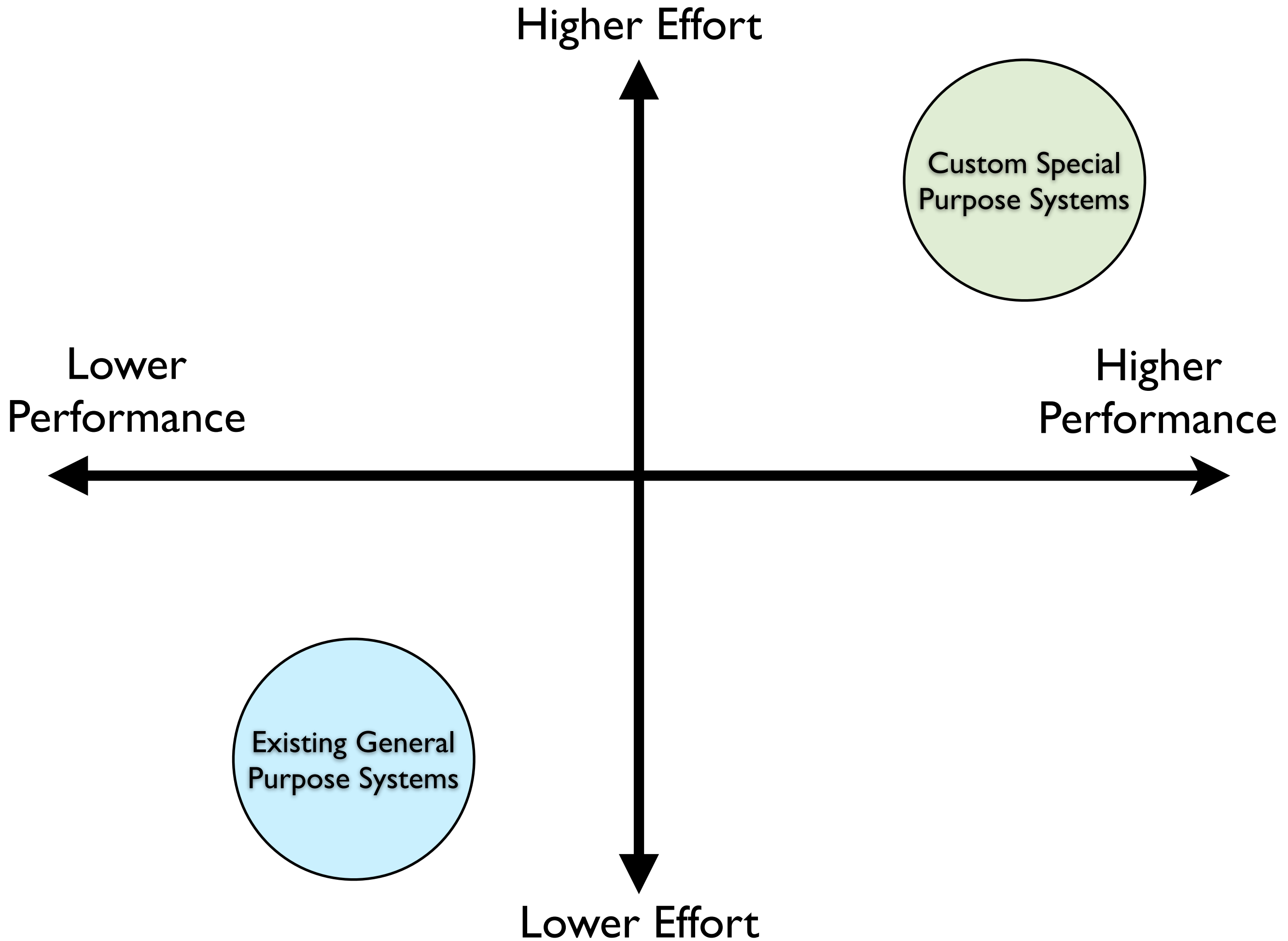
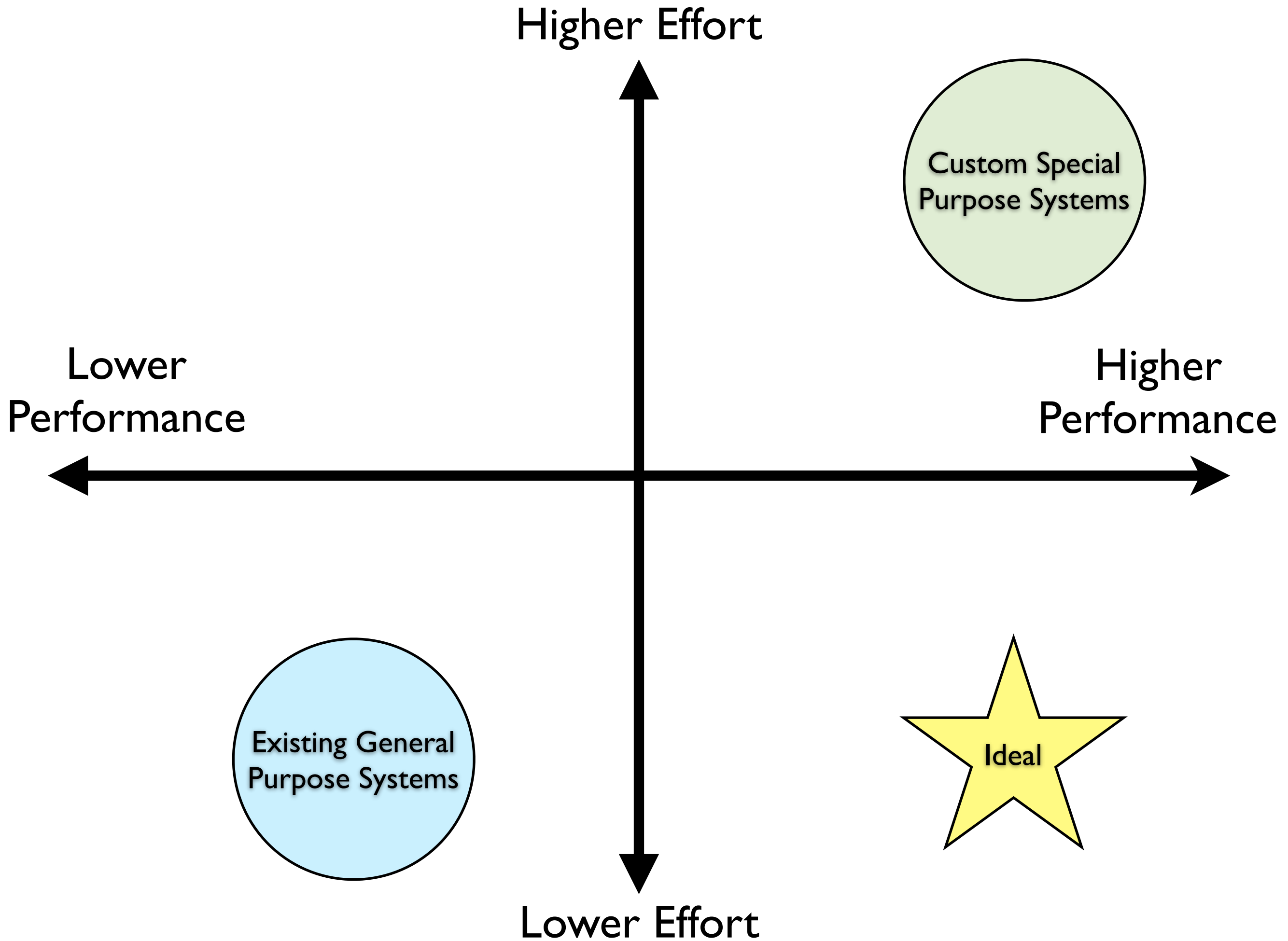




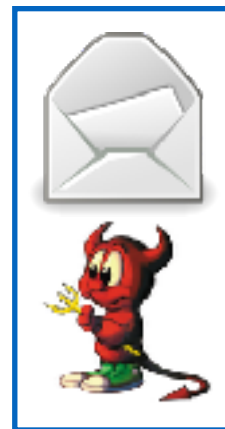
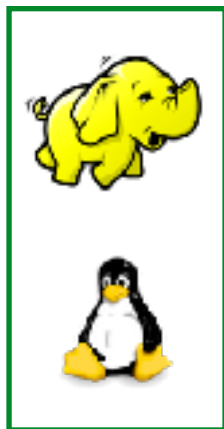
# EbbRT: A Framework for Building Per- Application Library Operating Systems

**Dan Schatzberg**, James Cadden, Han Dong, Orran Krieger, Jonathan Appavoo  
Boston University





Cloud Computing allows users to rent entire virtual machines to run their own OS and applications



Hypervisor

Hardware

Hypervisor

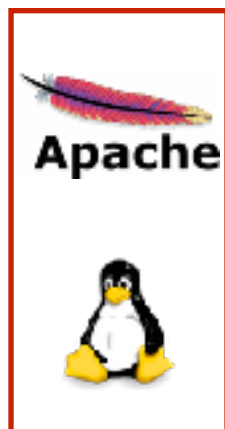
Hardware





Cloud Computing allows users to rent entire virtual machines to run their own OS and applications

The OS no longer needs to provide protection and isolation



Hypervisor

Hardware

Hypervisor

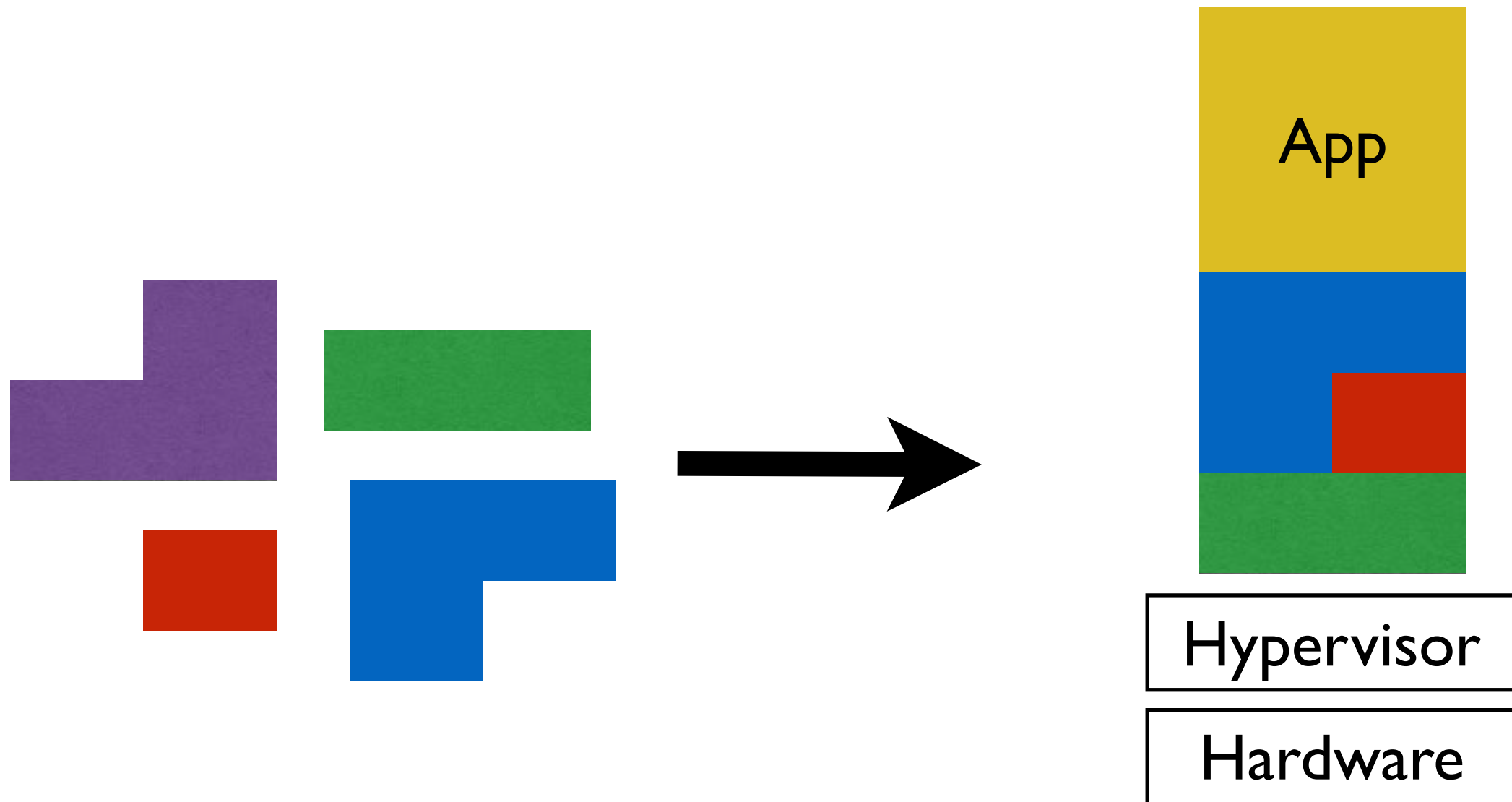
Hardware



# Elastic Building Block Runtime (EbbRT)

A framework for constructing per-application Library Operating Systems in order to achieve high performance with low effort.

Construct customized environments for individual applications with reusable components.



# Implementation Details

- Build bootable (by grub, kexec, qemu, etc.) ELF images using a modified GCC 5.3 toolchain
- Most code written in C++14 (~15kloc)
- Library of core system components
- Includes libc, libstdc++, boost libraries, and more



# Elastic Building Block Runtime (EbbRT) Architecture

# Elastic Building Block Runtime (EbbRT) Architecture

- I. Low-level, event-driven execution environment
  - Allow applications to be written more directly to hardware interfaces for high performance

# Elastic Building Block Runtime (EbbRT) Architecture

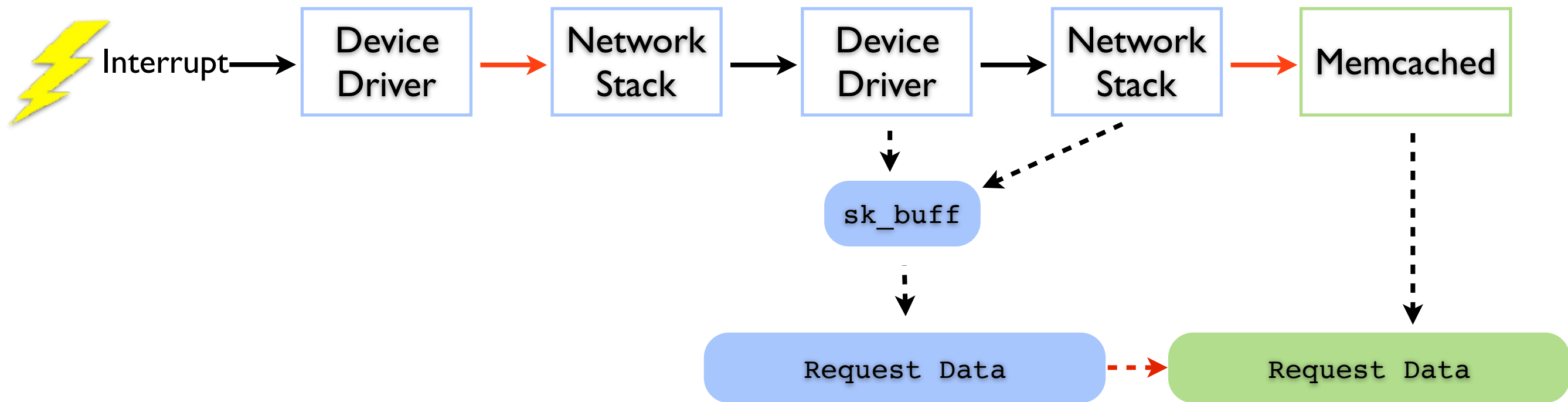
1. Low-level, event-driven execution environment
  - Allow applications to be written more directly to hardware interfaces for high performance
2. Heterogeneous distributed system architecture
  - Offload functionality to general purpose operating systems for compatibility

# Elastic Building Block Runtime (EbbRT) Architecture

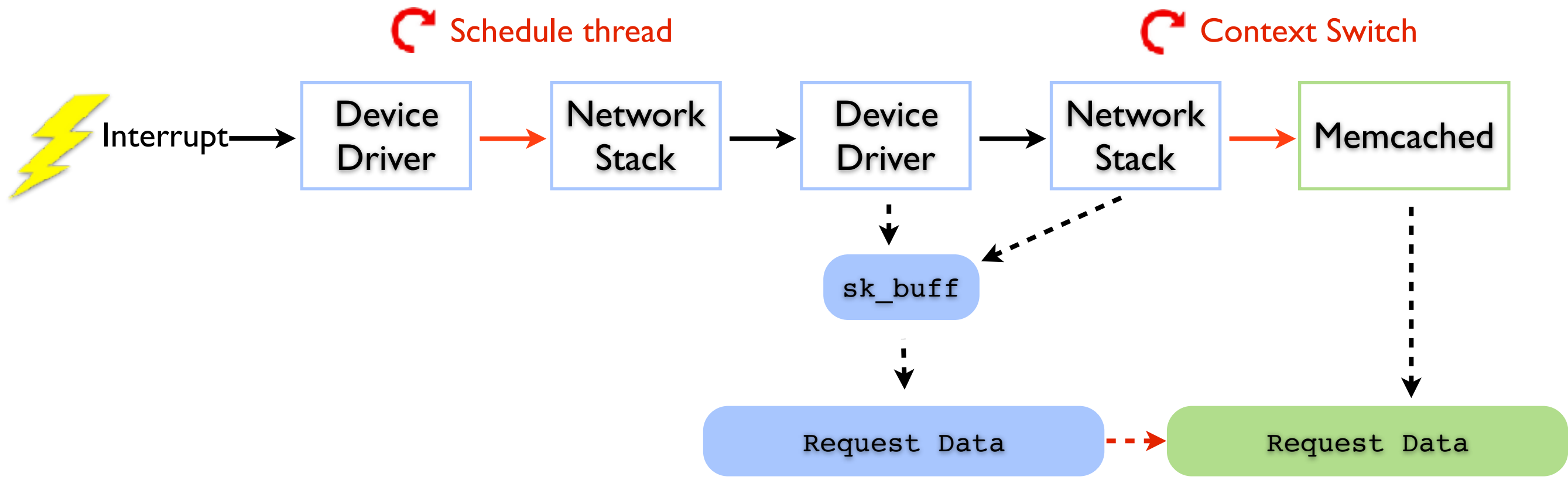
1. Low-level, event-driven execution environment
  - Allow applications to be written more directly to hardware interfaces for high performance
2. Heterogeneous distributed system architecture
  - Offload functionality to general purpose operating systems for compatibility
3. Elastic Building Blocks
  - Encapsulate system and application functionality in order to promote customization and reuse



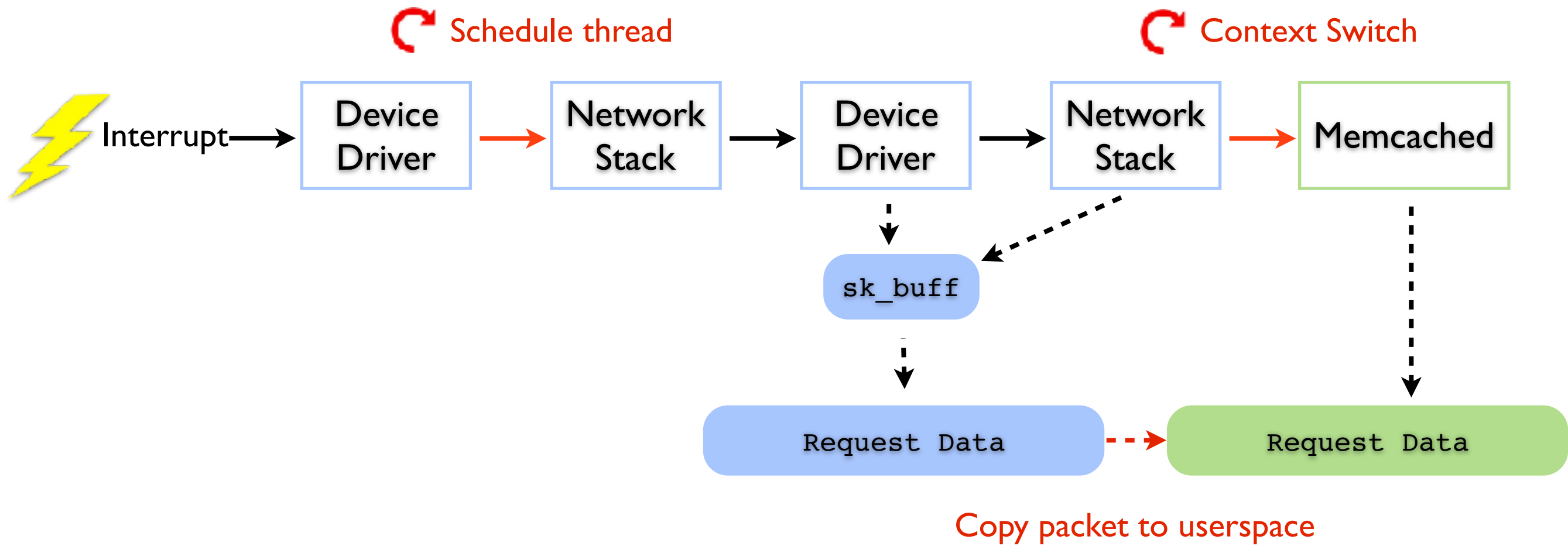
# Linux Memcached



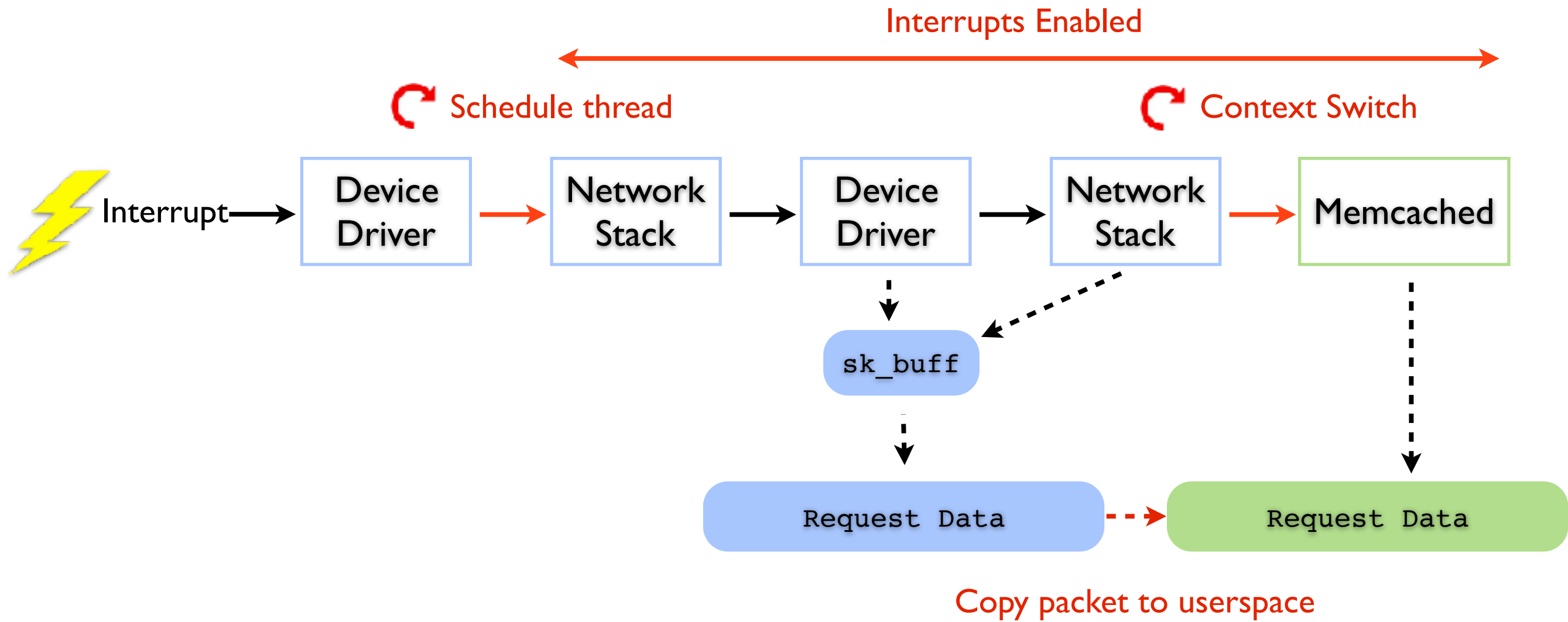
# Linux Memcached



# Linux Memcached

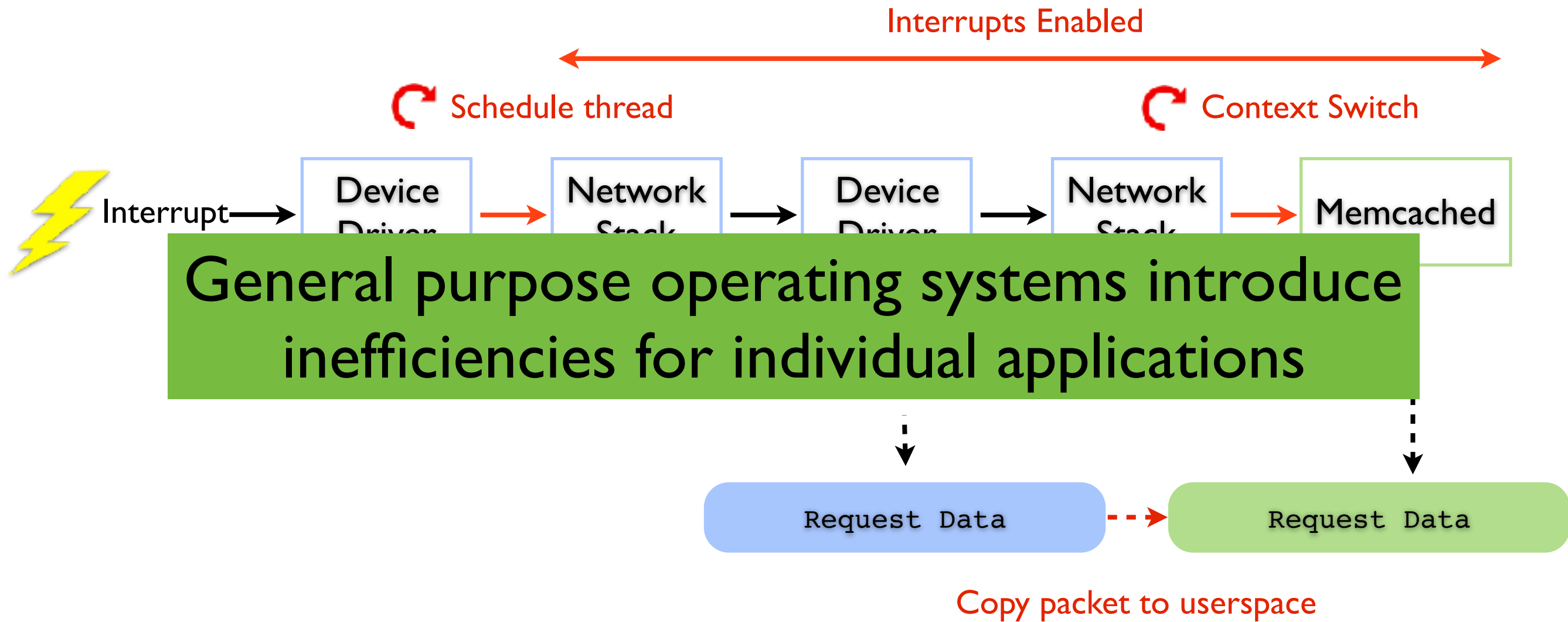


# Linux Memcached



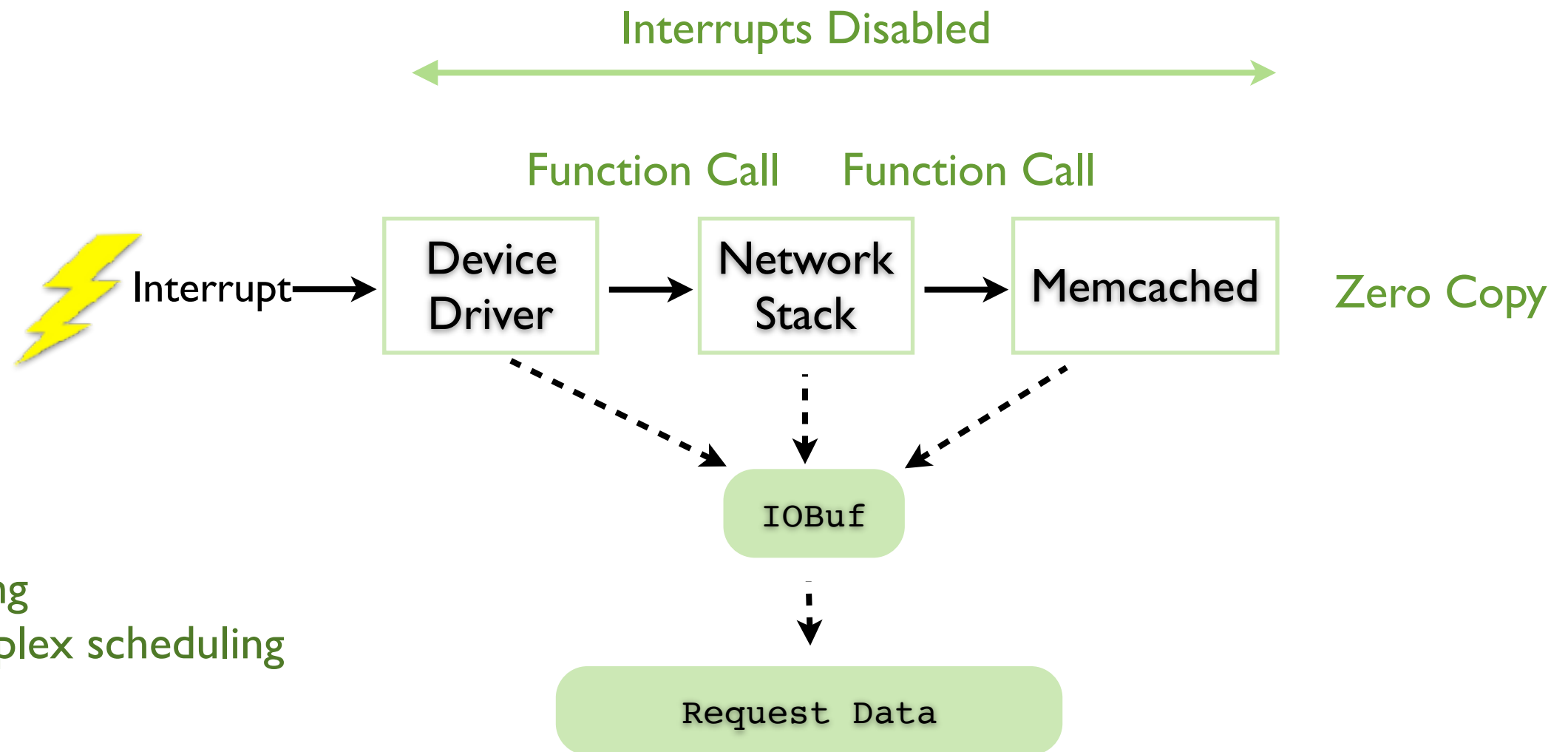


# Linux Memcached

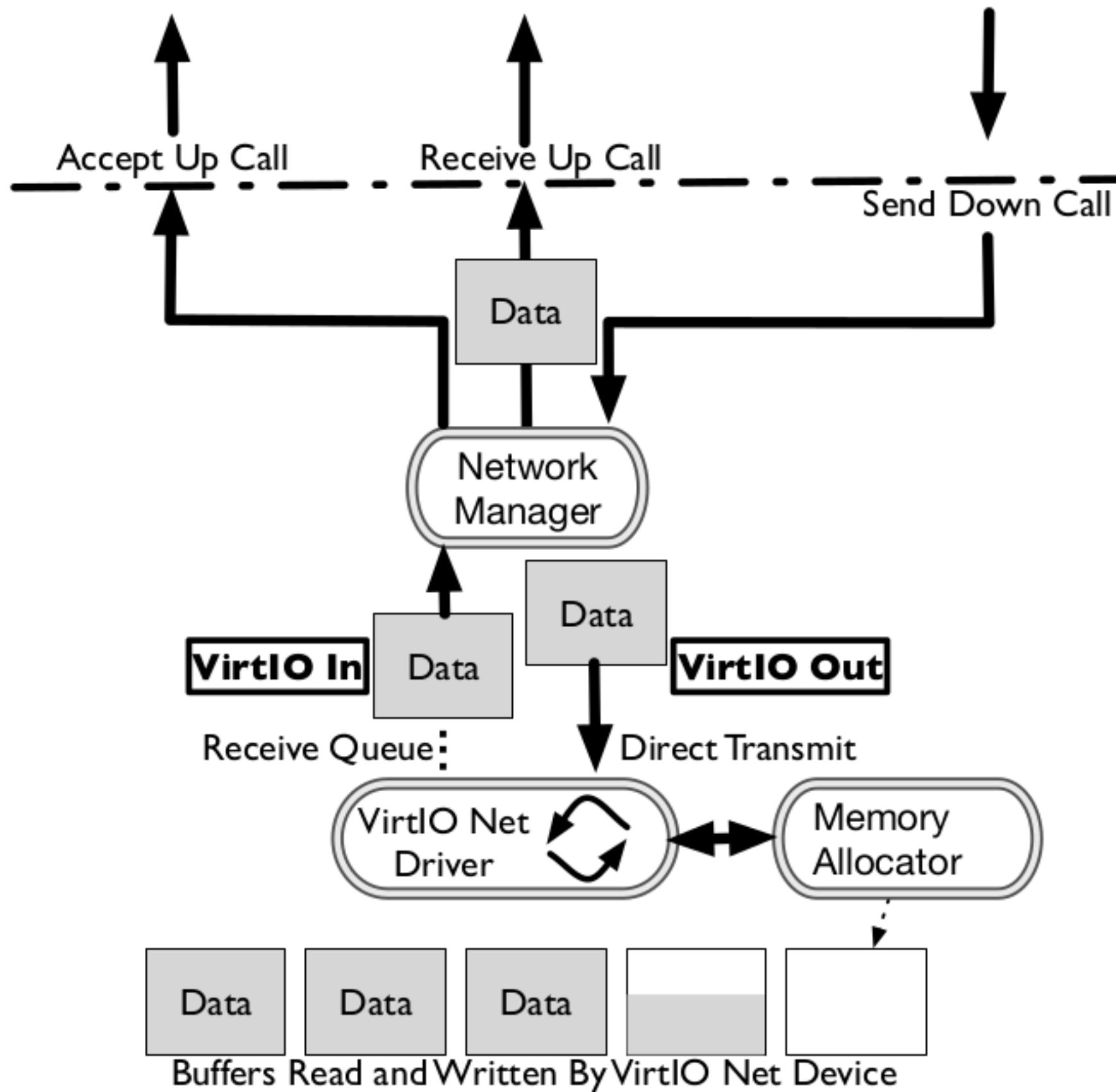


- EbbRT applications execute at highest privilege level
- Identity mapped memory with large pages
- Execution as non-preemptive events generated by hardware or software

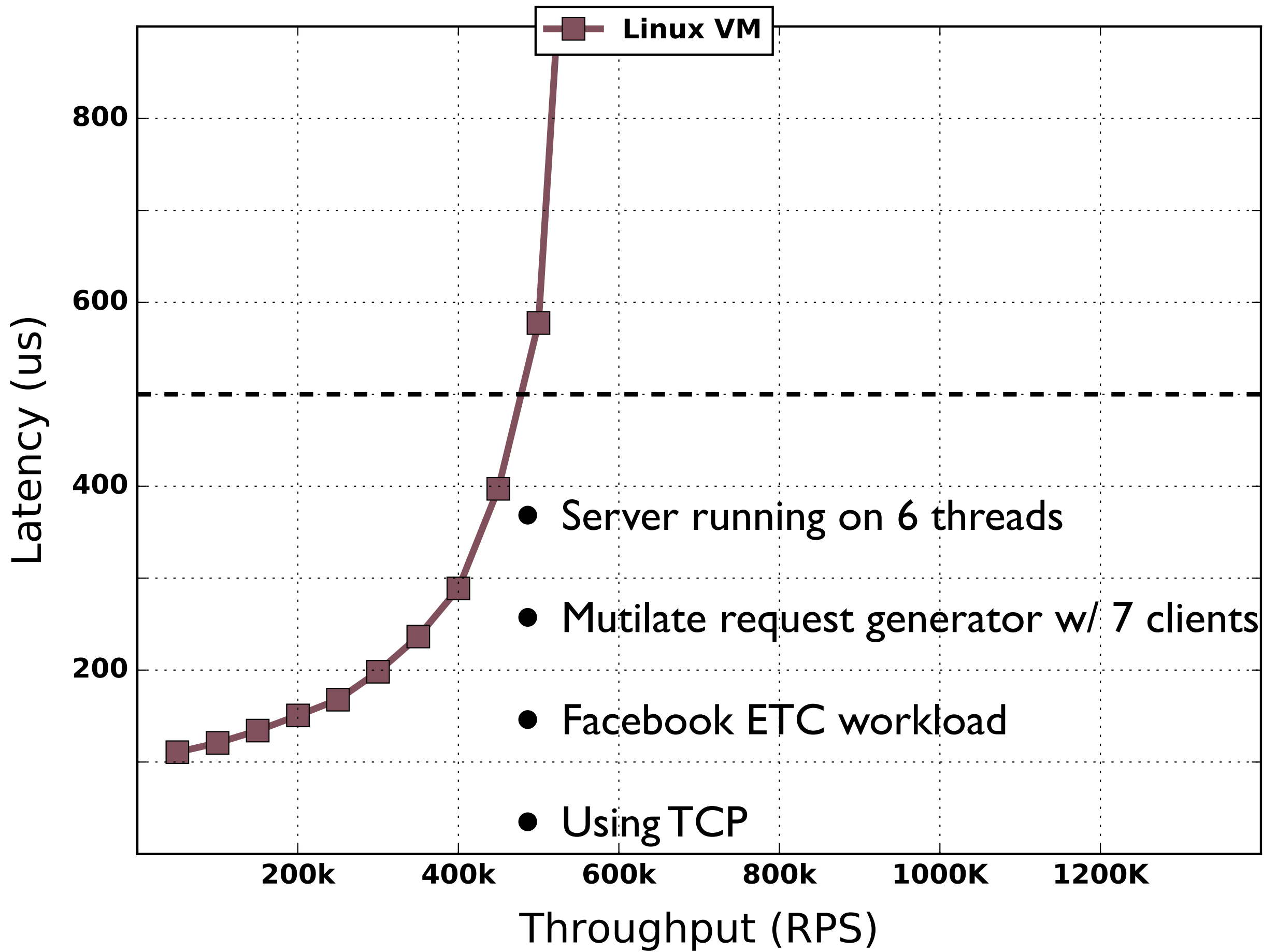
# EbbRT Memcached

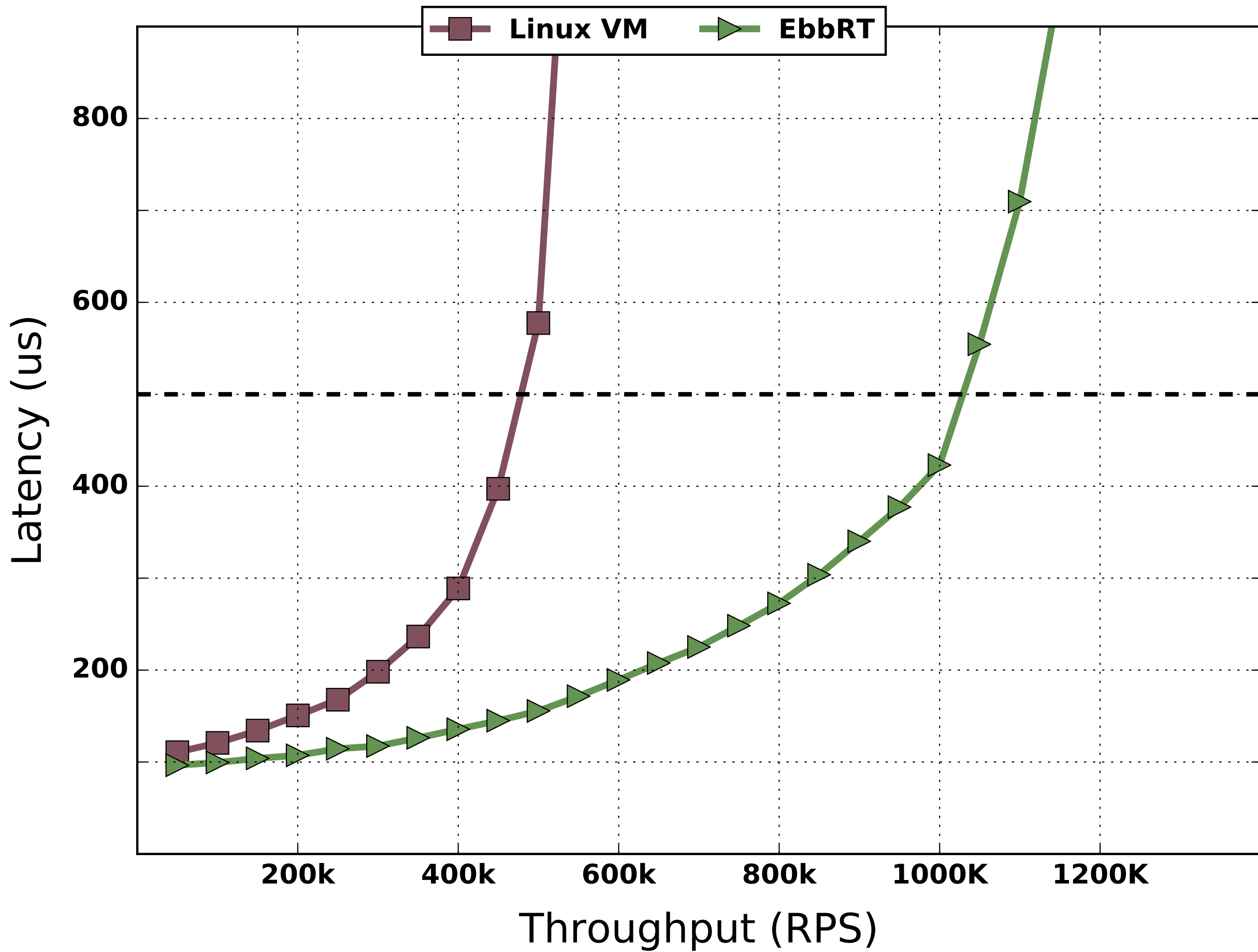


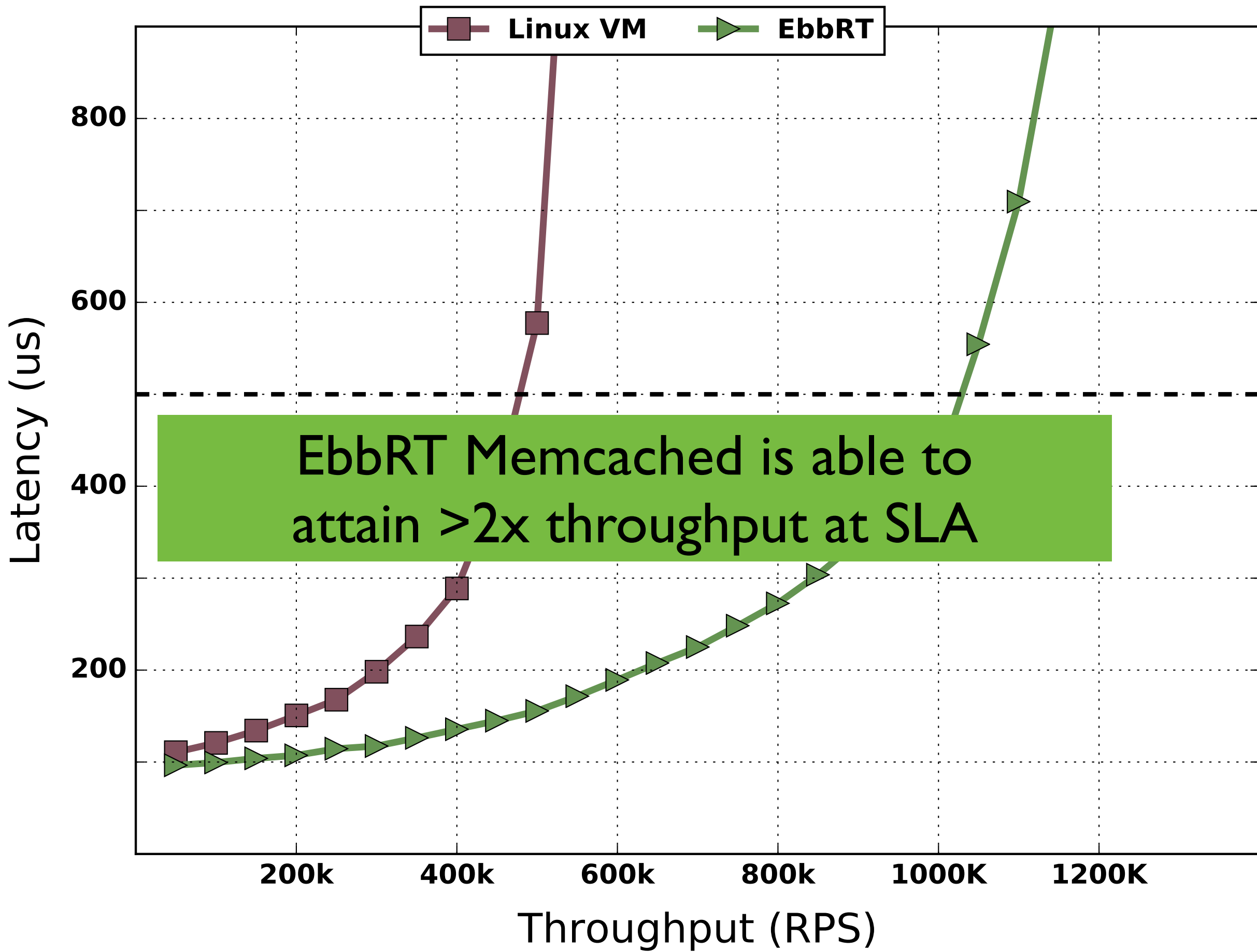
- No paging
- No complex scheduling

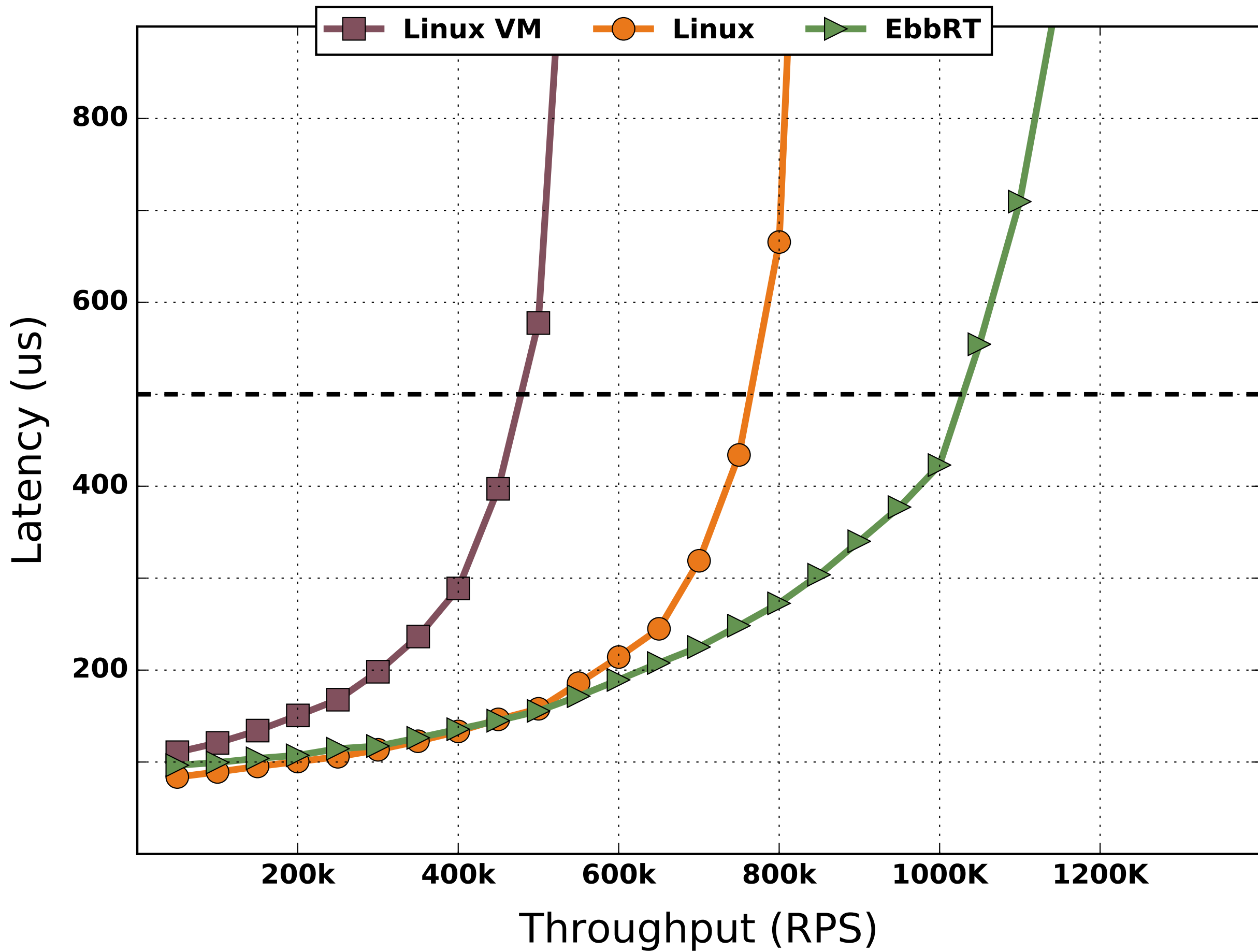














# Other Benefits

See paper for details

- Blocking semantics
- Read-Copy-Update (RCU)
- Core-local data structures

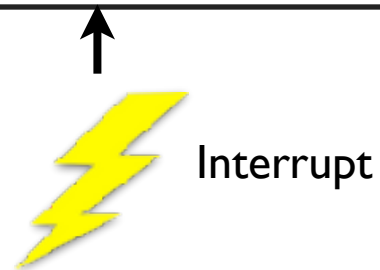
# Elastic Building Block Runtime (EbbRT) Architecture

1. Low-level execution environment
2. Heterogeneous distributed system architecture
3. Distributed object model



Network  
Processing

NIC Driver



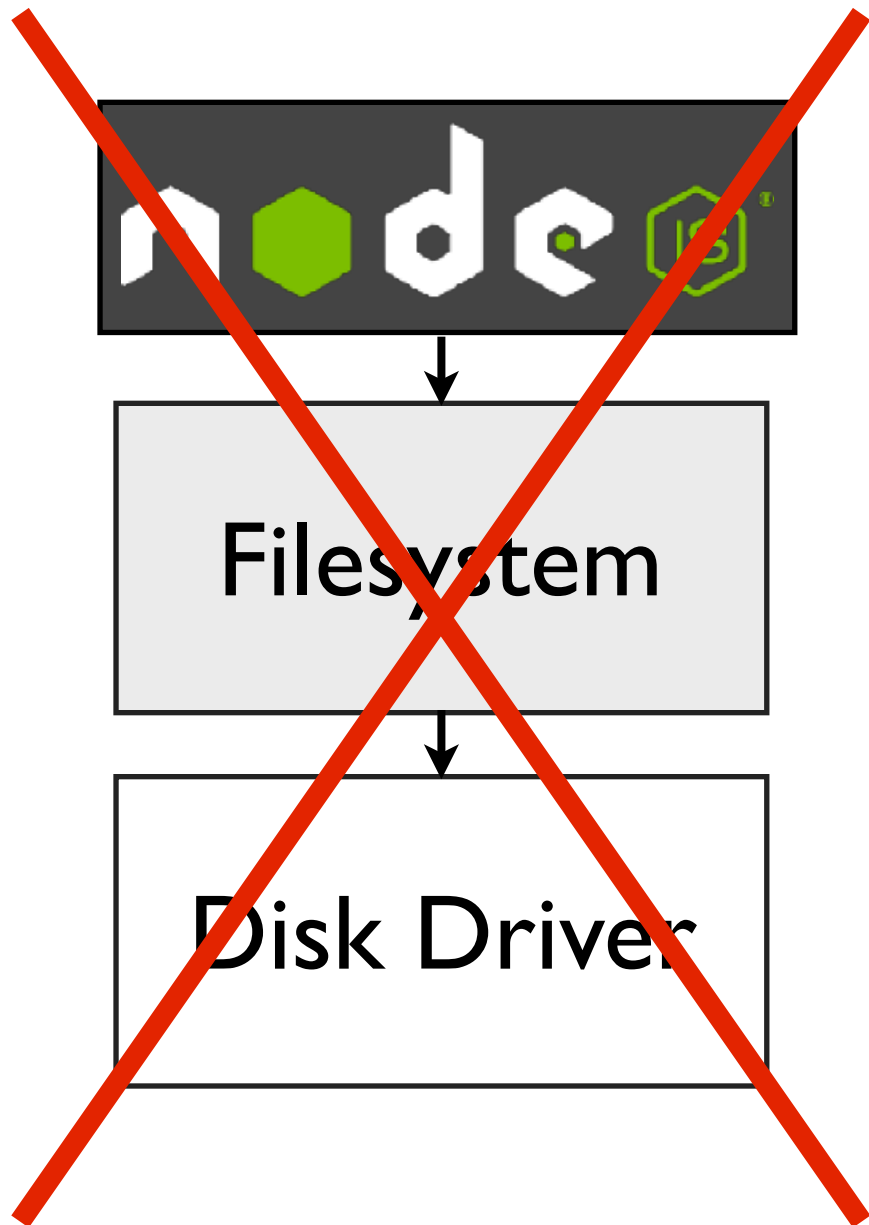
Reuse event-driven  
network stack to provide  
networking to Node.js



Filesystem

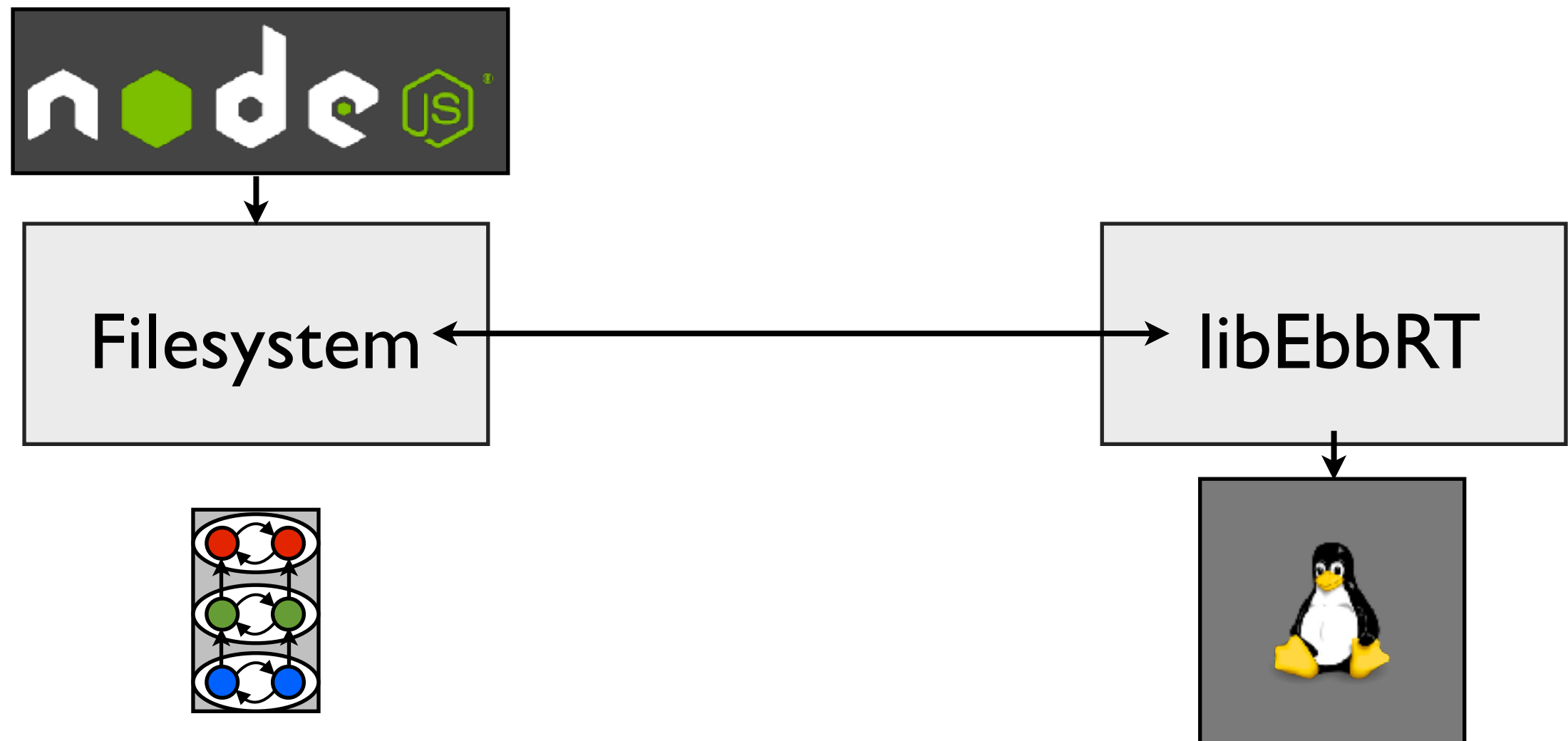


Disk Driver



Avoid duplicating  
functionality for  
compatibility

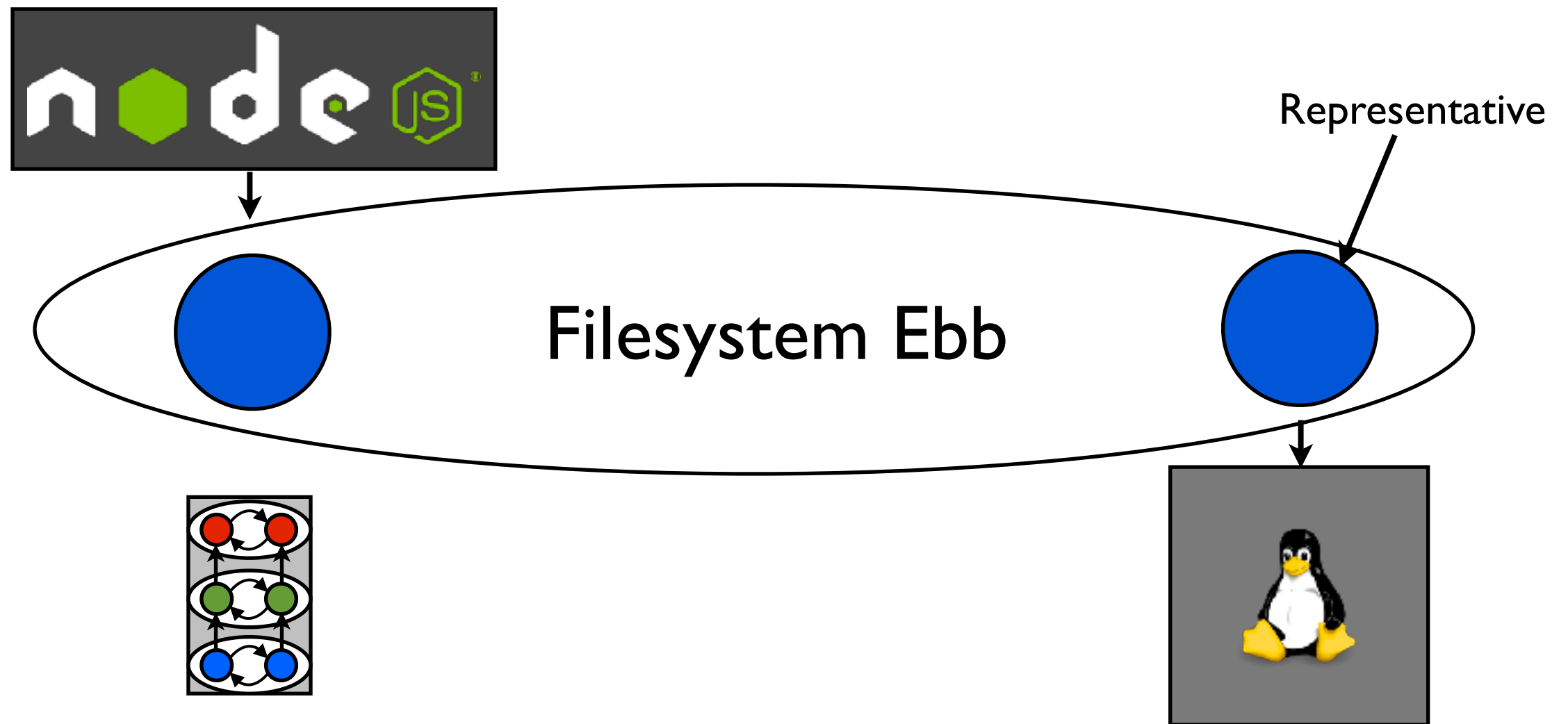
# Offload functionality for rapid development





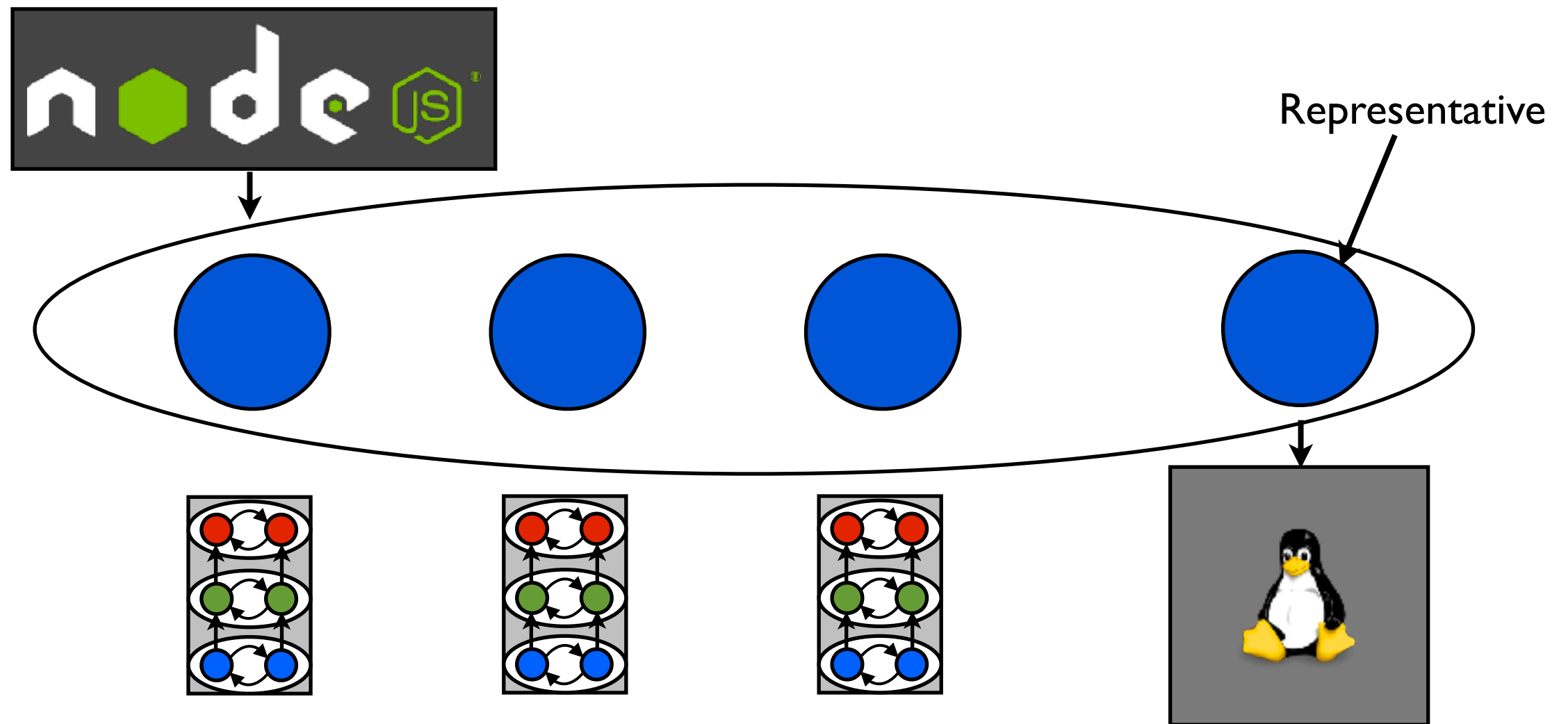
# Elastic Building Blocks

System-wide distributed objects

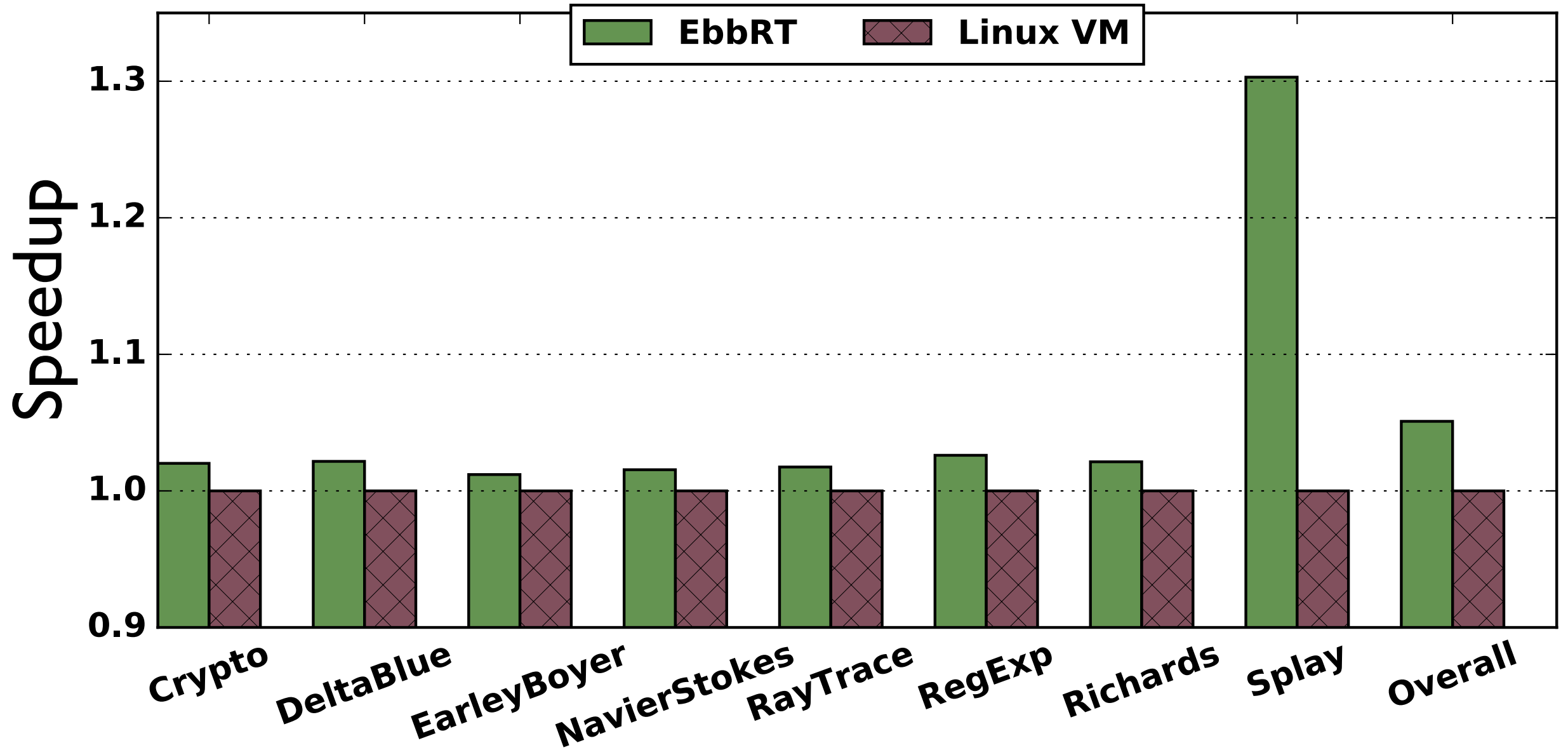


# Elastic Building Blocks

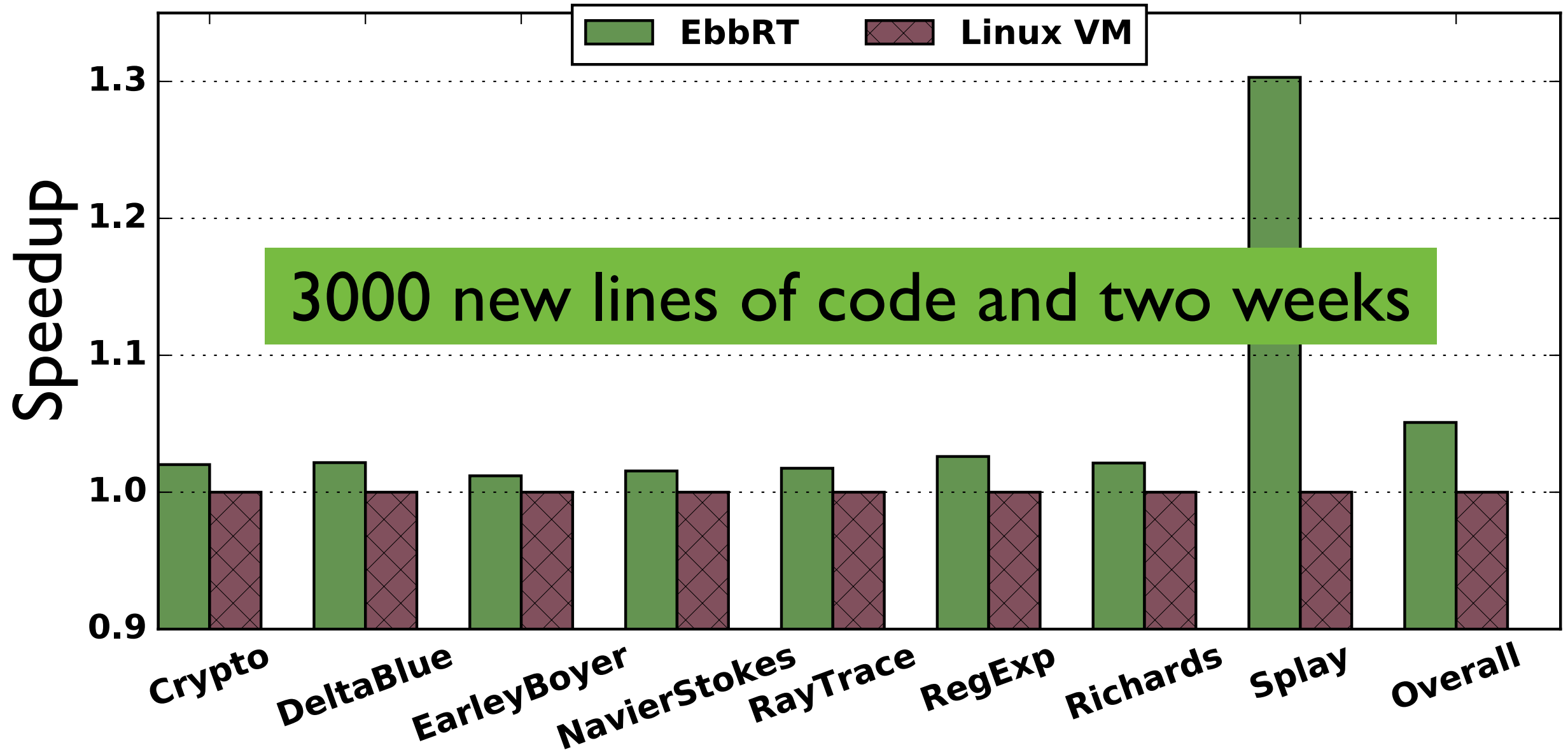
System-wide distributed objects



# V8 Javascript Benchmark Suite



# V8 Javascript Benchmark Suite



# Other Benefits

See paper for details

- Distributing EbbRT as a toolchain
- Providing language and library compatibility
- Language-level primitives (Lambdas, Futures, IOBufs)

# Elastic Building Block Runtime (EbbRT)

- Low-level execution environment enables applications to get much closer to the hardware
- Heterogeneous distributed system allows for functionality offload for incremental development
- Distributed object model encapsulates functionality, enabling customization and reuse



# Questions?

<https://github.com/sesa/ebbrt>

**Dan Schatzberg**, James Cadden, Han Dong, Orran Krieger, Jonathan Appavoo

[dschatz@bu.edu](mailto:dschatz@bu.edu)

[jmcadden@bu.edu](mailto:jmcadden@bu.edu)

[handong@bu.edu](mailto:handong@bu.edu)

[okrieg@bu.edu](mailto:okrieg@bu.edu)

[jappavoo@bu.edu](mailto:jappavoo@bu.edu)