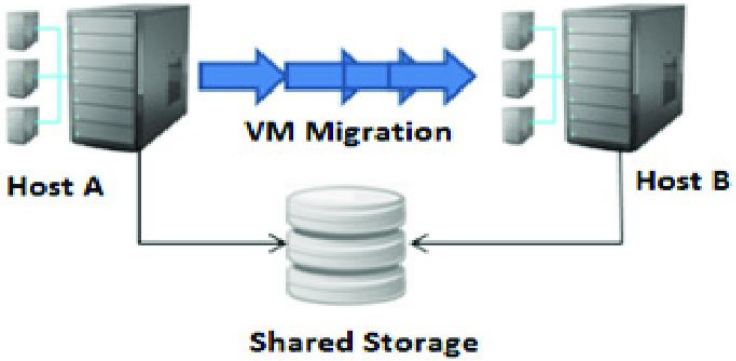
Performance Analysis of Live Migration using KVM



Debdoot Roy Chowdhury (23m0765)
Pranab Kumar Paul (23m0800)

Is Live Migration Effective? [Scope of this Project]

1. Is it effective than other prevalent methods? Why is it called Live?
2. By how much factor it reduces the downtime in compare to other methods?
3. Does Pre-Copy round affect VM’s normal performance?
4. How much iterations are needed to perform one such migration?
5. Does it provide the same throughput with heavy workloads?
6. Performance increase in compared Pure Stop and Copy

How to measure Downtime?

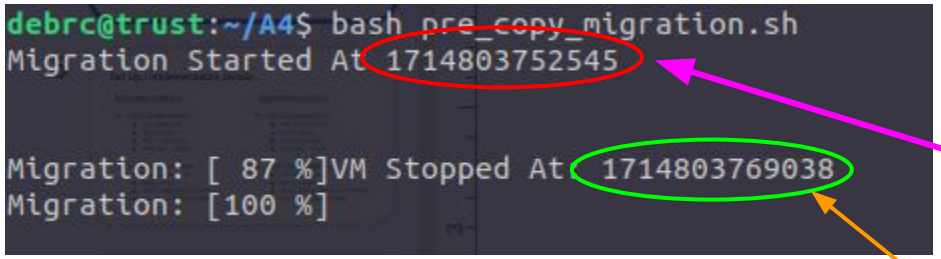
1. Ping to VM
    - Set up a Bridged Network for VM.
    - Prepare a Host which sends Ping continuously to the VM.
    - A noticeable time delay will be there when the VM undergoes downtime during Live Migration.

2. Application inside VM monitors UTC Timestamp
    - A bash script fetches current time of the system in infinite loop.
    - When VM experiences downtime it will not be able to fetch current time.
    - VM fetches current time again after turning on.

3. Using Virsh VM Status (Running/Paused/Shut Off)
    - *virsh domstate <domain_name>* returns the status of the VM in current host.
    - Downtime starts when the status of the VM in source host changes from running to paused/shutoff.
    - Downtime ends when the status of the VM in destination host changes to Running.

How to measure Migration Time?

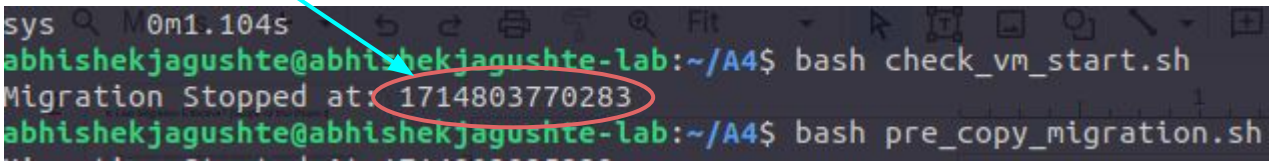
- Get start time when *virsh migrate* command is executed.
- Get finish time from the destination host when VM started running OR use the application inside the VM.



Migration Start Time

Downtime End Time

Downtime Start Time



Set Up / Implementation Details

Experiment Setup 1

- VM Configurations
  - OS: Debian 12
  - RAM - 2 GB
  - CPU - 1 Core(s)
  - Disk Size - 10GB
- Host Configurations
  - OS: Ubuntu 22.04
  - RAM - 8 GB
  - CPU - AMD Ryzen 5 (8 Cores)
  - Disk Size - 500 GB (SSD)

Experiment Setup 2

- VM Configurations
  - OS: Ubuntu 20.04
  - RAM - 4 GB
  - CPU - 2 Core(s)
  - Disk Size - 20 GB
- Host Configurations
  - OS: Ubuntu 22.04
  - RAM - 16 GB
  - CPU - Intel i5 13th Gen (16 Cores)
  - Disk Size - 500 GB (SSD)

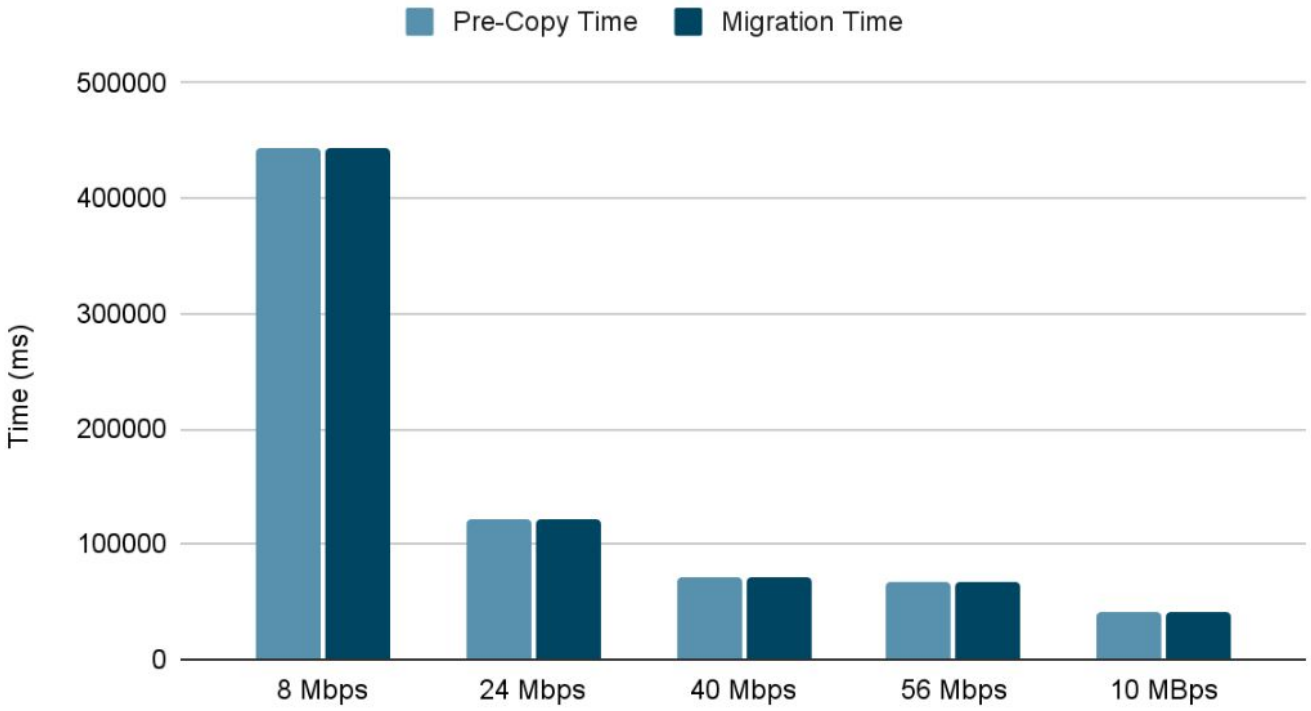
Set Up / Implementation Details

- Migrate Command
  virsh migrate --live --unsafe --verbose cs695 qemu+ssh://debrc@10.130.157.181/system

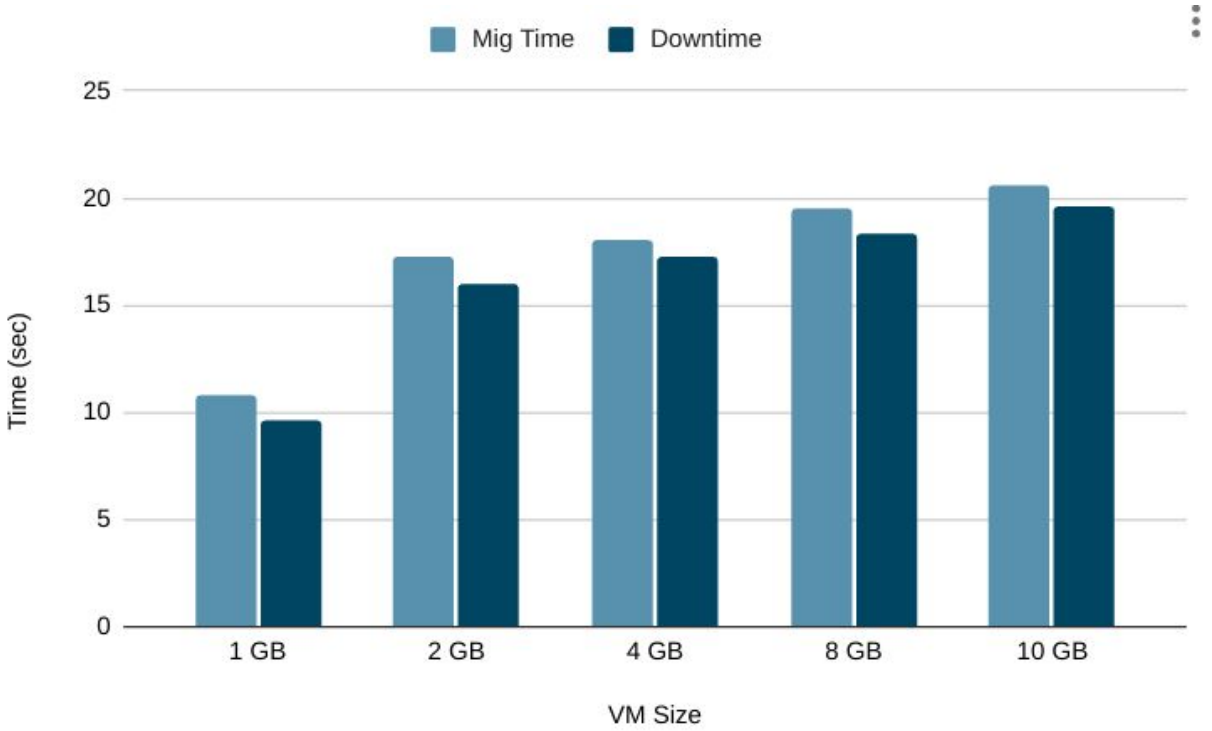
- Page Dirty Info
  virsh domjobinfo cs695

-

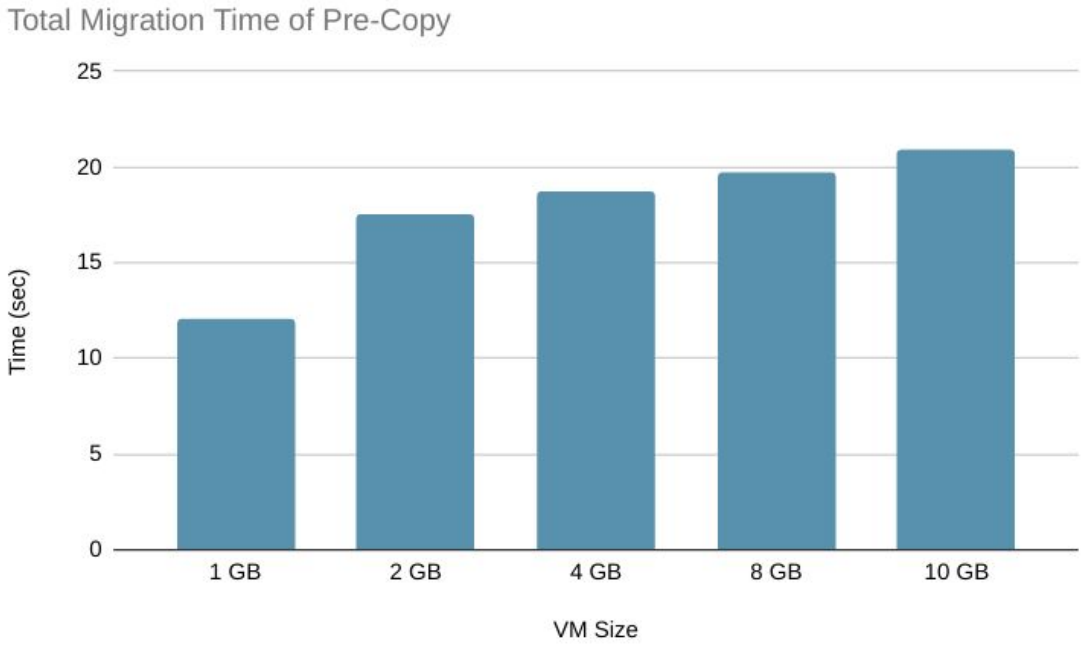
Analysis of Migration Time with Bandwidth



Stop and Copy Migration Time and Downtime

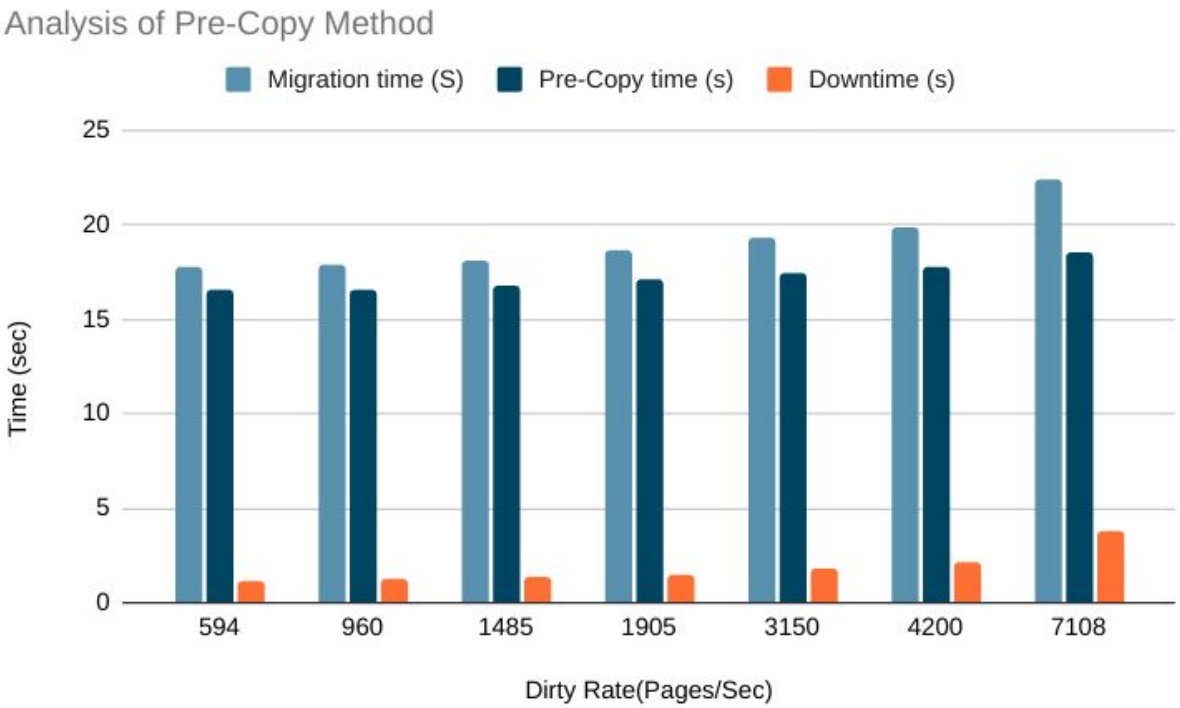


Migration Time of Pre-Copy

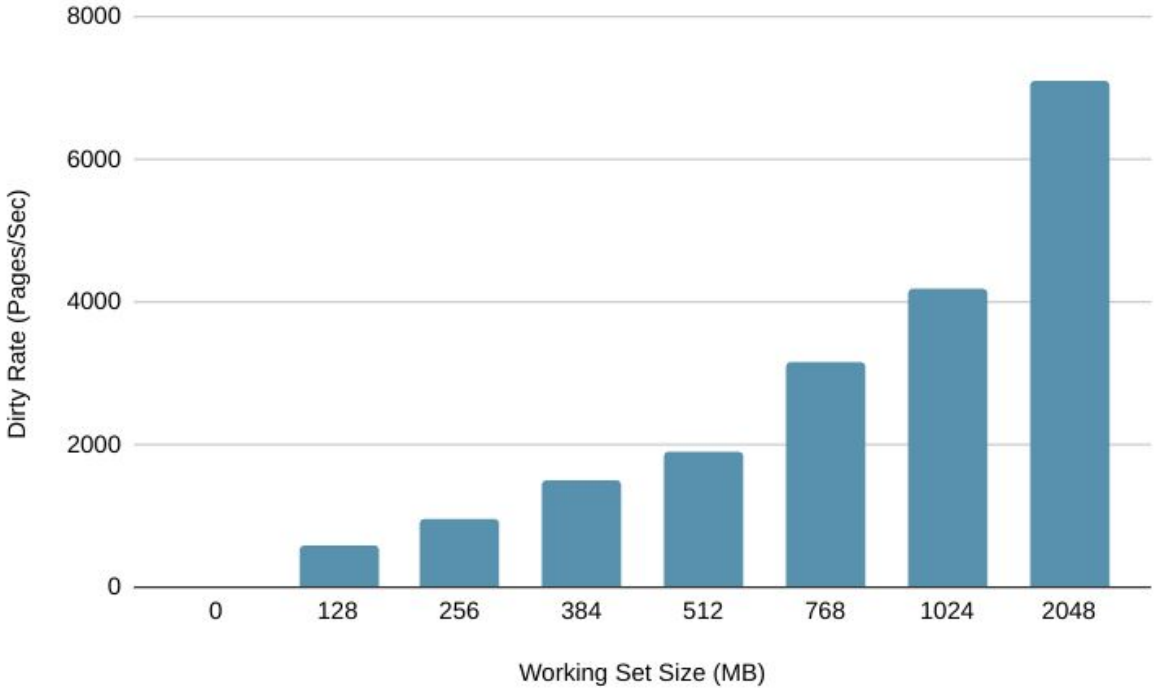


Downtime of Pre-Copy

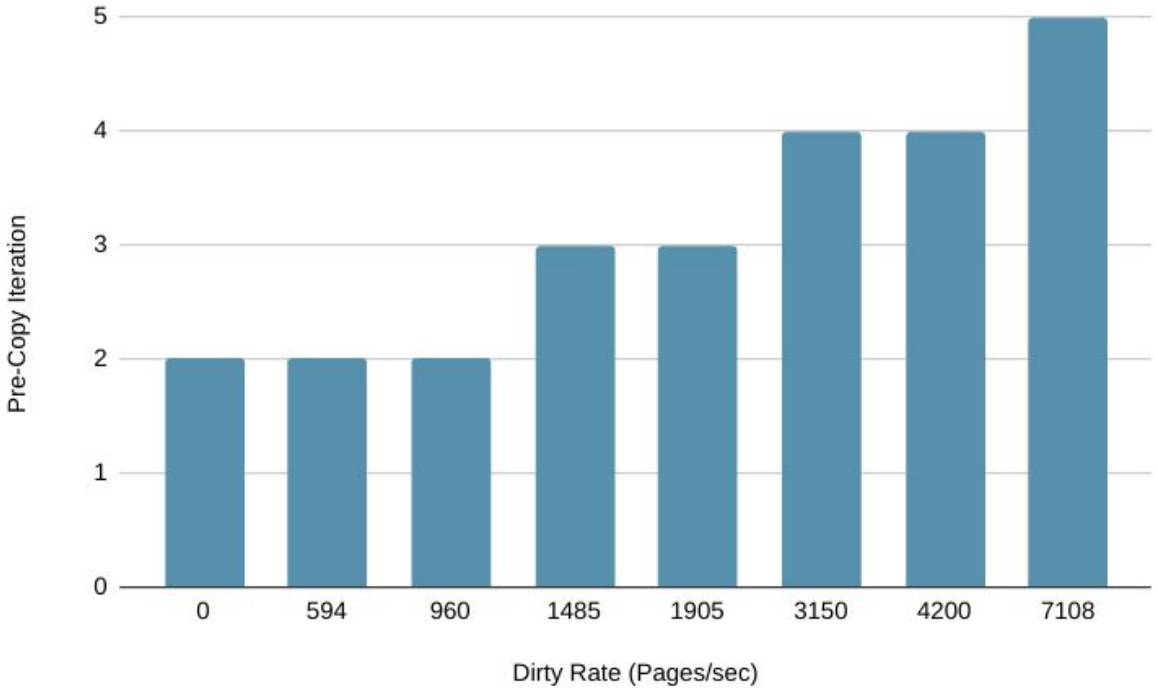
Analysis of Pre-Copy Method with Dirty Pages



Proof of Correctness of our loadtest



Analysis of Pre-Copy Iterations with Dirty Rate



Comparison of Pre-Copy and Stop-and-Copy