

Experimental Study of Virtual Machine Migration in Support of Reservation of Cluster Resources

Ming Zhao, Renato J. Figueiredo

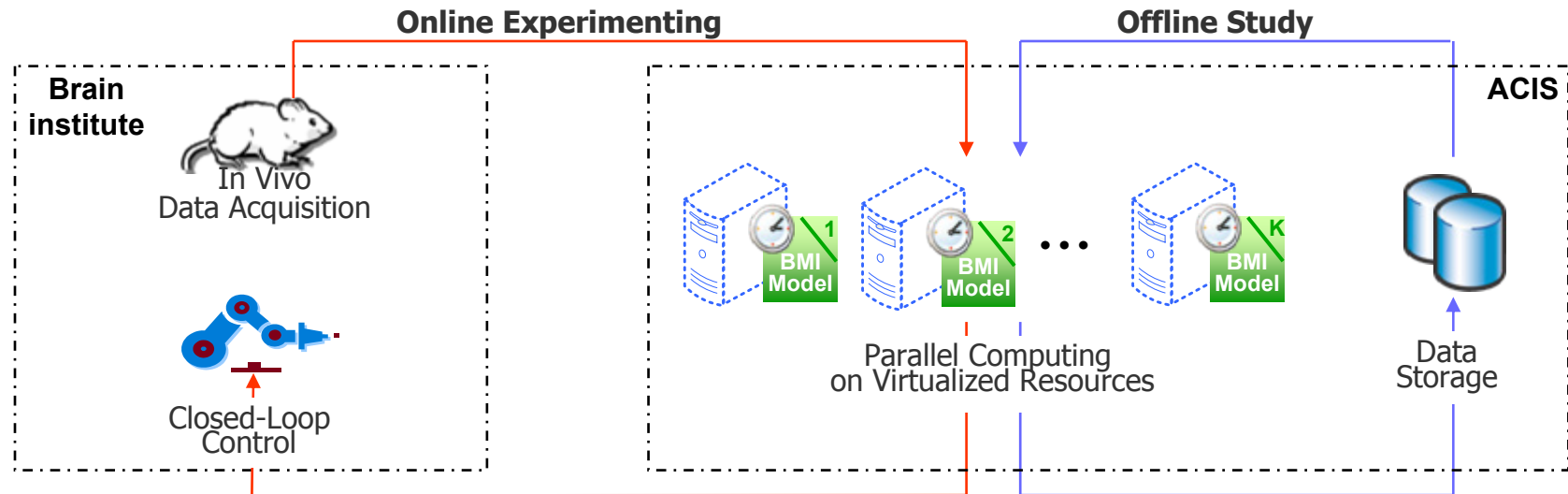
*Advanced Computing and Information Systems (ACIS)
Electrical and Computer Engineering*

University of Florida

{ming, renato}@acis.ufl.edu

Motivating Application

- Dynamic Data-driven Brain-machine Interface
 - Resource demanding parallel application
 - Time-critical during online experiments
 - Must finish each closed loop within 100ms
 - No stringent time requirement for offline study

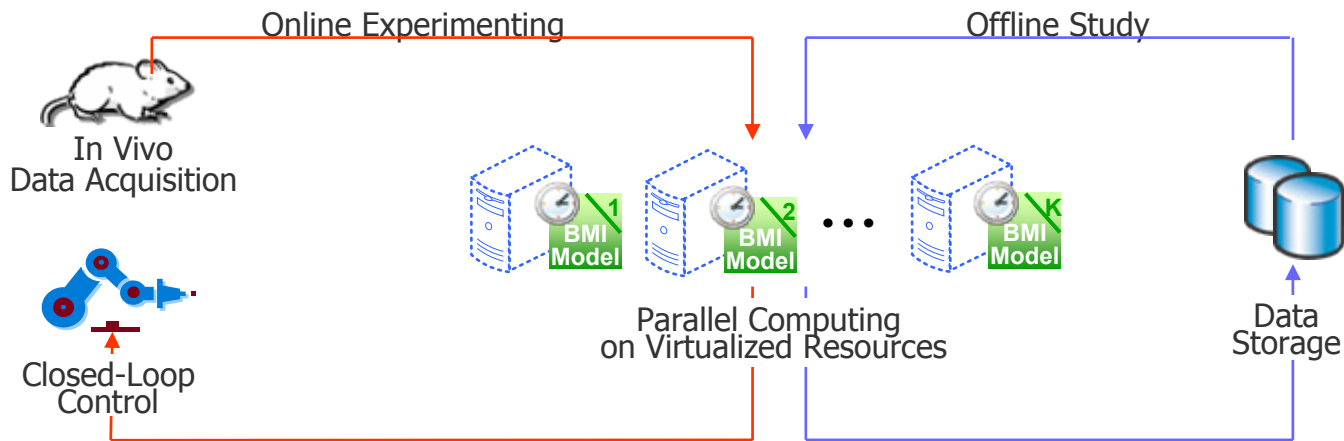


Dynamic Data-driven Brain-machine Interfaces

(<http://www.acis.ufl.edu/dddasbmi>)

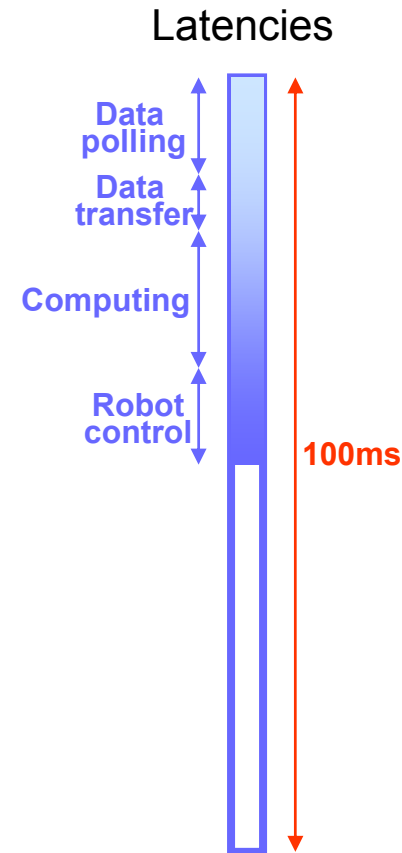
Motivating Application

- Motivation for VM-based resource reservation
 - Online experiments
 - Dedicate cluster resources for VMs running BMI models
 - Without reservation, deadline is often violated
 - Offline study
 - Share cluster resources with VMs running other workloads
 - Cluster reservation
 - Migrating exiting VMs to other resources



Dynamic Data-driven Brain-machine Interfaces

(<http://www.acis.ufl.edu/dddasbmi>)



Overview

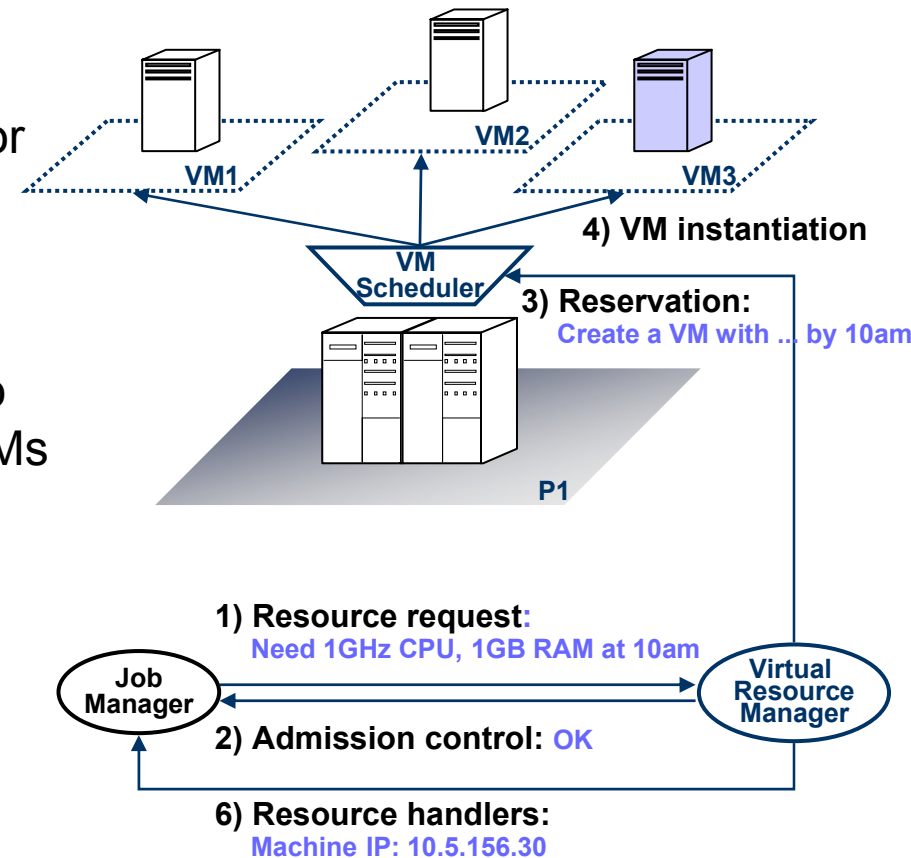
- Goal:
 - Efficient VM-based cluster resource reservation
- Challenge:
 - Vacate resources in time (to meet schedule) but not too early (to fully utilize resources)
- Solution:
 - Build a migration overhead model through experimental analysis

Outline

- Architecture
- Experimental analysis
 - Methodology
 - Migrating a single VM
 - Migrating a sequence of VMs
 - Migrating VMs in parallel
- Summary

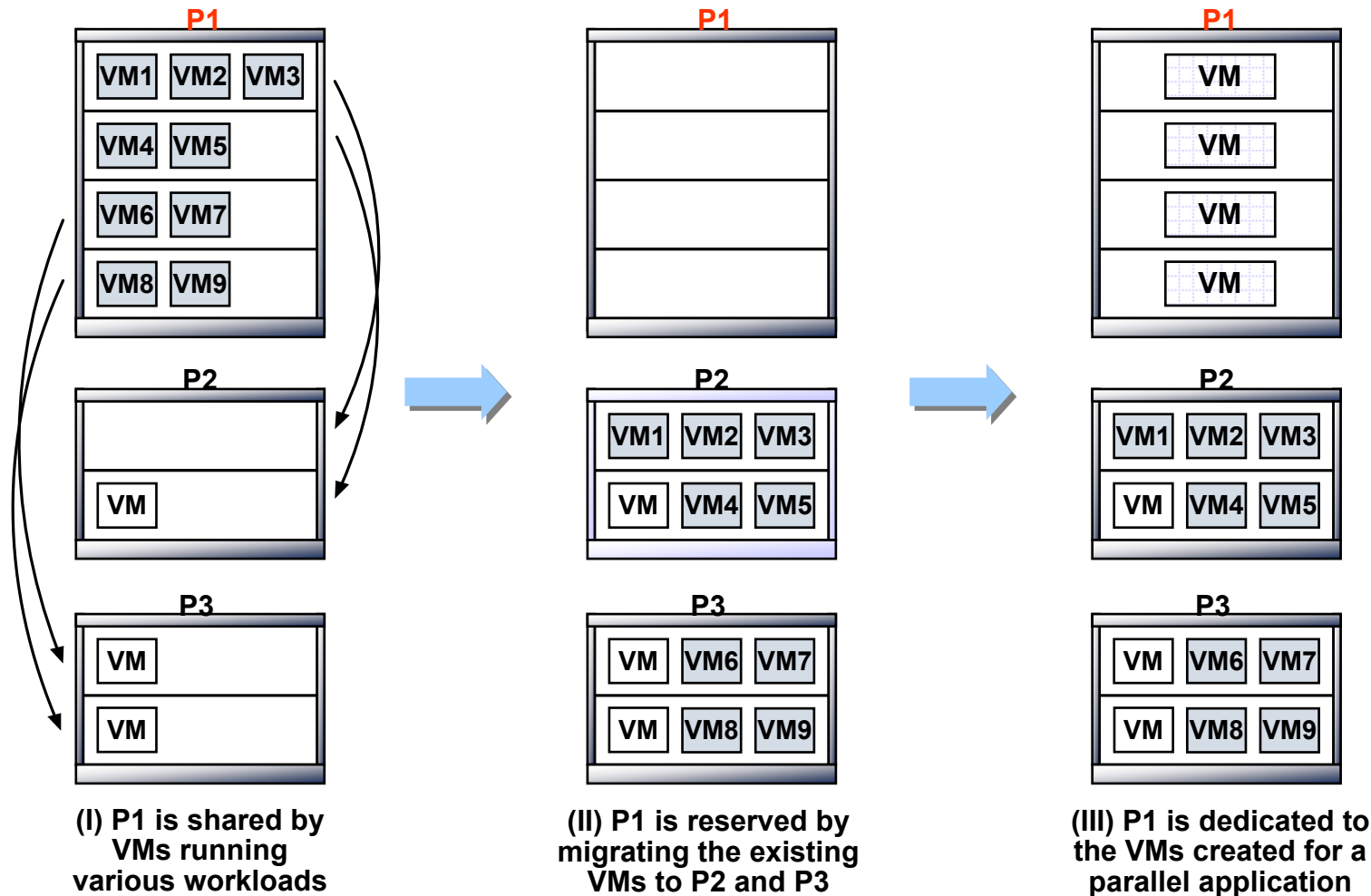
VM-based Resource Reservation

- Virtual resource manager
 - Global management of virtualized resources
 - Presents an abstracted interface for resource requests
 - Hides the complexity of using virtualized resources
 - Coordinates with VM schedulers to reserve, allocate resources with VMs
- Virtual machine scheduler
 - Per-host management of VMs
 - Presents a unified interface for managing VMs
 - Hides the complexity of using different VM software



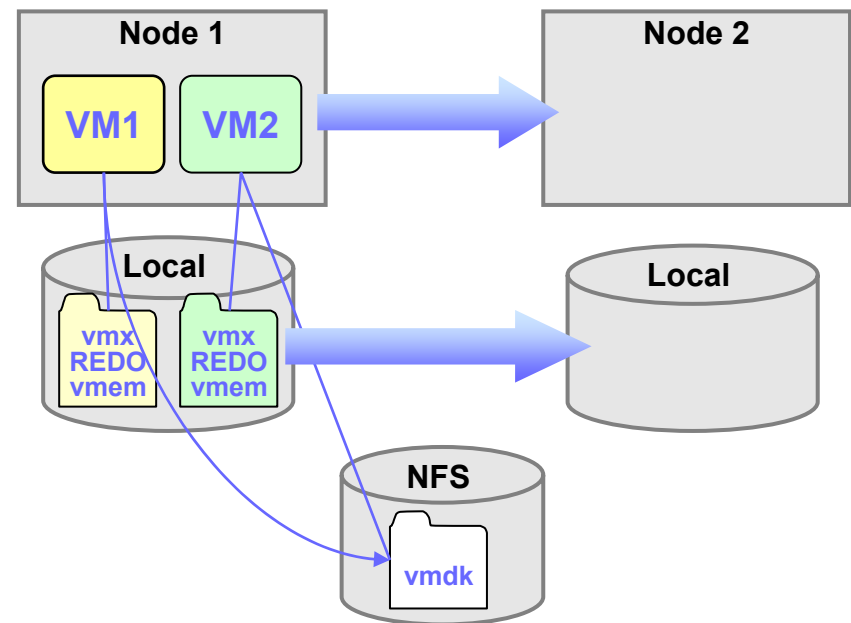
VM Migration based Cluster Vacating

- Example



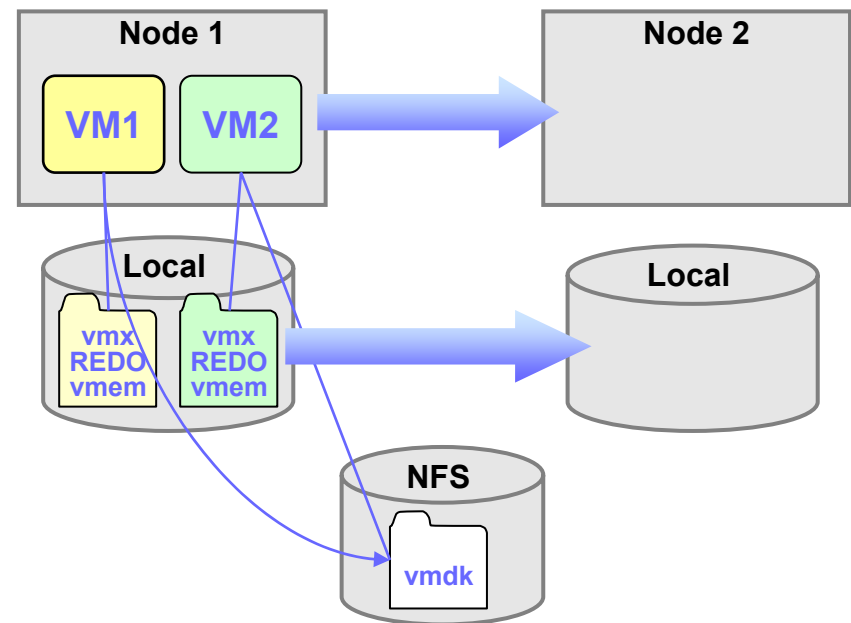
Experimental Setup

- Physical cluster
 - Each node has four CPUs (2.33GHz), 4GB memory
 - Gigabit Ethernet
- Virtual machines
 - VMware Server based
 - VM disk
 - Shared read-only image on NFS (.vmdk)
 - Independent changes (redo logs) on local disks (.REDO)
 - VM memory
 - Mapped to file (.vmem)
 - Stored on local disks



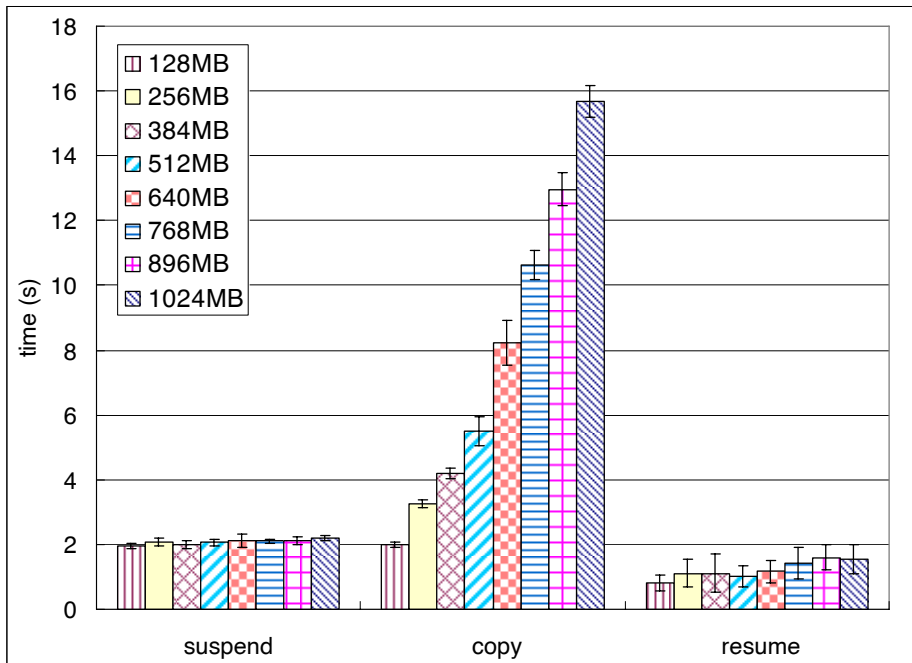
Experimental Setup

- Migration strategy
 - Suspend
 - Synchronizes memory state to file
 - Copy
 - Uses FTP to transfer memory state file, disk redo files
 - Maximum throughput is about 100MB/s
 - Resume
 - Restores memory state from file in foreground
 - Not based on VMware solutions (VMotion)

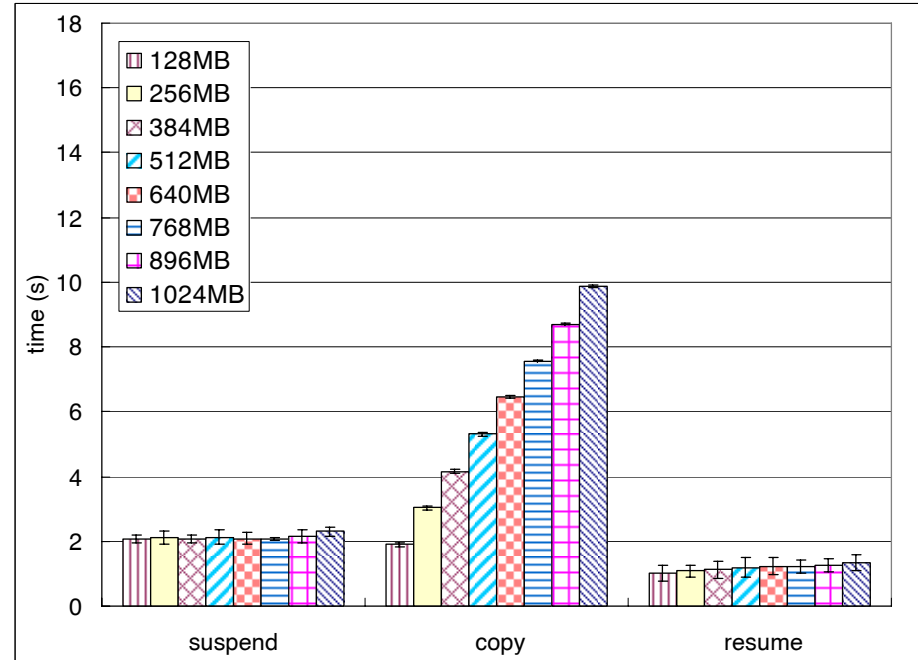


Migration Overhead of a Single VM

- Different memory sizes
 - Suspend and resume phases are fast
 - Copy time grows as memory size increases
- Different methods to store the migrated states
 - RAMFS removes the impact of disk I/Os to the migration process



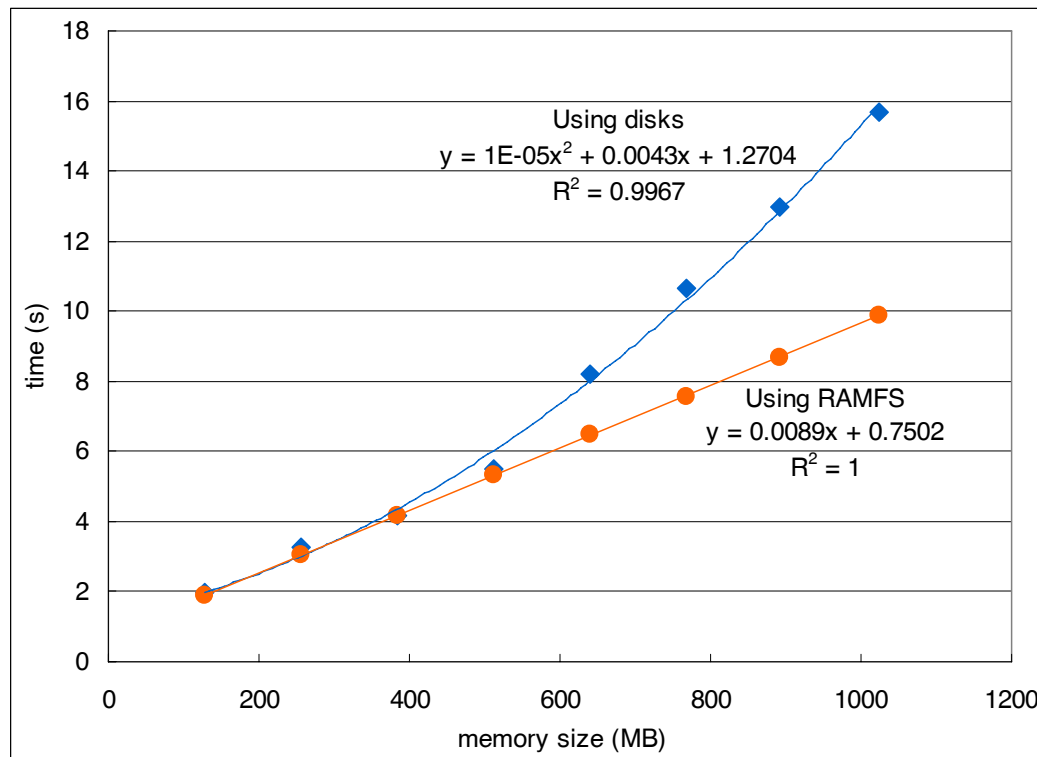
Using disks on the destination host to store migrated VM state files



Using RAMFS on the destination host to store migrated VM state files

Modeling Copy Phase

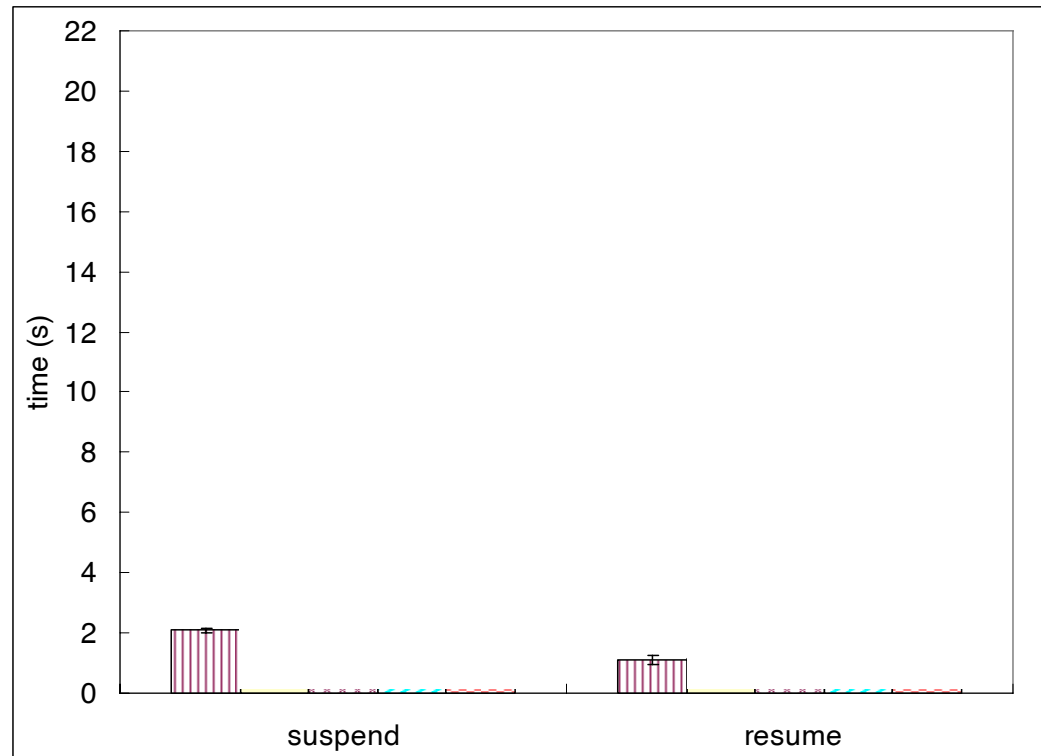
- Using regression methods
 - Polynomial for using disk, linear for using RAMFS
- RAMFS setup is used for all the following experiments



Using regression to model the copy phase

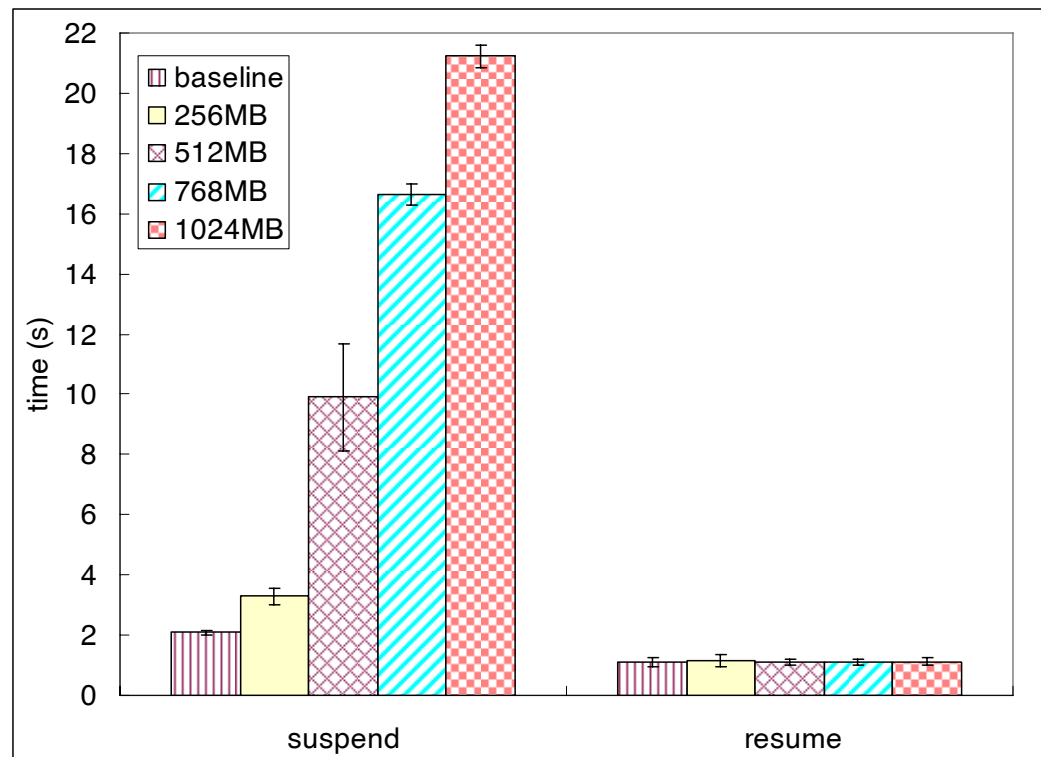
Modeling Suspend and Resume Phases

- Resume
 - Typically short: memory state is buffered after the copy phase
- Suspend
 - Typically short: memory state is frequently synchronized to file



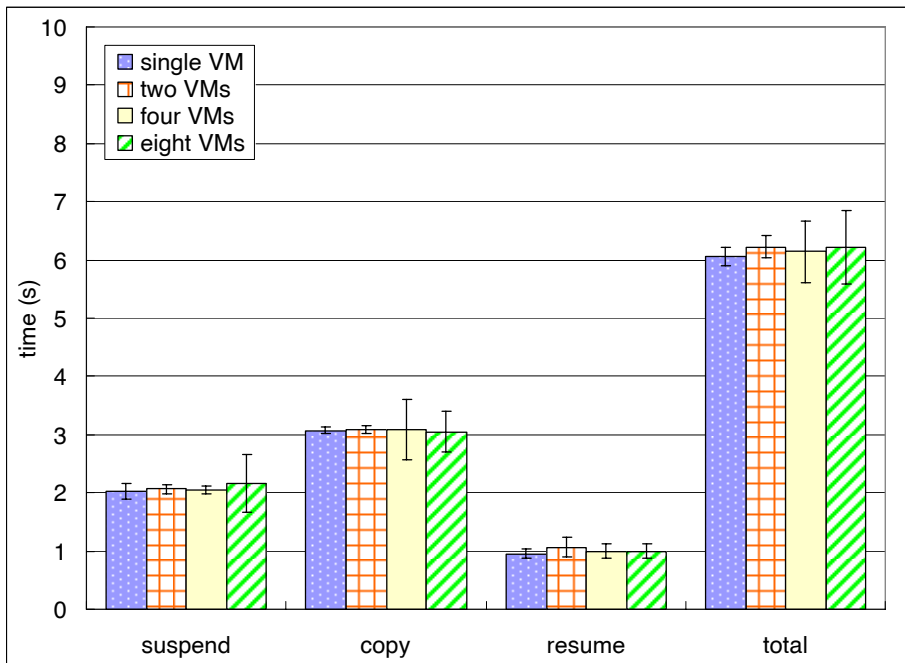
Modeling the Suspend and Resume Phases

- Running memory-intensive workload in the migrated VM
 - A “rogue” program that continuously modifies memory
 - Suspend time increases as more synchronization is needed

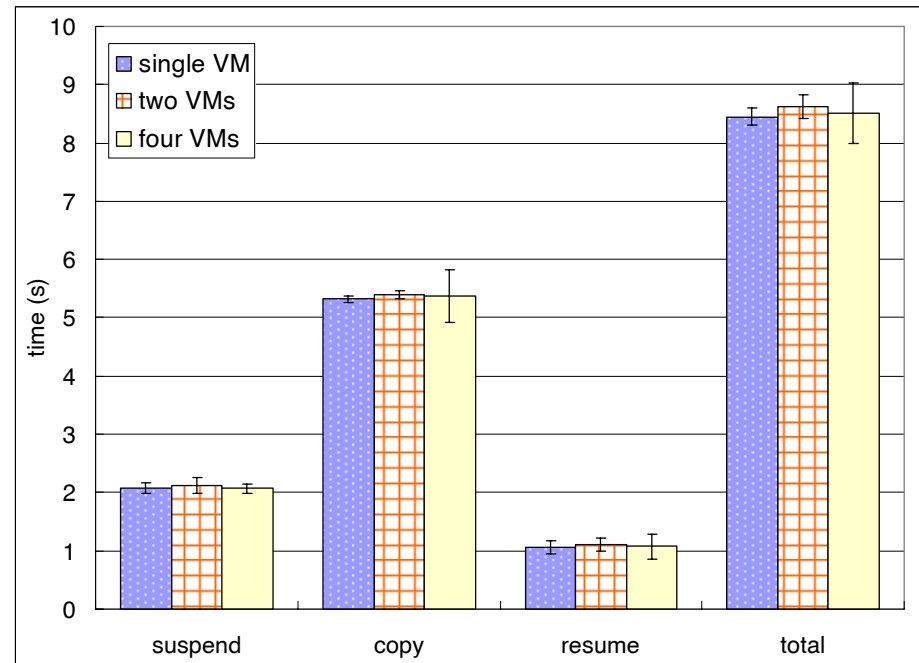


Migrating a Sequence of VMs

- Per-VM migration time is consistent regardless of the number of sequentially migrated VMs



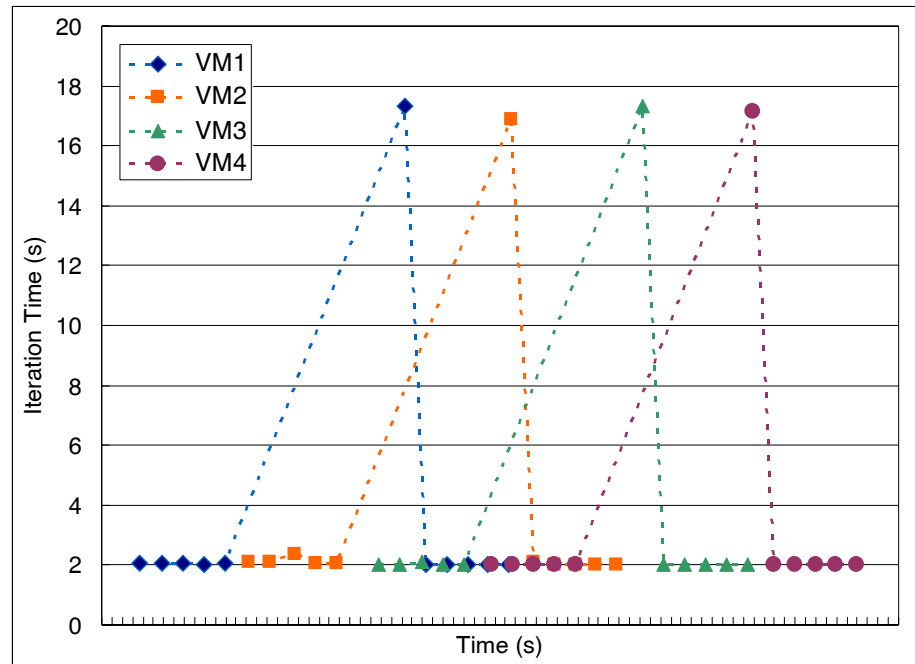
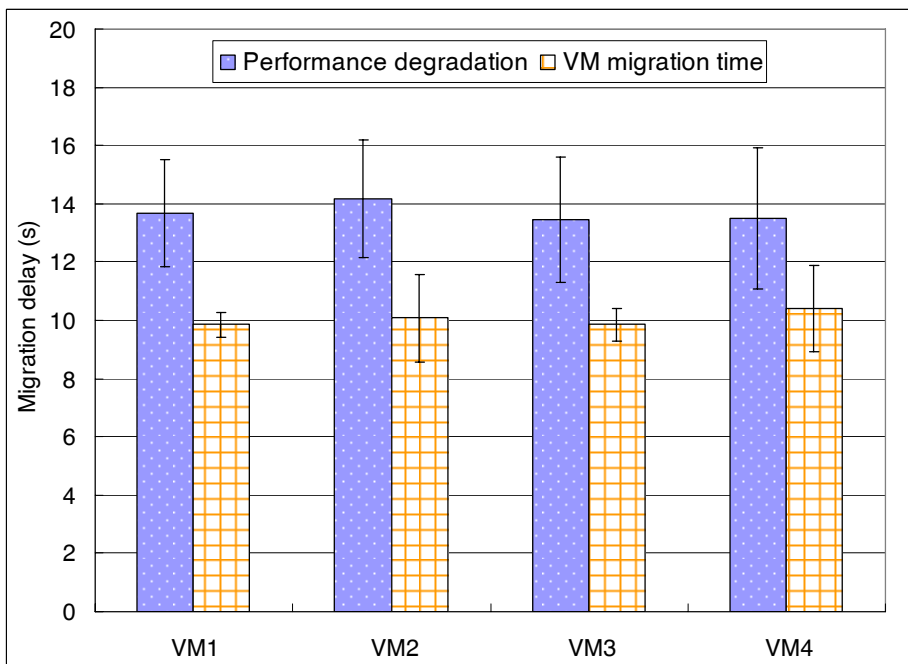
Per-VM migration time when migrating a sequence of 256MB-RAM VMs



Per-VM migration time when migrating a sequence of 512MB-RAM VMs

Migrating VMs with CPU-intensive Workload

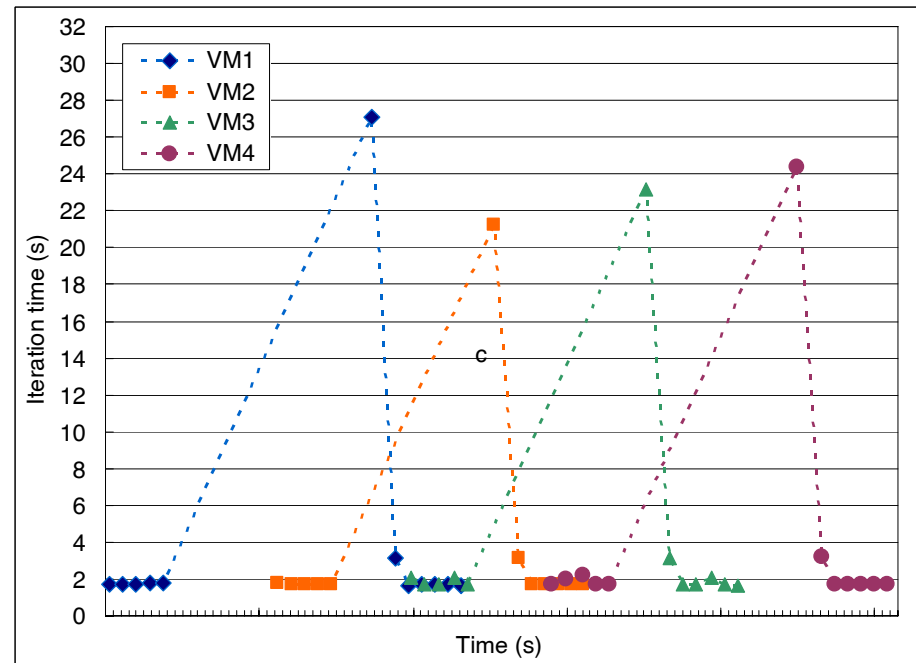
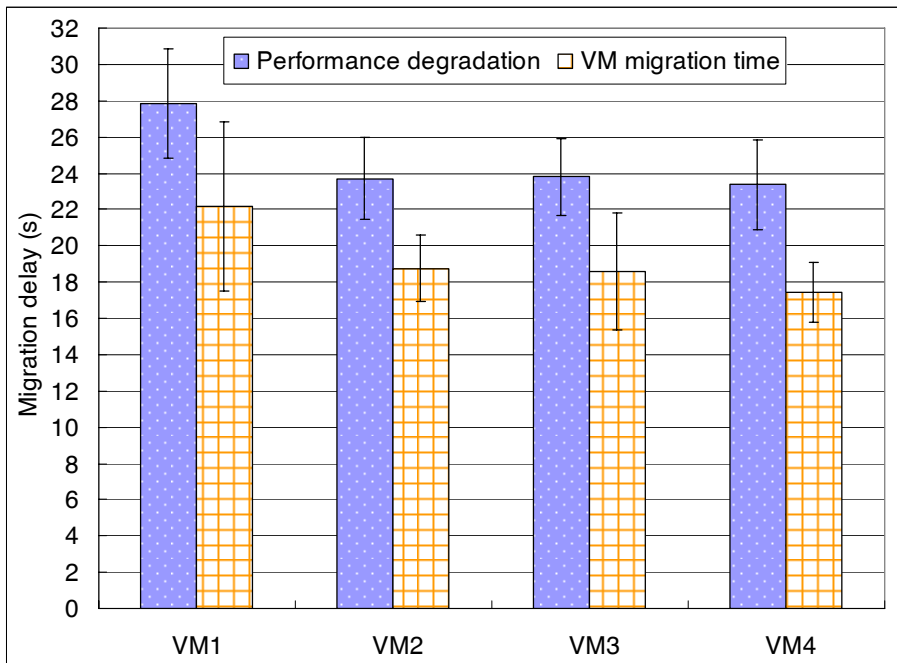
- CPU-intensive workload
 - Runs iteratively, consumes 100% of CPU
- Performance degradation
 - = Impacted iteration time – Regular iteration time
- Takes longer for applications in VMs to recover to full performance



Migrating four VMs in sequence; each has 512MB-RAM and runs a CPU-intensive benchmark

Migrating VMs with Memory-intensive Workload

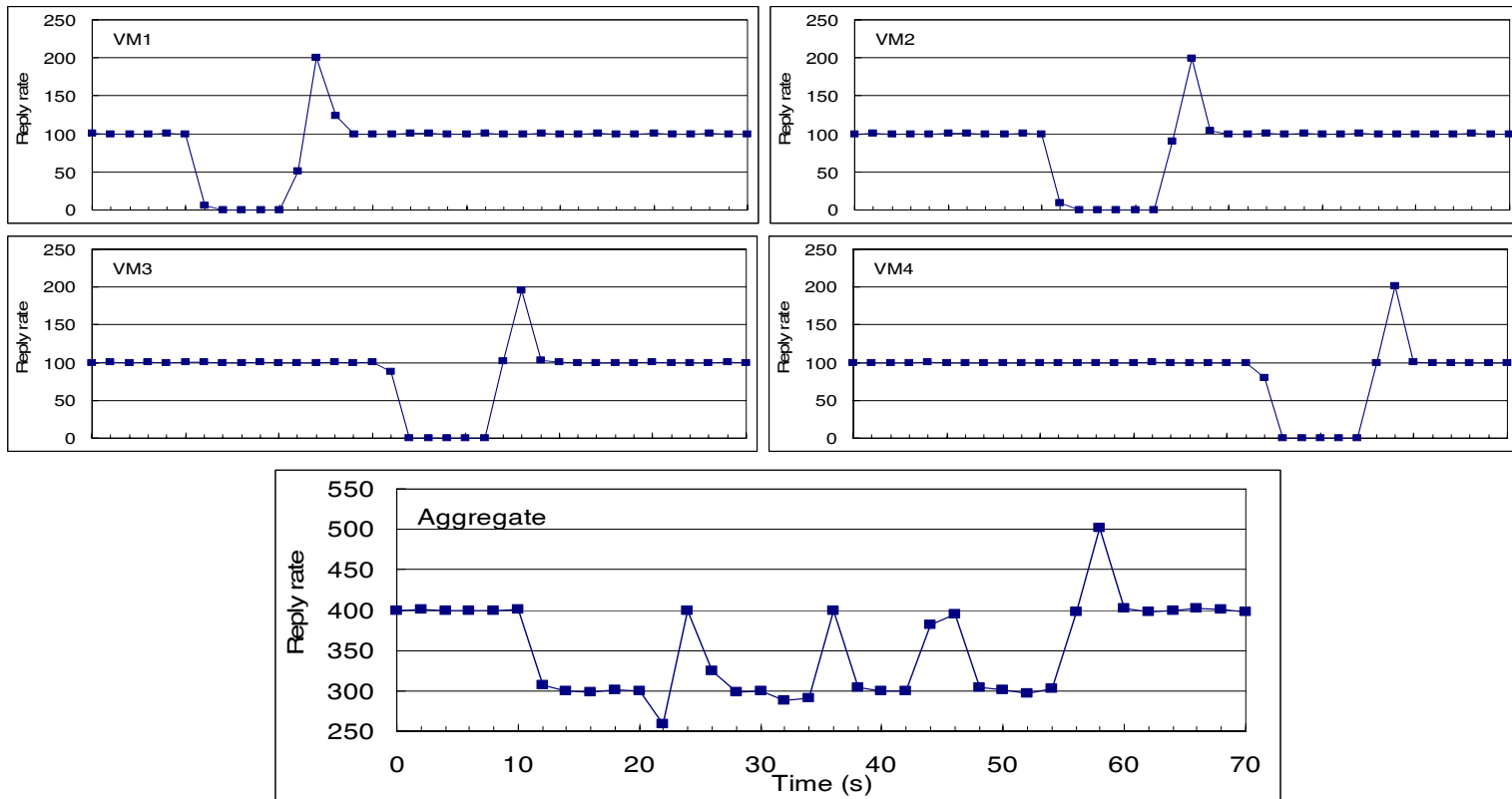
- Memory-intensive workload
 - Runs iteratively, touches 100% of memory
- Performance degradation
 - = Time of 2 impacted iterations – Regular iteration time * 2
- Migration is a more memory intensive process



Migrating four VMs in sequence; each has 512MB-RAM and runs a memory-intensive benchmark

Migrating VMs with Web Workloads

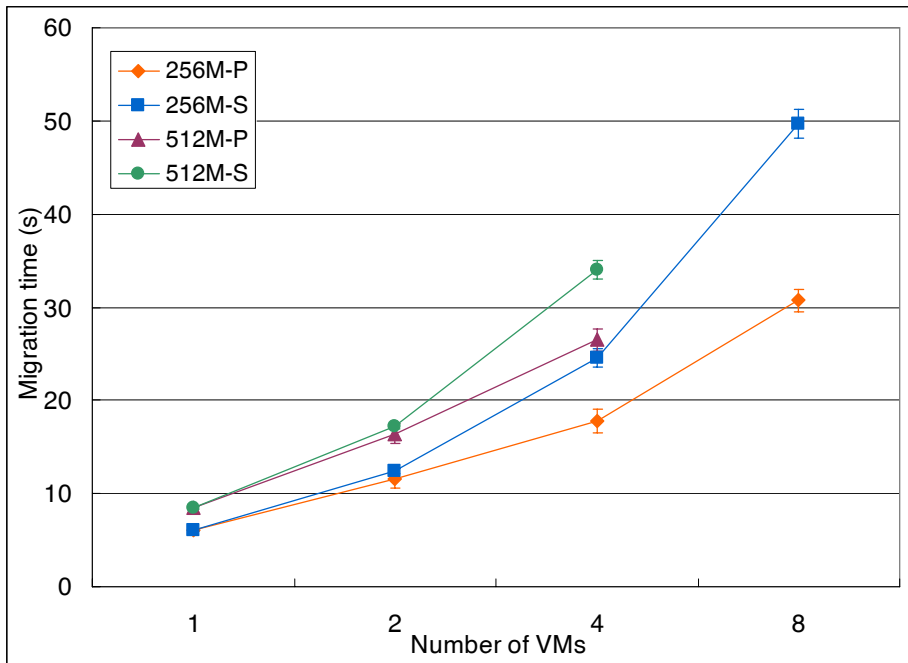
- Apache Web server
 - Serves constant-rate requests from outside clients (100 request/s each)
 - Takes longer for the throughputs to recover



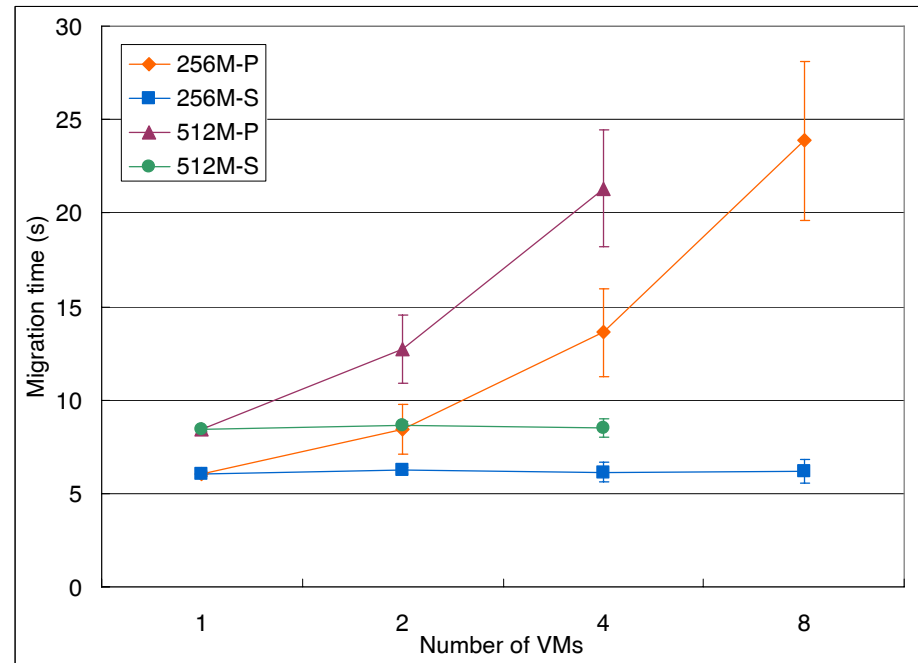
Migrating four VMs in sequence; each has 512MB-RAM and runs a Web server

Migrating VMs in Parallel

- Parallel migration has shorter total migration time
 - Faster vacating of cluster resources
- Sequential migration has shorter per-VM migration time
 - Less impact on the applications in the VMs



Total migration time



Per-VM migration time

Related Work

- VM-based resource management
 - In-VIGO
 - Virtual workspaces
 - Virtuoso
 - Shirako
- VM migration based resource reallocation
 - VIOLIN
 - Sandpiper
- VM migration optimizations
 - VMware VMotion
 - Xen live migration

Summary

- Conclusions:

- It is feasible to predict the migration time of VMs with different configurations and different numbers
- It takes longer for a VM's application to recover than the VM migration time
- Sequential migration has less impact on VMs' applications
- Parallel migration is faster for migrating multiple VMs

- Future work:

- Generalizes and validates the model across different physical platforms, with different VM technologies
- Considers modeling of live migrations

Acknowledgement

- DDDASBMI project team
 - <http://www.acis.ufl.edu/dddasbmi>
- Sponsorship from NSF
- Publication approval from VMware

- **Questions?**
 - ming@acis.ufl.edu
 - <http://www.acis.ufl.edu/~ming>