



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

---

Anusha Karur, Manar Alqarni, Muhanad Alghamdi

CSC541

Professor Tang

# Overview

---



- Introduction.
- The VM-based Resource Reservation.
- Experimental analysis.
  - Setup.
  - Migrating a Single VM.
  - Migrating a Sequence of VMs
  - Migrating Multiple VMs in Parallel
- Conclusions and future work.

# Introduction

---



- Virtual Machines (VM) have become very important.
- Modern virtual machine technologies allow a single physical server to be carved into multiple virtual resource containers, each delivering a powerful, secure, and isolated execution environment for applications.
- In addition to that VM environments can be customized to encapsulate the entire software and hardware platform needed by the applications.

# The VM-based Resource Reservation

---



- The reservations of CPU, memory and network resources for individual VM instances, as well as for VM clusters.
  
- The main goal is to enable an application to request the creation of virtual machines and clusters based on high-level specifications of both the VMs' environments and its desired QoS.

# VM-based Resource Reservation CONT.

---



- Allocating dedicated resources can lead to **resource inefficiencies**; VMs allow time-sharing of resources and can lead to **better resource utilization**.
- To implement such policy, all hosted VMs from the cluster to be reserved need to be vacating - through suspension, or if other resources are available, through migration.
- This preparation should be done in time to meet the reserved schedule, but cannot be too early and waste the resources that are useful to serve other tasks.

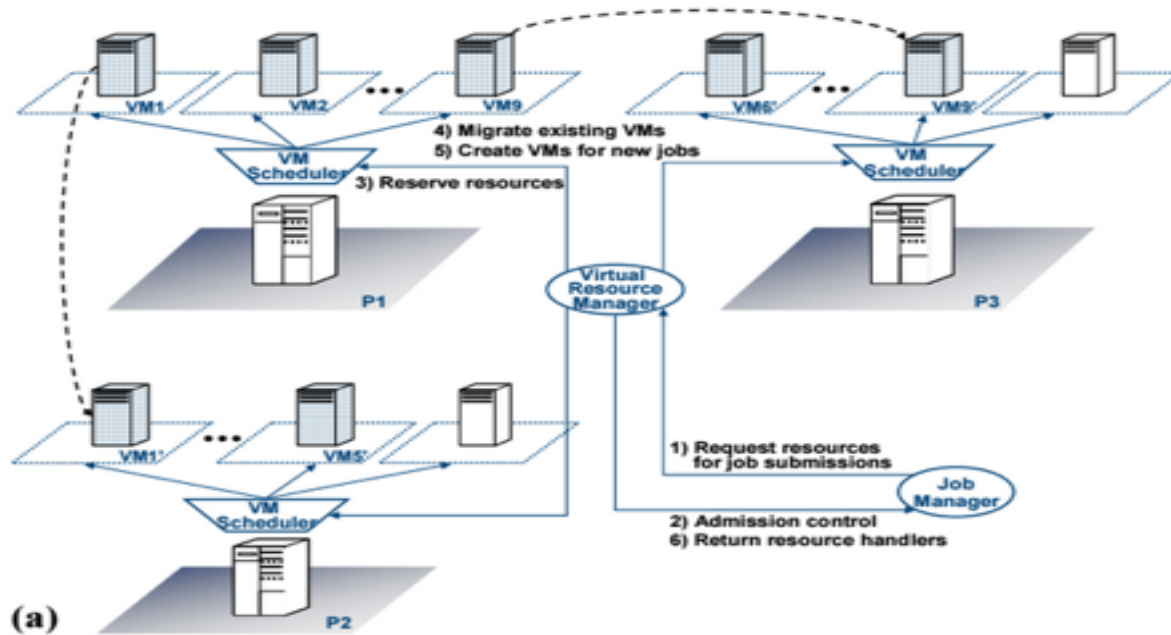
# VM-based Resource Reservation CONT.

---



- It provides a centralized management for the virtualized resources that are distributed across the physical hosts.
- It exposes an abstract interface to the resource clients and serves their resource requests.
- The clients can be unaware of that the resources are virtualized.
- They only need to specify the types and quantities of resources that are necessary for the scheduled jobs.
- It also supports the request of preparing a desired software environment on the reserved resources.

# VM-BASED RESOURCE RESERVATION



# Virtual Machine Scheduler

---



- A VM scheduler is on every physical server to manage the VMs that are hosted on it.
- It provides a unified interface for resource reservation, which allows the virtual resource manager to make resource reservations without knowing their underlying control mechanisms.
- **The virtual resource manager** only needs to specify the quantity and schedule of the necessary resources, and **the VM schedulers** are responsible to carry out the resource reservation and VM creations using the mechanisms provided by the VM software.



# VM Migration Based Resource Reservation

---



- The requested resources must be prepared in time to satisfy the requested schedule, but they should not be allocated too early and waste the resources that can potentially serve other tasks.
- The costs from VM-based resource reservation include both the overhead for migrating existing VMs and making the resources available for the new jobs, and the time needed to create and configure the desired environment with VMs.

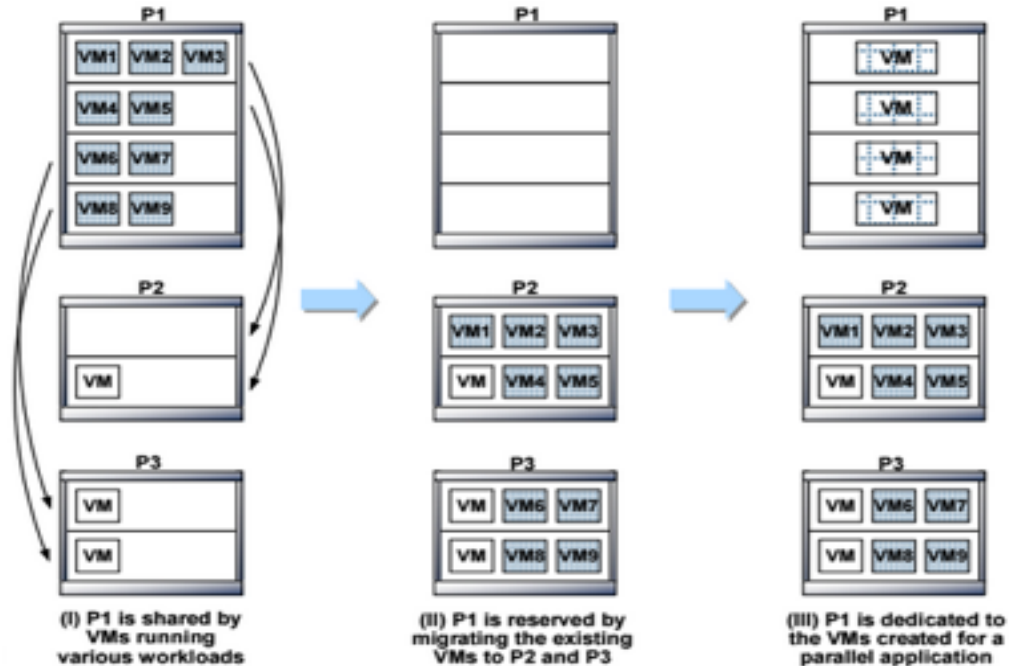
# VM Migration Based Resource Reservation

---



- The goal is to support parallel processing using dedicated resources when such an experiment takes place, while also allowing a cluster resource to be utilized by other workloads when such experiments are not taking place.
- The virtual resource manager needs to vacate a cluster for the parallel application, and move all the existing VMs, which are running various other workloads, to other hosts.

# VM Migration Based Resource Reservation



# EXPERIMENTAL ANALYSIS

---



- Setup
  - The VMs are hosted on a cluster of physical servers.
  - Each physical node has two dual-core processors and (4GB)memory, <--> Gigabit Ethernet.
  - Limitation of the physical nodes, the VM memory size is 1GB.
  - The VMs' virtual disks share the same read-only image, stored on a storage server and accessed through NFS.
  - Changes to the virtual disks from the VM executions are stored in the form of redo log files.
  - Running VMs' memory state is also mapped to files, which capture the VMs' memory snapshots when they are suspended.
  - In the absence of shared storage, disk redo logs and memory state files are stored on the local file system of the hosts.

# VM Migration Process

---



- 3 phases
  - Suspend
    - VM is suspended on the origin host, and its memory is captured to the memory state file
  - Copy
    - VM's configuration, memory state and disk redo files are transferred to the destination host through FTP
  - Resume
    - VM restores its memory state from the snapshot and then resumes its execution

# Migrating a Single VM

---



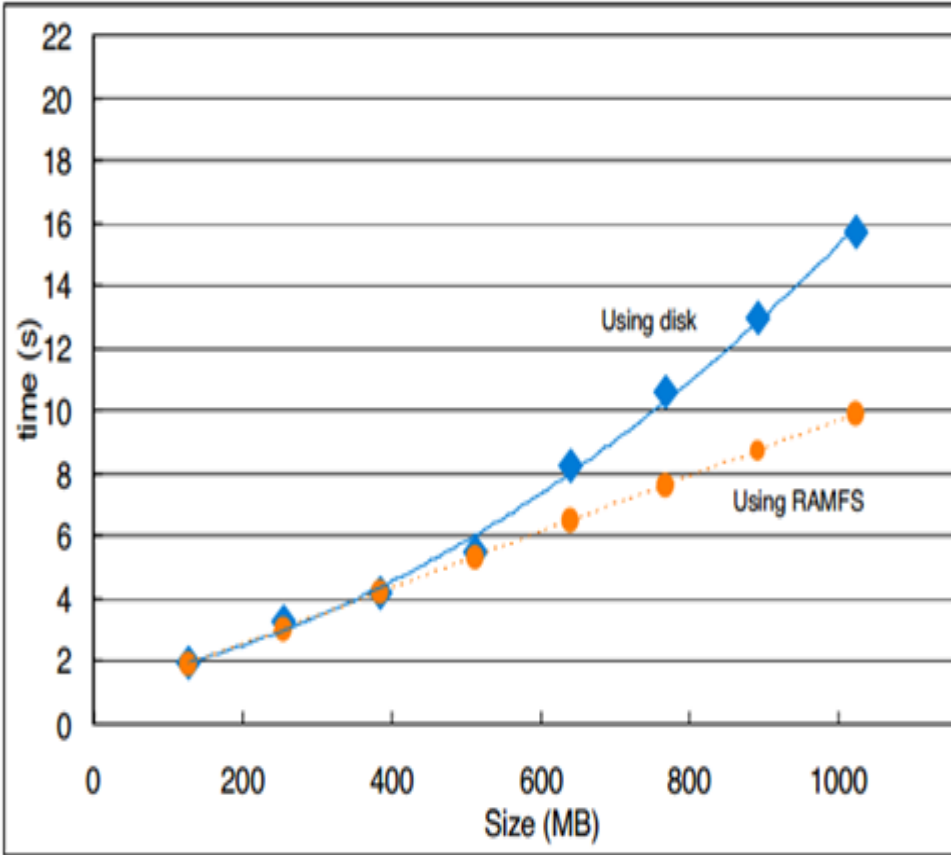
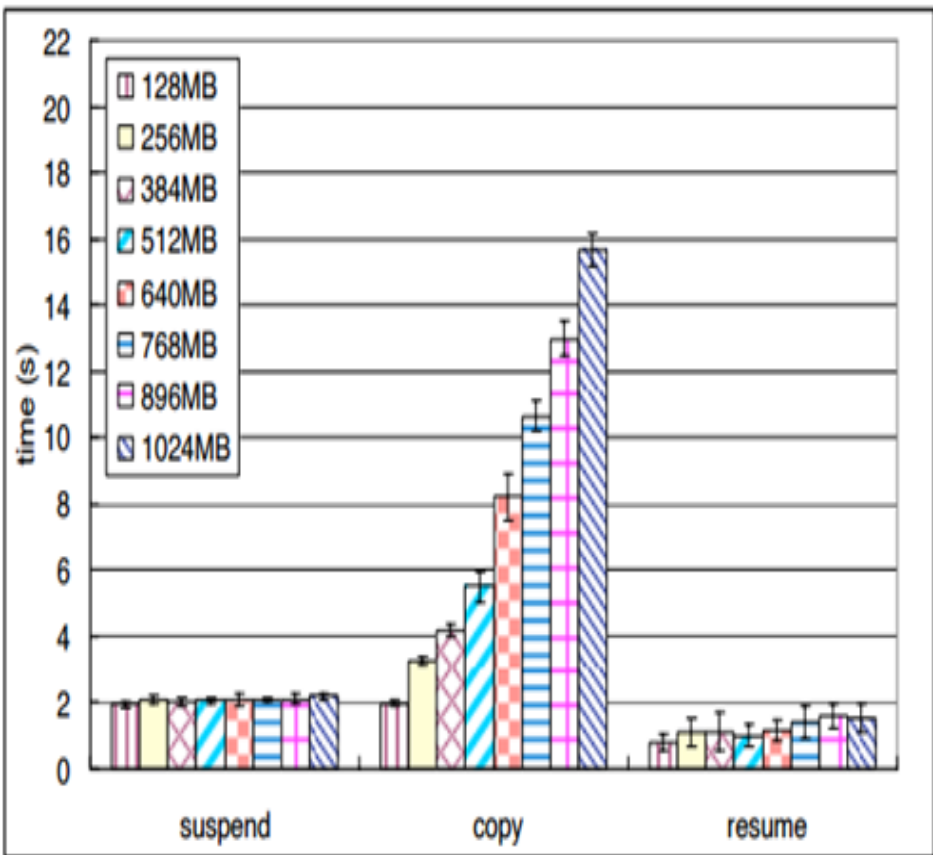
- It studies the 3 migration phases for a single VM, and analyzes its migration time with different VM configurations.
  - Experiments with Different Memory Sizes
  - Experiments with Memory-intensive Workloads

# Experiments with Different Memory Sizes

---



- VM's memory state file is major part that need be transferred
- The suspend and resume phases are relatively stable, & only increases slightly as the memory size increases,(time)
- More memory pages need be processed during these two phases.
- copy quickly grows and dominates the migration time for larger memory sizes.



**Figure 2:** The times needed for the three phases of migrating a single VM with different memory sizes. A local disk on the destination host was used to store the migrated VM states.

**Figure 3:** Using regression methods to model the relation between the time needed for the copy phase and the size of the VM memory.



# Regression Methods

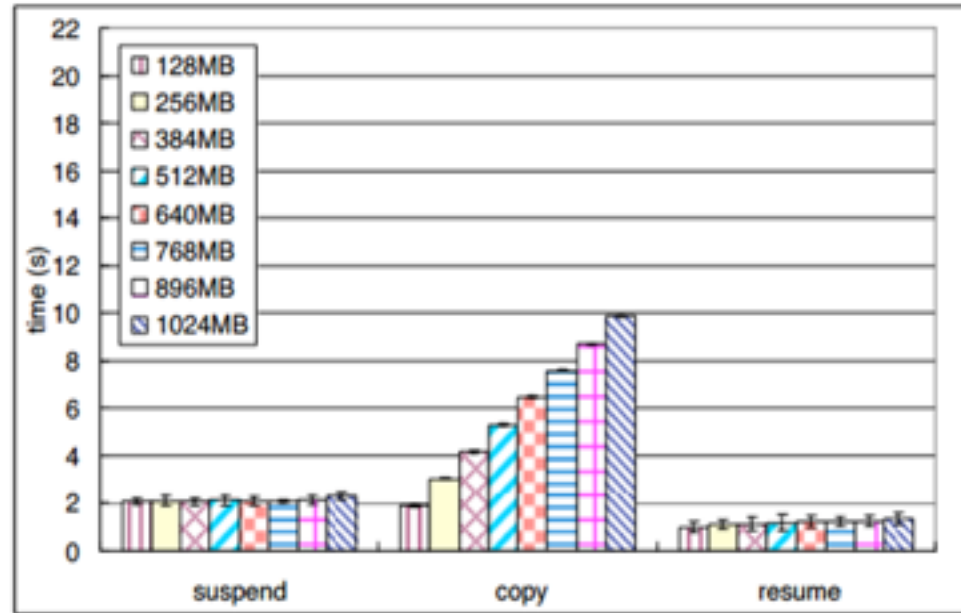
---



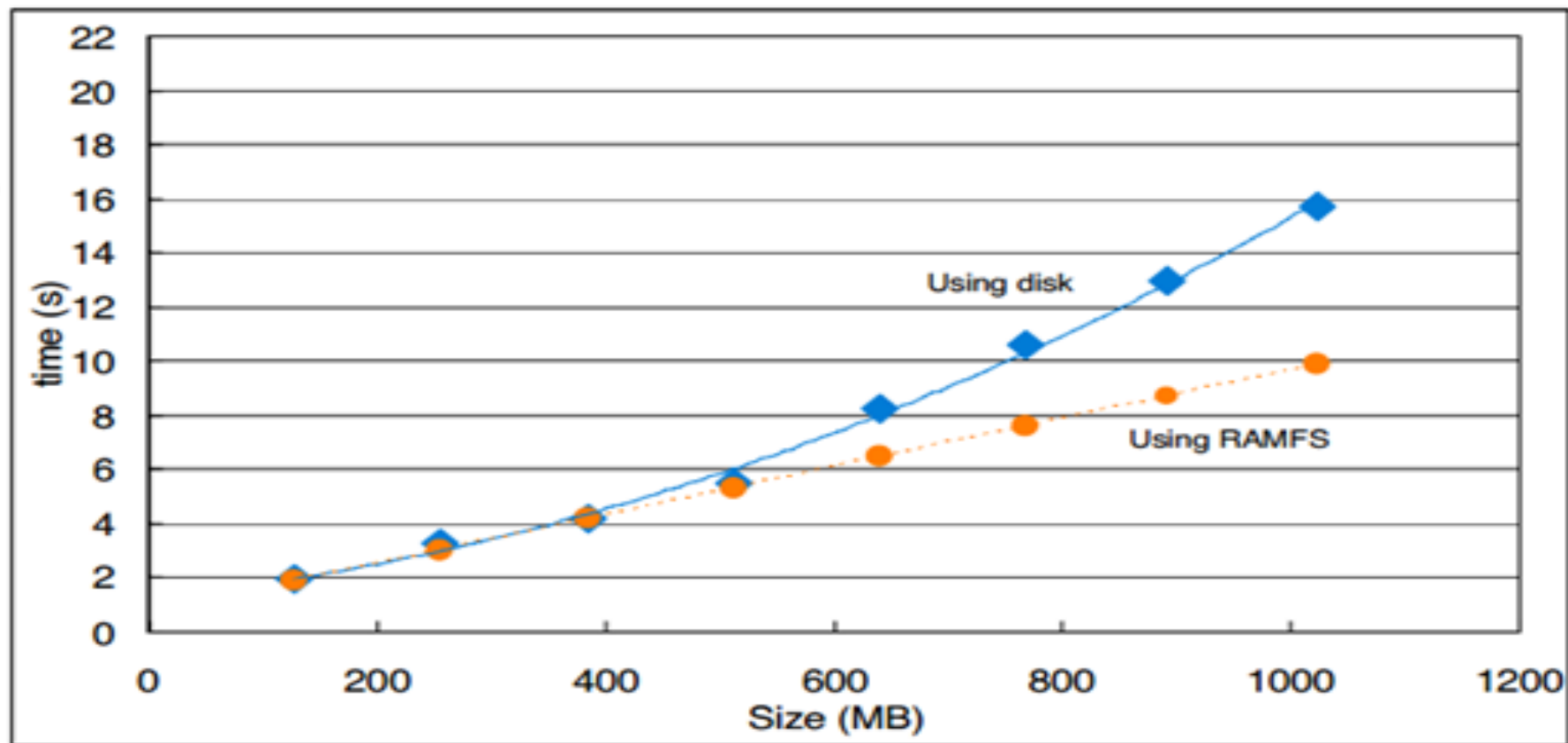
- Relationship between (time needed) copy phase and VM memory(size).
- Polynomial function can characterize this relationship best.
- Giving a VM migrating time and size, can predict other VMs migrating times
- Nonlinear model
- Small memory size, copy phase speed is limited by the network bandwidth (Gigabit/s);
- Greater memory sizes, large amount of dirty pages are buffered in memory during the copy phase,
- And the kernel forces to flush these data in foreground, which throttles the copy phase by the throughput of the disk (about 50MB/s).
- Kernel's I/O scheduling policy decides when and how to flush the data

# RAM-based file system (RAMFS)

- Separate the impact of the disk I/O on the migration process
- As the VM's memory state is not backed by stable storage,
- A viable recovery scheme is necessary for the VM in case of a crash.



**Figure 4: The times needed for the three phases of migrating a single VM with different memory sizes. A RAMFS on the destination host was used to store the migrated VM states.**



**Figure 3: Using regression methods to model the relationship between the time needed for the copy phase and the size of the VM memory.**

## Experiments with Different Memory Sizes: Conclusion

---



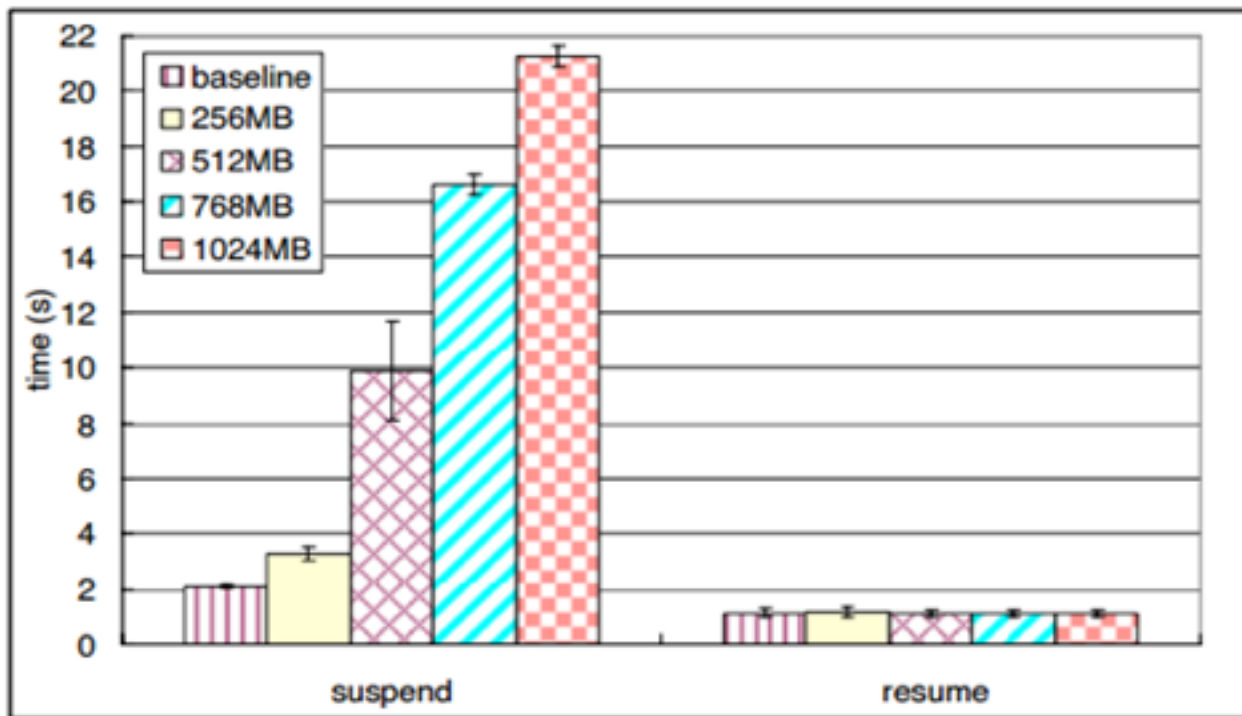
- VM's migration time for a particular memory size,
- Easy to predict the time for migrating a VM with other memory sizes.
- Applicable if the size of other VM state, e.g., the disk redo files, needs to be considered.
- Times for suspend and resume phases are small.
- The resume phase is very fast (copy phase is done),
  - VM's memory state is already buffered in memory, It doesn't require additional disk reads.
- The suspend time is also usually short,
  - running VM's memory is frequently synchronized with its memory state file, and It also doesn't involve many disk writes.

# Experiments with Memory-intensive Workloads

---



- If a considerable amount of memory pages need be synchronized (VM is suspended), It take longer.
- Program which continuously touches a given amount of memory was intentionally started in a VM that was under migration
- The resume and suspend times, with different amounts of modified memory, are compared to the baseline performance when this program is not used



**Figure 5: The times needed for the suspend and resume phases of migrating a VM with different amounts of continuously modified memory.**

# Migrating a Sequence of VMs

---



- Predict the total migration time based on the time for a single VM?
  - Experiments with Different Number of VMs
  - numbers of VMs are migrated consecutively.
  - memory size of 256MB.
  - memory Size of 512MB.
  - Using A local disk on the destination host.
  - Using A RAMFS on the destination host.

# Migrating a Sequence of VMs

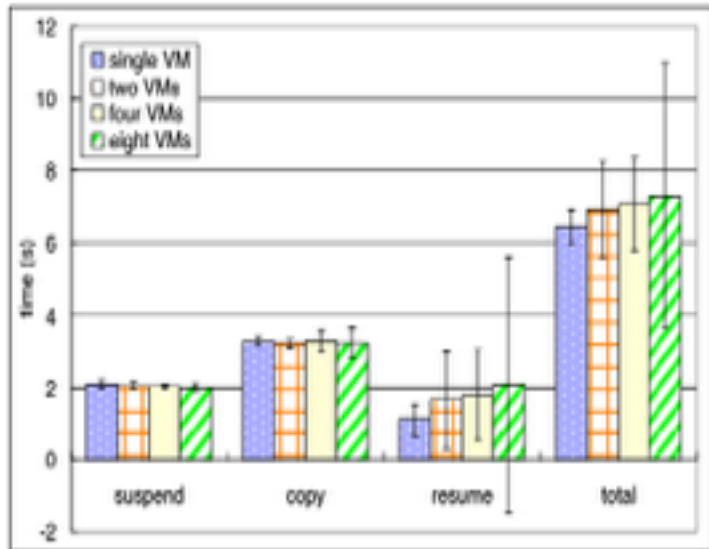


Figure 6: The per-VM migration times when different numbers of VMs, each with a 256MB memory, were migrated in sequence. A local disk on the destination host is used to store the migrated VM state.

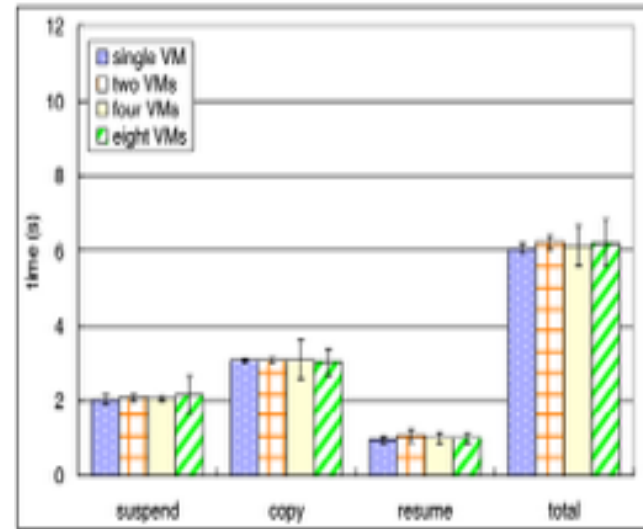


Figure 7: The per-VM migration times when different numbers of VMs, each with a 256MB memory, were migrated in sequence. A RAMFS on the destination host was used to store the migrated VM state.



# Migrating a Sequence of VMs

---



-The Suspend and copy are not affected.

-The Resume becomes slower

This is also because of the aforementioned flushing of dirty data from copying a VM's state files.

- It affects the VM migration and the following VMs' migrations.
- Such a problem as in the previous experiment was solved using RAMFS.
  - migration time becomes very consistent regardless of the number of migrated VMs.
  - A very similar problem with the 512MB memory VMs.
  - RAMFS setup was used for all the following experiments.

# Migrating a Sequence of VMs

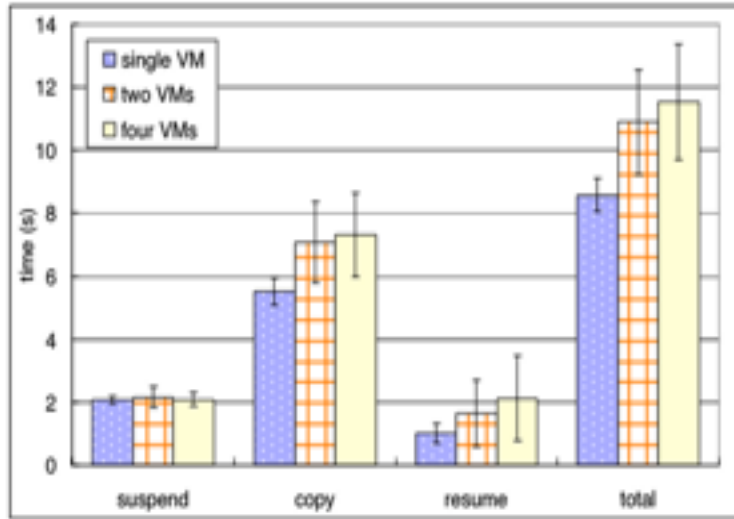


Figure 8: The per-VM migration times when different numbers of VMs, each with a 512MB memory, were migrated in sequence. A local disk on the destination host was used to store the migrated VM state.

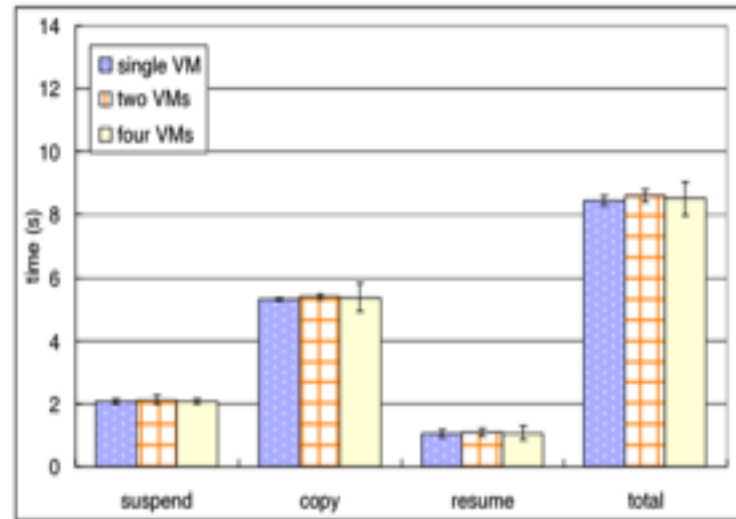


Figure 9: The per-VM migration times when different numbers of VMs, each with a 512MB memory, were migrated in sequence. A RAMFS on the destination host was used to store the migrated VM state.

# Experiments with Different Workloads

---



- study the interference between the workloads running inside of the VMs and the migration process.
  - 1- CPU-intensive workload
    - CPU-intensive workload, which is adapted from the Freebench's Distray benchmark
    - It runs iteratively, where each cycle takes exactly 2 seconds (100% of CPU).
      - The performance degradation = the runtime of the iteration being affected by the migration,  
minus the regular iteration time.
  - 2- memory-intensive.
    - The second workload uses a memory-intensive program that is similar to the one used **Experiments with Memory-intensive Workloads**.
    - It runs iteratively, (Modifies the whole memory and sleeps 1 second).
    - The average iteration time is 1.75 second.
    - the performance degradation is the sum of these two iterations' times, minus twice of the regular iteration time.

# Experiments with Different Workloads

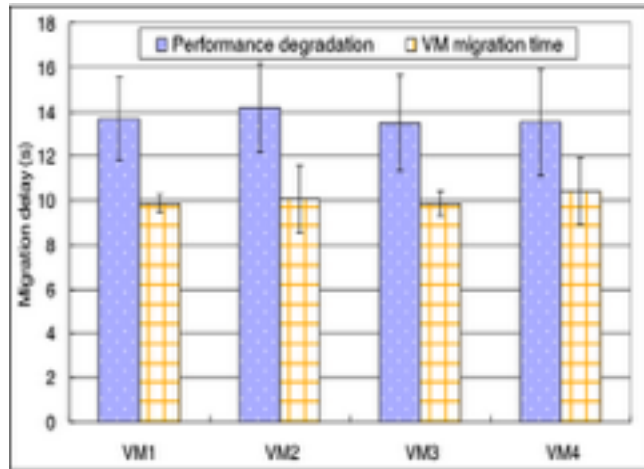


Figure 10: The migration time and performance degradation when four VMs were migrated in sequence, each with 512MB memory and a CPU-intensive benchmark running inside.

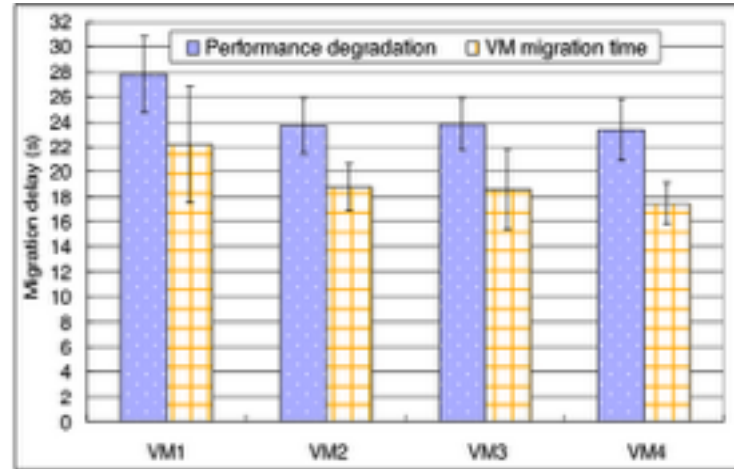


Figure 12: The migration time and performance degradation when four VMs were migrated in sequence, each with 512MB memory and a memory-intensive benchmark running inside.

# Experiments with Different Workloads

---



- The performance degradation time (in CPU case) is longer than the VM migration time by around 2 seconds , which means that it takes additional time for the benchmark to recover to its full performance after the its VM is migrated.
- The performance degradation(Memory case) experienced by the program is longer than the VM migration time by around 5 seconds.
- The migration has a greater impact on memory-intensive program than the CPU-intensive workload.
- The first VM's migration takes more time than the other ones.
- This is because suspending the VMs for this memory-intensive program involves a considerable amount of disk writes, and the first VM's migration has an additional start-up overhead from initiating the I/Os.



# Migrating Multiple VMs in Parallel

migrating multiple VMs in parallel and studies its benefits and costs in comparison to sequential migration.

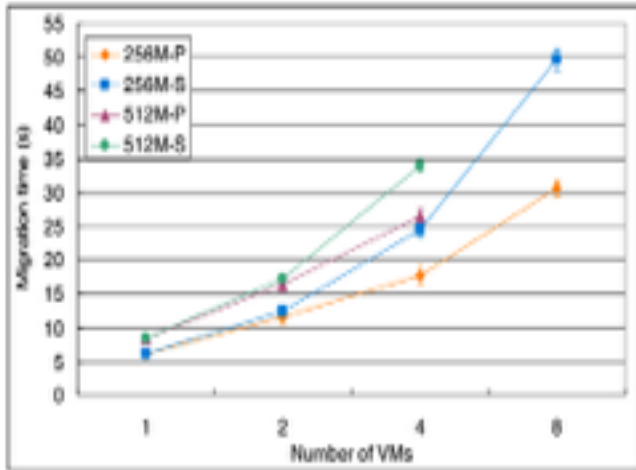


Figure 15: The comparison of total migration times for a sequence of VMs between parallel and sequential migration.

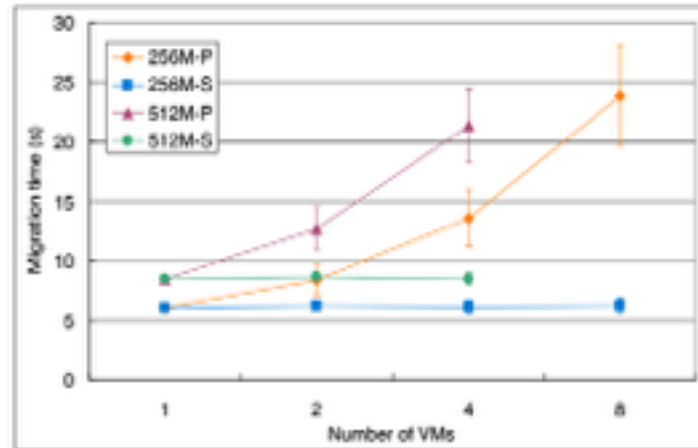


Figure 16: The comparison of per-VM migration times for a sequence of VMs between parallel and sequential migration.

# The comparison results

---



- The speedup is 1.4 times for 4 VMs and 1.6 times for 8 VMs.
  - For VMs with 512MB memory size, the speedup is 1.3 times for 4 VMs, which is less than that of the smaller VMs.
  - the parallel migration has a much worse per-VM migration time than the sequential one. For VMs with 256MB memory, the slowdown is 2.2 times for 4 VMs and 3.8 times for 8 VMs. For VMs
  - with 512MB memory size, the slowdown is 1.5 times for 2VMs and 2.5 times for 4 VMs.
- 
- there is a tradeoff between increasing the speed of migrating multiple VMs and reducing the impact on the performance of the VMs. These different migration strategies can be selected based on the optimization needs.

# conclusion

---



- VM migration is key to realizing VM-based resource reservation.
- In order to make an efficient reservation, we need to understand the overhead.
- The paper went over different migration strategies (Single, sequential, and parallel).
- Some of the findings:
  - An accurate estimation of the migration time for a number of VMs is possible given the measurement of a single VM's migration time.
  - The performance degradation period caused by a VM's migration is relatively longer than the migration time.
  - Different migration strategies can be selected for different optimizations.
  - parallel migration can deliver better speed when migrating multiple VMs, and sequential migration can reduce the performance overhead for the applications that are running on the migrated VMs.





---

**Questions ?**