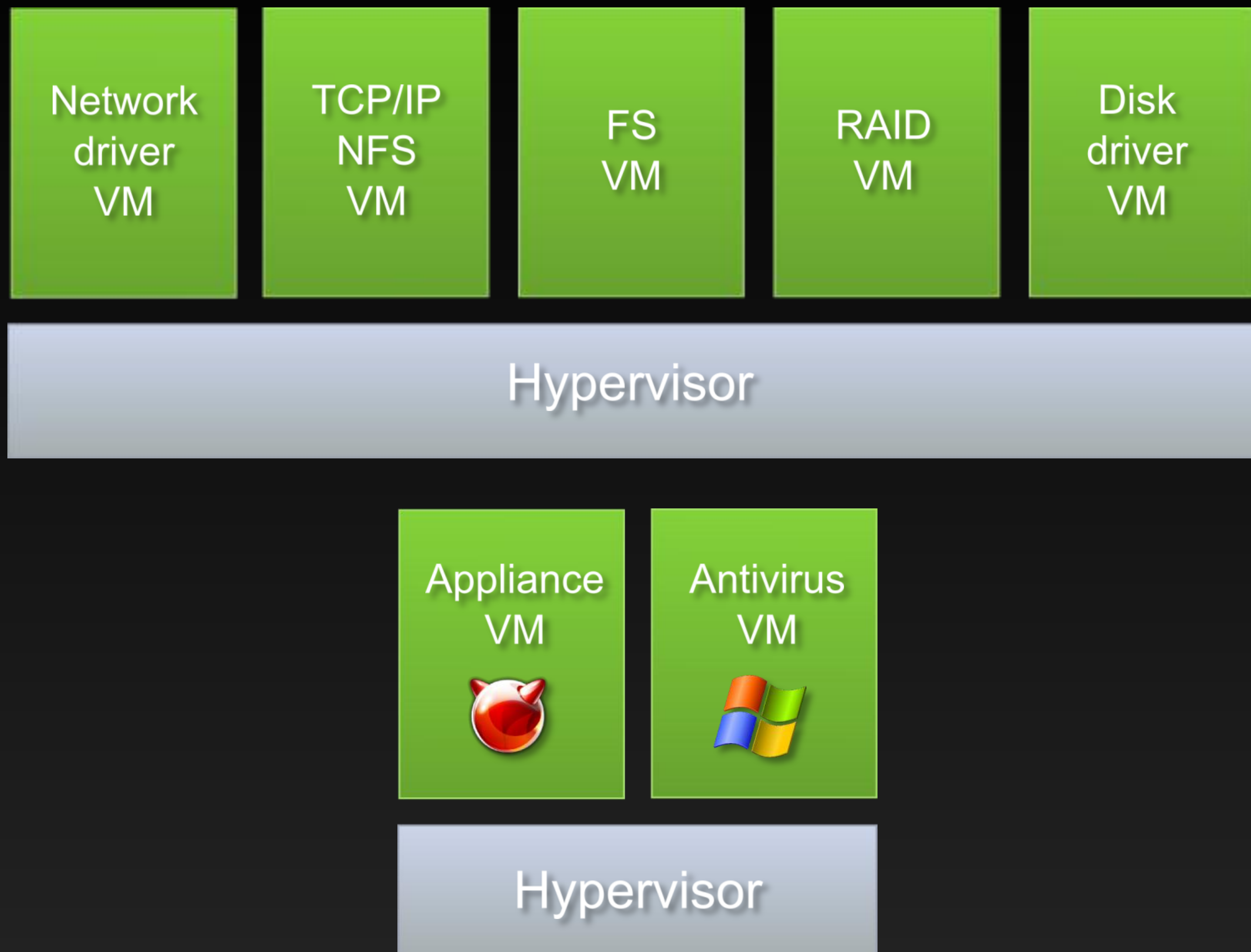
Example Appliance

- Monolithic kernel
- Kernel components

Problems:

- Fault isolation
- Performance isolation
- Resource provisioning

Split architecture



Benefits of virtualization

- High availability
  - Fault-isolation
  - Micro-reboots
  - Partial functionality in case of failure
- Performance isolation
- Resource allocation
  - Consolidation and load balancing, VM migration
- Non-disruptive updates
  - Hardware upgrades via VM migration
  - Software updates as micro-reboots
- Computation to data migration

Main Problem: Performance

Is it possible to match performance of a monolithic environment?

- Large amount of data movement between components
  - Mostly cross-core
  - Connection oriented (established once)
  - Throughput optimized (asynchronous)
  - Coarse grained (no one-word messages)
  - Multi-stage data processing
- Main cost contributors
  - Transitions to hypervisor
  - Memory map/copy operations
  - Not VM context switches (multi-cores)
  - Not IPC marshaling

Main Insight: Relaxed Trust Model

- Appliance is built by a single organization
- Components:
  - Pre-tested and qualified
  - Collaborative and non-malicious
- Share memory read-only across VMs!
- Fast inter-VM communication
  - Exchange only pointers to data
    - No hypervisor calls (only cross-core notification)
    - No memory map/copy operations
  - Zero-copy across entire appliance

Contributions

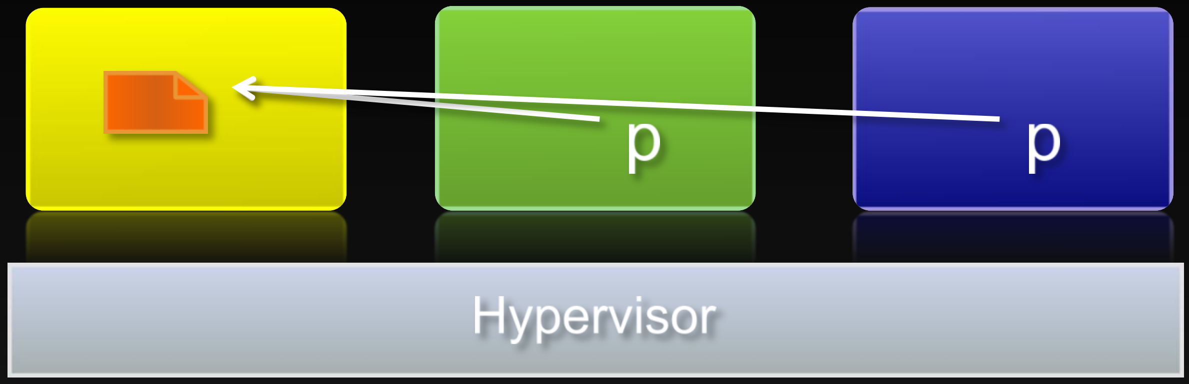
- Fast inter-VM communication mechanism
- Abstraction of a single address space for traditional systems
- Case study
  - Realistic microkernelized network attached storage

Design

Design Goals

- Performance
  - High-throughput
- Practicality
  - Minimal guest system and hypervisor dependencies
  - No intrusive guest kernel changes
- Generality
  - Support for different communication mechanisms in the guest system

Transitive Zero Copy

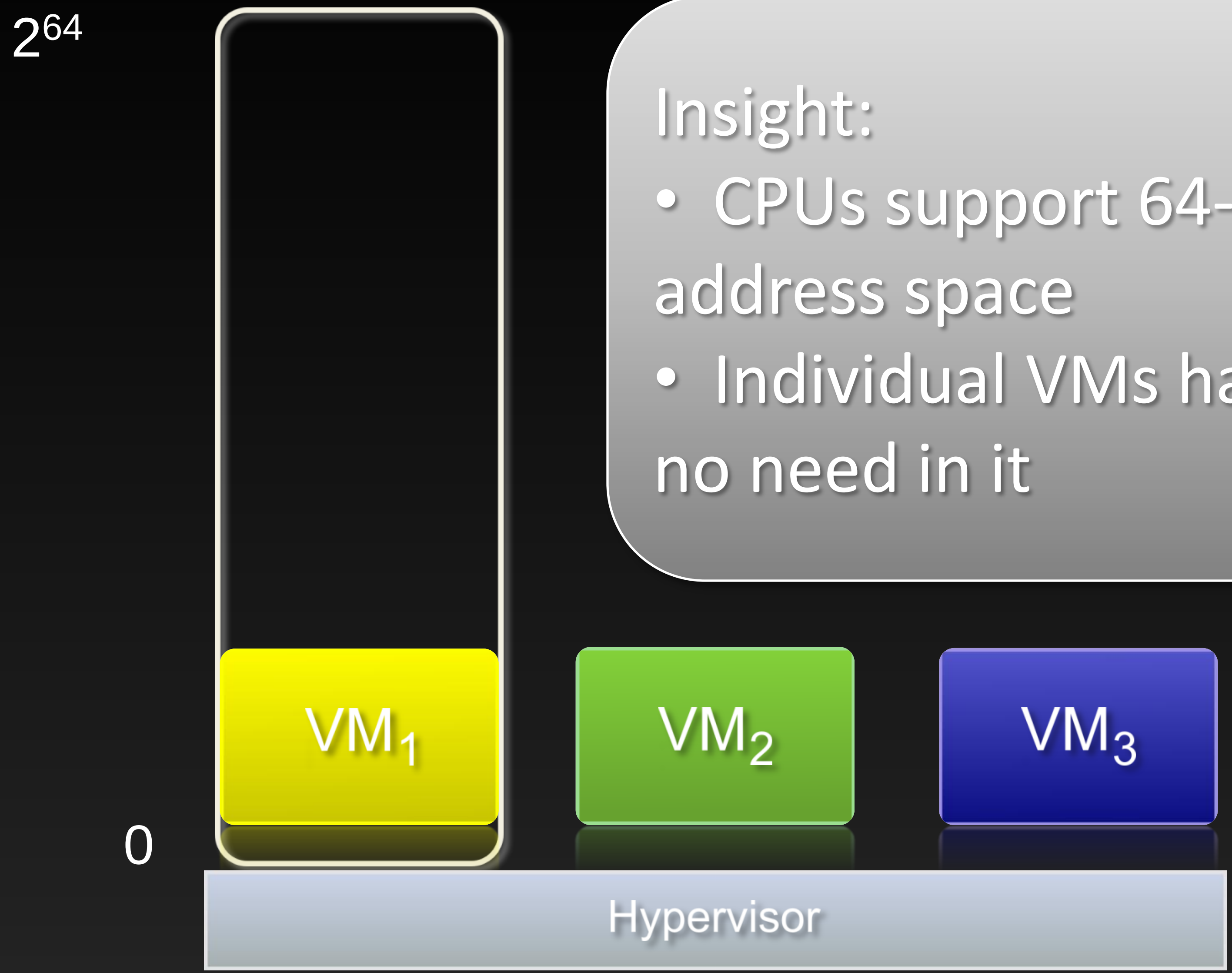


- Goal
  - Zero-copy across entire appliance
  - No changes to guest kernel
- Observation
  - Multi-stage data processing

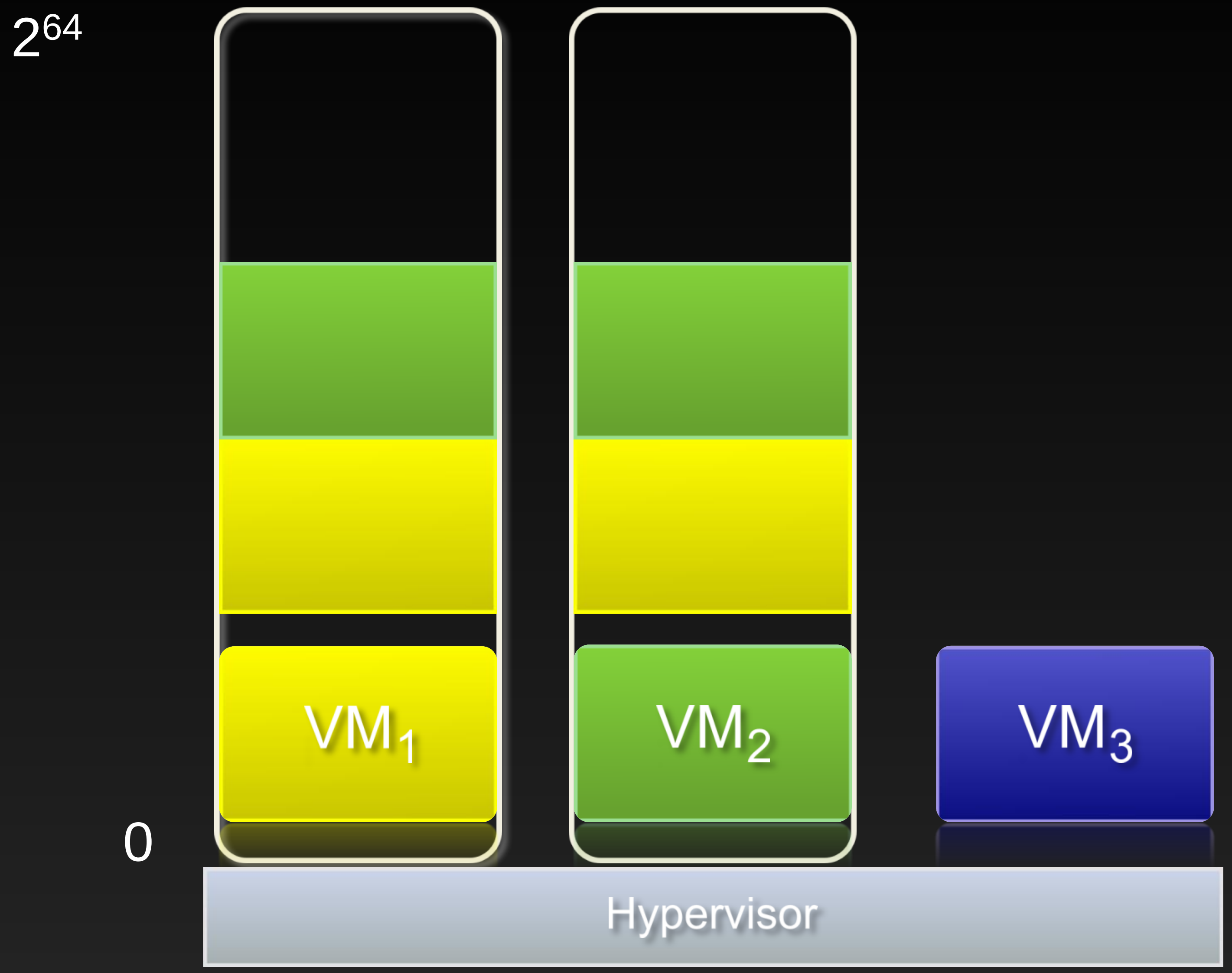
Pseudo Global Virtual Address Space

Insight:

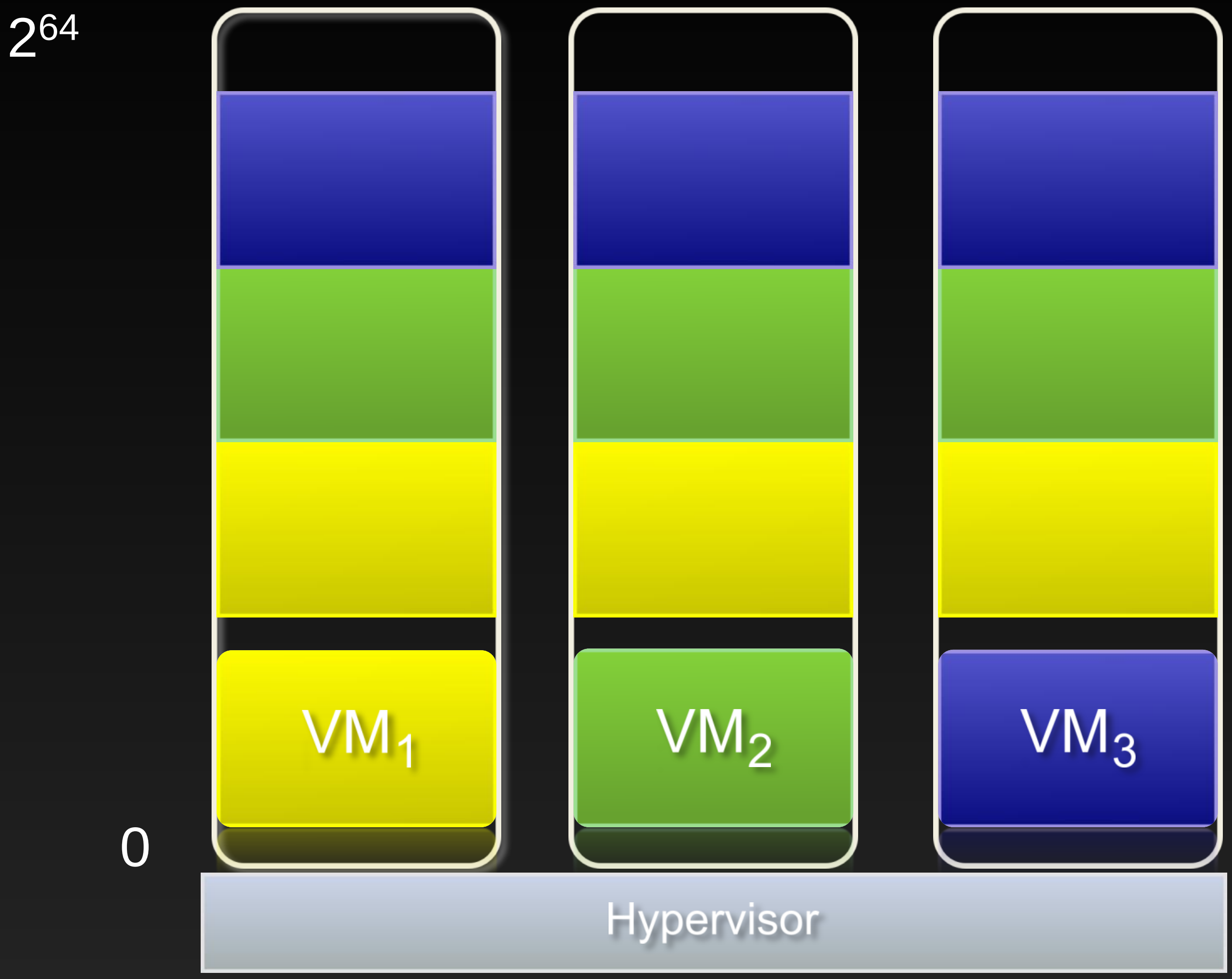
- CPUs support 64-bit address space
- Individual VMs have no need in it



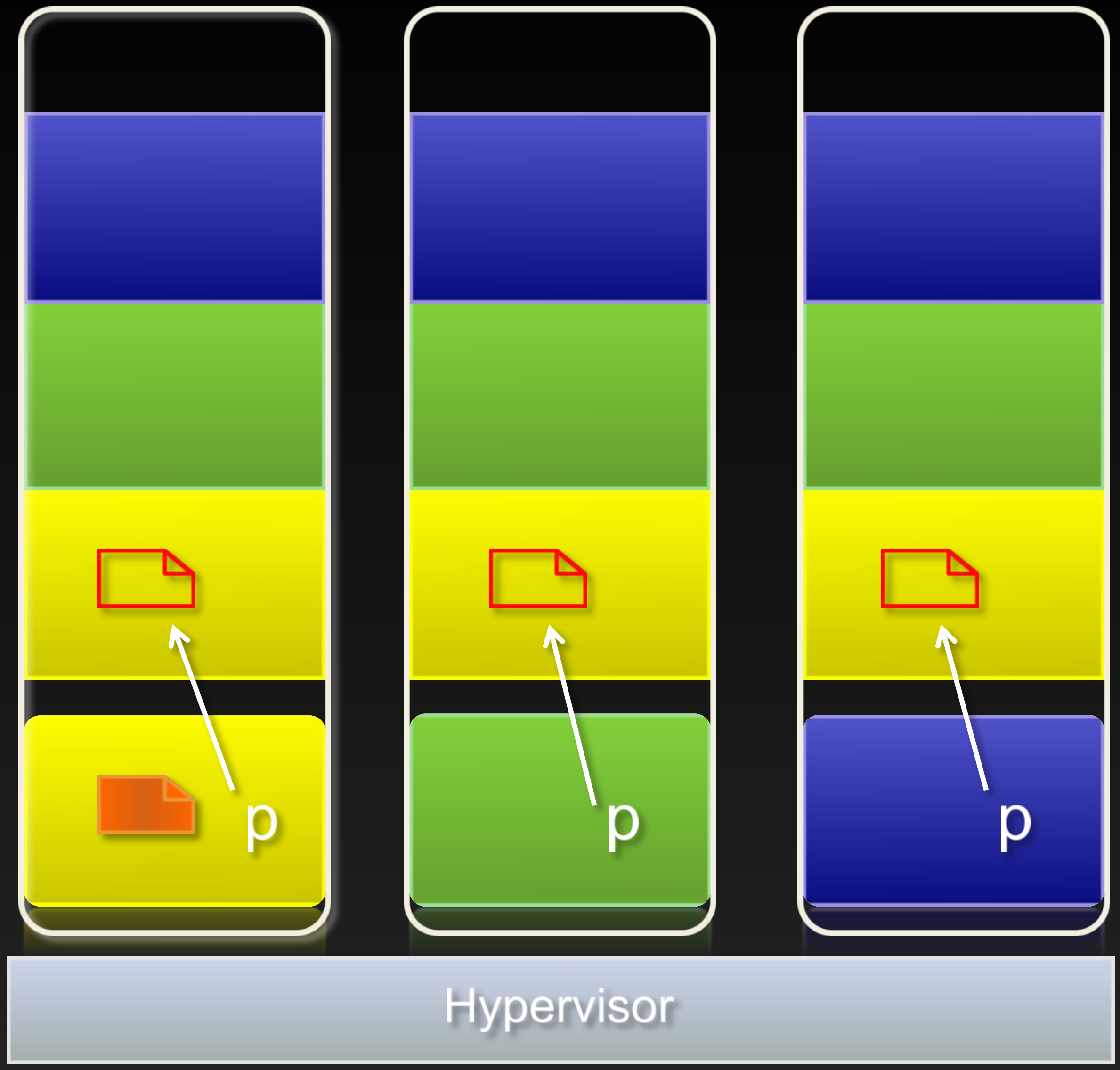
Pseudo Global Virtual Address Space



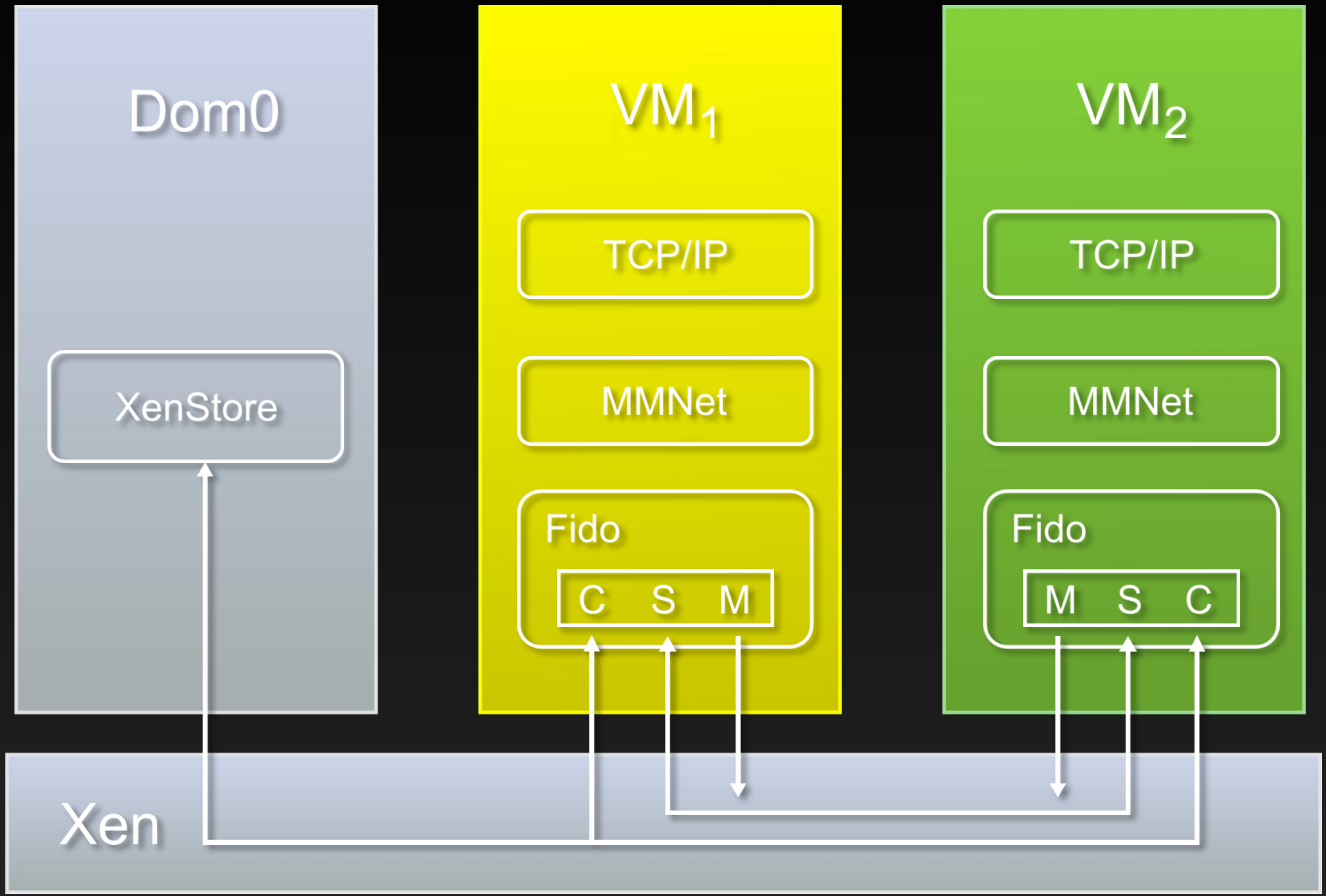
Pseudo Global Virtual Address Space



Transitive Zero Copy

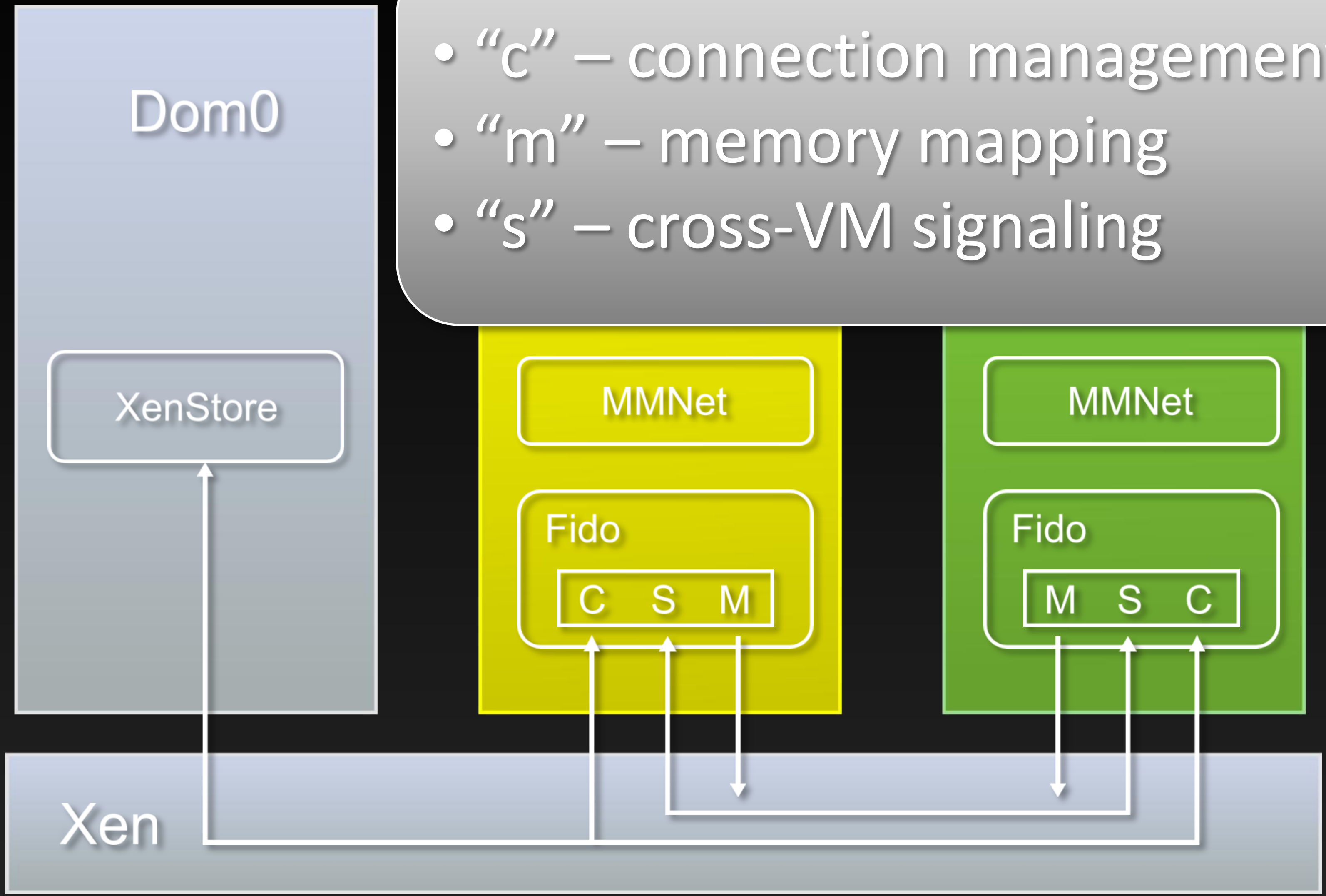


Fido: High-level View



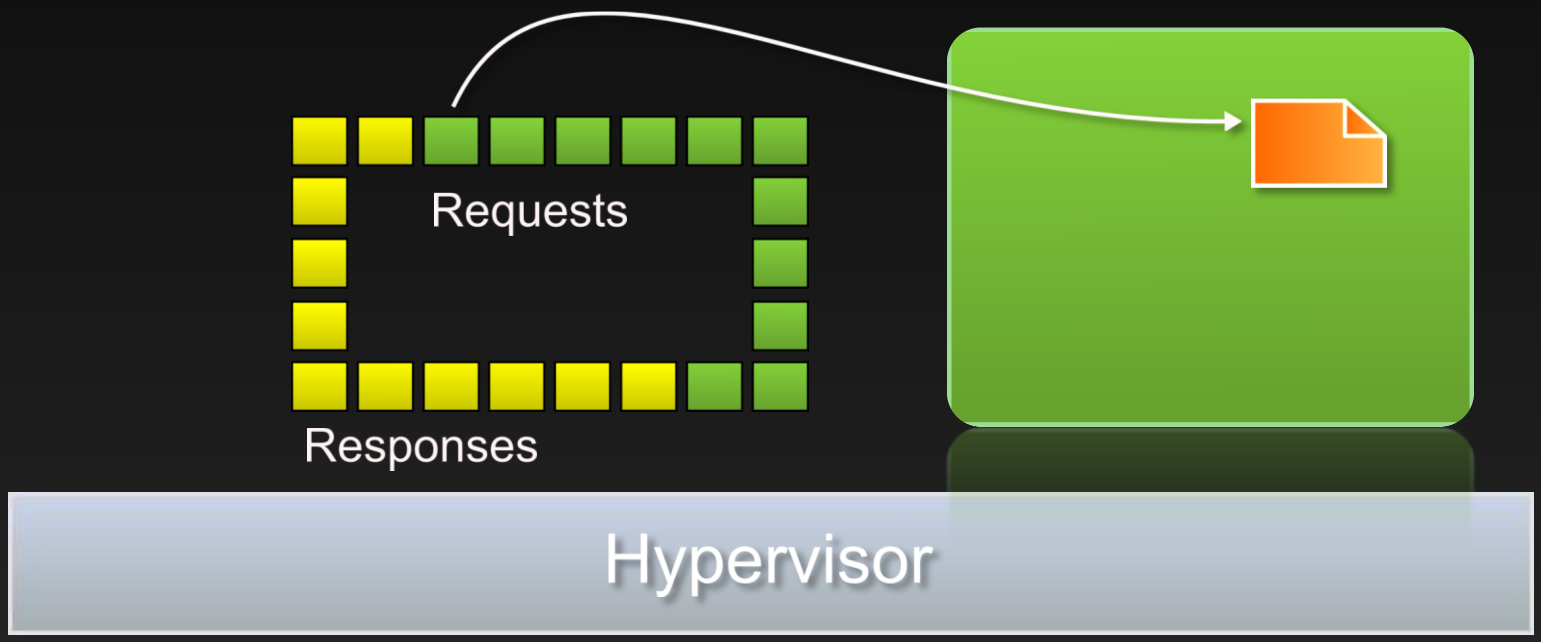
Fido: High-level View

- "c" – connection management
- "m" – memory mapping
- "s" – cross-VM signaling



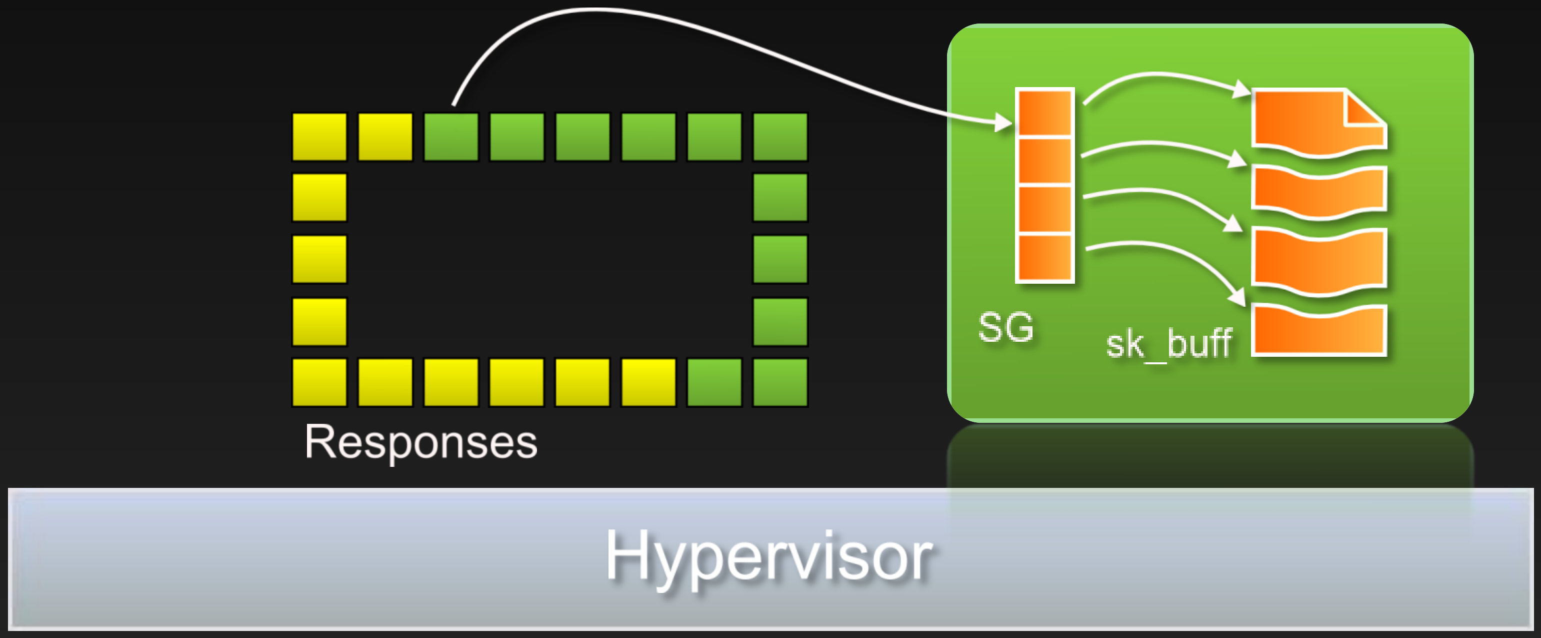
IPC Organization

- Shared memory ring
  - Pointers to data



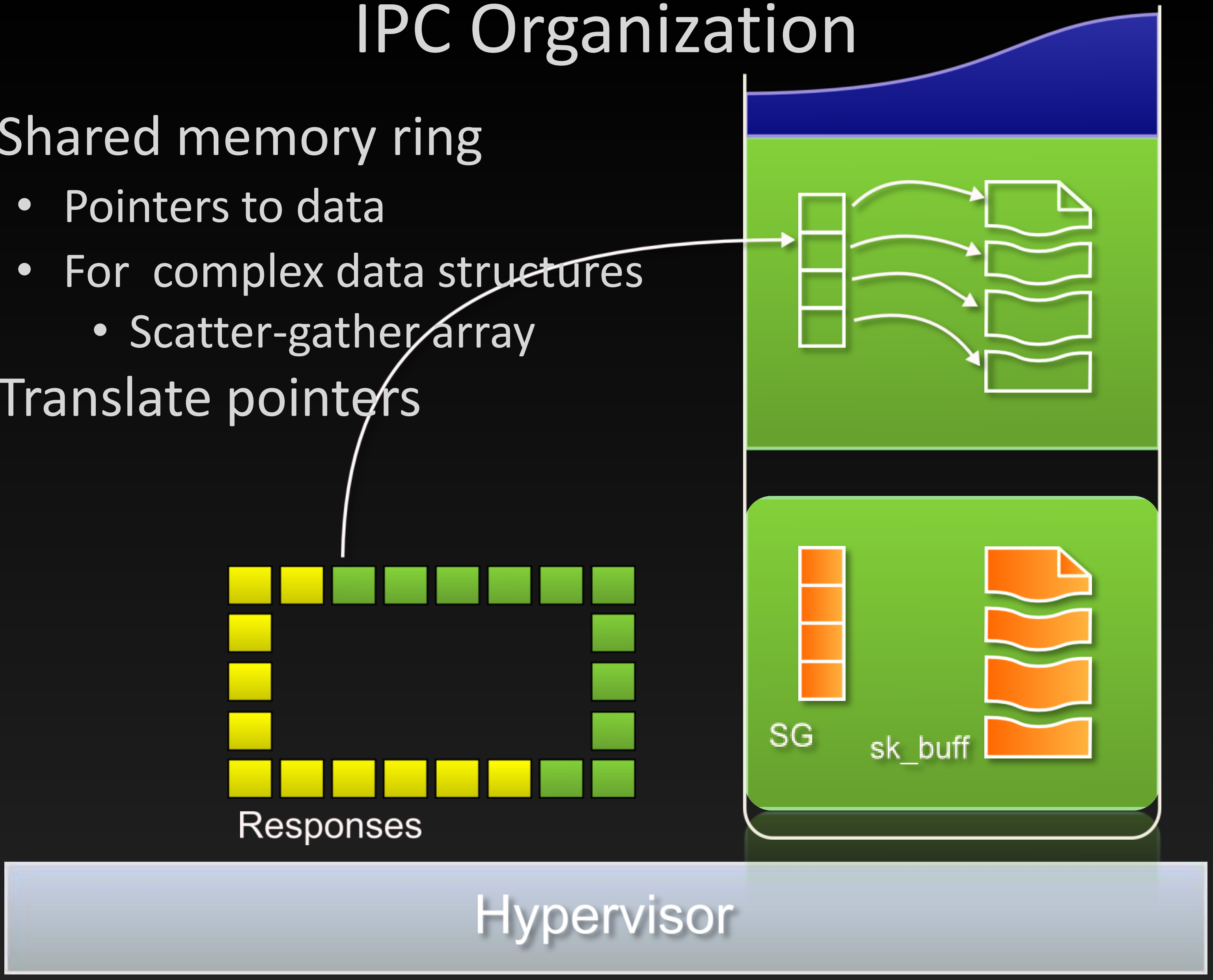
IPC Organization

- Shared memory ring
  - Pointers to data
  - For complex data structures
    - Scatter-gather array



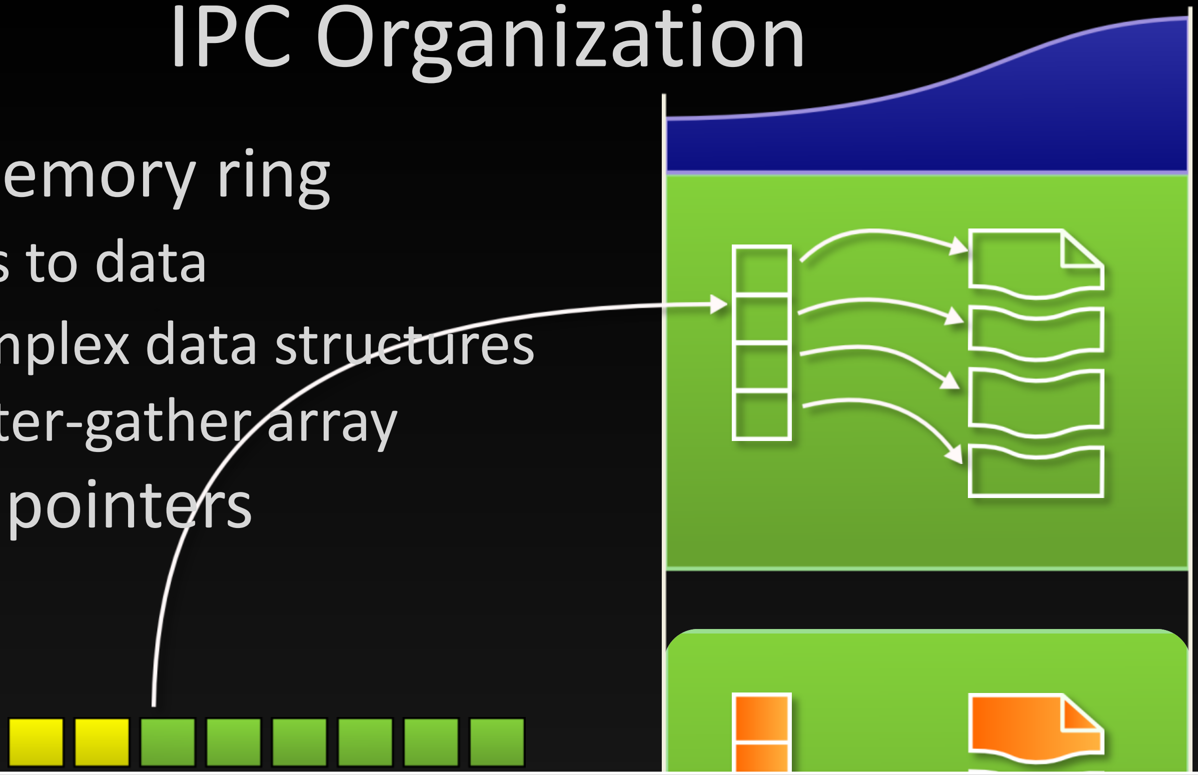
IPC Organization

- Shared memory ring
  - Pointers to data
  - For complex data structures
    - Scatter-gather array
- Translate pointers



IPC Organization

- Shared memory ring
  - Pointers to data
  - For complex data structures
    - Scatter-gather array
- Translate pointers



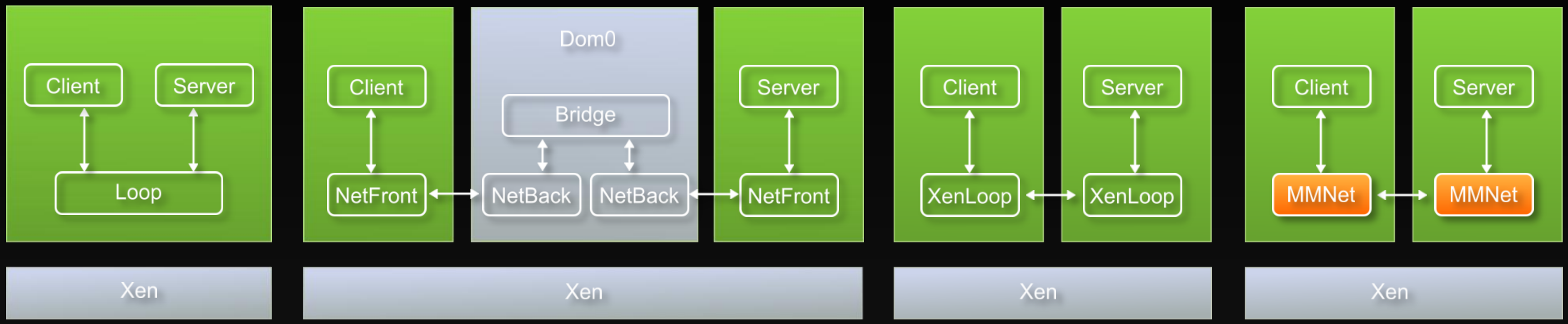
- Signaling:
  - Cross-core interrupts (event channels)
  - Batching and in-ring polling

Fast device-level communication

- MMNet
  - Link-level
  - Standard network device interface
  - Supports full transitive zero-copy
- MMBlk
  - Block-level
  - Standard block device interface
  - Zero-copy on write
  - Incurs one copy on read

Evaluation

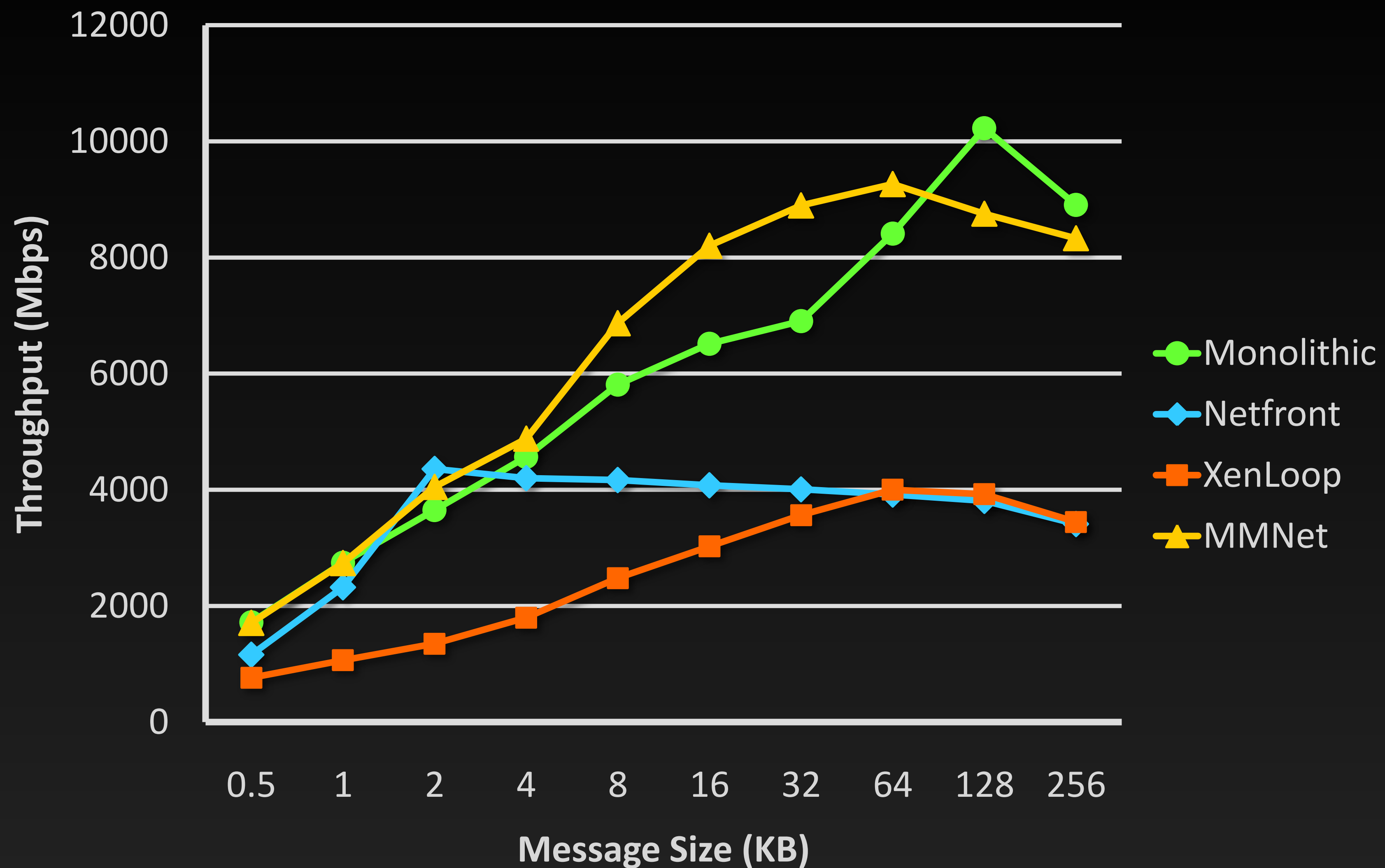
MMNet Evaluation



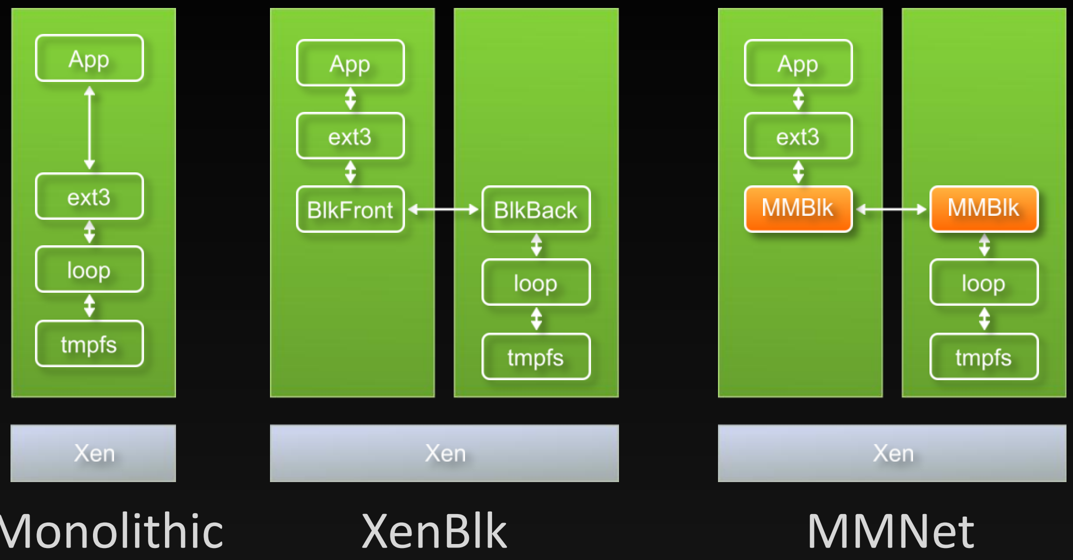
Loop NetFront XenLoop MMNet

- AMD Opteron with 2 2.1GHz 4-core CPUs (8 cores total)
- 16GB RAM
- NVidia 1Gbps NICs
- 64-bit Xen (3.2), 64-bit Linux (2.6.18.8)
- Netperf benchmark (2.4.4)

MMNet: TCP Throughput

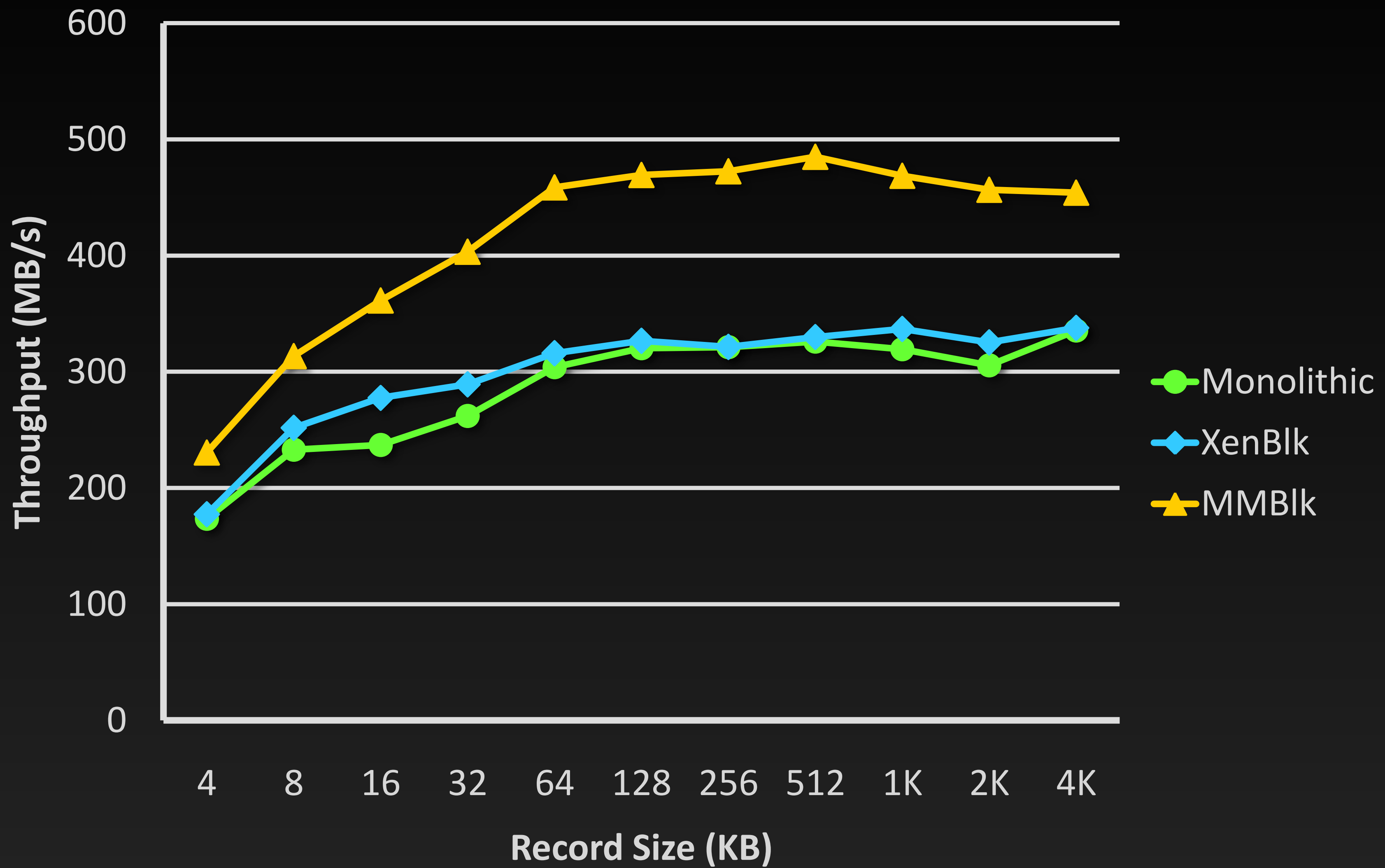


MMBlk Evaluation



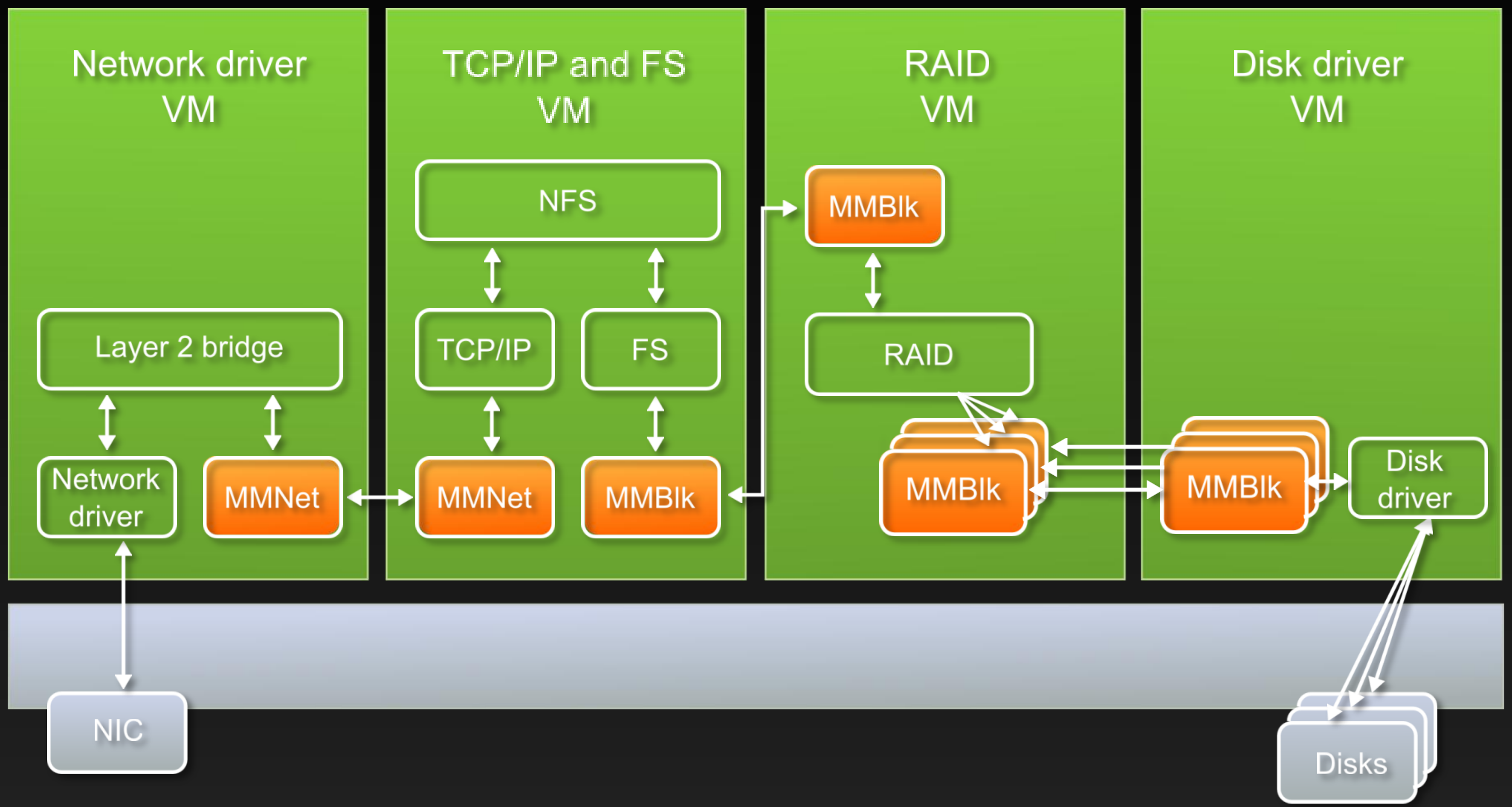
- Same hardware
  - AMD Opteron with 2 2.1GHz 4-core CPUs (8 cores total)
  - 16GB Ram
  - NVidia 1Gbps NICs
- VMs are configured with 4GB and 1GB RAM
- 3 GB in-memory file system (TMPFS)
- IOZone benchmark

MMBlk Sequential Writes

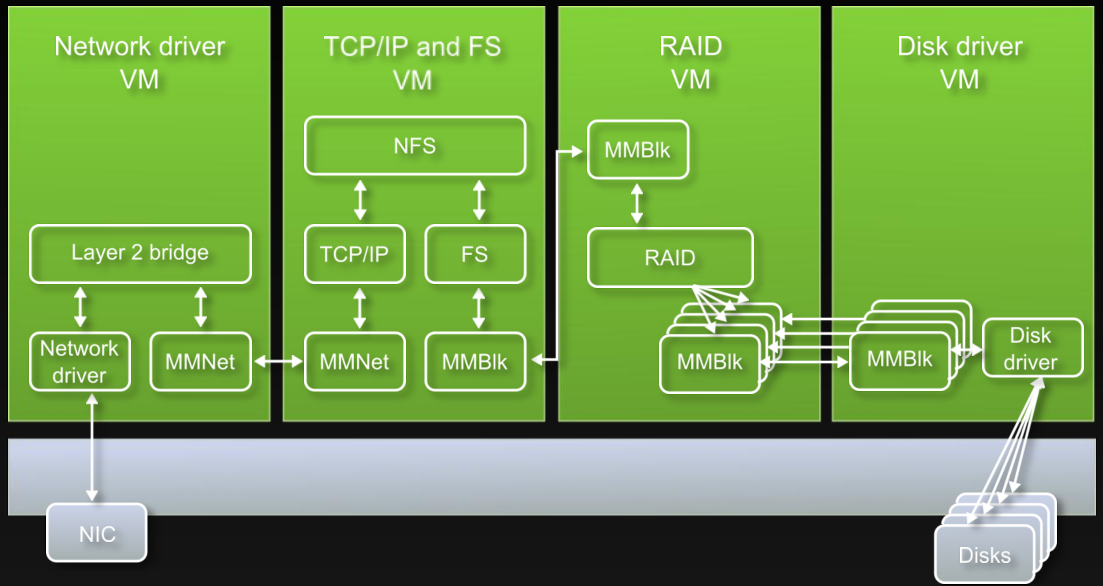


Case Study

Network-attached Storage

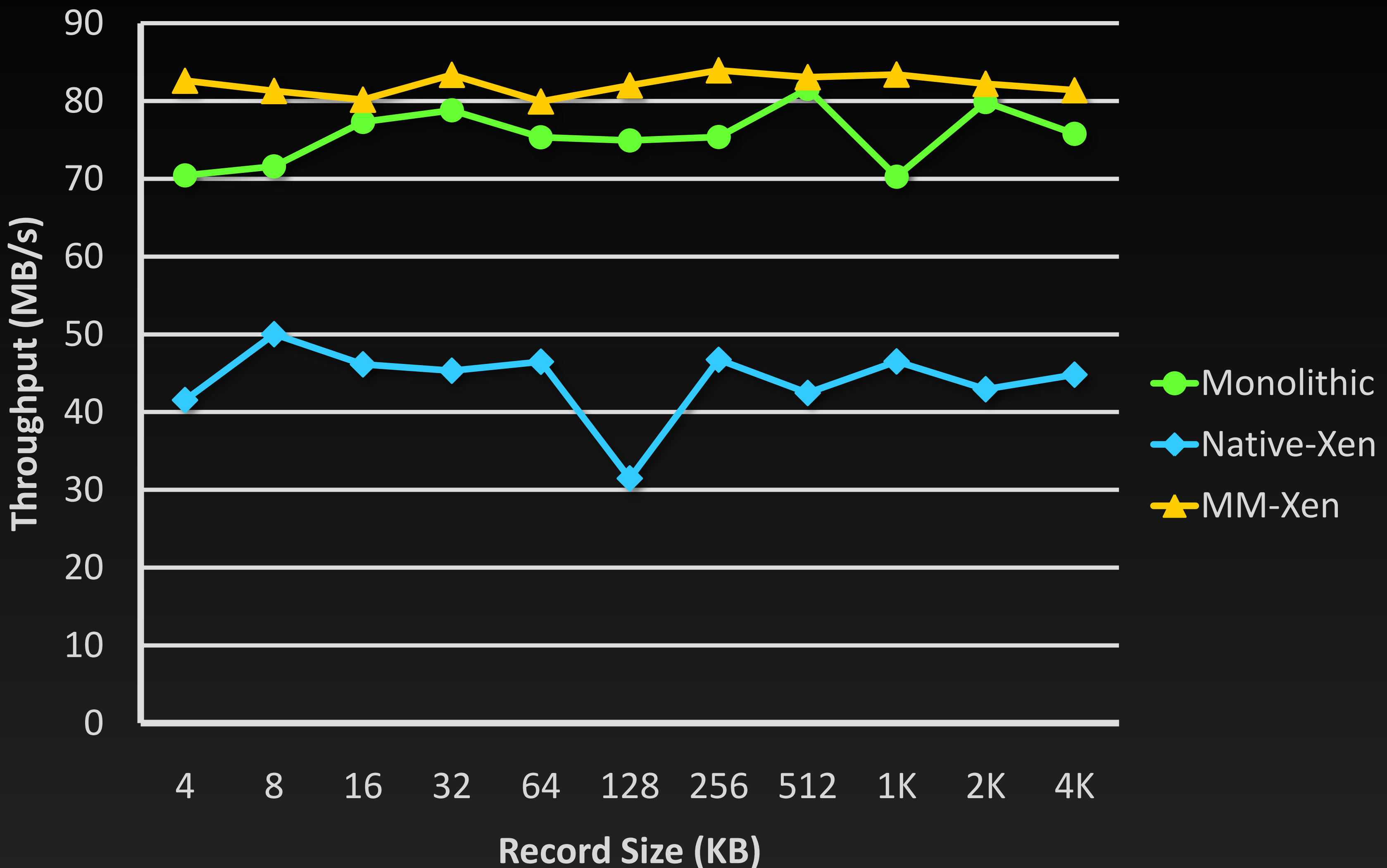


Network-attached Storage

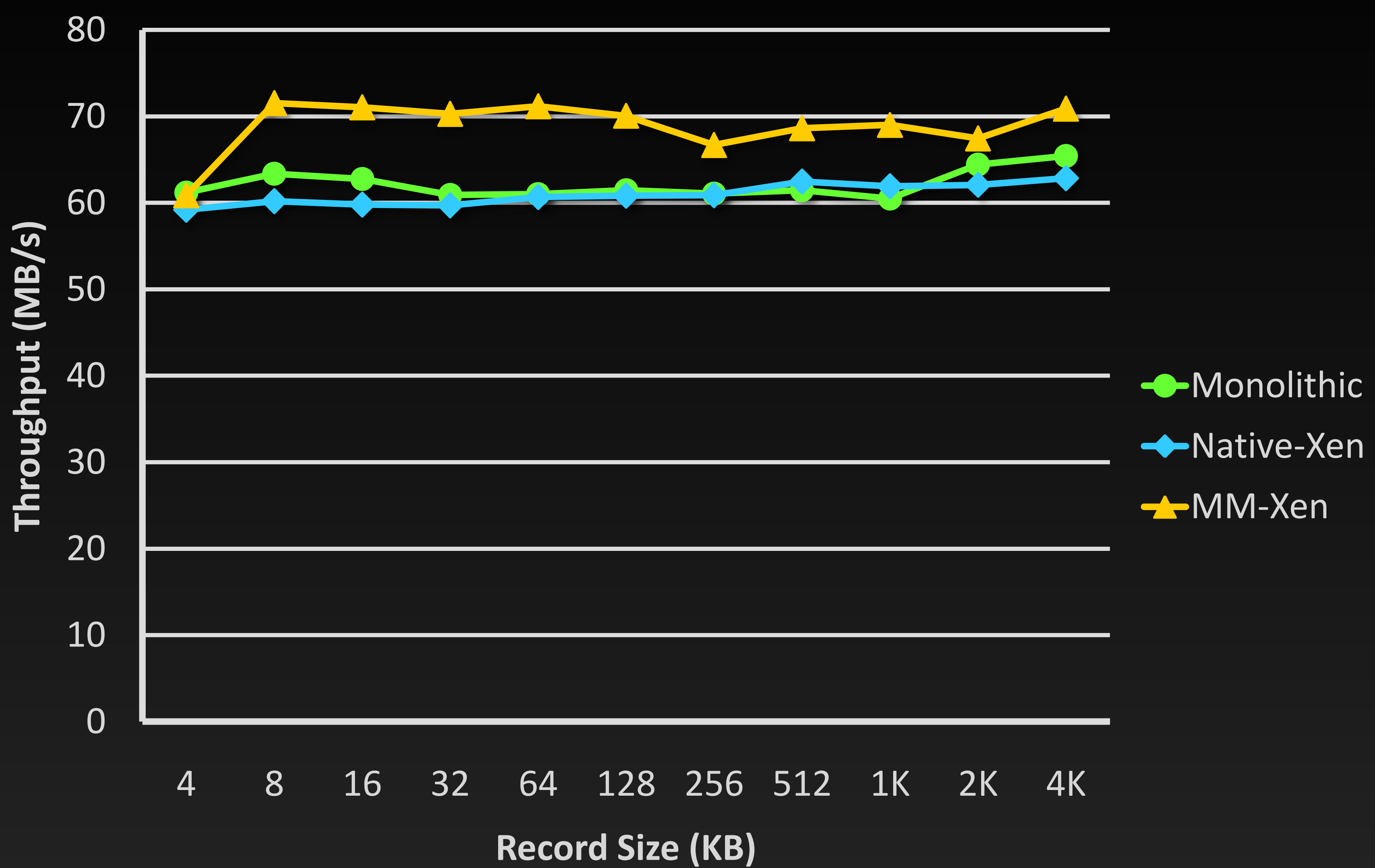


- RAM
  - VMs have 1GB each, except FS VM (4GB)
  - Monolithic system has 7GB RAM
- Disks :
  - RAID5 over 3 64MB/s disks
- Benchmark
  - IOZone reads/writes 8GB file over NFS (async)

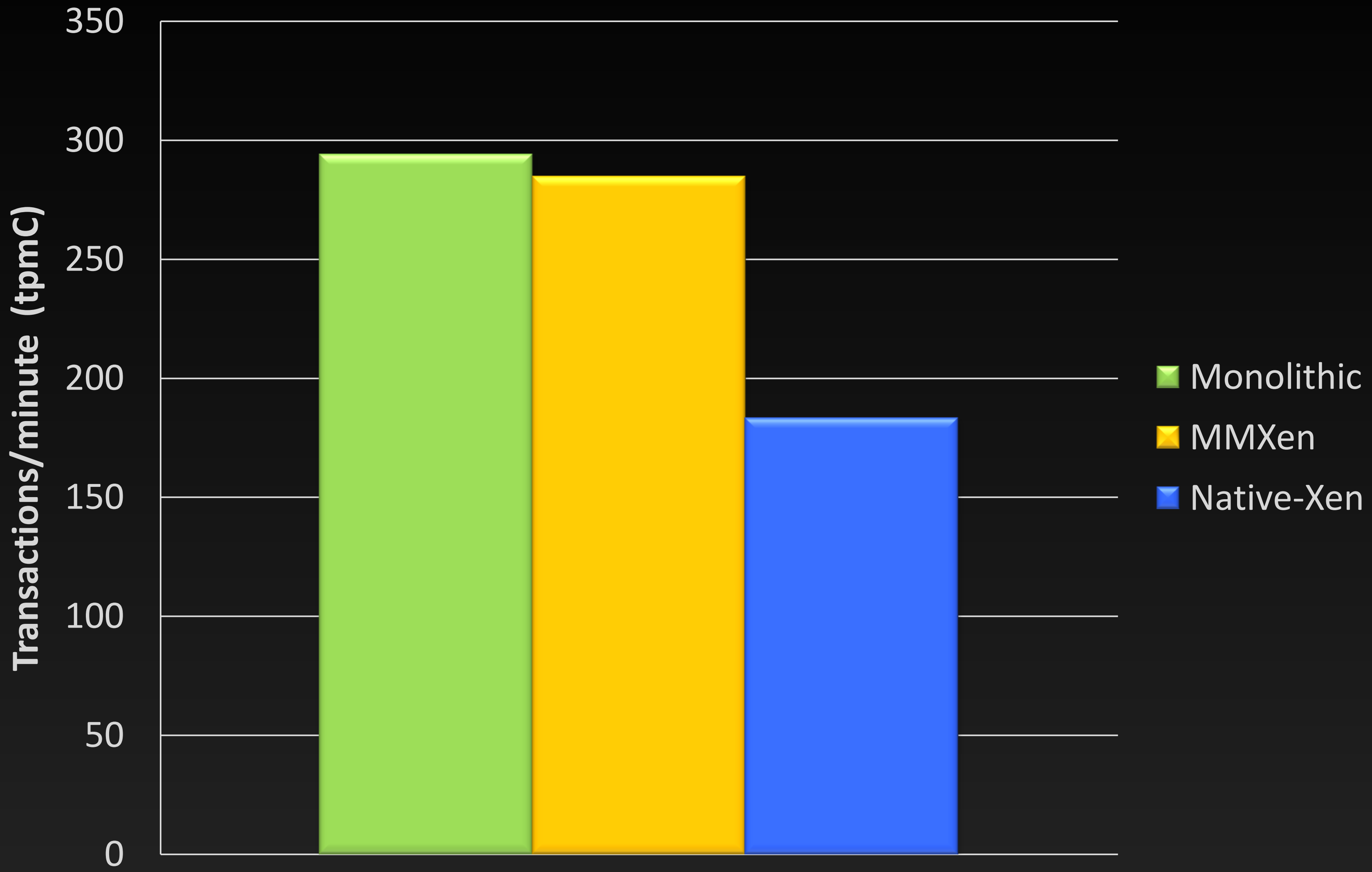
Sequential Writes



Sequential Reads



TPC-C (On-Line Transactional Processing)



Conclusions

- We match monolithic performance
  - “Microkernelization” of traditional systems is possible!
- Fast inter-VM communication
  - The search for VM communication mechanisms is not over
- Important aspects of design
  - Trust model
    - VM as a library (for example, FSVA)
  - End-to-end zero copy
    - Pseudo Global Virtual Address Space
- There are still problems to solve
  - Full end-to-end zero copy
  - Cross-VM memory management
  - Full utilization of pipelined parallelism

Thank you.

aburtsev@flux.utah.edu

Backup Slides

Related Work

- Traditional microkernels [L4, Eros, CoyotOS]
  - Synchronous (effectively thread migration)
  - Optimized for single-CPU, fast context switch, small messages (often in registers), efficient marshaling (IDL)
- Buffer management [Fbufs, IOLite, Beltway Buffers]
  - Shared buffer is a unit of protection
  - Fast-forward – fast cache-to-cache data transfer
- VMs [Xen split drivers, XWay, XenSocket, XenLoop]
  - Page flipping, later buffer sharing
  - IVC, VMCI
- Language-based protection [Singularity]
  - Shared heap, zero-copy (only pointer transfer)
- Hardware acceleration [Solarflare]
- Multi-core OSes [Barrelfish, Corey, FOS]