The Design and Evolution of Live Storage Migration in VMware ESX

Ali Mashtizadeh, VMware, Inc.

Emré Celebi, VMware, Inc.

Tal Garfinkel, VMware, Inc.

Min Cai, VMware, Inc.

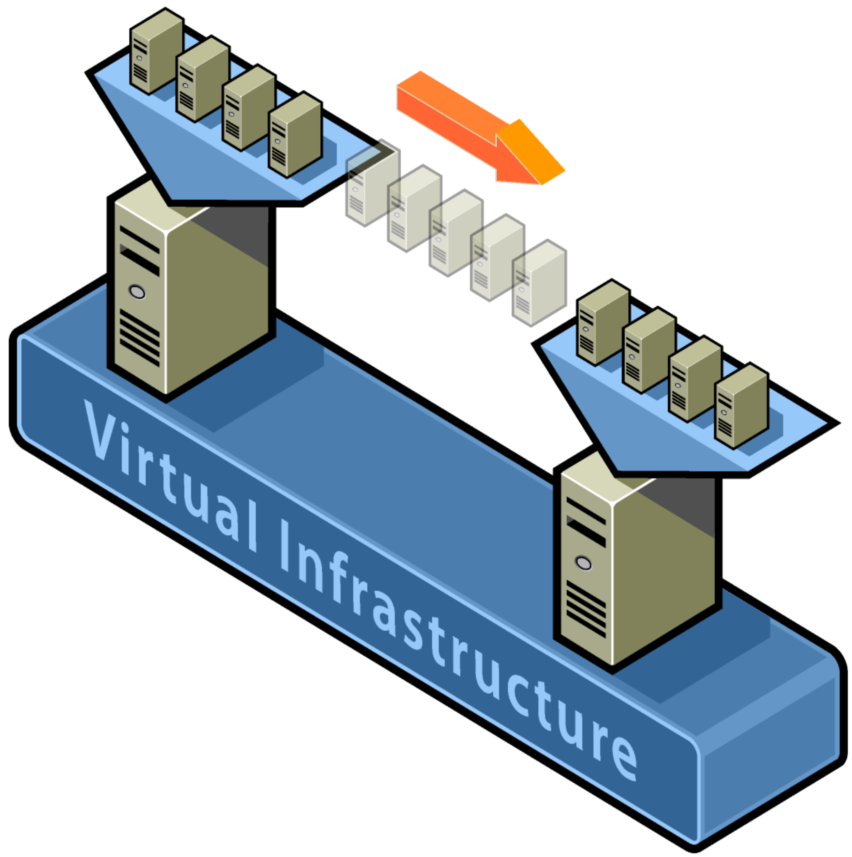
Agenda

- **What is live migration?**
- Migration architectures
- Lessons

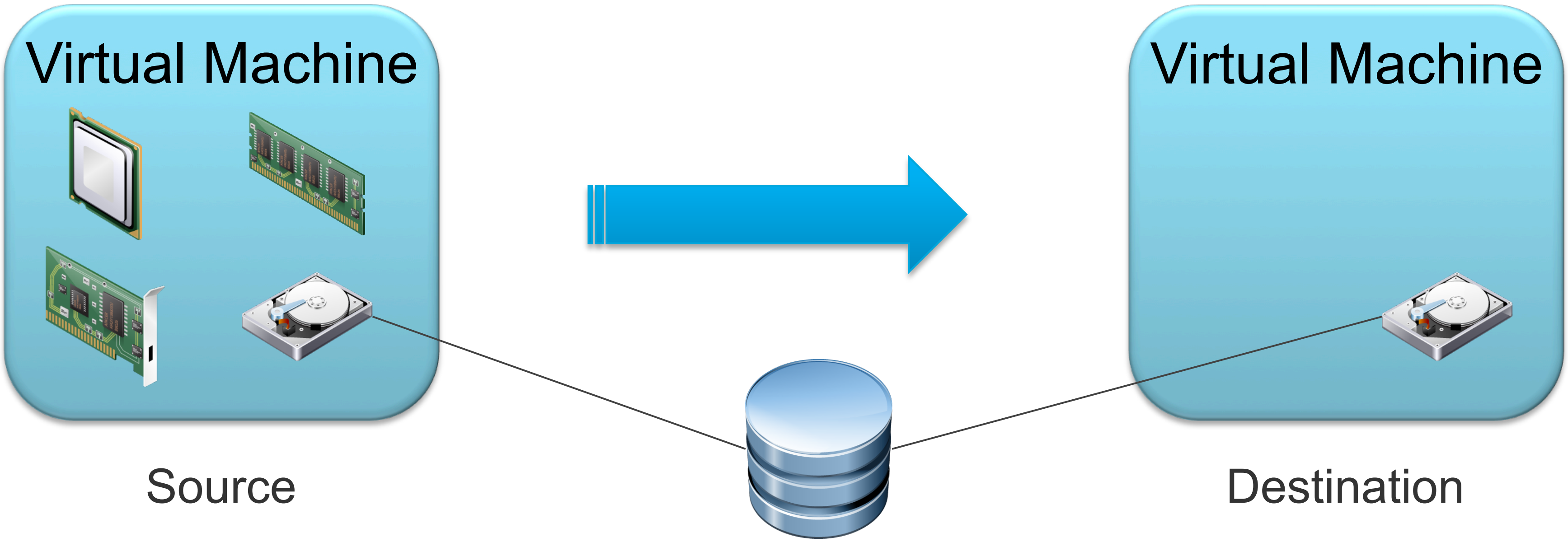
What is live migration (vMotion)?

- **Moves a VM between two physical hosts**
- **No noticeable interruption to the VM (ideally)**

- **Use cases:**
  - Hardware/software upgrades
  - Distributed resource management
  - Distributed power management



Live Migration



- **Disk is placed on a shared volume (100GBs-1TBs)**
- **CPU and Device State is copied (MBs)**

- **Memory is copied (GBs)**
  - Large and it changes often → Iteratively copy

Live Storage Migration

- **What is live storage migration?**
  - Migration of a VM’s virtual disks

- **Why does this matter?**
  - VMs can be very large
  - Array maintenance means you may migrate all VMs in an array
  - Migration time in hours

Live Migration and Storage Live Migration – a short history

- **ESX 2.0 (2003) – Live migration (vMotion)**
  - Virtual disks must live on shared volumes

- **ESX 3.0 (2006) – Live storage migration lite (Upgrade vMotion)**
  - Enabled upgrade of VMFS by migrating the disks
- **ESX 3.5 (2007) – Live storage migration (Storage vMotion)**
  - Storage array upgrade and repair
  - Manual storage load balancing
  - Snapshot based

- **ESX 4.0 (2009) – Dirty block tracking (DBT)**
- **ESX 5.0 (2011) – IO Mirroring**

Agenda

- What is live migration?
- **Migration architectures**
- Lessons

Goals

- **Migration Time**
  - Minimize total end-to-end migration time
  - Predictability of migration time

- **Guest Penalty**
  - Minimize performance loss
  - Minimize downtime

- **Atomicity**
  - Avoid dependence on multiple volumes (for replication fault domains)

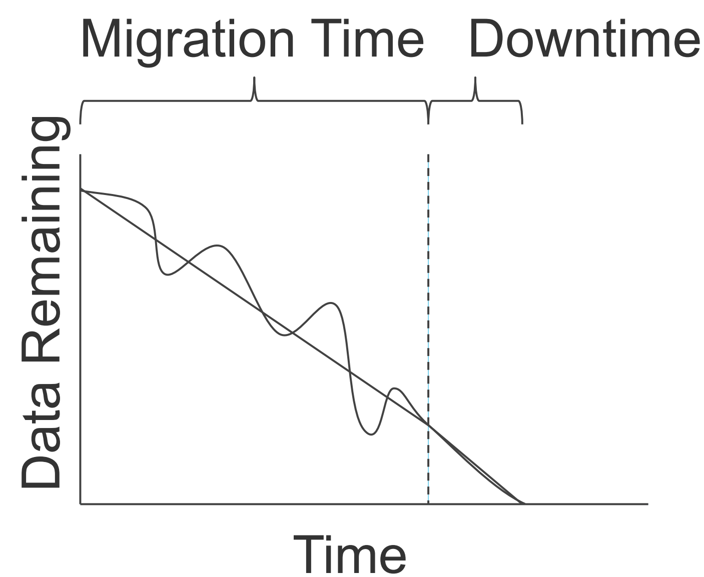
- **Guarantee Convergence**
  - Ideally we want migrations to always complete successfully

Convergence

- **Migration time vs. downtime**

- **More migration time → more guest performance impact**
- **More downtime → more service unavailability**

- **Factors that effect convergence:**
  - Block dirty rate
  - Available storage network bandwidth
  - Workload interactions



- **Challenges:**
  - Many workloads interacting on storage array
  - Unpredictable behavior

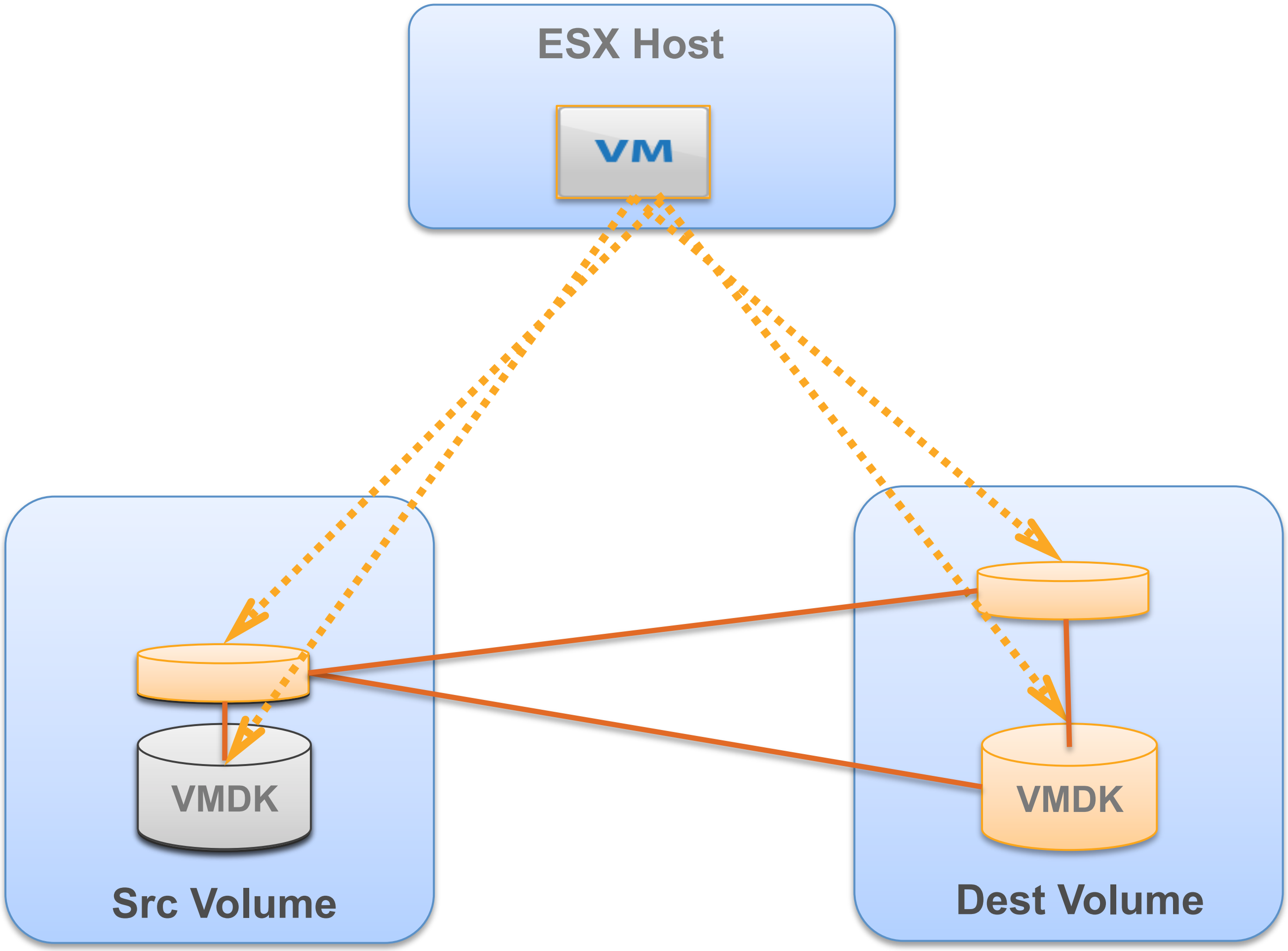
Migration Architectures

- **Snapshotting**

- **Dirty Block Tracking (DBT)**
  - Heat Optimization

- **IO Mirroring**

Snapshot Architecture – ESX 3.5 U1



Synthetic Workload

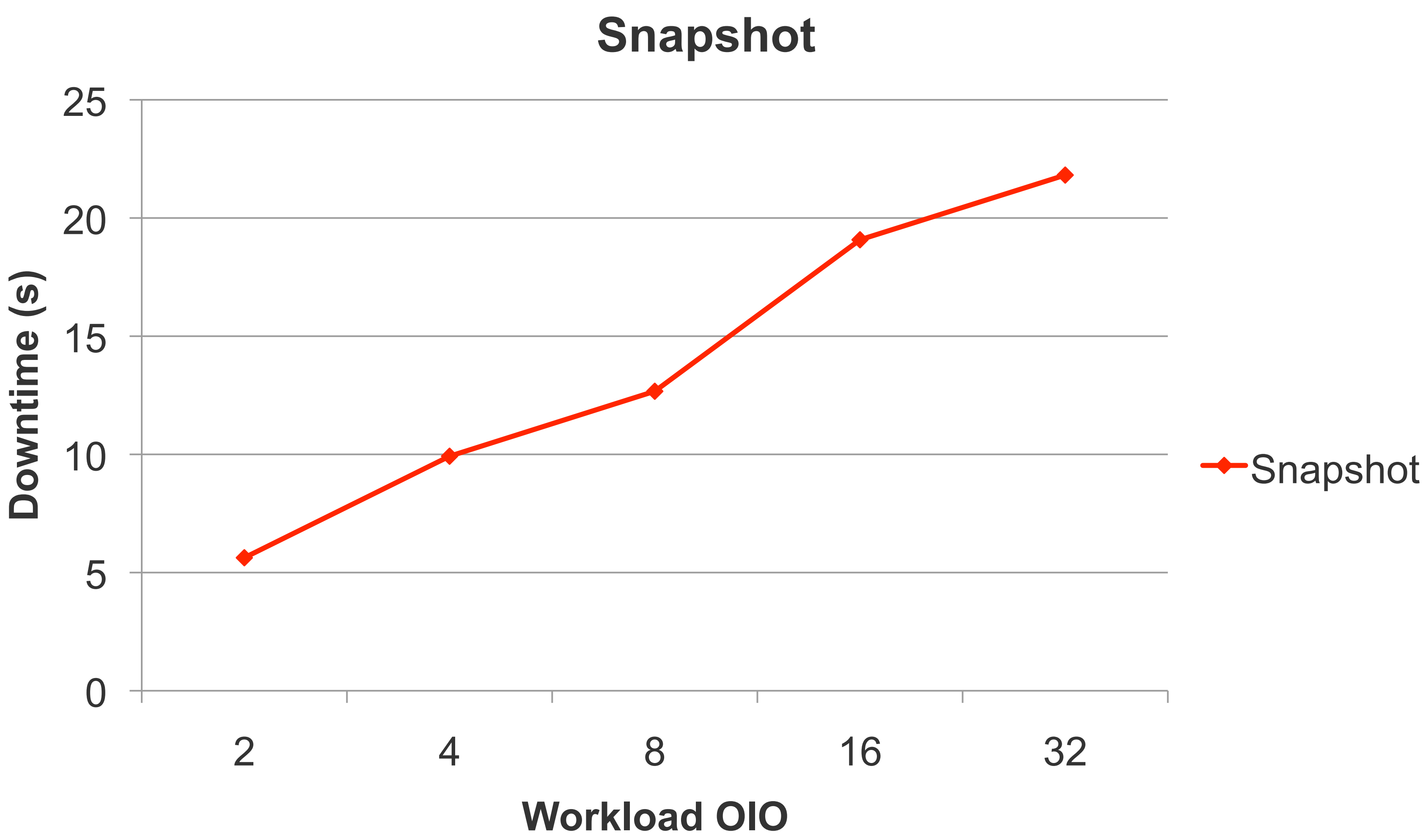
- **Synthetic Iometer workload (OLTP):**
  - 30% Write/70% Read
  - 100% Random
  - 8KB IOs
  - Outstanding IOs (OIOs) from 2 to 32

- **Migration Setup:**
  - Migrated both the 6 GB System Disk and 32 GB Data Disk

- **Hardware:**
  - Dell PowerEdge R710: Dual Xeon X5570 2.93 GHz
  - Two EMC CX4-120 arrays connected via 8Gb Fibre Channel

Migration Time vs. Varying OIO

Downtime vs. Varying OIO



Total Penalty vs. Varying OIO

Snapshot Architecture

- **Benefits**
  - Simple implementation
  - Built on existing and well tested infrastructure
- **Challenges**
  - Suffers from snapshot performance issues
  - Disk space: Up to 3x the VM size
  - Not an atomic switch from source to destination
    - A problem when spanning replication fault domains
  - Downtime
  - Long migration times

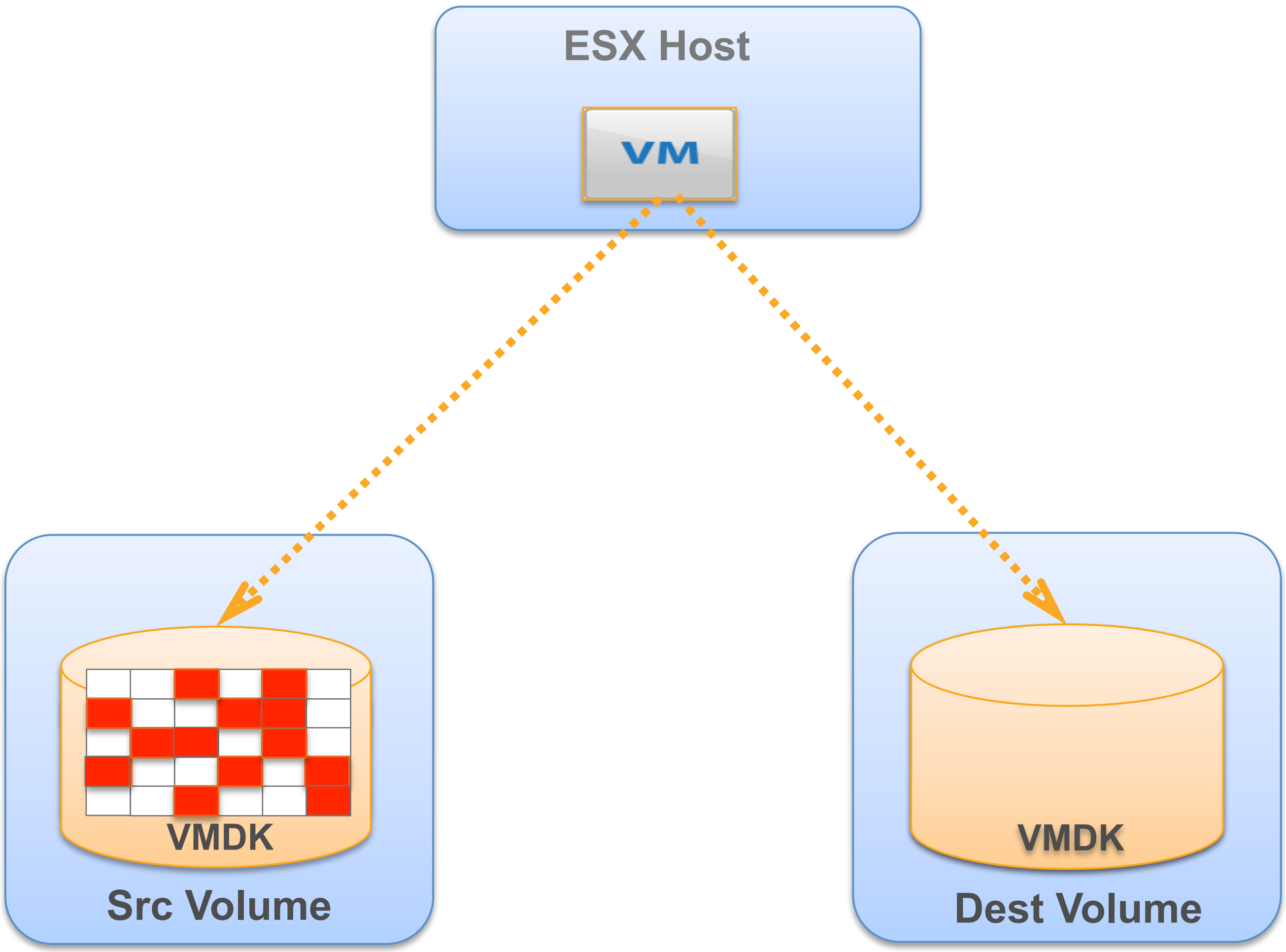
Snapshot versus Dirty Block Tracking Intuition

- **Virtual disk level snapshots have overhead to maintain metadata**
- **Requires lots of disk space**

- **We want to operate more like live migration**
  - Iterative copy
  - Block level copy rather than disk level – enables optimizations

- **We need a mechanism to track writes**

Dirty Block Tracking (DBT) Architecture – ESX 4.0/4.1

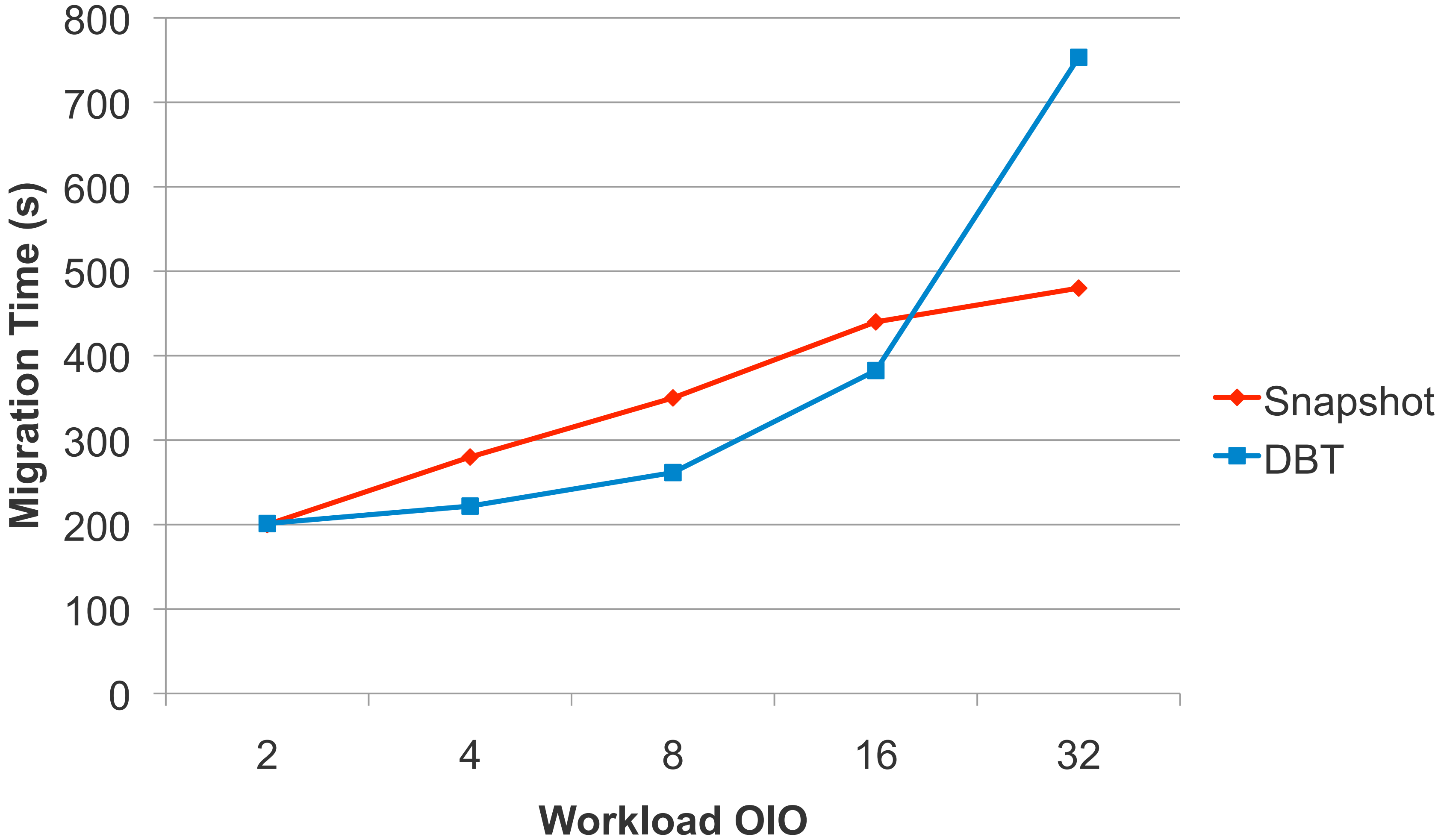


Data Mover (DM)

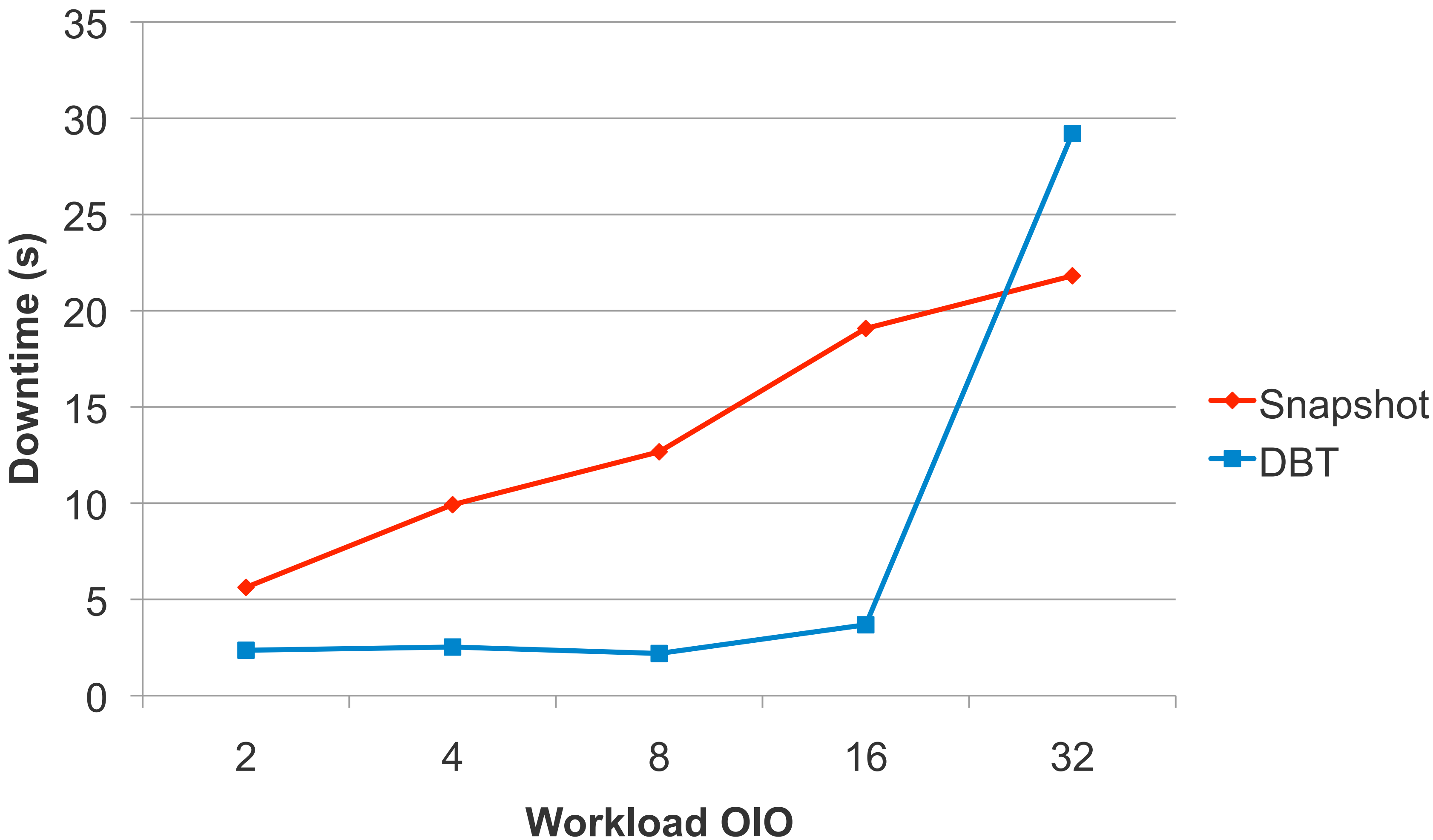
- **Kernel Service**
  - Provides disk copy operations
  - Avoids memory copy (DMAs only)

- **Operation (default configuration)**
  - 16 Outstanding IOs
  - 256 KB IOs

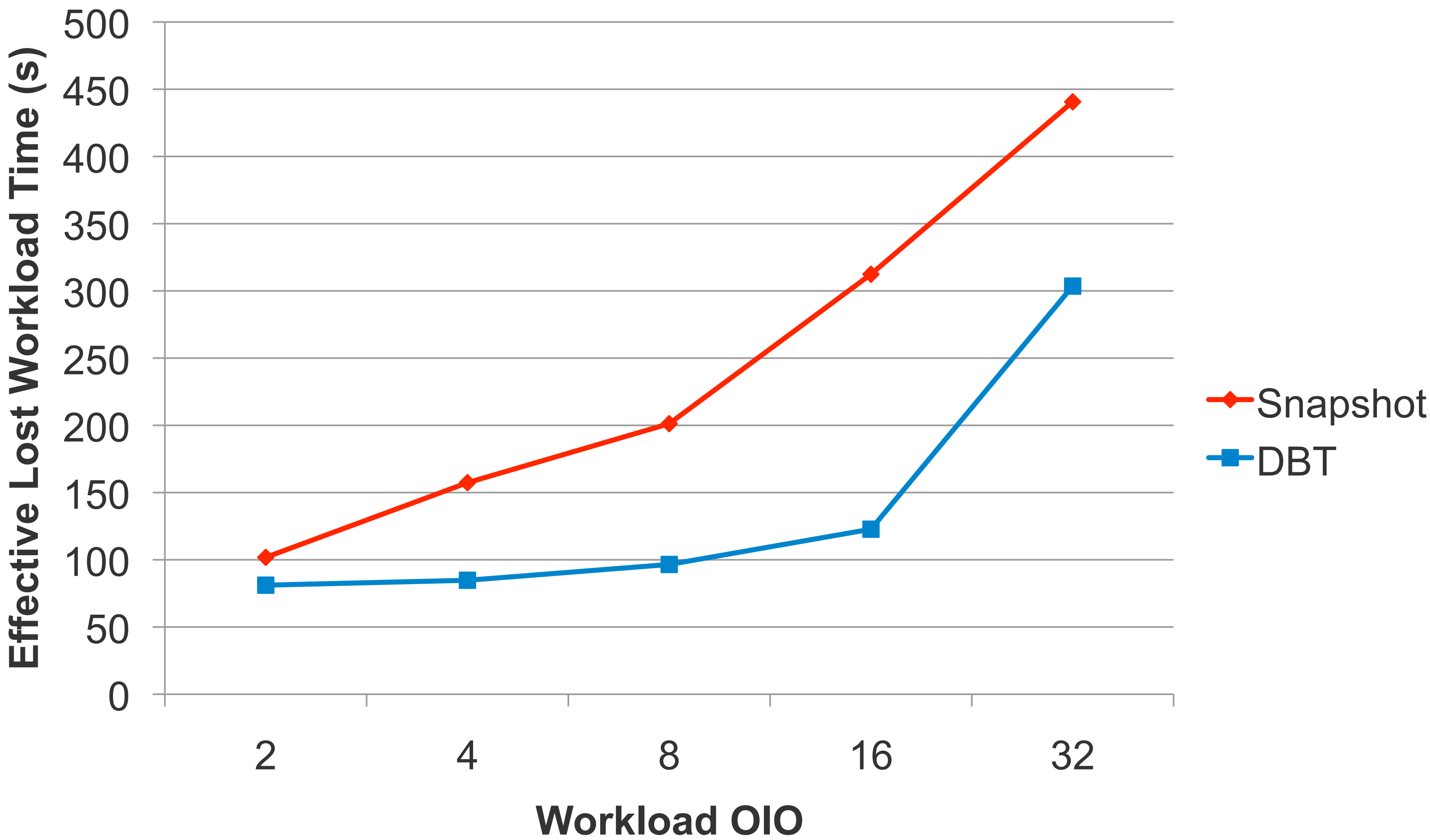
Migration Time vs. Varying OIO



Downtime vs. Varying OIO



Total Penalty vs. Varying OIO

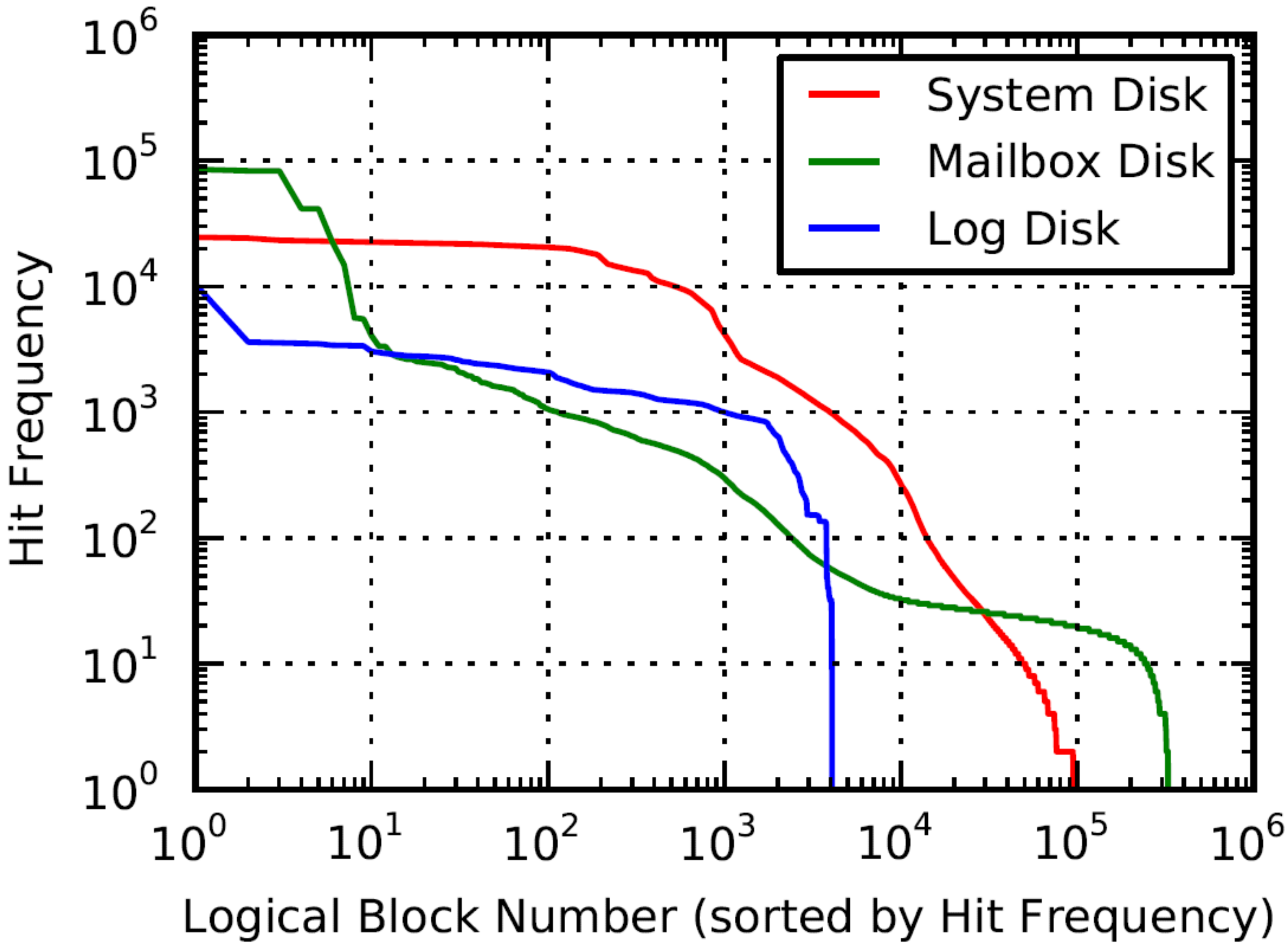


Dirty Block Tracking Architecture

- **Benefits**
  - Well understood architecture based similar to live VM migration
  - Eliminated performance issues associated with snapshots
  - Enables block level optimizations
  - Atomicity

- **Challenges**
  - Migrations may not converge (and will not succeed with reasonable downtime)
    - Destination slower than source
    - Insufficient copy bandwidth
  - Convergence logic difficult to tune
  - Downtime

Block Write Frequency – Exchange Workload



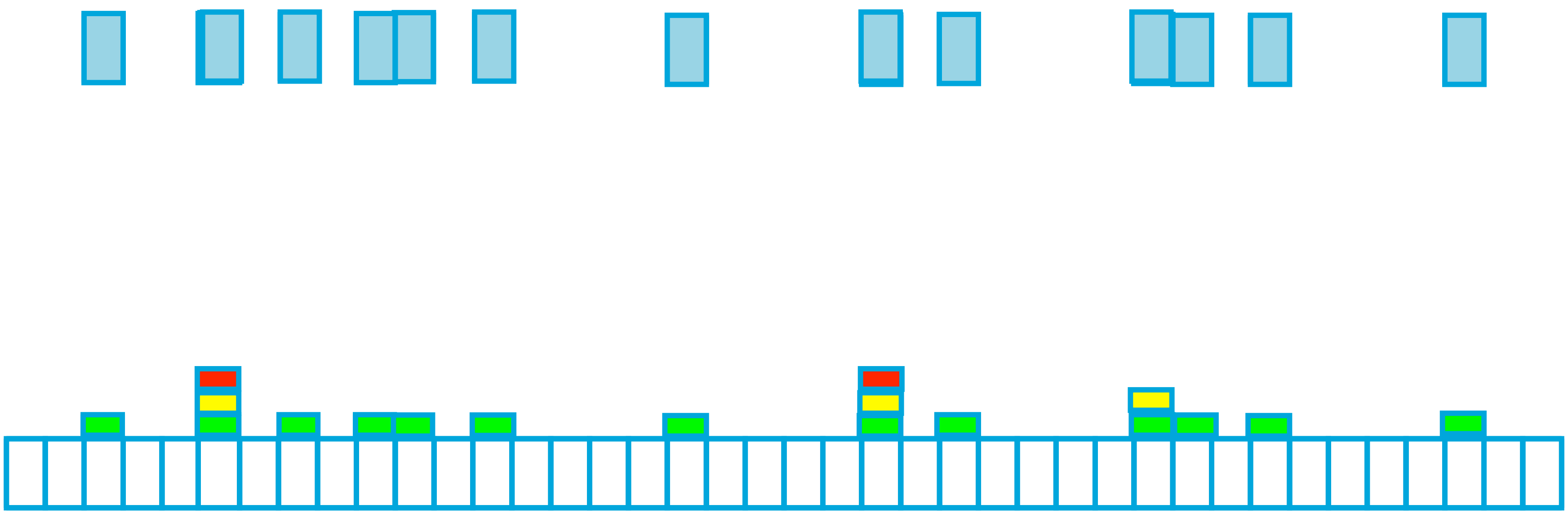
Heat Optimization – Introduction

- **Defer copying data that is frequently written**

- **Detects frequently written blocks**
  - File system metadata
  - Circular logs
  - Application specific data

- **No significant benefit for:**
  - Copy on write file systems (e.g. ZFS, HAMMER, WAFL)
  - Workloads with limited locality (e.g. OLTP)

Heat Optimization – Design



vmware

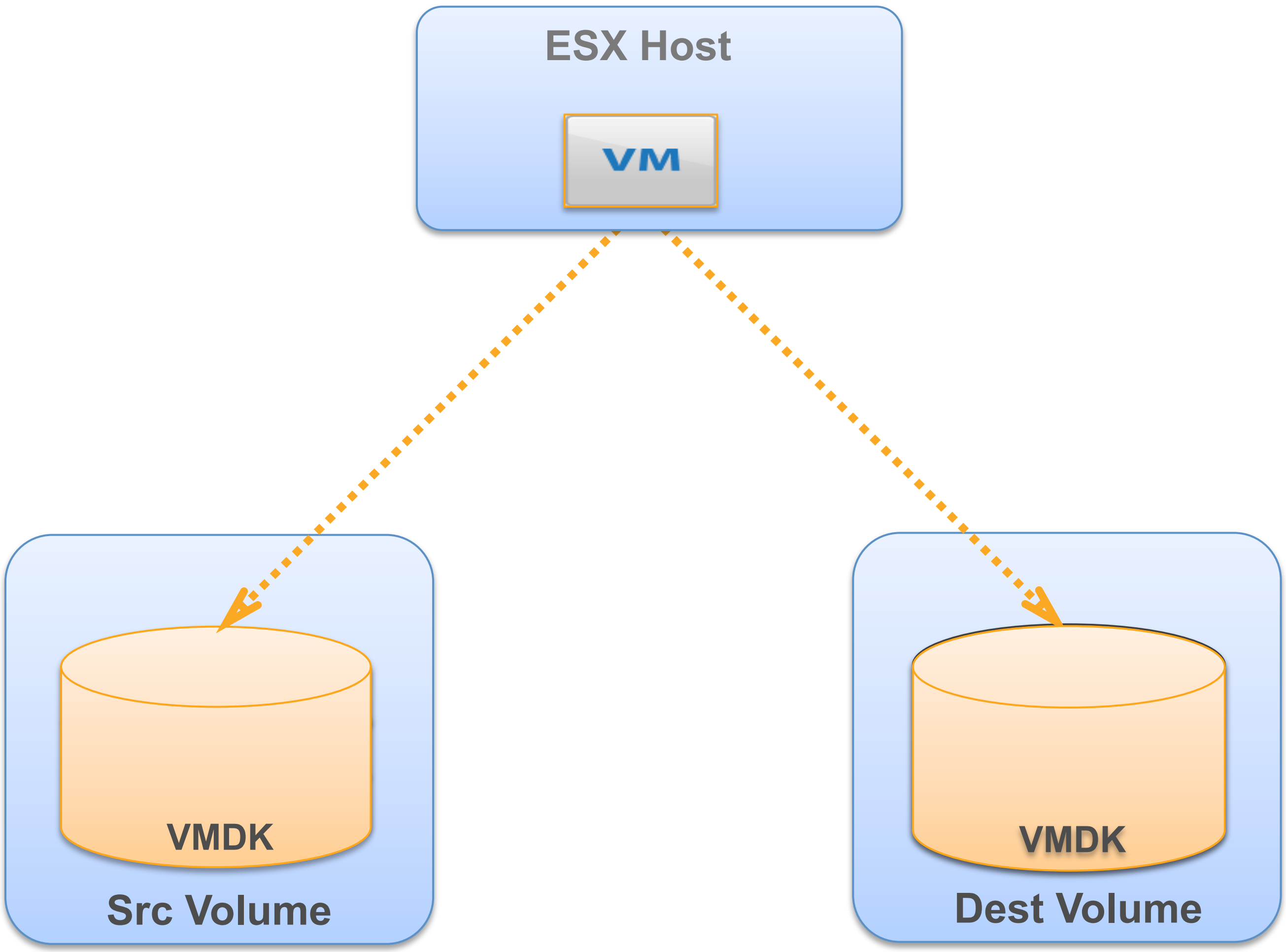
DBT versus IO Mirroring Intuition

- **Live migration intuition – intercepting all memory writes is expensive**
  - Trapping interferes with data fast path
  - DBT traps only first write to a page
  - Writes a batched to aggregate subsequent writes without trap

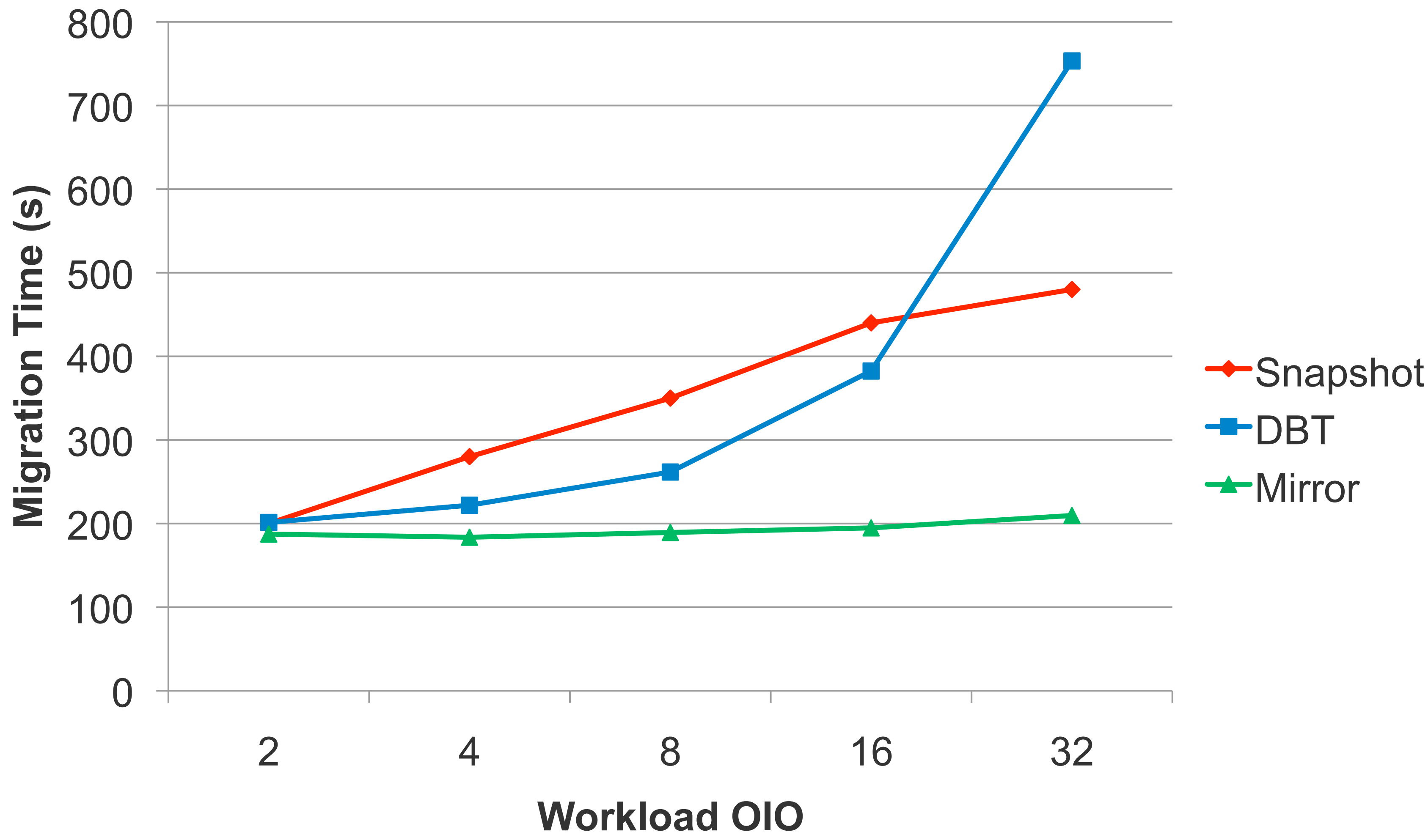
- **Intercepting all storage writes is cheap**
  - IO stack processes all IOs already

- **IO Mirroring**
  - Synchronously mirror all writes
  - Single pass copy of the bulk of the disk

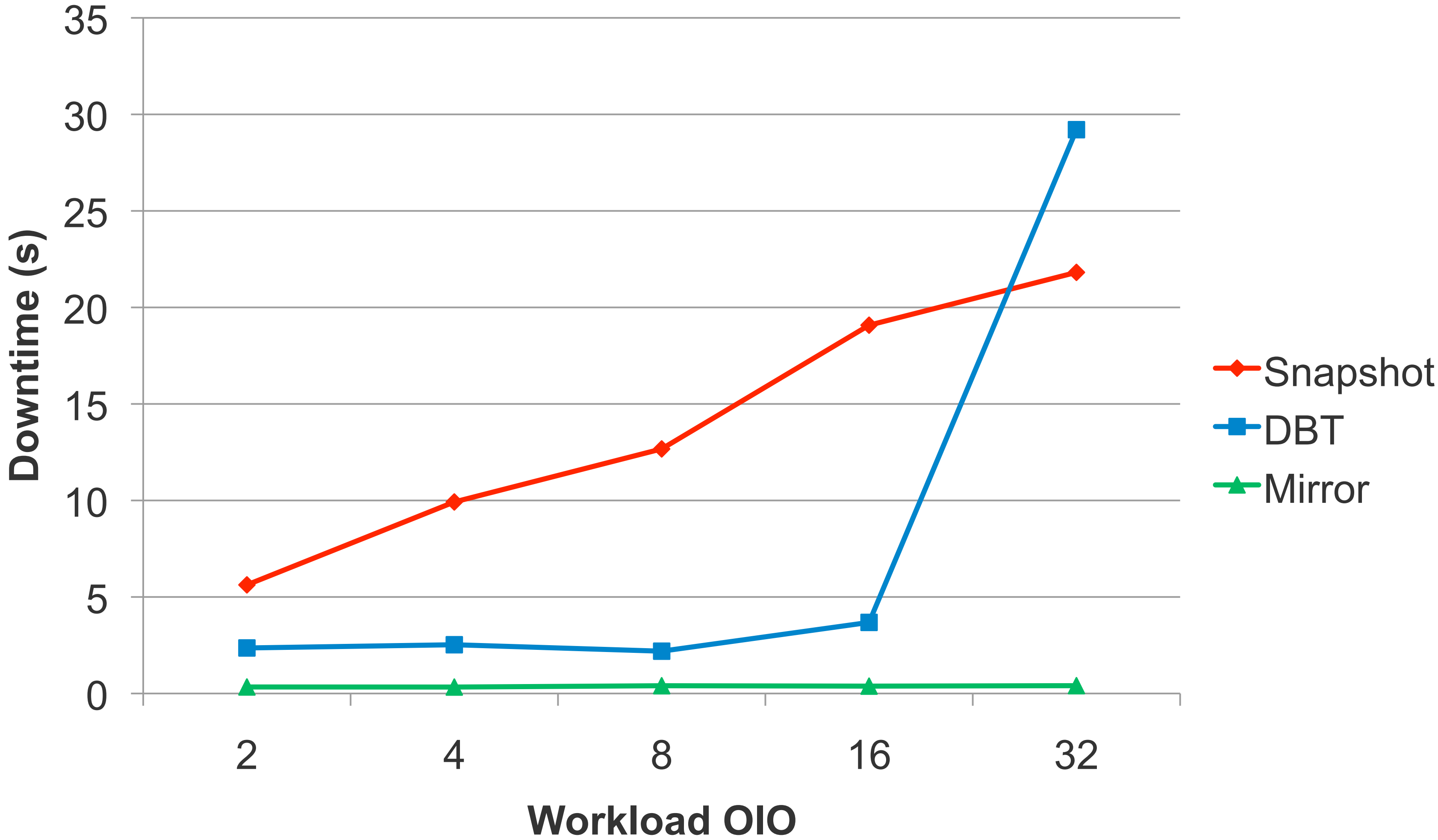
IO Mirroring – ESX 5.0



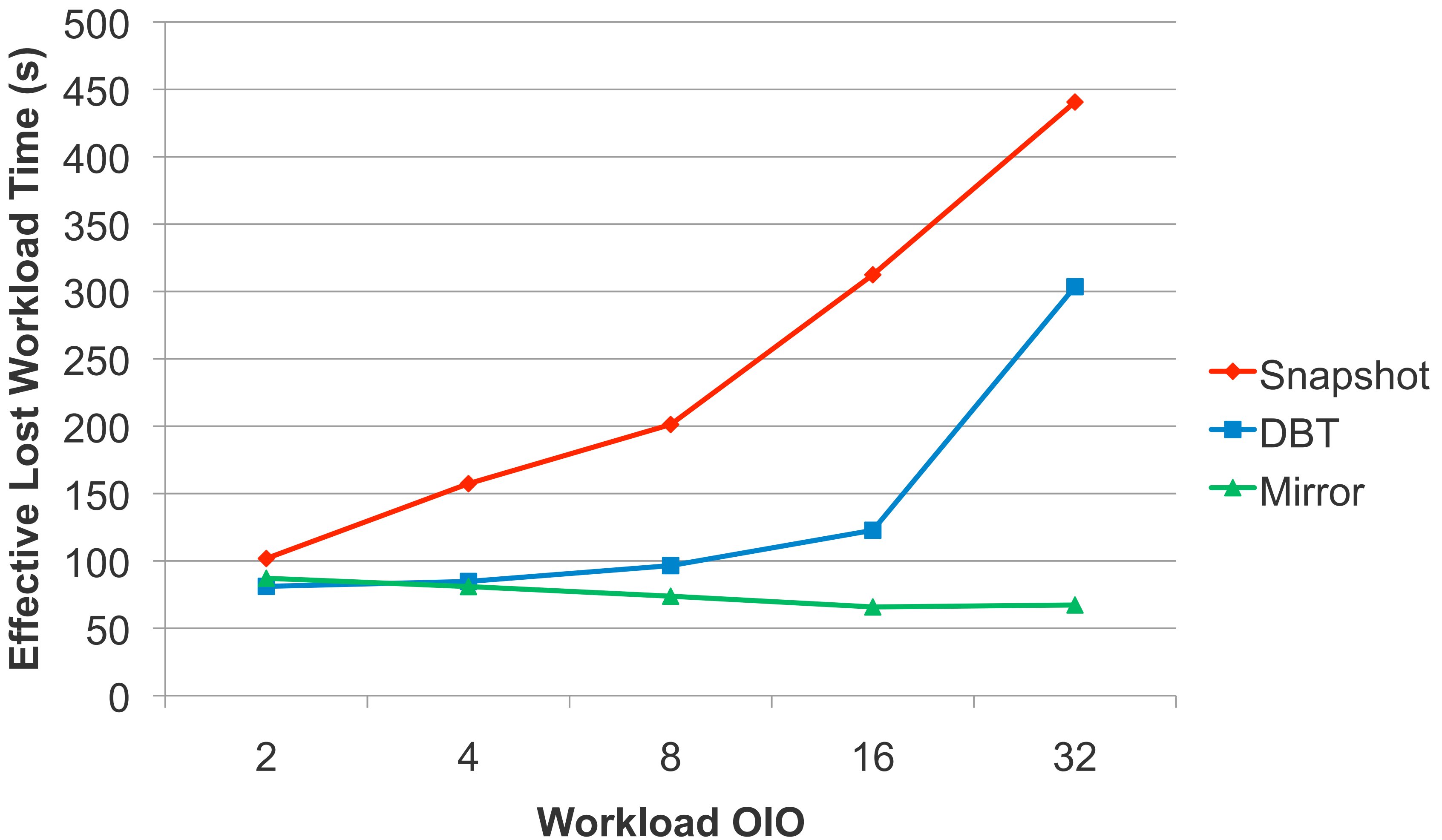
Migration Time vs. Varying OIO



Downtime vs. Varying OIO



Total Penalty vs. Varying OIO

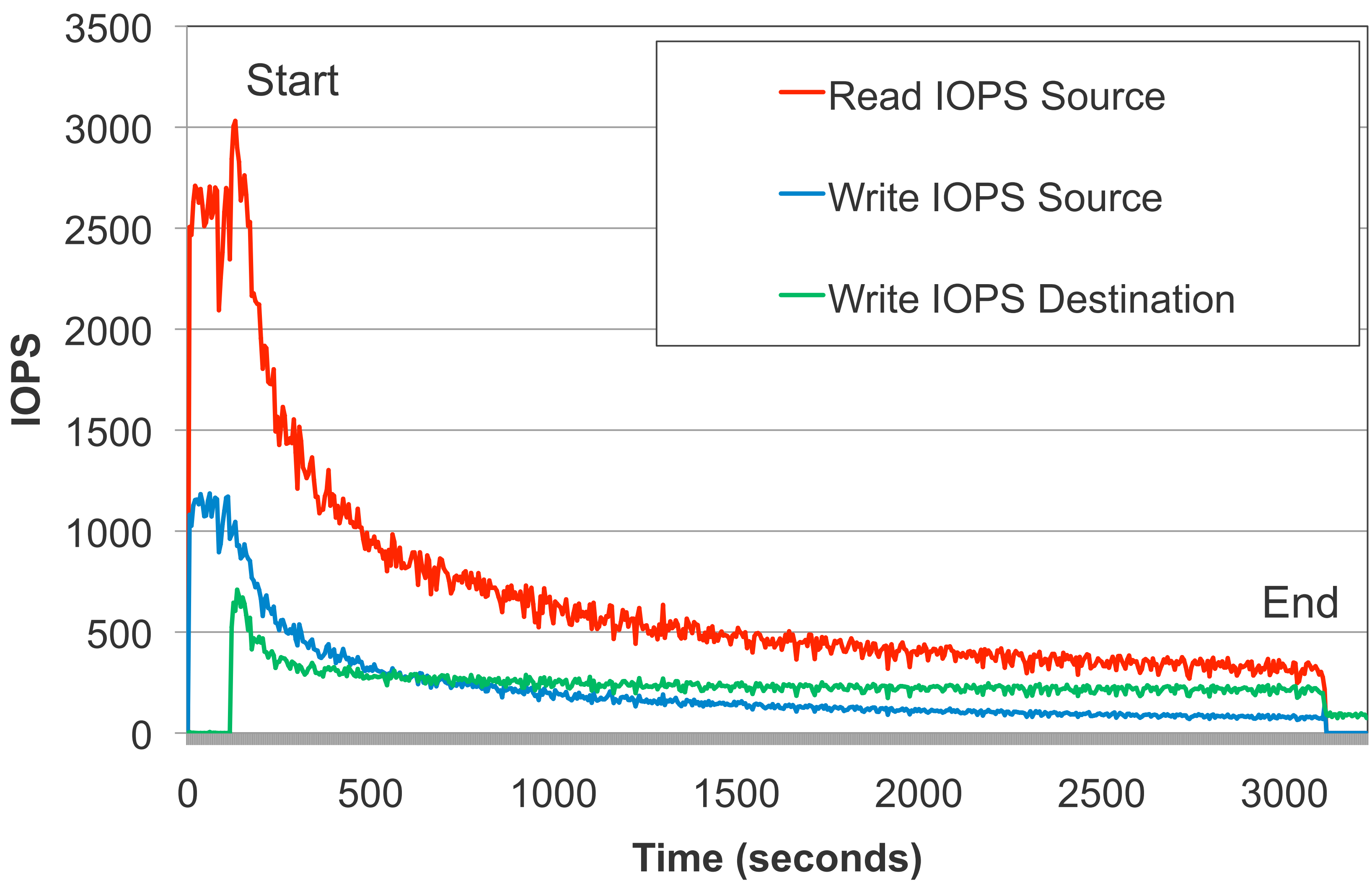


IO Mirroring

- **Benefits**
  - Minimal migration time
  - Near-zero downtime
  - Atomicity
- **Challenges**
  - Complex code to guarantee atomicity of the migration
  - Odd guest interactions require code for verification and debugging

Throttling the source

IO Mirroring to Slow Destination



Agenda

- What is live migration?
- Migration architectures
- Lessons

Recap

- **In the beginning live migration**

- **Snapshot:**
  - Usually has the worst downtime/penalty
  - Whole disk level abstraction
  - Snapshot overheads due to metadata
  - No atomicity

- **DBT:**
  - Manageable downtime (except when OIO > 16)
  - Enabled block level optimizations
  - Difficult to make convergence decisions
  - No natural throttling

Recap – Cont.

- **Insight: storage is not memory**
  - Interposing on all writes is practical and performant

- **IO Mirroring:**
  - Near-zero downtime
  - Best migration time consistency
  - Minimal performance penalty
  - No convergence logic necessary
  - Natural throttling

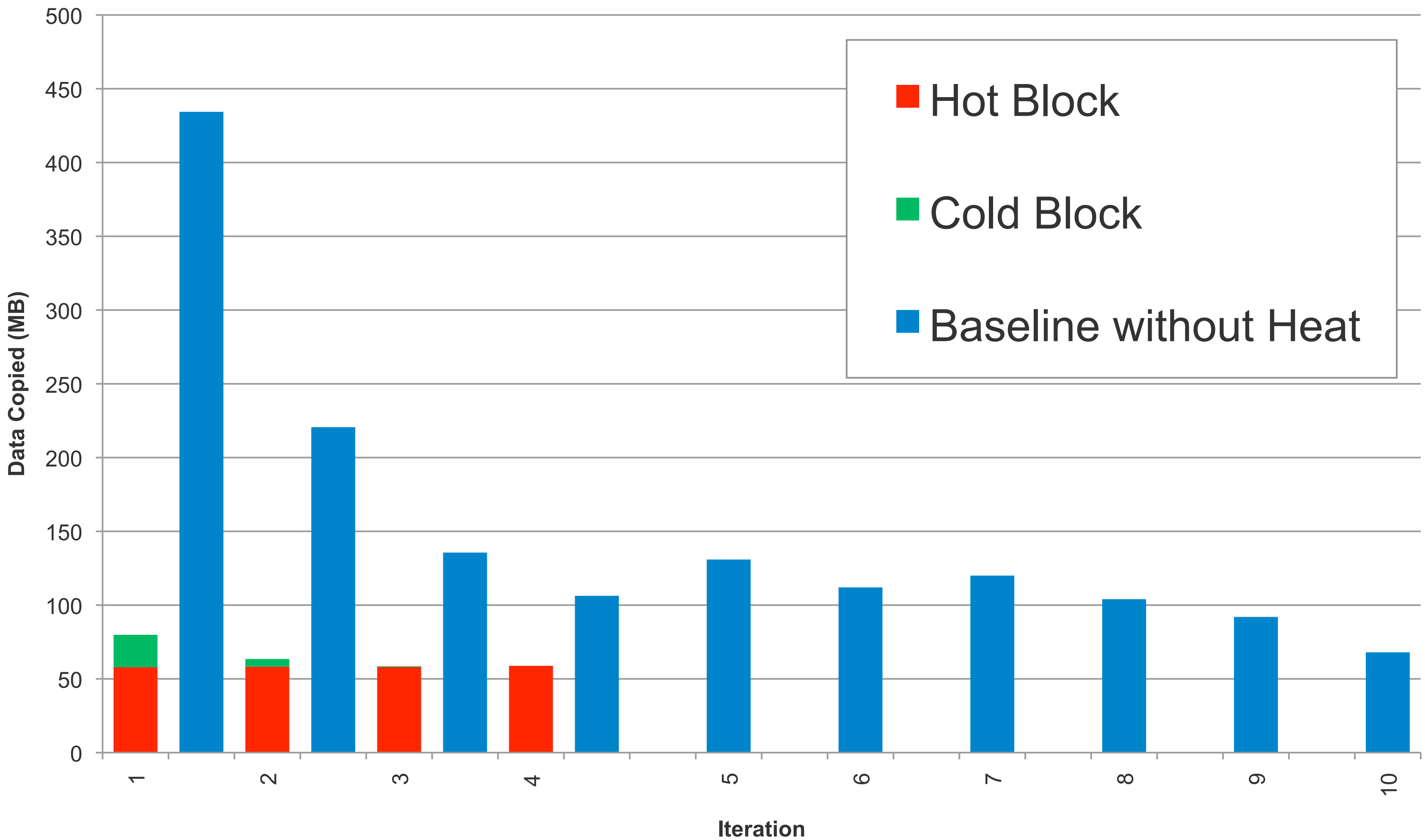
Future Work

- **Leverage workload analysis to reduce mirroring overhead**
  - Defer mirroring regions with potential sequential write IO patterns
  - Defer hot blocks
  - Read ahead for lazy mirroring
- **Apply mirroring to WAN migrations**
  - New optimizations and hybrid architecture

Thank You!

Backup Slides

Exchange Migration with Heat



Exchange Workload

- **Exchange 2010:**
  - Workload generated by Exchange Load Generator
  - 2000 User mailboxes
  - Migrated only the 350 GB mailbox disk

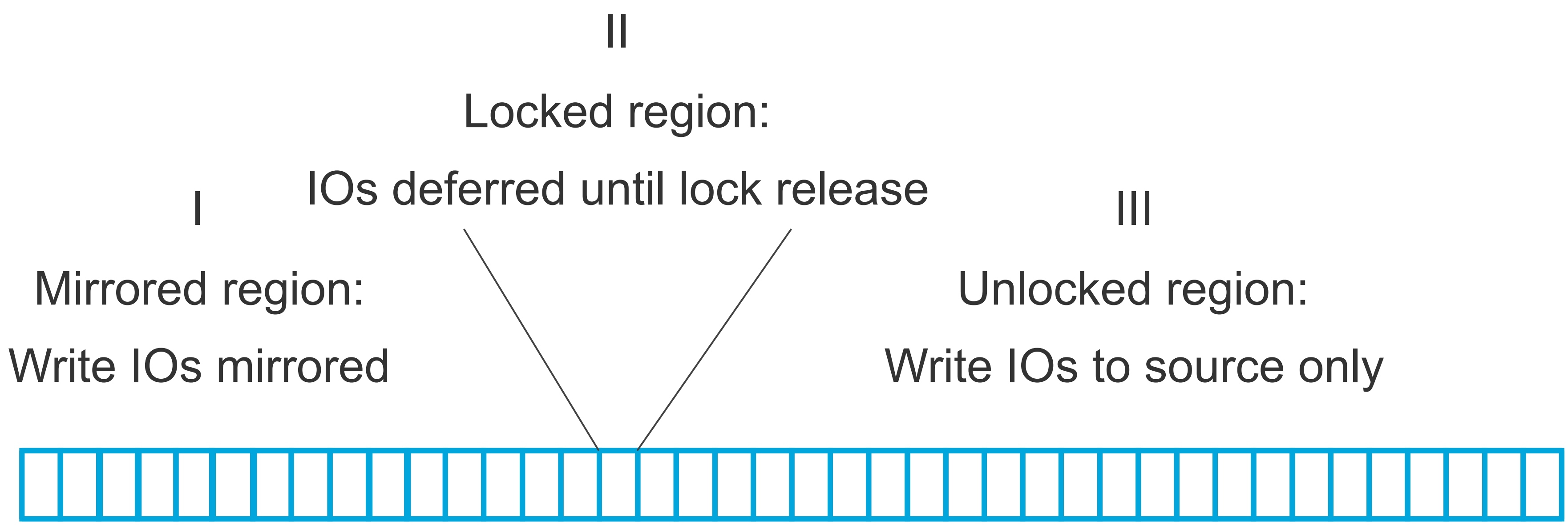
- **Hardware:**
  - Dell PowerEdge R910: 8-core Nehalem-EX
  - EMC CX3-40
  - Migrated between two 6 disk RAID-0 volumes

Exchange Results

Type	Migration Time	Downtime
DBT	2935.5	13.297
Incremental DBT	2638.9	7.557
IO Mirroring	1922.2	0.220
DBT (2x)	Failed	-
Incremental DBT (2x)	Failed	-
IO Mirroring (2x)	1824.3	0.186

IO Mirroring Lock Behavior

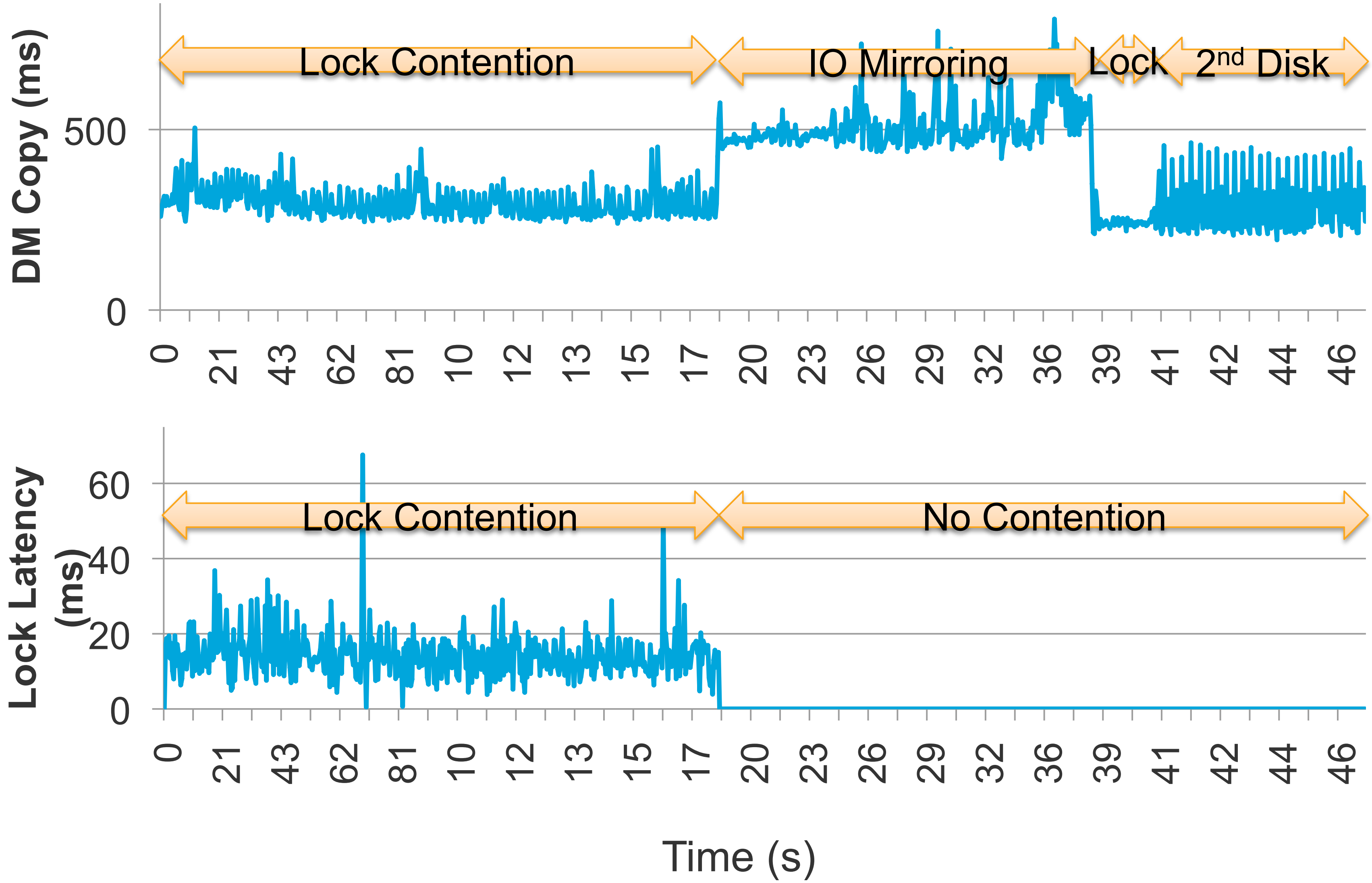
- **Moving the lock region**
  1. Wait for non-mirrored inflight read IOs to complete. (queue all IOs)
  2. Move the lock range
  3. Release queued IOs



Non-trivial Guest Interactions

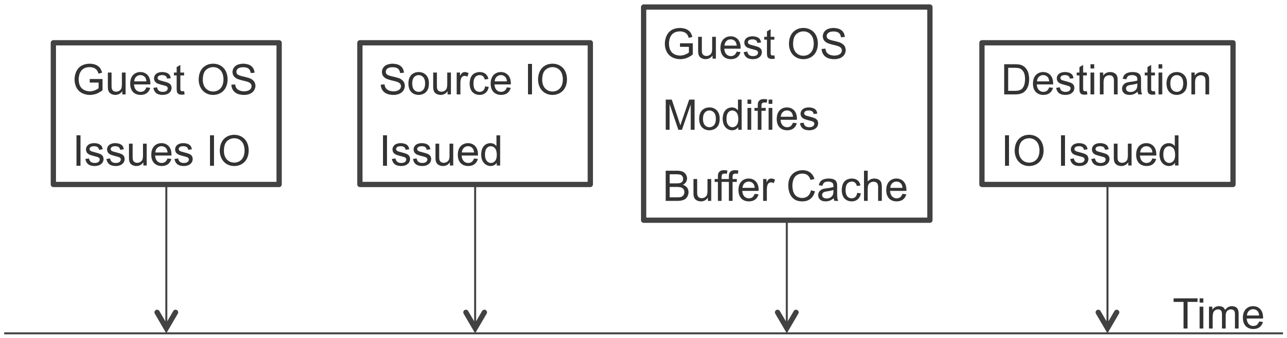
- Guest IO crossing disk locked regions
- Guest buffer cache changing
- Overlapped IOs

Lock Latency and Data Mover Time



Source/Destination Valid Inconsistencies

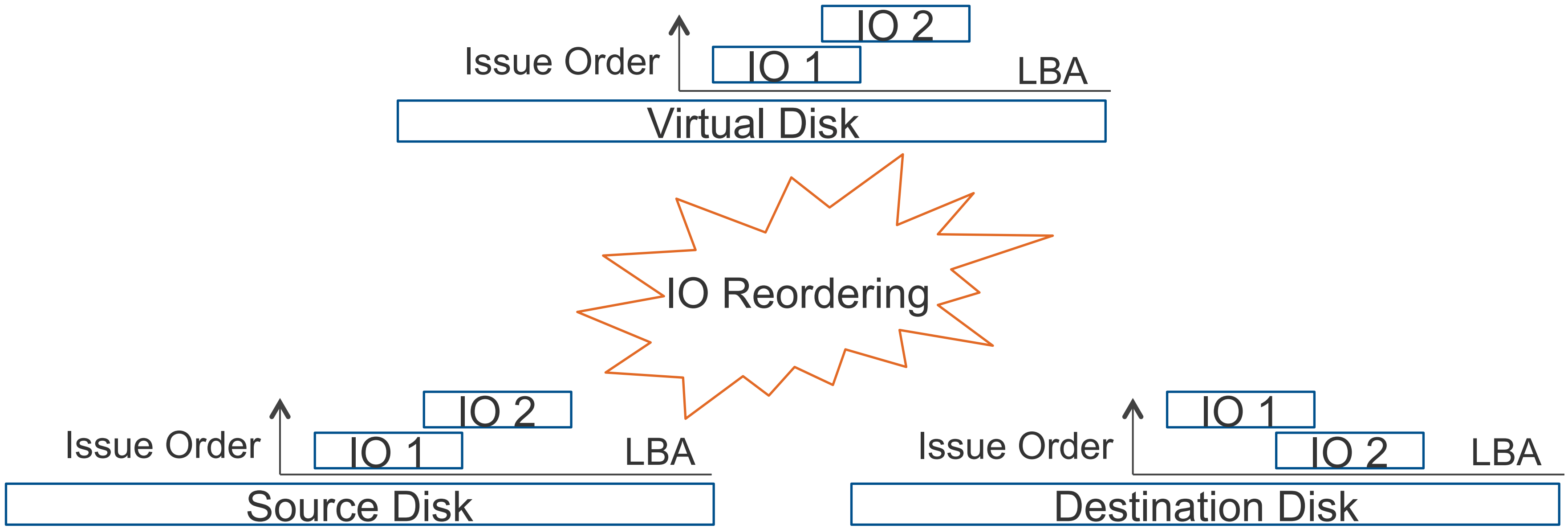
- **Normal Guest Buffer Cache Behavior**



- **This inconsistency is okay!**
  - Source and destination are both valid crash consistent views of the disk

Source/Destination Valid Inconsistencies

- **Overlapping IOs (Synthetic workloads only)**



- **Seen in Iometer and other synthetic benchmarks**
- **File systems do not generate this**

Incremental DBT Optimization – ESX 4.1