# Mega-VM: A Memory Enhancing Framework for Datacenters

Rashid Tahir*, Gohar Irfan†, Bilal Bakht†, Hashim Sharif*, Fareed Zaffar† and Matthew Caesar*

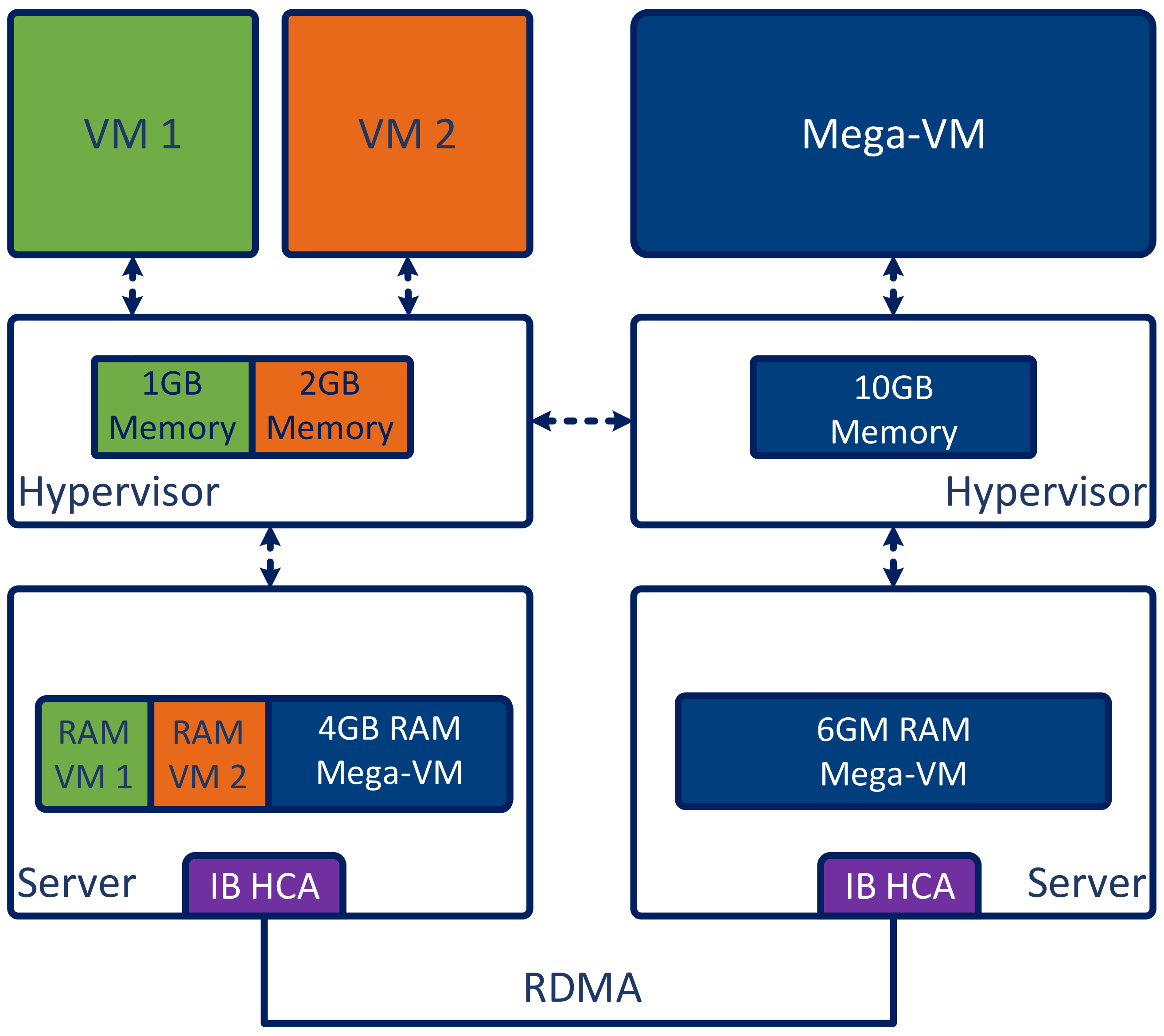
University of Illinois at Urbana Champaign* Lahore University of Management Sciences†

## The Problem: Clouds, Workloads and Memory Requirements

- Datacenters must support an increasingly diverse set of memory-intensive workloads.
- Scale-out solutions either necessitate a port of the problem or require OS modifications. They suffer from memory fragmentation and waste costly network bandwidth.
- Scale-up solutions are costly and have limited upgradeability. They cannot handle job sizes beyond a “cross-over” point.
- Additionally, some workloads are better suited to scale-out whereas others have higher performance on scale-up.

## Mega-VM: Pooled Memory with Central Abstractions

- Peering hypervisors provide VMs with centralized memory abstractions on top of large pools of physically distributed memory.
- “Logically scaled-up” model implemented on top of cheap off-the-shelf scale-out server.
- Requires no modification to any other layer in the software stack.
- Removes resource fragmentation by allocating small wasted resources to VMs hosted on other servers.
- Relies on low-latency interconnects supporting RDMA and Nested Paging.

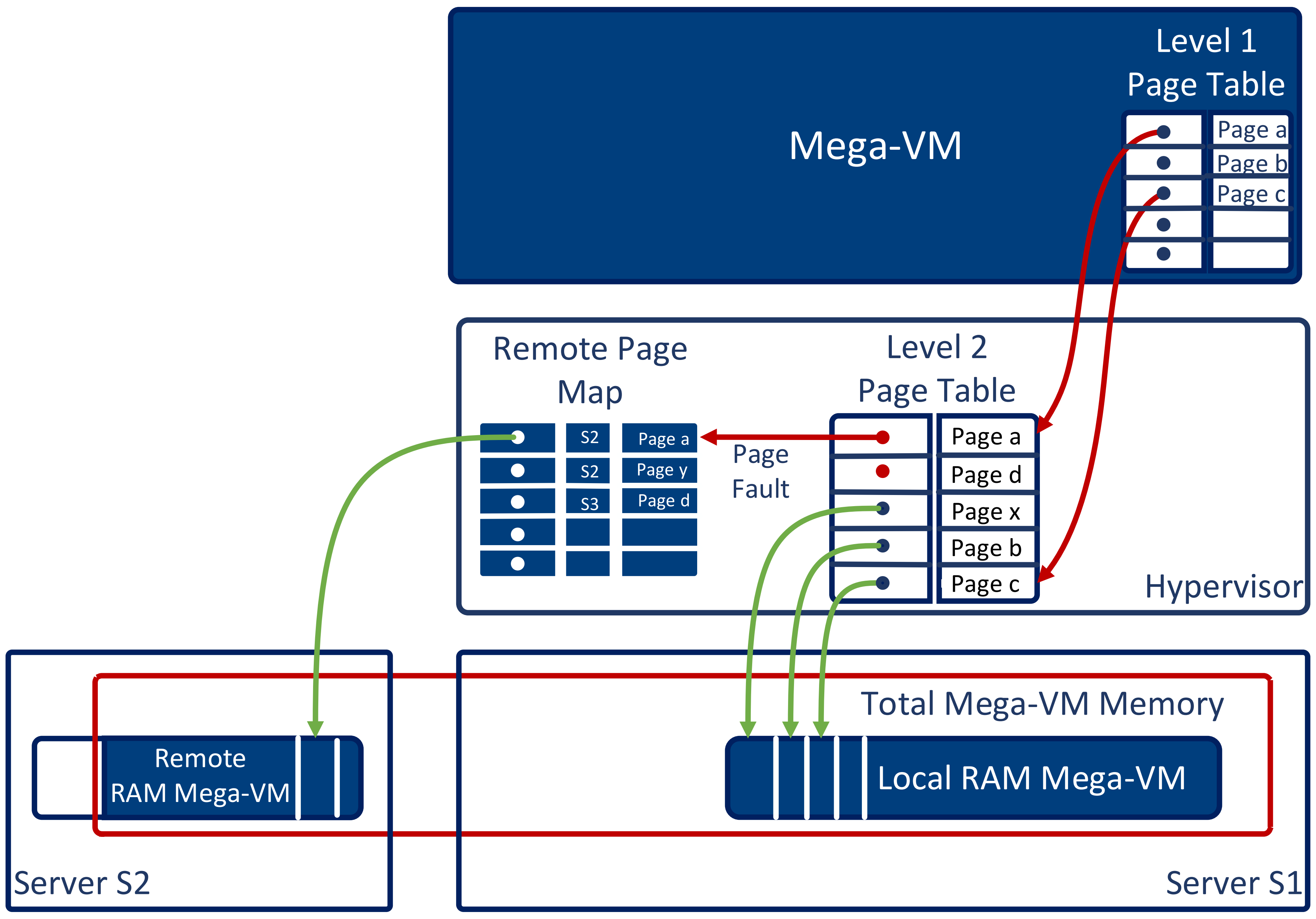


## Mega-VM: Latency Between Servers

- Low-latency between servers is achieved via state-of-the-art interconnects based on InfiniBand and PCI express.
- Empirically achieved 900 nanosecs with Mellanox ConnectX-4 HCAs over InfiniBand.
- RDMA and RMA protocols bypass the networking stack further reducing latencies.
- Secondary interconnects remove congestion caused by internet traffic and provide dedicated RDMA channels across servers.

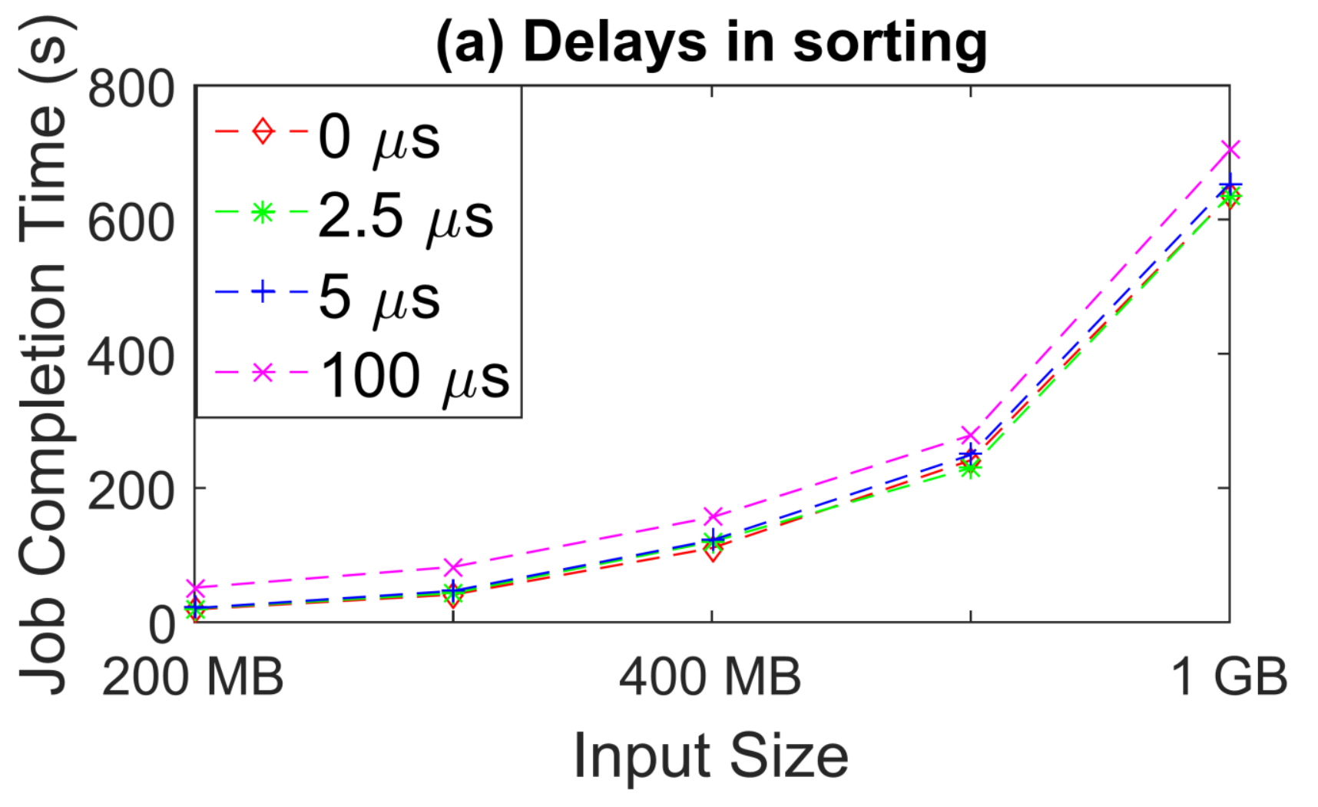
## Mega-VM: Consctruction

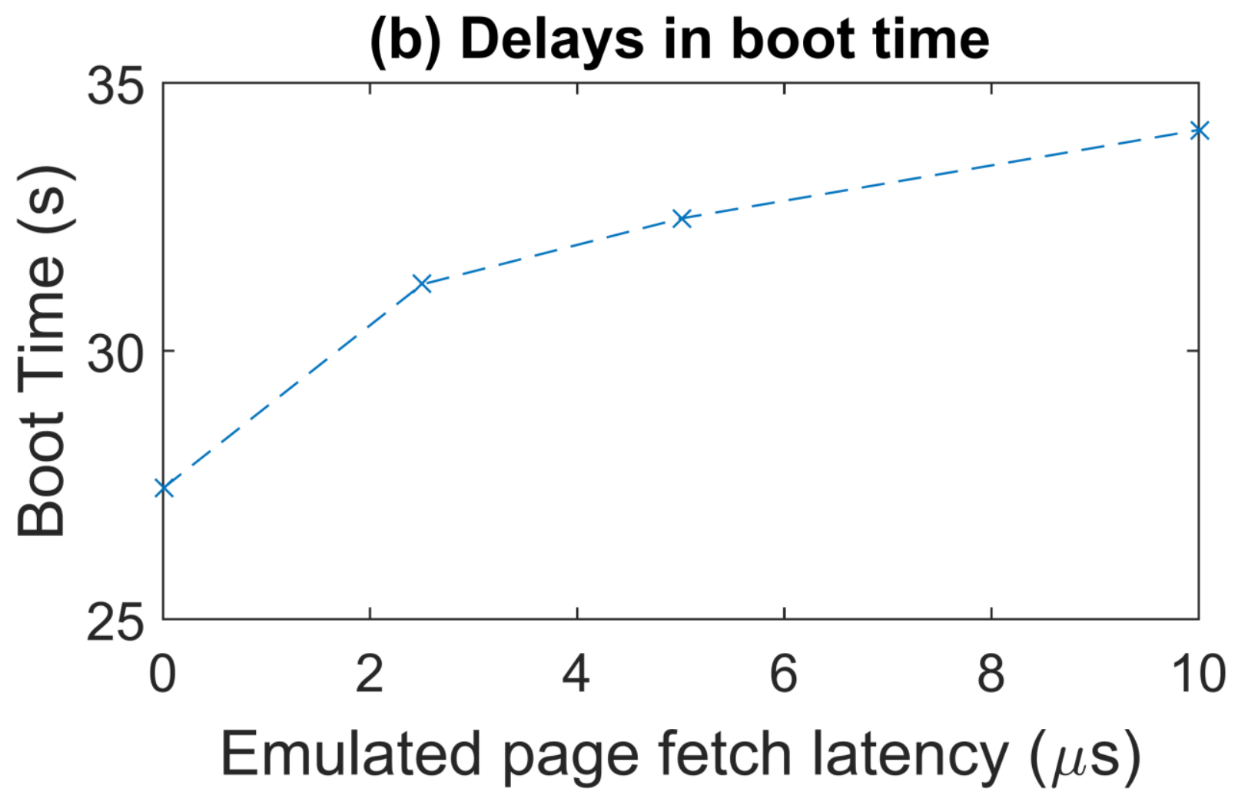
- Hypervisor assigns a large virtual RAM to the guest kernel by trapping BIOS calls from the OS initialization routines providing the guest kernel with the illusion of a large address space.
- Memory book-keeping is done via a lookup table, known as the Remote Page Map (RPM) maintained by the hypervisor.
- A second level page fault (VM to hypervisor) traps to the hypervisor, which quickly fetches the remote page using RDMA.

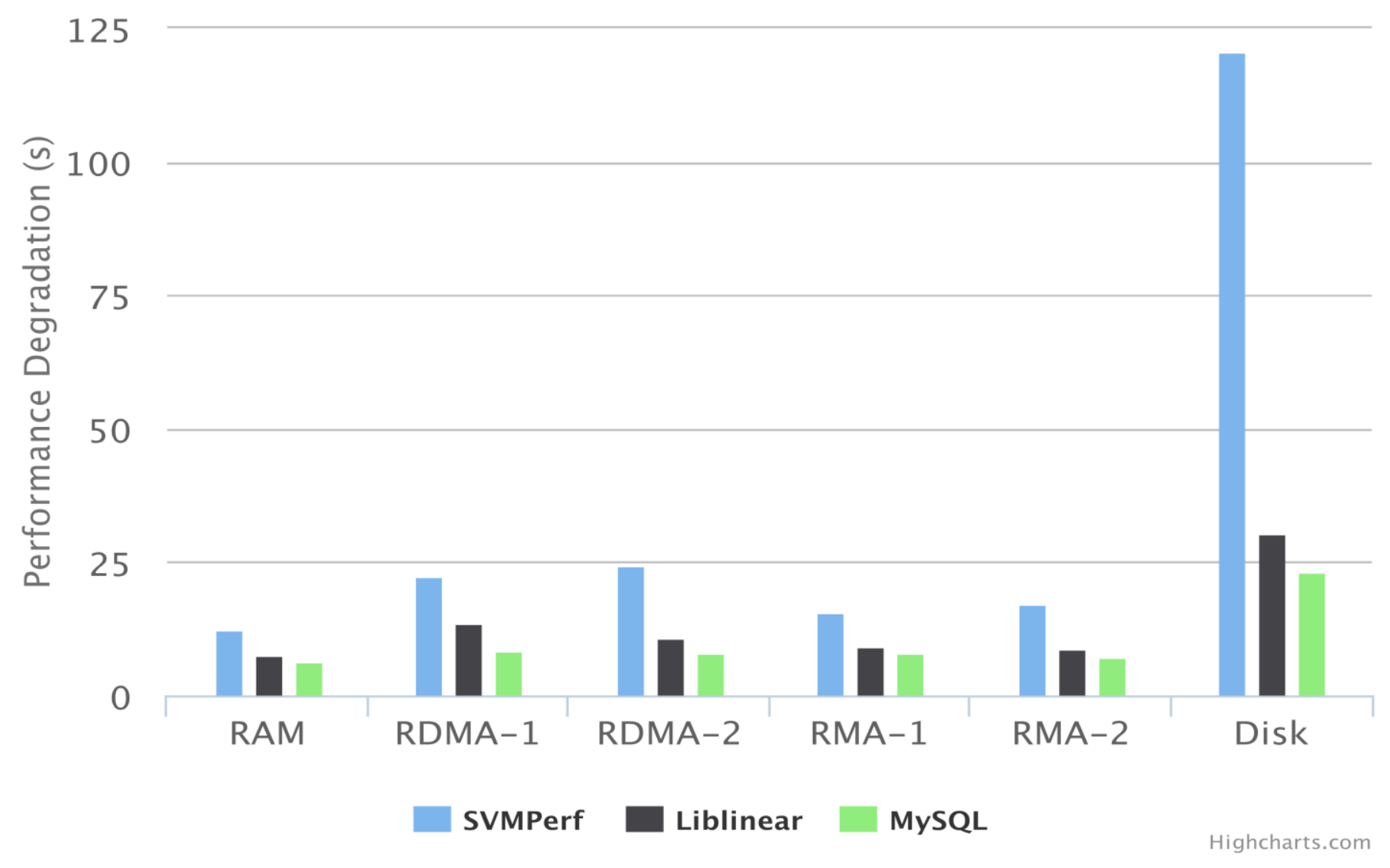


## Mega-VM: Evaluation

- Modified Xen and Bochs to emulate page fetch latencies for each second-level page fault.
- Measured delays in job completion times for sorting and booting (<1.1x) on Xen.
- Measured delays for job completion times for SVM-perf classifier package, Liblinear Logistic Regression tool and MySQL database joins (<1.2x-2x) on Bochs. Also implemented Multinode LRU and a simple page prefetching algorithm.


(a) Delays in sorting


(b) Delays in boot time


Performance Degradation (s)
RAM
RDMA−1
RDMA−2
RMA−1
RMA−2
Disk
SVMPerf
Liblinear
MySQL
Highcharts.com