INTERIM

Traffic-Aware Live VM Migration

Venudi Dayaratne
20000286 | 2020/CS/028

Supervisor: Dr. Dinuni K. Fernando

Table of contents

01 Background

02 Related Work

03 Research Gap

04 Research Questions

05 Objectives

06 Scope

07 Research Approach

08 Progress

09 Proposed Algorithm

10 Next Steps

11 Evaluation

01

Background

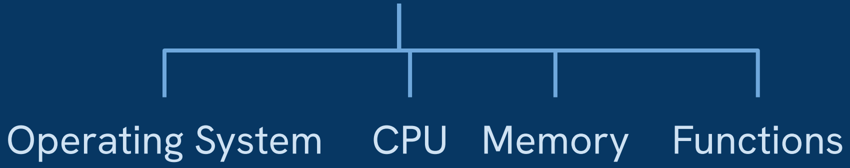
Background

Virtualization

Allows multiple machines to access the same physical resources simultaneously.

Virtual Machine (VM)

emulates



Operating System

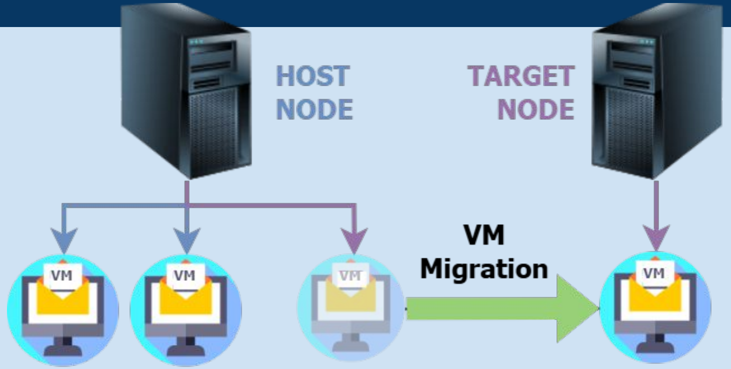
CPU

Memory

Functions

VM Migration

Moving a VM from one physical node to another.



VM Migration

Cold/ Non-live migration

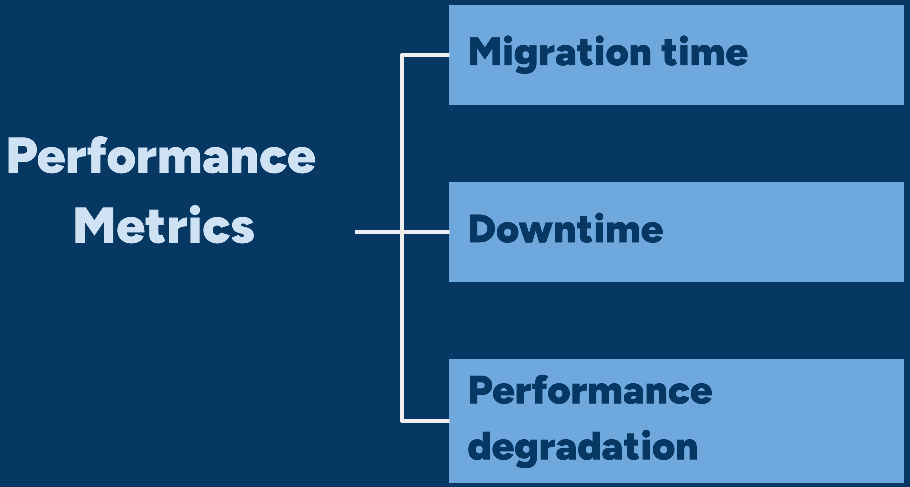
Stops the VM, migrates VM to the destination and resumes.

Live migration

Migrates a VM instantaneously while it's running.

- Pre-Copy
- Post-Copy
- Hybrid

Live Migration



Performance Metrics

Migration time

Time required to complete the entire migration process.

Downtime

Duration during which the services provided by the VMs become inaccessible to users.

Performance degradation

Slowdown of the applications running on the migrating VMs during the migration.

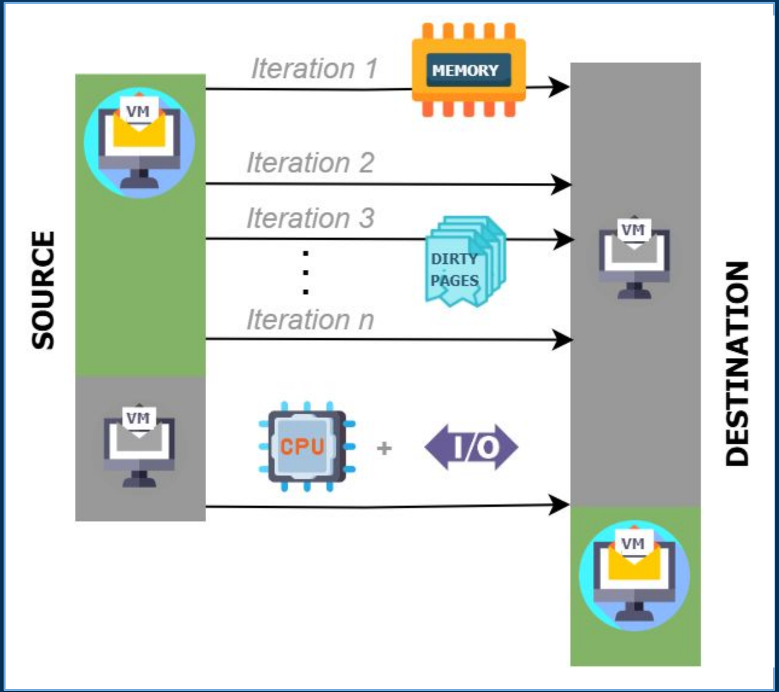
Live Migration (1)

Pre-Copy Migration

Memory transfer.

Stop-and-copy.

Starts running at destination.



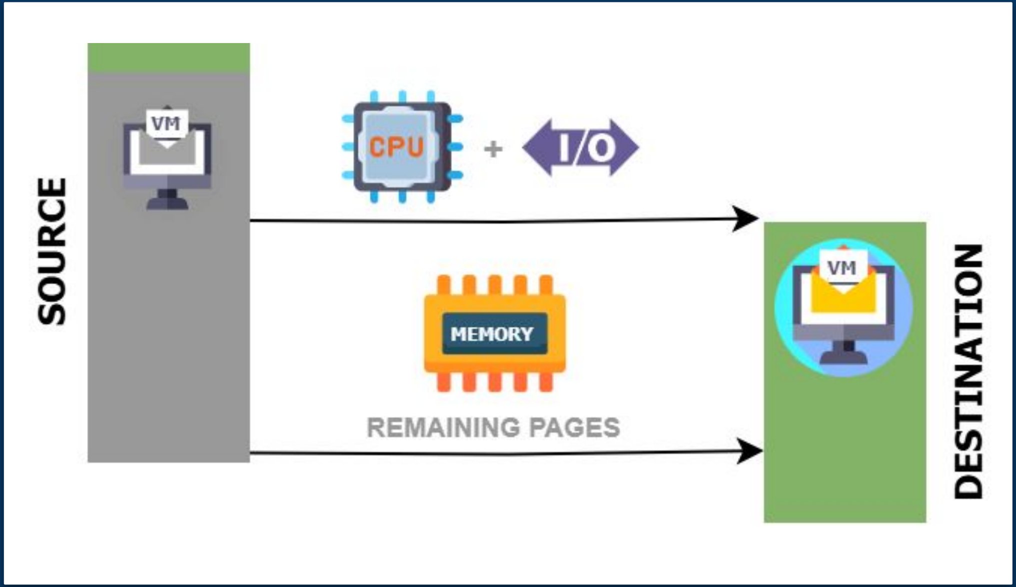
Live Migration (2)

Post-Copy Migration

Suspends VM at source and resumes it at the destination.

Source proactively pushes remaining pages.

Fetches page faults.



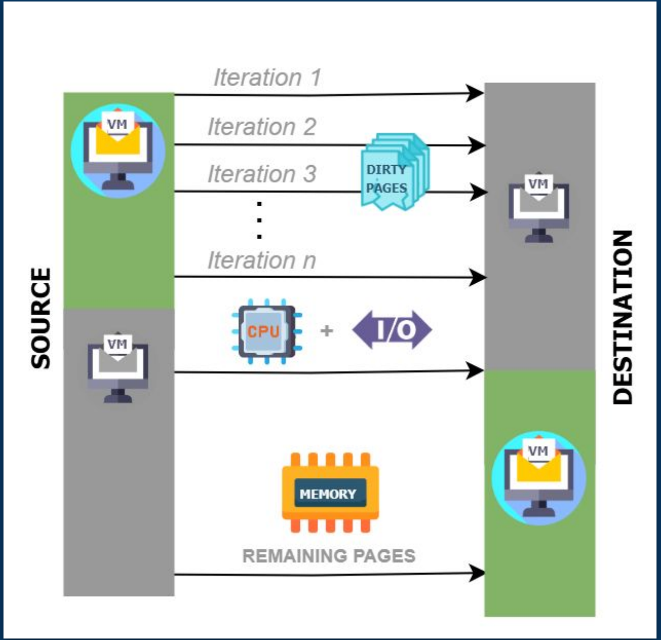
Live Migration (3)

Hybrid Migration

Pre-copy: copies dirty pages from source to destination.

Stop-and-copy.

Post-copy.



Source: Sahni, Shashank and Vasudeva Varma (2012). "A hybrid approach to live migration of virtual machines". In: 2012 IEEE international conference on cloud computing in emerging markets (CCEM). IEEE, pp. 1–5.

Live Migration (4)

Multiple VM Migration

Serial Migration	Gang Migration
One VM after the other.	Concurrently.
Full migration bandwidth can be used.	Migration bandwidth is shared among VMs.

Trade-off

Lower migration time.	Greater migration time.
Greater total downtime.	Lower total downtime.

Sources: Sun, Gang, et al. "A new technique for efficient live migration of multiple virtual machines." Future Generation Computer Systems 55 (2016): 74-86, 5] F. Callegati, W. Cerroni, Live migration of virtualized edge networks: analytical modeling and performance evaluation, in: 2013 IEEE SDN4FNS, 2013, pp. 1-6.

Live Migration (5)

Priority (Urgency) Level

- How quickly the VMs need to be migrated.
- Priority level can be specified by system administrators or assigned considering Service Level Agreements (SLAs) and migration time violations.

For example,

Breakdown - migration need to occur as fast as possible.

Routine server maintenance - not as urgent, but less service disruption expected.

Live Migration (6)

Service Level Agreements (SLAs)

Formal contracts between cloud service providers and customers.

Define expected level of service delivery

- Performance
- Availability
- Response time

Traffic Contention Problem

Traffic

Migration Traffic

Pre-copy has **predominantly outgoing** traffic at the **source.**

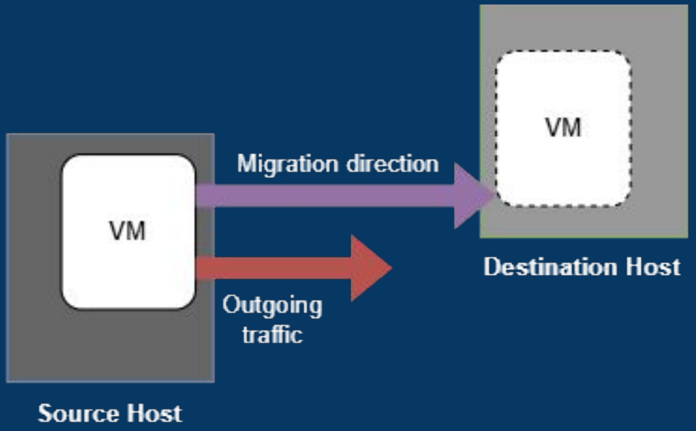
Post-copy has **predominantly incoming** traffic at the **destination.**

Workload Traffic

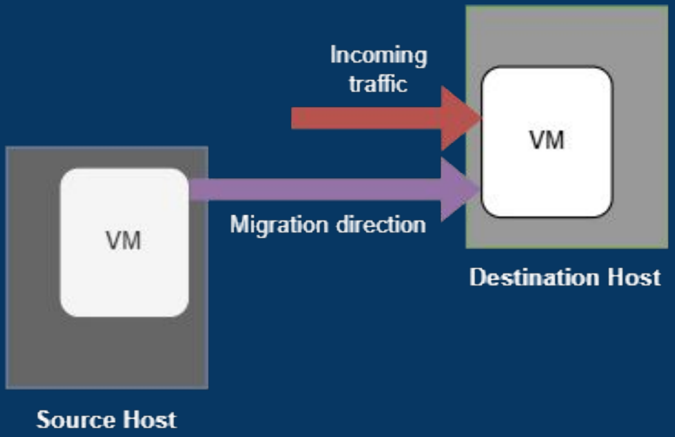
Incoming traffic - data packets received by a system or network interface.

Outgoing traffic - data packets transmitted from a network interface.

Traffic Contention Problem (2)



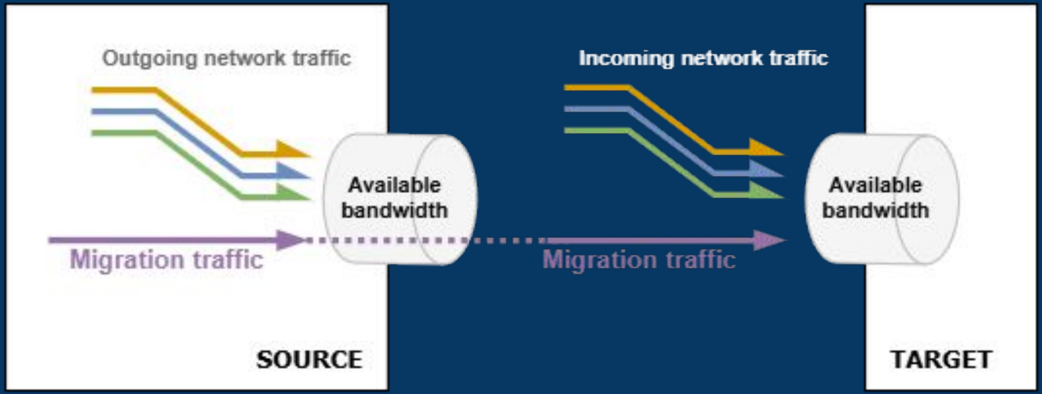
Pre-Copy



Post-Copy

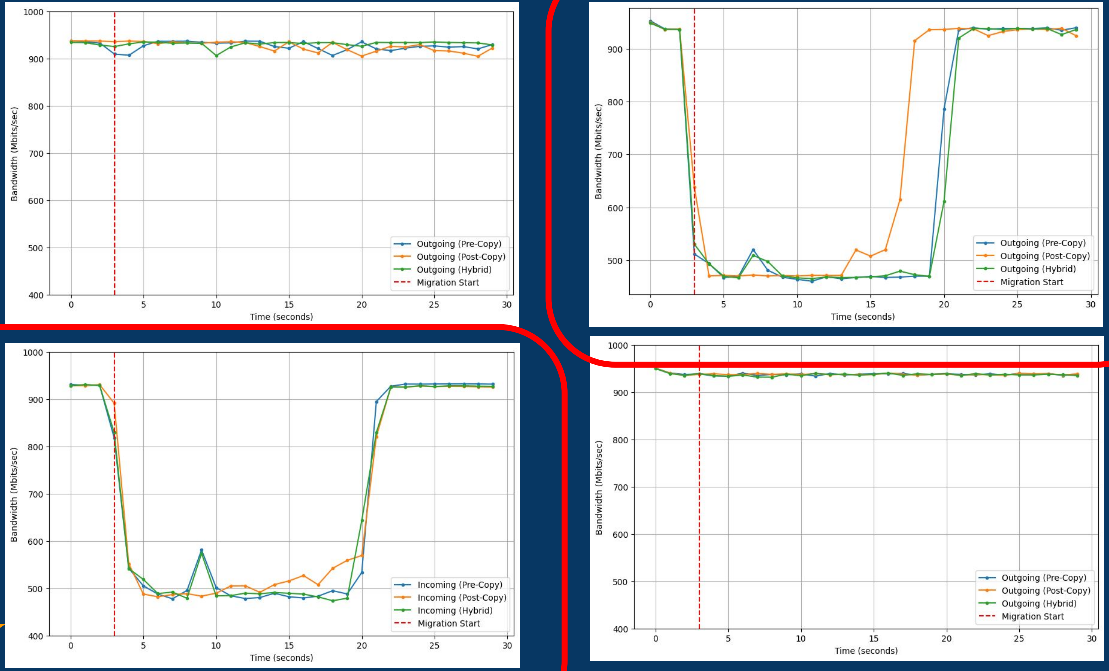
Traffic Contention Problem

Migration traffic and workload traffic **share** the same **network bandwidth**. When they occur simultaneously, they **compete** for the limited network resources.



Preliminary Experimental Study

Bandwidth Fluctuation During Migrations with iPerf Traffic



Source

Destination

Incoming traffic

Outgoing traffic

This traffic contention problem,

- ⏫ Increases migration time
- ⏬ Degrades application performance

02

Related Work

Related Work

Choosing where to migrate to:

1. ***AppAware*** (Shrivastava et al., IEEE INFOCOM, 2011) - optimizes migration destinations based on traffic volume, network topology, and server resource constraints.
2. ***S-CORE*** (Tso et al., IEEE ICDCS, 2014) - minimizes overall communication between VMs by treating VM communication as an optimization problem and using a distributed migration technique that adapts to traffic changes.
3. ***ShareOn*** (Maheshwari et al., IEEE Access, 2018) - selects target nodes for migration based on current traffic conditions to minimize network traffic.

Related Work (1)

Choosing which VM to migrate:

4. ***ST-LVM-LB*** (Kanniga Devi, Murugaboopathi, and Muthukannan, Cluster Computing, 2018) - migrates the least loaded VMs, considering network bandwidth and current flow.

5. ***NTVMM*** (Fu et al., ITNEC, 2019) - reduces traffic and network load through algorithms for VM selection and placement.

Related Work (2)

Dynamically adapting based on traffic:

6. **Traffic-aware algorithm** (Deshpande and Keahey, Future Generation Computer Systems, 2017) - Monitors application and migration traffic to decide between pre-copy or post-copy migration techniques.

7. **Adaptive migration algorithm selection framework** (Cui, Zhu, et al., IJPE, 2020) - Uses fuzzy clustering to categorize VMs based on business traffic and select the most appropriate migration algorithm.

Related Work (3)

Priority level:

8. ***SOLive*** (Fernando, P. Yang, and Lu, IEEE INFOCOM, 2020) - Approach that considers migration urgency in reserving bandwidth.

9. VM selection based on priority (Nadeem, Elazhary, and Fadel, IJACSA, 2018).

Related Work (4)

Traffic shaping:

10. ***mVM Scheduler*** (Kherbache, Madelaine, and Hermenier, IEEETCC, 2017) - monitors traffic patterns and network capacity to determine optimal migration start times and bandwidth allocation.
11. SDN controllers offer real-time traffic flow management, bandwidth allocation, and route adjustments (Manzalini et al., IEEE Communications, 2013).
12. ***Geometric Programming Model*** adapts bandwidth to different phases of migration using a cost function (Cerroni and Esposito, IEEETCC, 2016).
13. Pre-copy migration techniques, which progressively increase transfer rates (Choudhary et al., JoCCASA, 2017).

03

Research Gap

Research Gap

Need for a live VM migration algorithm that considers, SLA violations and migration priority, by adhering to traffic contention.

Lacking research on the decision-making process for performing serial vs parallel migrations **in the context of network contention**.

Less focus on hybrid live migration to address the traffic contention problem in live gang migration.

04

Research Questions

Research Questions

01 What is the optimal migration strategy to reduce the migration time and minimize network contention, considering how quickly the VM needs to be migrated?

02 How can we efficiently migrate VMs in gang migration or serial migration by considering network contention and the migration priority to reduce migration time, downtime, and application performance degradation?

03 Can the integration of the hybrid copy technique enhance the efficiency of live multiple VM migration when considering traffic contention?

05

Objectives

Objectives

1. Identify key network metrics that affect VM migration performance.

2. Create an adaptive decision-making algorithm that uses network metrics, migration priority and SLA requirements to optimize live multiple VM migration.

3. Investigate the impact of the hybrid-copy approach on the adaptability of live multiple VM migration techniques to different traffic conditions.

06

Scope

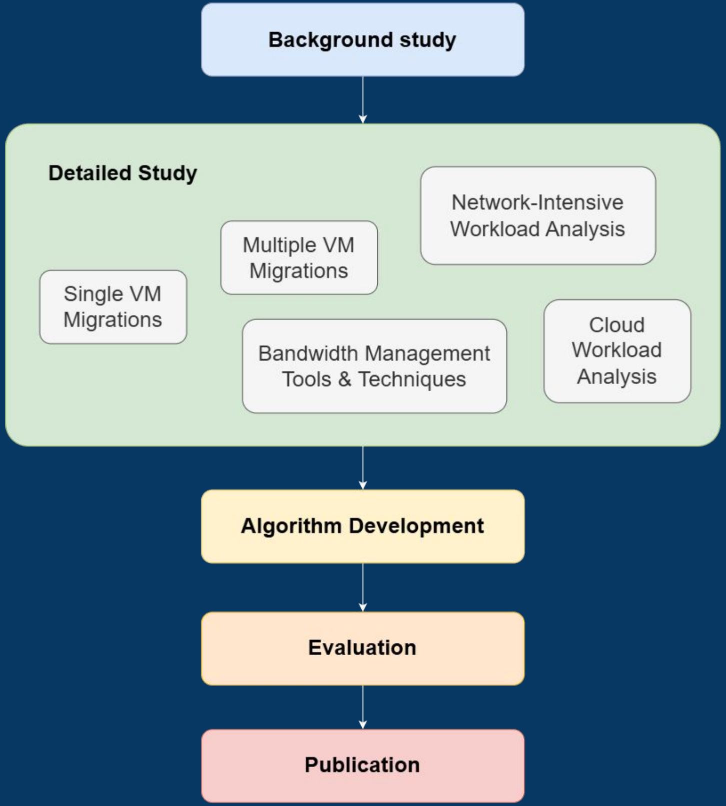
In Scope

- Live VM migration.
  - LAN migrations.
  - KVM/QEMU virtualization environment.
  - Single NIC supported server with a single port for migration and application traffic.
  - Linux host OS.
  - Single and multiple VM migration.
- Analyzing the effect of traffic on live VM migration.
- Developing an optimized algorithm for **traffic-sensitive** live migration of multiple **co-located** VMs considering the **priority** and **SLA violations** of the migration.
- Evaluating migration performance for migration time, downtime and application performance degradation.

07

Research Approach

Research Methodology



Background study

Detailed Study

Single VM Migrations

Multiple VM Migrations

Network-Intensive Workload Analysis

Bandwidth Management Tools & Techniques

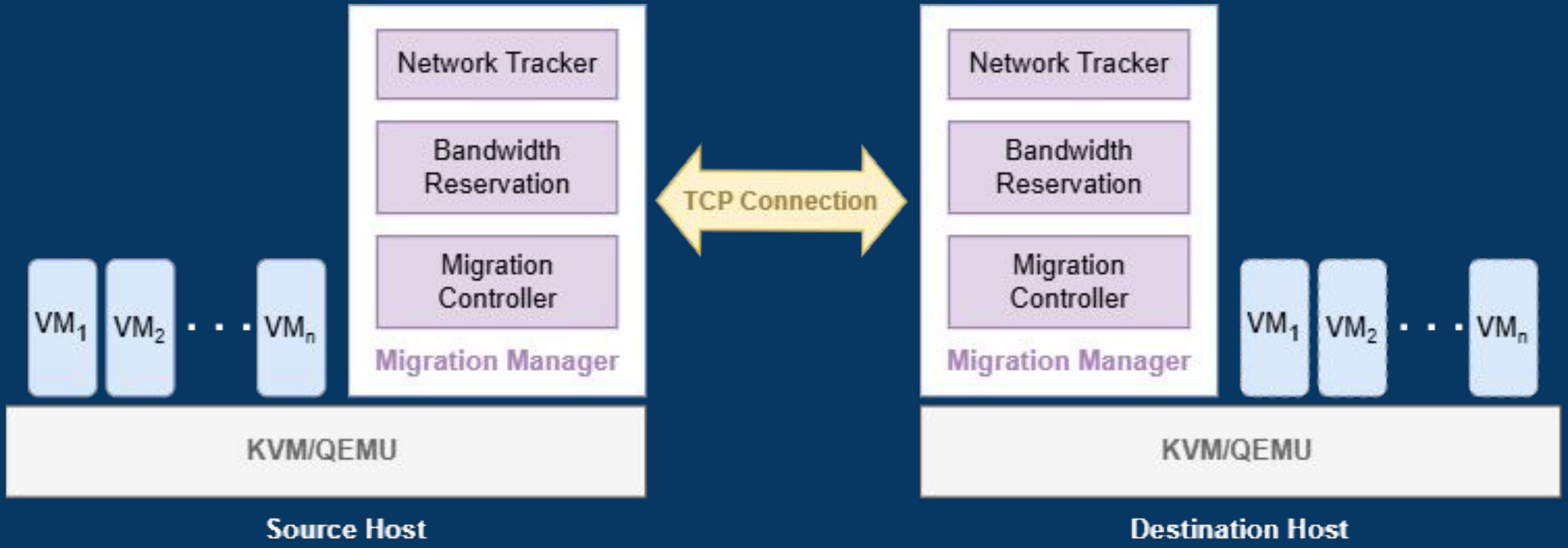
Cloud Workload Analysis

Algorithm Development

Evaluation

Publication

Proposed Architecture



08

Progress

Preliminary Study

Measurements

- Available bandwidth
- Total migration time
- Downtime

Traffic types

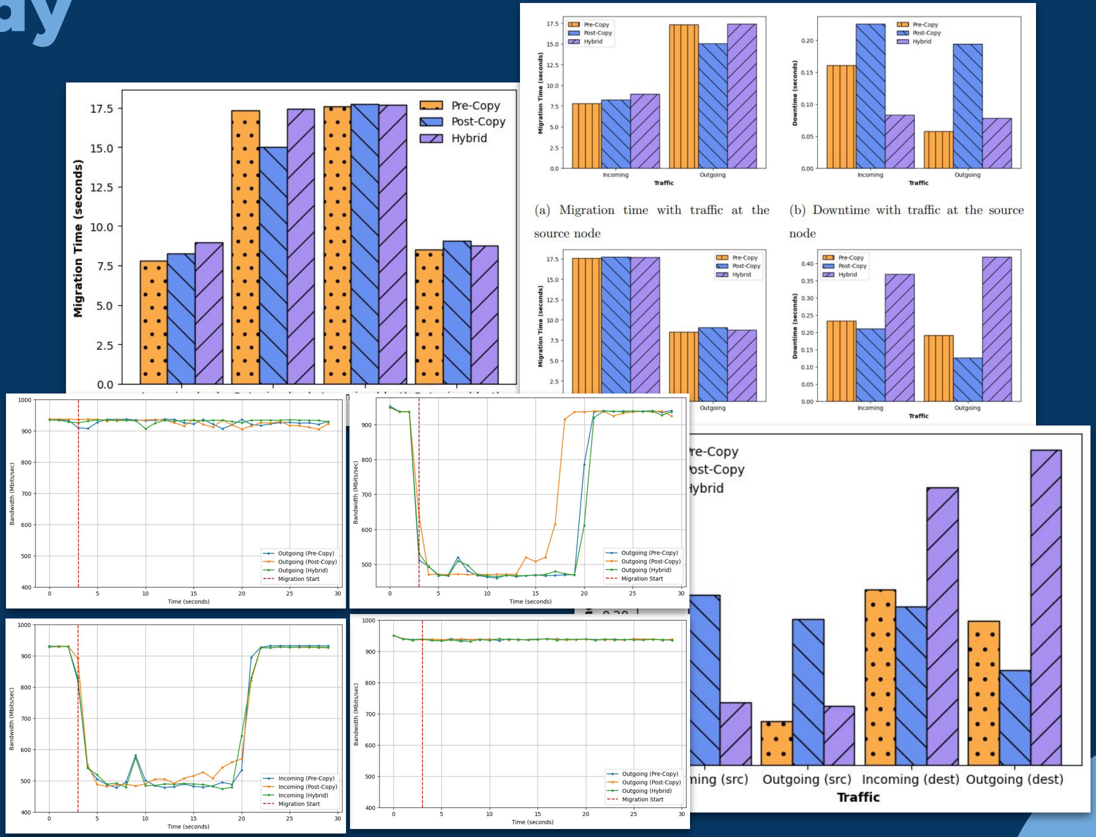
- Incoming
- Outgoing

At

- Source (src) node
- Destination (dest) node
- Migrating VMs

Tools used

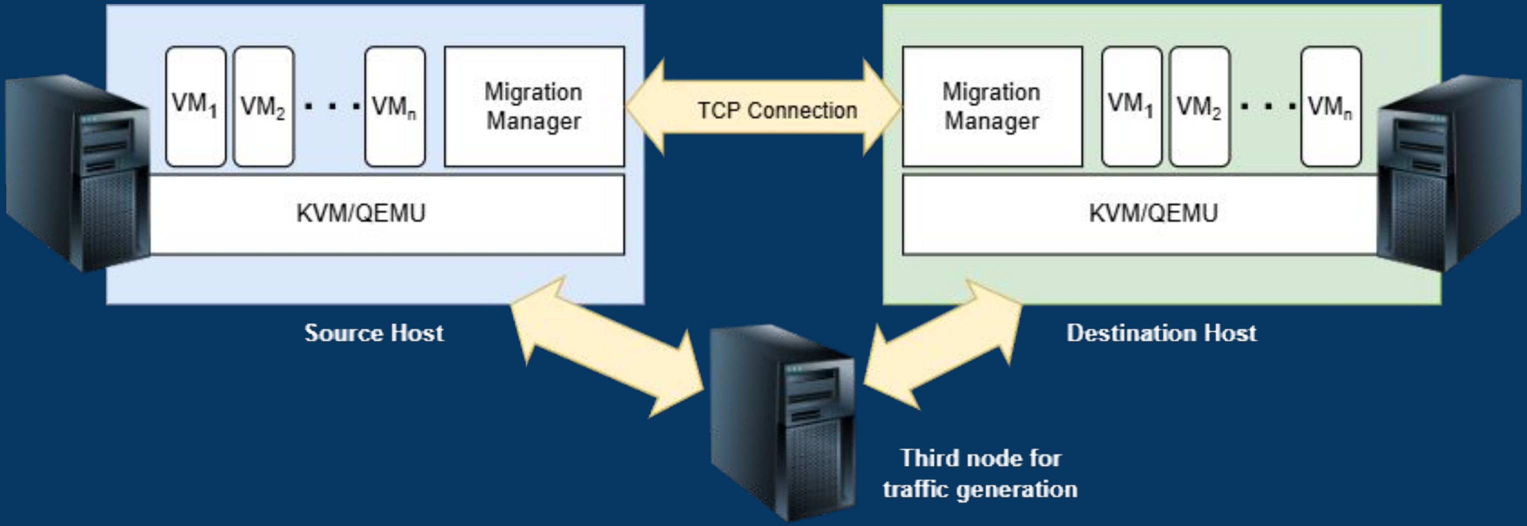
- iPerf3
- Memcached



(a) Migration time with traffic at the source node

(b) Downtime with traffic at the source node

High-level Architecture of the Testbed



Hosts (source and destination)

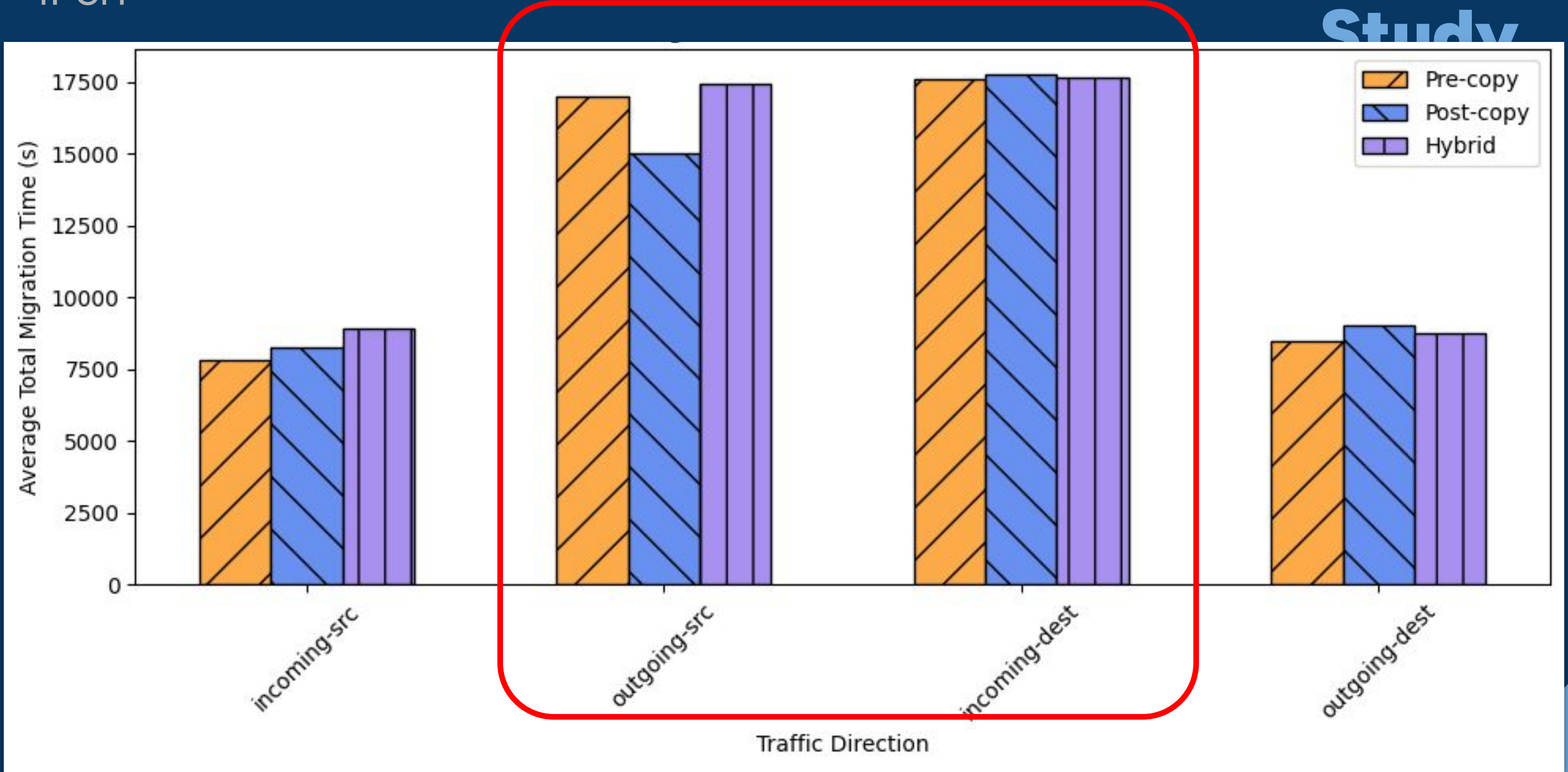
- 12-core Intel Xeon processors
- 16GB RAM
- Ubuntu Servers

- Connected with Gigabit ethernet
- QEMU-KVM hypervisor
- VMs with Linux OSs

Migration Time

iPerf

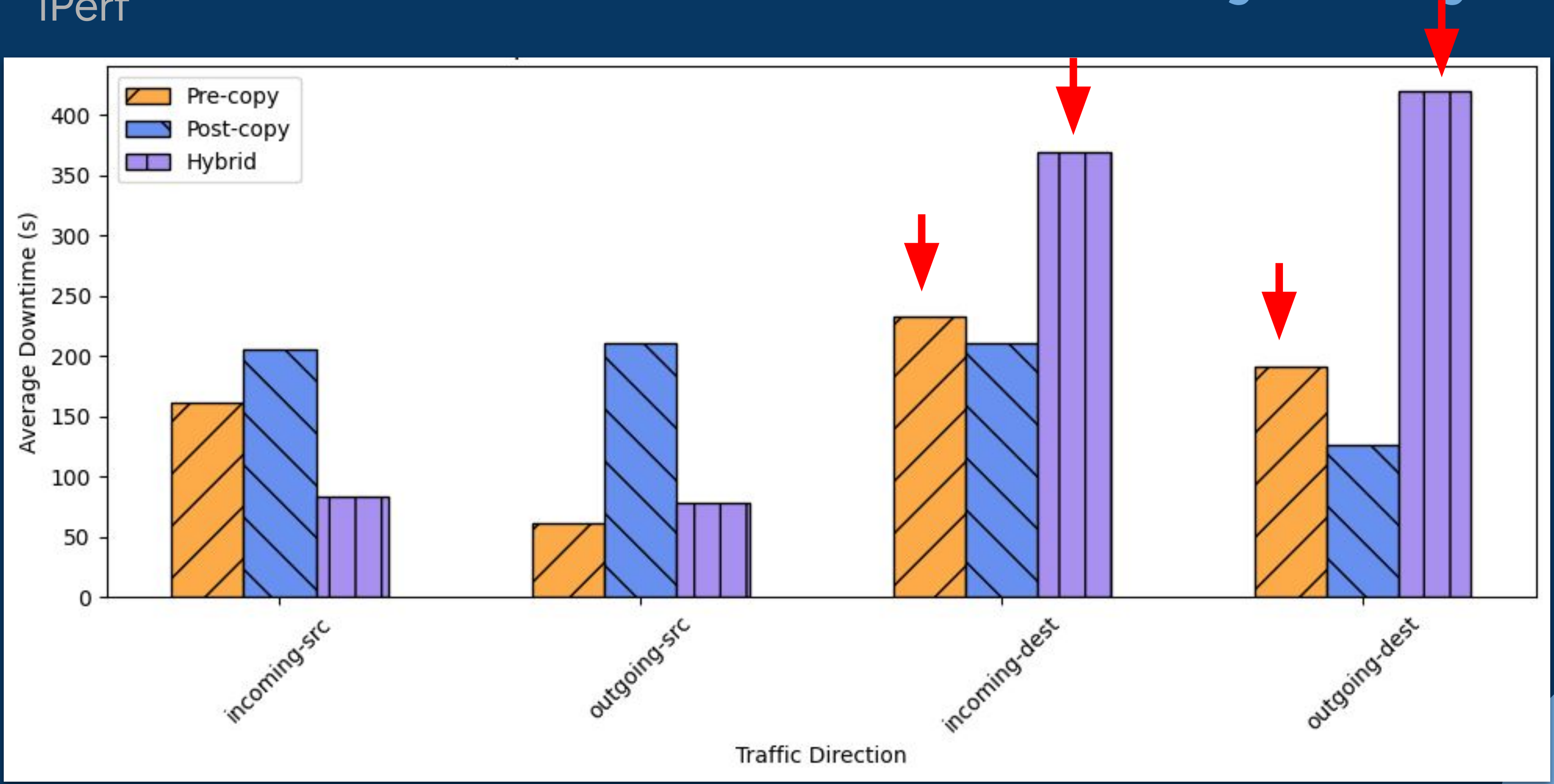
Preliminary Study



Downtime

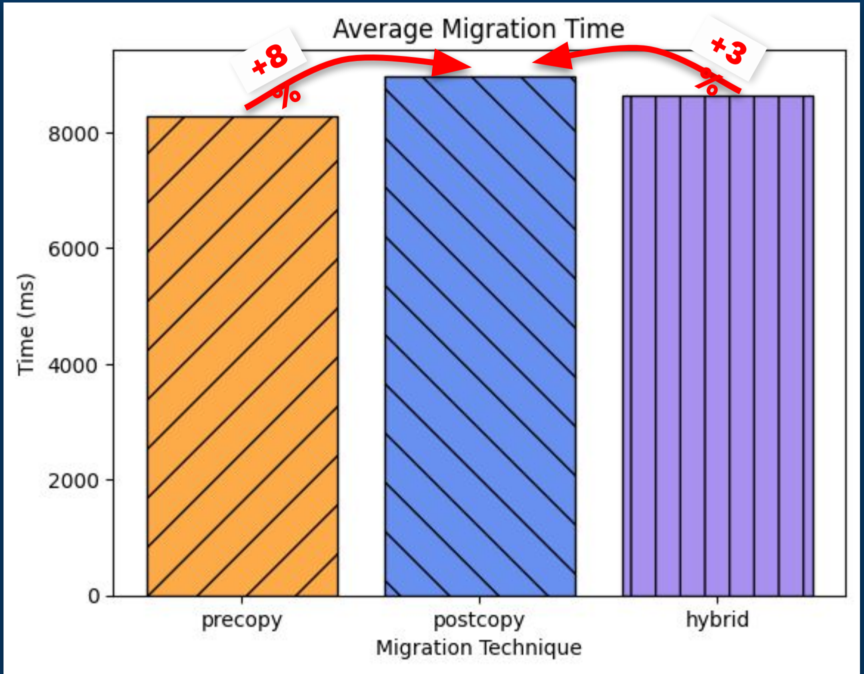
iPerf

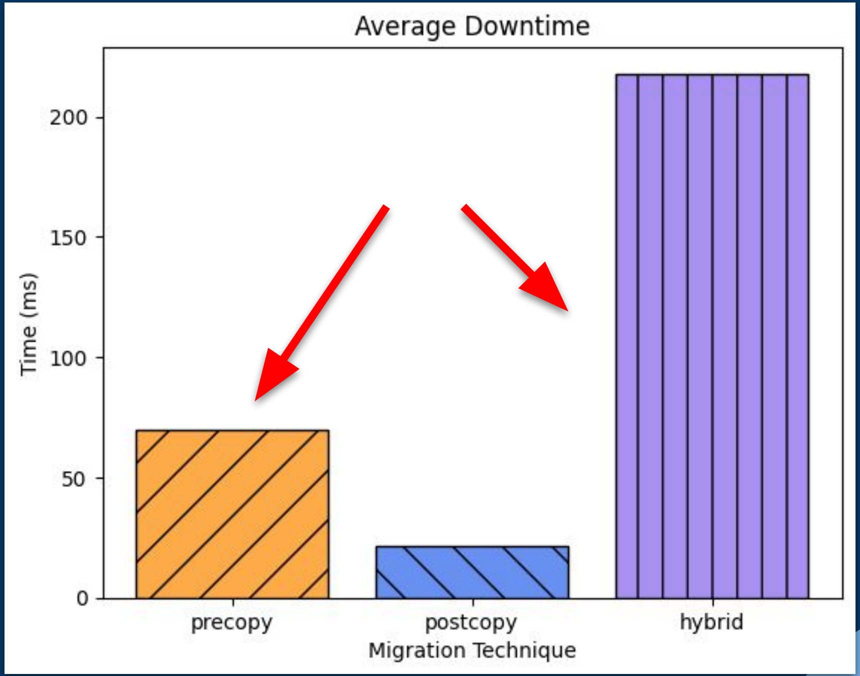
Preliminary Study



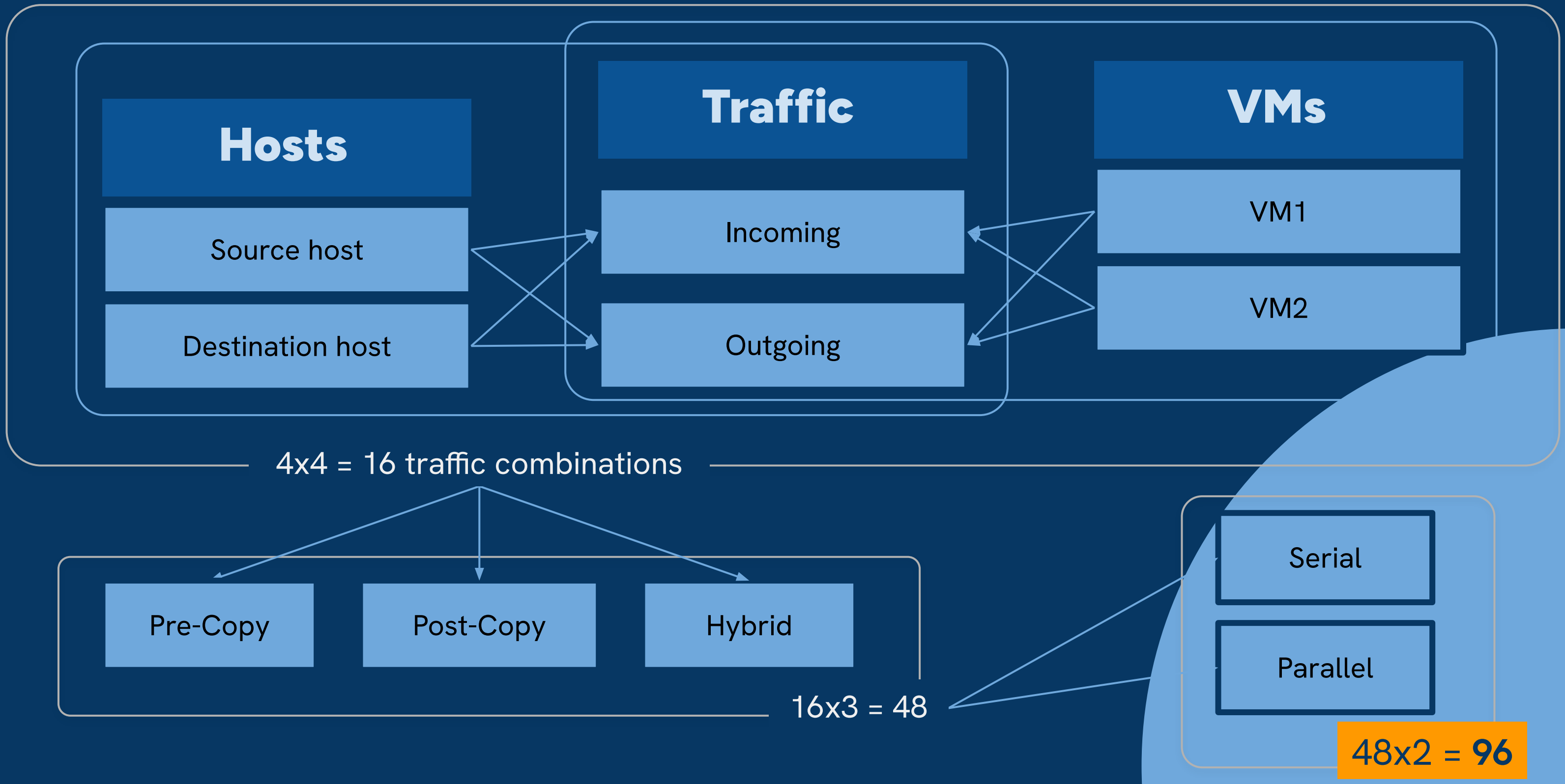
Preliminary Study

Memcached





Migrating Multiple VMs Under Different Traffic Conditions



Hosts

Source host

Destination host

Traffic

Incoming

Outgoing

VMs

VM1

VM2

4x4 = 16 traffic combinations

Pre-Copy

Post-Copy

Hybrid

16x3 = 48

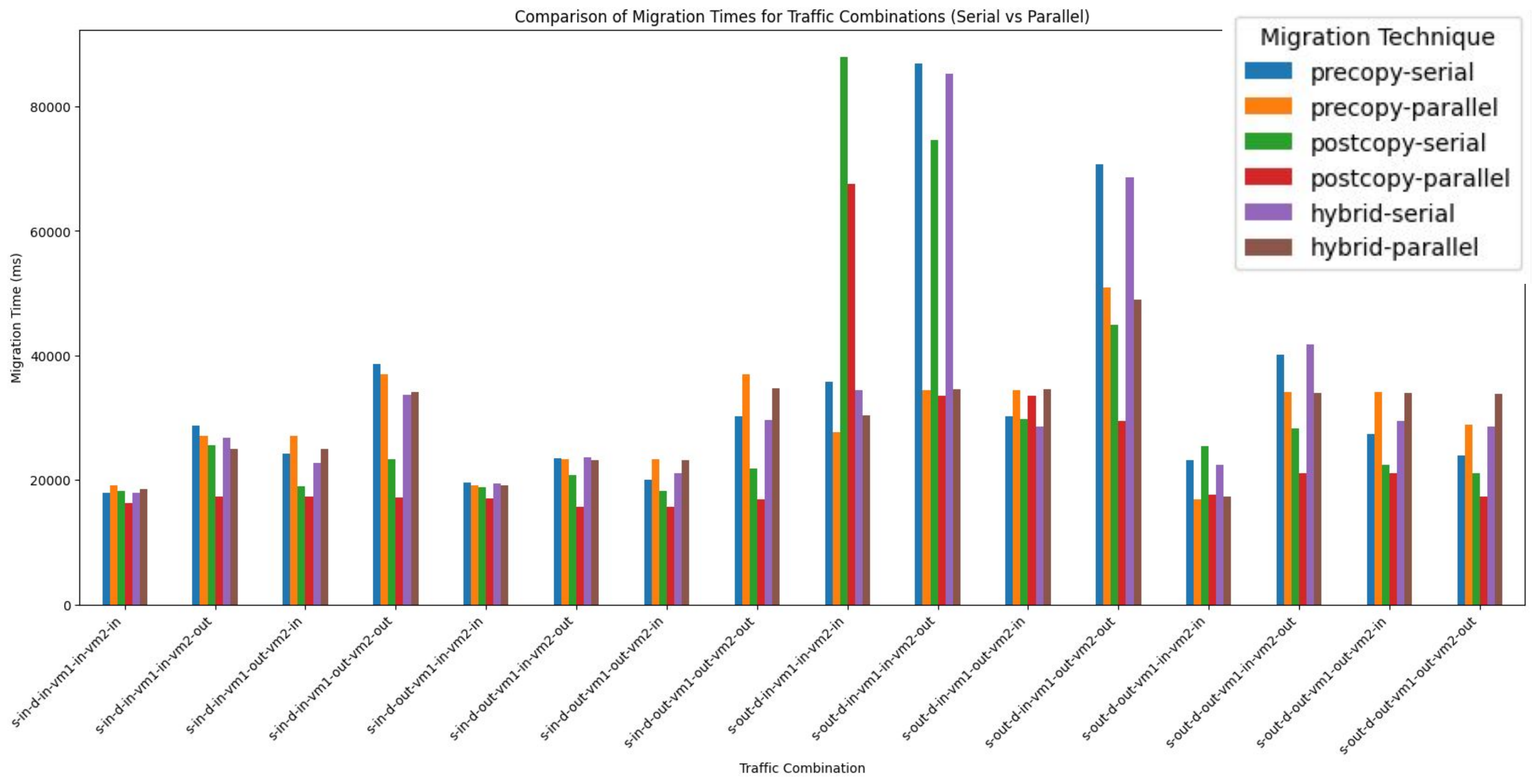
Serial

Parallel

48x2 = **96**

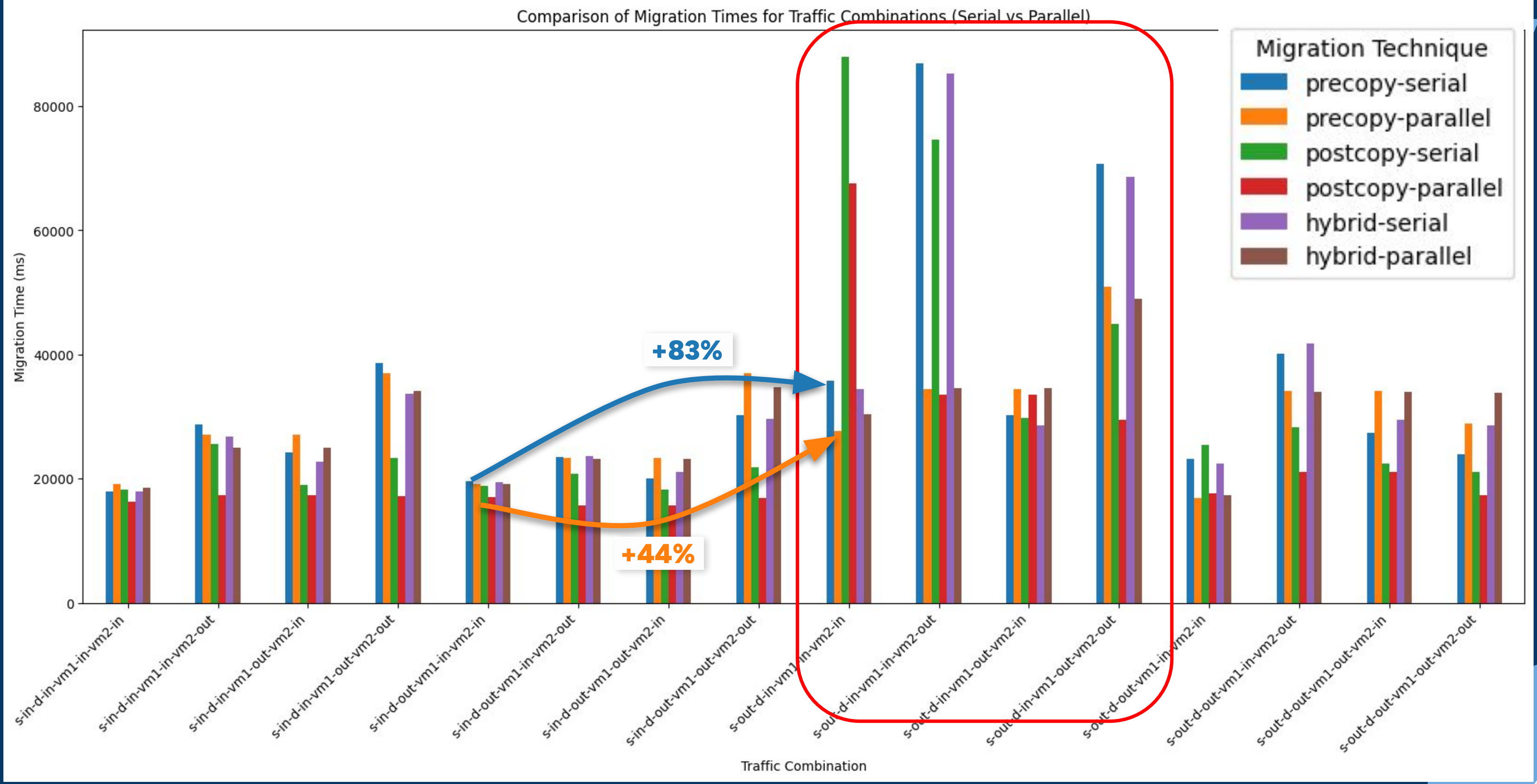
Serial vs Parallel: Total Migration Time (TMT)

Preliminary Study



Serial vs Parallel: Total Migration Time (TMT)

Preliminary

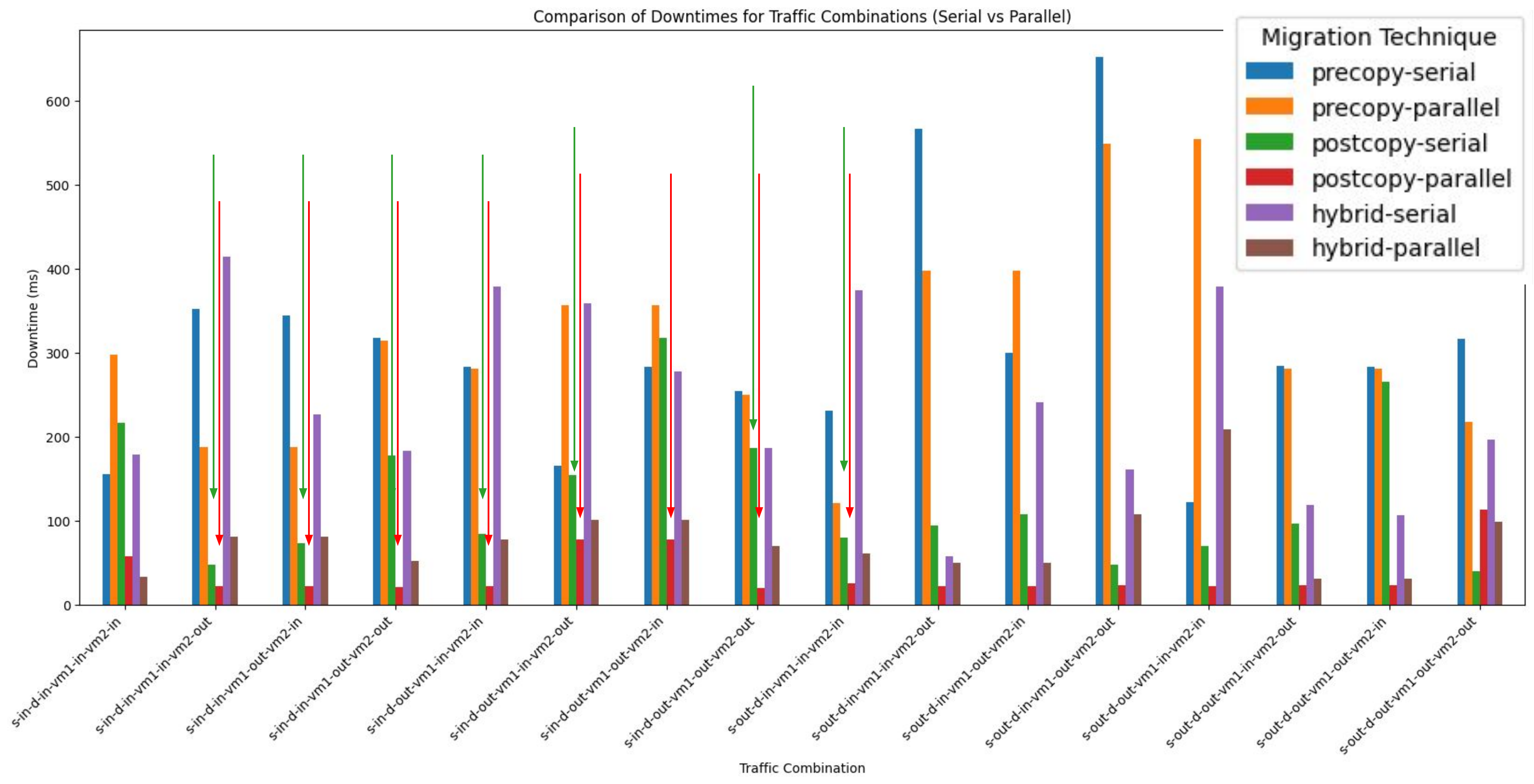


+83%

+44%

Serial vs Parallel: Downtime (DT)

Preliminary Study



Preliminary Study

The experiments revealed the following techniques for each traffic combination,

Src: Inbound, Dest: Inbound ● **VM in**: Hybrid ● **VM out:** Post-copy	**Src: Outbound, Dest: Outbound** ● **VM in**: Post-copy ● **VM out**: Post-copy
Src: Inbound, Dest: Outbound ● **VM in**: Post-copy ● **VM out:** Post-copy	**Src: Outbound, Dest: Inbound** ● **VM in**: ○ Low priority: Pre-copy ○ High priority: Hybrid ● **VM out**: Post-copy

09

Proposed Algorithm

[Tentative]

Proposed Algorithm

Migration Algorithm

Step 1: Categorize VMs based on traffic patterns

Step 2: Select migration technique

Step 3: Determine migration strategy for each VM

Step 4: Reserve bandwidth and migrate VMs

   a. high priority
   b. medium priority
   c. low priority vm list strategy

Step 5: Finalize migration

Proposed Algorithm

Migration Algorithm

Step 1: **Categorize VMs based on traffic patterns**

Step 2: Select migration technique

Step 3: Determine migration strategy for each VM

Step 4: Reserve bandwidth and migrate VMs

a. high priority

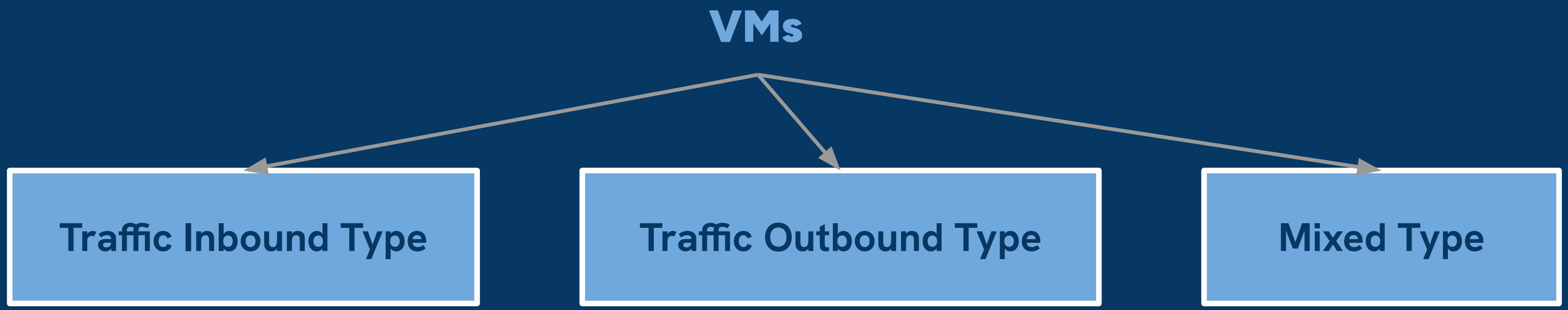
b. medium priority

c. low priority vm list strategy

Step 5: Finalize migration

Proposed Algorithm

Step 1: Categorize VMs based on traffic patterns



VMs

Traffic Inbound Type

Traffic Outbound Type

Mixed Type

Proposed Algorithm

Migration Algorithm

Step 1: Categorize VMs based on traffic patterns

Step 2: **Select migration technique**

Step 3: Determine migration strategy for each VM

Step 4: Reserve bandwidth and migrate VMs

a. high priority

b. medium priority

c. low priority vm list strategy

Step 5: Finalize migration

Proposed Algorithm

Step 2: Select migration technique (Pre-copy, post-copy, hybrid)

Src: Inbound, Dest: Inbound ● **VM in:** Hybrid ● **VM out:** Post-copy	**Src: Outbound, Dest: Outbound** ● **VM in:** Post-copy ● **VM out:** Post-copy
Src: Inbound, Dest: Outbound ● **VM in:** Post-copy ● **VM out:** Post-copy	**Src: Outbound, Dest: Inbound** ● **VM in:** ○ Low priority: Pre-copy ○ High priority: Hybrid ● **VM out:** Post-copy

**Mixed type VMs will be considered separately and handled as a separate condition.*

Proposed Algorithm

Migration Algorithm

Step 1: Categorize VMs based on traffic patterns

Step 2: Select migration technique

Step 3: Determine migration strategy for each VM

Step 4: Reserve bandwidth and migrate VMs

a. high priority
b. medium priority
c. low priority vm list strategy

Step 5: Finalize migration

Proposed Algorithm

Step 3: Determine migration strategy for each VM (serial, parallel)

- **High Priority** - parallel
- **Low priority, low contention** - parallel
- **Low priority, high/moderate contention** - serial

Proposed Algorithm

Migration Algorithm

Step 1: Categorize VMs based on traffic patterns
Step 2: Select migration technique
Step 3: Determine migration strategy for each VM
Step 4: **Reserve bandwidth and migrate VMs**

a. high priority VMs
b. medium priority VMs
c. low priority VMs

Step 5: Finalize migration

Proposed Algorithm

Step 4: Bandwidth Reservation Algorithm

Input: VMs, priority level, lower limit and upper limit to maintain SLA.

- Capture current bandwidth
- Calculate available bandwidth, workload bandwidth, and transfer bandwidth (Estimated using *QEMU*)
- Bandwidth reservation

Proposed Algorithm

Bandwidth Reservation (High Priority)

1. Reserve lower limit for workload traffic (to prioritize migration).
2. Reserve entire available bandwidth (total bandwidth - lower limit) for the migration.
3. Start migration.

Bandwidth Reservation (Low Priority)

1. Reserve whatever available bandwidth for the migration.
2. Start migration.

Proposed Algorithm

Bandwidth Reservation (Medium Priority)

1. Reserve at most upper limit for workload traffic
2. Reserve the remaining bandwidth for the migration
3. Start migration
4. While the migration is running,
    a. Periodically check available bandwidth
    b. If utilized bandwidth is reduced, use the extra bandwidth for migration (reserved bandwidth for workload does not reduce below the lower limit)

Proposed Algorithm

Migration Algorithm

Step 1: Categorize VMs based on traffic patterns

Step 2: Select migration technique

Step 3: Determine migration strategy for each VM

Step 4: Reserve bandwidth and migrate VMs

a. high priority VMs

b. medium priority VMs

c. low priority VMs

Step 5: Finalize migration

10

Next Steps

Next steps

- Collect more data to strengthen the migration decisions taken by the proposed algorithm.
- Empirically revise and improve the traffic sensitive migration algorithm.
- Experiment with traffic shaping tools (e.g: Trickle, tc, wondershaper) to evaluate bandwidth.
- Experiment with different workloads such as transactional, analytical and AI/ML workloads.
- Implement the proposed algorithm on QEMU version 8.1.2.
- Evaluate the developed algorithm to quantify performance improvements or degradations introduced by the traffic-sensitive migration strategy compared to traditional techniques.

11

Evaluation

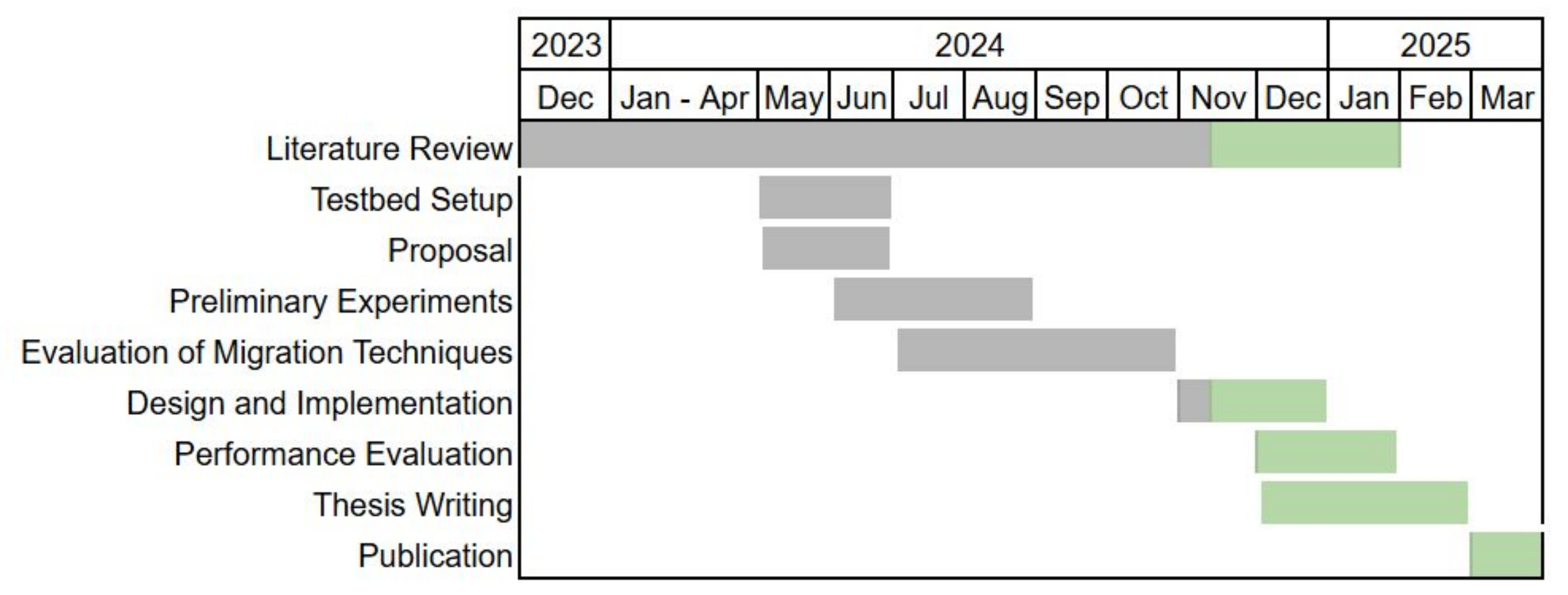
Evaluation

- Environment will be same as the testbed setup displayed earlier.
- Workload generation
  - Transactional (TPC-C, HammerDB)
  - Analytical (TPC-H, BigBench)
  - Content delivery (Citron, wrk2)
  - Development and testing (Phoronix Test Suite, Jenkins Plugins)
  - Batch processing (HiBench, TPCx BB)
  - AI/ML (MLPerf, DeepBench)
  - Archival (SPEC SFS2014, fio)

Evaluation

1. Experiment with traditional techniques without traffic awareness
    - Randomized trials
    - Migrate VMs under varying network conditions
2. Experiment with the traffic-sensitive algorithm integrated to QEMU
3. Evaluate with the baseline approaches

Timeline


2023
2024
2025
Dec
Jan - Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Literature Review
Testbed Setup
Proposal
Preliminary Experiments
Evaluation of Migration Techniques
Design and Implementation
Performance Evaluation
Thesis Writing
Publication

References

- Cui, Yong, Liang Zhu, et al. (2020). “An adaptive traffic-aware migration algorithm selection framework in live migration of multiple virtual machines”. In: International Journal of Performability Engineering 16 (2), pp. 314–324. issn: 09731318. doi: 10.23940/ijpe.20.02.p14.314324
- Cui, Yong, Zhenjie Yang, et al. (2017). “Traffic-aware virtual machine migration in topology-adaptive dcn”. In: IEEE/ACM Transactions on Networking 25.6, pp. 3427–3440
- Deshpande, Umesh and Kate Keahey (July 2017). “Traffic-sensitive Live Migration of Virtual Machines”. In: Future Generation Computer Systems 72, pp. 118– 128. issn: 0167739X. doi: 10.1016/j.future.2016.05.003
- Fernando, Dinuni, Ping Yang, and Hui Lu (2020). “SDN-based Order-aware Live Migration of Virtual Machines”. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, pp. 1818–1827. doi: 10.1109/INFOCOM41043. 2020.9155415
- Fu, Xiong et al. (2019). “Network traffic based virtual machine migration in cloud computing environment”. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, pp. 818–821.
- Kanniga Devi, R, G Murugaboopathi, and M Muthukannan (2018). “Load monitoring and system-traffic-aware live VM migration-based load balancing in cloud data center using graph theoretic solutions”. In: Cluster Computing 21.3, pp. 1623–1638.
- Maheshwari, Sumit et al. (2018). “Traffic-aware dynamic container migration for real-time support in mobile edge clouds”. In: 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, pp. 1–6
- Nadeem, Hanan A, Hanan Elazhary, and Mai A Fadel (2018). “Priority-aware virtual machine selection algorithm in dynamic consolidation”. In: International Journal of Advanced Computer Science and Applications 9.11
- Nasim, Robayet and Andreas J Kassler (2015). “Network-centric performance improvement for live VM migration”. In: 2015 IEEE 8th International Conference on Cloud Computing. IEEE, pp. 106–113
- Sahni, Shashank and Vasudeva Varma (2012). “A hybrid approach to live migration of virtual machines”. In: 2012 IEEE international conference on cloud computing in emerging markets (CCEM). IEEE, pp. 1–5.
- Shrivastava, Vivek et al. (2011). “Application-aware virtual machine migration in data centers”. In: 2011 Proceedings IEEE INFOCOM. IEEE, pp. 66–70
- Son, Jungmin and Rajkumar Buyya (2018). “Priority-aware VM allocation and network bandwidth provisioning in software-defined networking (SDN)-enabled clouds”. In: IEEE Transactions on Sustainable Computing 4.1, pp. 17–28
- Tso, Fung Po et al. (2014). “Scalable traffic-aware virtual machine management for cloud data centers”. In: 2014 IEEE 34th International Conference on Distributed Computing Systems. IEEE, pp. 238–247.

In summary

- Develop a **traffic sensitive, adaptive decision making** algorithm
  - Addressing the **traffic contention problem**
  - Considering **migration priority, SLA violations**
- Investigate the impact of the **hybrid-copy** approach on the adaptability to traffic of **live multiple VM migration** techniques.

Thank you!