Traffic-Aware Live Virtual Machine Migration

Research Proposal

SCS 4224: Final Year Project in Computer Science

Venudi Dayaratne

20000286

June 19, 2024

Declaration

This proposal is my original work and has not been submitted at this or any other university or institute. To the best of my knowledge, it does not include any material authored or published by others, except where appropriately cited in the text.

Name : H.B.V.D. Dayaratne
Registration No. : 2020/CS/028
Index No. : 20000286

Signature

Supervisor : Dr. (Mrs). D.K. Fernando

19-6-2024

Signature

Contents

List of Figures

List of Tables

Abbreviations

CDC Cloud Data Center. 2

LAN Local Area Network. 14

MILP Mixed Integer Linear Programming. 10

NIC Network Interface Card. 2, 13, 14

NTVMM Network Traffic-based Virtual Machine Migration algorithm. 10

OS Operating System. 1, 13, 15

QoS Quality of Service. 5

SLA Service Level Agreement. 5

SLAs Service Level Agreements. 5

VM Virtual Machine. 1–6, 8, 10, 11, 13, 15, 16

VMs Virtual Machines. 1, 3, 4, 6–8, 10, 11, 13–15

WAN Wide Area Network. 1, 14

WWS Writable Working Set. 3, 6

1 Introduction

In today's digital landscape, the need for on-demand and scalable computing resources has driven the adoption of cloud computing. Virtualization enables the creation of virtual images of physical resources, allowing multiple machines to access them simultaneously. A Virtual Machine (VM) emulates a physical machine with its own Operating System (OS), CPU, memory, and functions. A physical machine, known as a node in virtualization, serves as a server hosting one or many VMs. These VMs can be resized and relocated—a process known as VM migration (Bahrami, Haghighat, and Gholipoor n.d.).

Migrating a VM instantaneously while it is running is called *Live Migration*. In contrast, the simultaneous live migration of multiple co-located VMs is referred to as *Gang Live Migration*. This process typically involves transferring the VM's memory, disk storage, and network connections from the source node to the destination (Le 2020). Within a Local Area Network (LAN), VM migrations do not require the migration of disk storage due to the use of Network Attached Storage (NAS) (Jo et al. 2013). However, migrations over a Wide Area Network (WAN) necessitate the transfer of both the disk state and memory (Wood, Ramakrishnan, et al. 2014).

There are three (3) traditional live migration techniques namely pre-copy (Clark et al. 2005; Nelson, Lim, Hutchins, et al. 2005), post-copy (Hines and Gopalan 2009; Hines, Deshpande, and Gopalan n.d.), and hybrid-copy (Sahni and Varma 2012). These techniques are designed for single VM migrations. However, when adapting these techniques for gang migration, if all the VMs monotonously adopt a single migration algorithm, it overlooks the traffic. The traffic generated due to the VM migration adds to the existing application traffic leading to inefficiencies.

1.1 Motivation

Live migration of VMs in a Cloud Data Center (CDC) can occur for various reasons such as to balance load (Padala et al. 2007; Wood, Shenoy, et al. 2007), handling hardware failures (Nagarajan et al. 2007), performing hardware maintenance (Devi, Aruna, Priya, et al. 2011), and saving power and energy through consolidation (Nathuji and Schwan 2007; Hu et al. 2008). In all these scenarios, it is crucial that the migration is performed swiftly.

When the migration occurs in the same Network Interface Card (NIC) interface as the VM, the VM workload and migration traffic share the bandwidth. This causes a network contention posing a significant challenge in live migration, specifically in pre-copy and post-copy techniques. In pre-copy migration, the migration traffic competes with the outgoing network traffic whereas the migration traffic competes with the destination's incoming traffic, in post-copy (Deshpande and Keahey 2017). This interference negatively affects the performance of applications on the migrating VM, slows down the migration and increases downtime. The magnitude of network traffic has increased significantly with the continuous evolution of CDCs. This has only added to the network traffic further slowing down the migration.

The above challenges can be addressed gracefully in live VM migration if network traffic is considered.

1.2 Background

1.2.1 Live VM Migration

In live migration, the memory pages which belong to the VM and disk block are copied. The VM's runtime state (CPU, disk I/O, memory state) should be moved into the host node while the VM is still running (Clark et al. 2005).

1.2.1.1 Pre-copy Technique

In the pre-copy live VM migration technique, while the VM operates on the source, its memory pages are moved to the destination. The measurement taken to pause the *memory transfer phase* of the migration is identifying a Writable Working Set (WWS) (Hines, Deshpande, and Gopalan n.d.). Once this milestone is reached this phase stops and thus begins the next phase of the pre-copy migration, the stop-and-copy phase. The VM is temporarily halted facilitating the CPU state and the dirty memory pages to be transferred to the target host. Finally, all connections in the open network are reconfigured to the new host.

1.2.1.2 Post-copy Technique

Post-copy migration copies the non-pageable memory, network states, and minimal CPU to the destination host and resumes the VM at the destination. Following this, the remaining pages are fetched (Bahrami, Haghighat, and Gholipoor n.d.).

1.2.1.3 Hybrid-copy Technique

To mitigate the performance degradation associated with the post-copy approach and the downtime characteristic of the pre-copy method, the hybrid-copy migration technique was developed. This method integrates both pre-copy and post-copy approaches. Initially, while the VM operates on the source host, its memory is copied to the destination as in the first round of the pre-copy technique. Subsequently, the VM is transferred to the destination host and resumes execution. Any page faults occurring thereafter are handled by requesting the dirty memory pages from the source, similar to the post-copy technique (Sahni and Varma 2012).

1.2.2 Multiple VM Migration

Virtual machines residing on the same physical host are known as co-located VMs. These VMs communicate and complement each other's services and executing on the same physical machine reduces communication costs (Huang et al. 2007) and improves consolidation (Wood, Shenoy, et al. 2007). For reasons such as saving

	Preparation Time	Migration Time	Downtime	Migrated Data
Pre-copy	VM memory transfer + dirty memory page transfer iterations	Depends on bandwidth, iterations, dirty rate	CPU + remaining dirty memory	Multiples of RAM size
Post-copy	Negligible	Depends on the rate of page faults	CPU + virtual device states	The exact size of the RAM
Hybrid	Page table scanning + working set identification	Depends on dirty rate, bandwidth	CPU + virtual device states + remaining dirty memory	More than a single RAM size, lesser than or equal to Pre-Copy (depends on dirty rate)

Table 1: Migration Technique Comparison (Altahat et al. 2020)

energy and maintaining system performance, the need for migrating multiple such correlated VMs(Sun, Liao, Zhao, et al. 2015) arises. These migrations can be carried out either serially or parallelly (Callegati and Cerroni 2013; Chang, Walters, and Wills 2013; Sun, Liao, Anand, et al. 2016).

In the *serial migration* strategy, VMs are migrated one after the other using the migration approaches proposed for single VM migration. The live migration of multiple Virtual Machines parallelly is called *live gang migration* (Deshpande, Wang, and Gopalan 2011). As VMs are migrated concurrently in this approach, the bandwidth available for migration is shared by the VMs.

In parallel migration, the total migration time is longer than in serial migration, but the downtime is shorter. This suggests a tradeoff: parallel migration has the advantage of less downtime, making it better for maintaining service availability. However, serial migration has a shorter overall migration time, making it more efficient in terms of communication resources and transmission overhead (Callegati and Cerroni 2013)

Priority level refers to how quickly multiple VMs need to be migrated. Highly prioritized migrations may require gang migration, which, despite its impact on performance, minimizes downtime. For less prioritized migrations, a serial approach (migrating VMs one after another) might be more suitable, providing a more controlled and gradual process. System administrators can specify the priority level for each migration (Fernando, P. Yang, and Lu 2020) or priority can be assigned considering the SLA violations and migration time violations (Tsakalozos et al. 2017).

Different migrations have varying time requirements and corresponding deadlines, especially when dealing with multiple VMs (Nadeem, Elazhary, and Fadel 2018). Some virtual network functions (VNFs) require rapid migration to maintain low end-to-end latency, while web services with higher latency tolerance can be scheduled over a longer period. For example, in case of a breakdown, the VM migration needs to occur as quickly as possible in contrast to routine server maintenance which isn't as urgent but needs to be done with the least service disruption. Service Level Agreements (SLAs) and Quality of Service (QoS) considerations are crucial in determining whether to use serial or parallel migration for multiple VMs and in determining the bandwidth to be reserved for the migration. Depending on the use case, migrations may need to meet specific deadlines or involve trade-offs between priority level and performance.

1.2.3 Performance Metrics

Performance metrics serve as an essential evaluation to assess the live VM migration methods as shown in Table 1. Among the numerous metrics identified, total migration time, downtime, total migrated data, migration overhead, application degradation, preparation time, and resume time certain metrics stand out as most commonly used (Voorsluys et al. 2009; Altahat et al. 2020; Soni and Kalra 2013; Hines and Gopalan 2009; Kuno, Nii, and Yamaguchi 2011). Specifically,

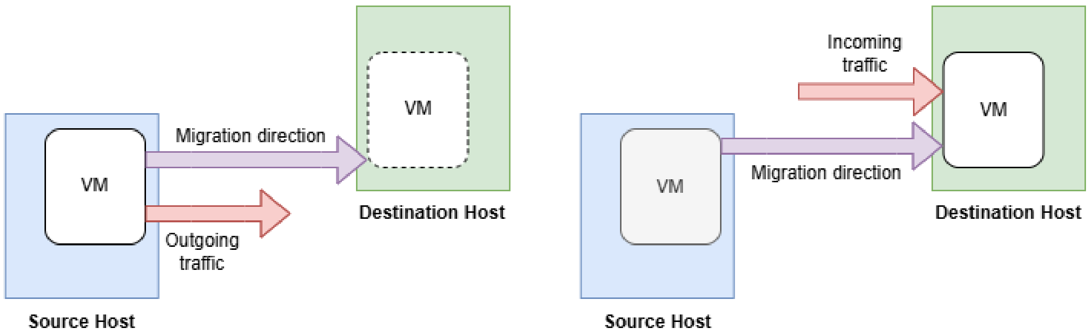
- Total migration time - the time required to complete the entire migration process. In gang migration, it's the time between from the beginning of the migration process at the source to the completion of the last VM's migration at the destination.
- Downtime - the duration during which the services provided by the VMs become inaccessible to users. The VM's CPU state and the WWS are transferred during this time and therefore the CPU execution is fully suspended.
- Total migrated data - the volume of migrated data.
- Application performance degradation - the slowdown of the applications running on the migrating VMs during the migration.

1.2.4 Traffic Contention Problem

In the context of migration, traffic can be classified between workload traffic and migration traffic. As for workload traffic, it includes workloads of the migrating VM, other co-located VMs or business traffic generated by applications that reside in VM or the host.

1.2.4.1 Migration traffic

- Pre-copy Migration iteratively transfers the state data (CPU, memory, and I/O state) of a VM from the source host to the destination host while the VM continues to run on the source. The traffic is primarily outbound, as data flows from the source to the destination (shown in Figure 1a).
- In Post-copy Migration, the VM is immediately moved to the destination host, and any missing data pages are fetched from the source as needed. The traffic is primarily inbound, with data being requested from the source to the destination (shown in Figure 1b).



(a) Pre-Copy

(b) Post-Copy

Figure 1: Traffic directions in Pre-Copy and Post-Copy migrations

1.2.4.2 Application Traffic

- **Incoming traffic** refers to the data packets received by a system or network interface. In VMs and applications, incoming traffic typically includes requests, data, or communications initiated by external entities (Cui, Zhu, et al. 2020). This traffic is critical for the operation of client-server applications and services hosted within the VMs, as it represents the demand side of the service interactions.

- **Outgoing traffic** refers to the data packets transmitted from a system or network interface. For VMs and applications, outgoing traffic includes responses, data, or communications sent out from the VMs to external entities. Outgoing traffic is essential for delivering services and content from the VMs to the end users, forming the supply side of the service interactions (Deshpande and Keahey 2017).

1.2.4.3 Traffic Contention Problem

Both migration traffic and workload traffic share the same network bandwidth (shown in Figure 1). When they occur simultaneously, they compete for the limited network resources (ibid.). In pre-copy migration, the outbound migration traffic competes with the outbound application traffic originating from VMs at the source node (shown in Figure 2a), which can degrade both the migration process and the application performance (shown in Figure 3). In post-copy migration, the inbound migration traffic at the destination competes with the inbound application traffic to VMs (shown in Figure 2d, leading to similar degradation in performance and increased migration time (shown in Figures 3) (Cui, Zhu, et al. 2020).

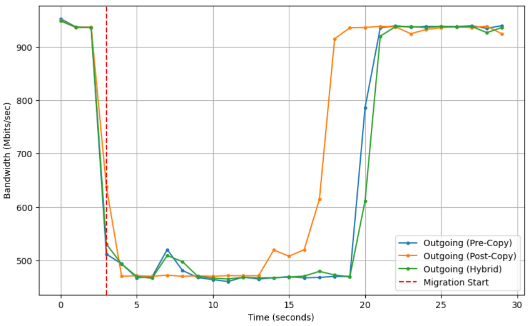
Prolonged migration duration due to network contention can delay the completion of the migration process, thereby delaying the release of resources at the source host. In gang migration scenarios, where multiple VMs are migrated concurrently, the combined migration traffic can severely impact the available bandwidth, further worsening the contention problem.

Network-bound applications experience degraded performance due to the reduced bandwidth available for their traffic, leading to slower response times and potentially affecting the quality of service. Both client-server applications and other network-intensive applications running on the same host can suffer from reduced throughput and increased latency due to this directional traffic contention.

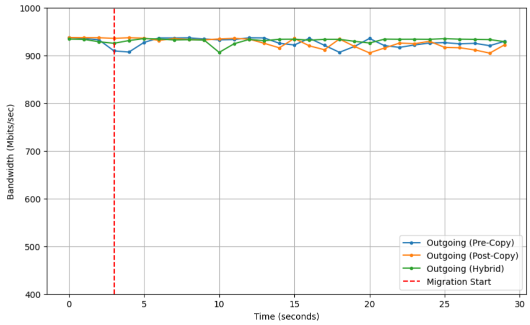
1.3 Related Work

Shrivastava et al. 2011 proposed *AppAware*, a live migration approach that considers network topology, communication patterns of VMs, and physical server limitations to optimize migration destinations based on application-specific communication patterns. Similarly, Cui, Z. Yang, et al. 2017 introduced a topology-adaptive Data Center Network (DCN) that constructs adaptive network topologies based on VM demands and traffic, dynamically adjusting to meet current demands. Tso et al. 2014 proposed SCORE, an approach to minimize overall communication between VMs by treating VM communication as an optimization problem and using a distributed migration technique that adapts to traffic changes. These approaches focus on optimizing network topology and communication patterns to reduce traffic.

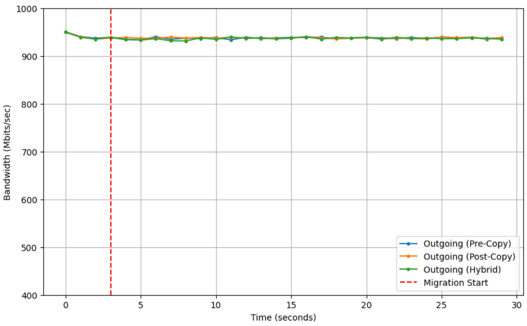
A system and traffic-aware live VM migration algorithm for load balancing (ST-LVM-LB), a graph-based approach to monitor resources and load balance by migrating the least loaded VMs, considering network bandwidth and current flow



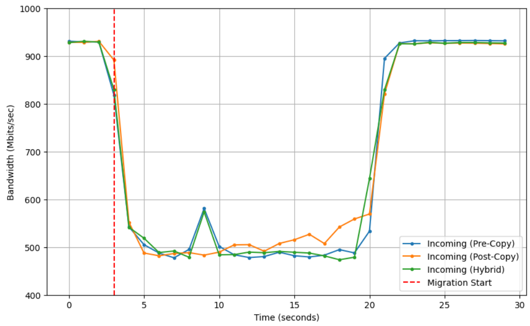
(a) Outgoing traffic at the source node



(b) Incoming traffic at the source node



(c) Outgoing traffic at the destination node



(d) Incoming traffic at the destination node

Figure 2: Bandwidth fluctuation by live migration with the presence of traffic

is proposed in Kanniga Devi, Murugaboopathi, and Muthukannan 2018. Similarly, Fu et al. 2019 introduced NTVMM, which reduces traffic and network load through algorithms for VM selection and placement, focusing on the highest traffic-generating VMs and optimizing their placement. Liu et al. 2015 addressed network congestion in cross-site live migration of multiple VMs by framing it as a MILP problem, considering migration and inter-VM communication traffic, and enhancing the algorithm with *MinUti-O* to reduce complexity. Nasim and Kassler 2015 proposed using multipath-TCP and queue management strategies to optimize traffic-aware live migration, reducing downtime and migration time. These approaches focus on load balancing, resource monitoring, and managing network congestion through various optimization techniques.

Maheshwari et al. 2018 introduced ShareOn, a traffic-aware container migration algorithm that selects target nodes for migration based on current traffic conditions to minimize network traffic. Deshpande and Keahey 2017 proposed a traffic-aware VM live migration algorithm that monitors application and migration traffic to decide between pre-copy or post-copy migration techniques. Cui, Zhu, et al. 2020 proposed an adaptive migration algorithm selection framework using fuzzy clustering to categorize VMs based on business traffic and select the most appropriate migration algorithm. These approaches dynamically adapt based on traffic, optimizing migration performance through adaptive techniques.

Fernando, P. Yang, and Lu 2020 proposed a live migration approach that considers migration urgency in reserving bandwidth for the migration. Nadeem, Elazhary, and Fadel 2018 factors the priority for VM selection, by selecting to migrate only low-priority tasks, keeping the high-priority ones at the host. Dalvandi, Gurusamy, and Chua 2015 proposed an algrithm that takes the time requested for the VMs, for placement and routing in migrations. Haidri et al. 2022 considered the deadline of the request in selecting (**vms**) to balance the load and ensure the request deadlines are met and Son and Buyya 2018 for VM/flow placement.

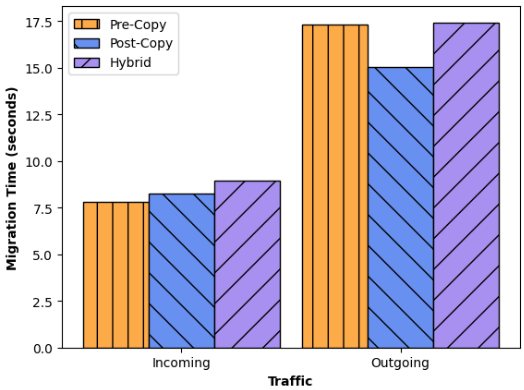
Ongoing research in this field includes several algorithms that consider the effect of traffic on live VM migration. These approaches include selecting traditional migration techniques based on incoming and outgoing traffic, reducing overall migration traffic, and planning the sequence of migrations. Additionally, there are proposed approaches with adaptive network topologies relevant to migrations, including load balancing. Moreover, some recent studies have introduced the concept of migration urgency, which prioritizes the migration process based on how quickly the VMs need to be moved. Despite these advances and numerous promising traffic-sensitive live VM migration algorithms, several challenges and potential performance improvements remain unexplored, particularly in effectively integrating migration urgency with traffic-aware strategies to optimize overall network performance.

With numerous promising traffic-sensitive live VMs proposed, several challenges and performance improvements are yet to be addressed and explored.

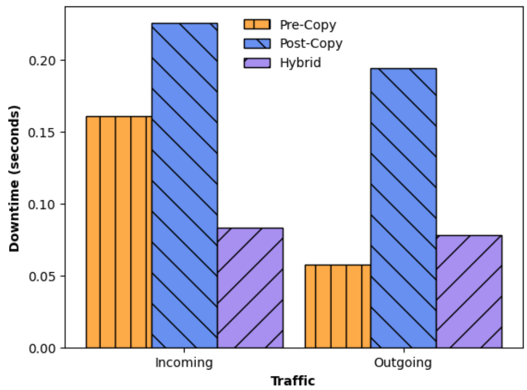
1.4 Research Gap

Currently, there are certain algorithms that consider traffic contention and some that focus on urgency. However, there is a need for algorithms that take both factors into account. Such an approach would dynamically choose migration strategies based on traffic contention (using pre-copy or post-copy techniques) and priority. This would accommodate tasks that require minimal migration time, those that prioritize reduced downtime, and those willing to trade off between migration time and downtime for better application performance.

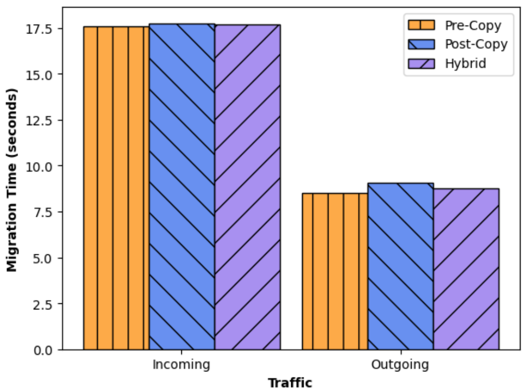
Furthermore, comprehensive studies on the decision-making process for performing serial versus parallel migrations are lacking. This research aims to fill that gap by developing algorithms that consider both network traffic and urgency to optimize migration strategies, thereby reducing migration time, downtime, and application performance degradation.



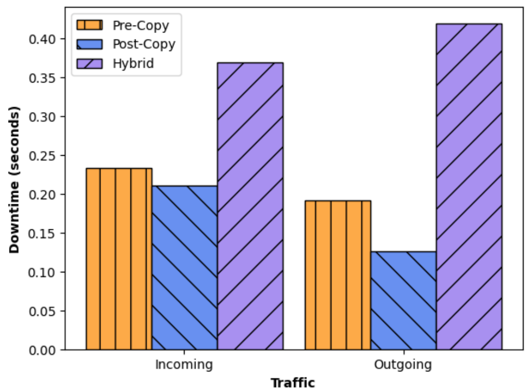
(a) Migration time with traffic at the source node



(b) Downtime with traffic at the source node



(c) Migration time with traffic at the destination node



(d) Downtime with traffic at the destination node

Figure 3: Total migration time and downtime with different traffic types

2 Research Questions

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

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

3. Can the integration of the hybrid copy technique enhance the efficiency of live gang migration when considering traffic contention?

3 Scope

3.1 In Scope

- Live Migration of VMs
  - The focus is on live migration within a KVM/QEMU virtualization environment.
  - Single NIC supported server with a single shared port for migration and application traffic
  - The host OS used for the servers is Linux
  - Includes the live migration of a single VM from one host to another.
  - Covers scenarios involving the simultaneous live migration of multiple VMs between hosts.
- 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 of the migration.

- Evaluating migration performance using industry-expected benchmarks (Sysbench (*Sysbench* n.d.), Memcached (*Memcached* n.d.), iPerf (*iPerf* n.d.)) for migration time, downtime and application performance degradation.

3.2 Out of Scope

- WAN migrations
  - Tests are carried out currently for LAN environments. These techniques can be adapted to WAN environments.
- Servers with a dedicated NIC for migration
  - Some servers use separate NIC interfaces for the application and migration traffic (Fernando, Terner, et al. 2019). In these situations, the migration does not cause any contention with the business traffic. However, this condition is quite rare.

4 Aims & Objectives

4.1 Aim

To develop an adaptive algorithm for traffic-sensitive live migration of multiple VMs that is migration priority aware, and optimizes overall network performance.

4.2 Objectives

- Identify key network metrics (e.g. bandwidth, latency, packet loss) that affect VM migration performance.
- Create an adaptive decision-making algorithm that uses network metrics and migration deadlines to optimize live multiple VM migration.
- Investigate the impact of the hybrid-copy approach on the adaptability of live gang migration techniques.

5 Significance of the Research

This research is significant as it aims to develop an adaptive algorithm for traffic-sensitive live migration of multiple VMs, which is both migration priority aware and optimized for overall network performance. The focus is on making informed decisions about whether to use serial or parallel migration while also addressing the network contention problem.

Additionally, investigating the hybrid-copy approach will provide insights into enhancing the adaptability of live gang migration techniques. Ultimately, this research aims to reduce migration time, downtime, and application performance degradation, thereby improving service provisioning and communication resource efficiency.

6 Research Approach

1. **Preliminary Study:** The initial step involves collecting data using a testbed to analyze how performance metrics fluctuate with changing traffic under different live migration techniques.

2. **Testbed Setup:** As illustrated in Figure 4, the setup consists of two (2) physical servers interconnected with Gigabit Ethernet and has Ubuntu Server as the host OS. KVM/QEMU (*Kernel Virtual Machine* n.d.) is the hypervisor chosen and it has VMs with Linux-based guest OSs. A single network link for both business traffic and migration traffic.

3. **Evaluation of Migration Methodologies Against Traffic:** Analyse the data collected using the testbed. Evaluate the effect of traffic, application and migration, incoming and outgoing traffic, on the migration time, downtime and the application performance.

4. **Develop a traffic-sensitive VM migration algorithm:** Develop an algorithm that optimizes the selection between serial and parallel migration

strategies and the selection of optimal migration techniques based on traffic patterns and migration urgency requirements.

5. **Performance evaluation:** Evaluate the algorithm against traditional migration techniques and existing traffic-aware live multiple VM migration approaches.

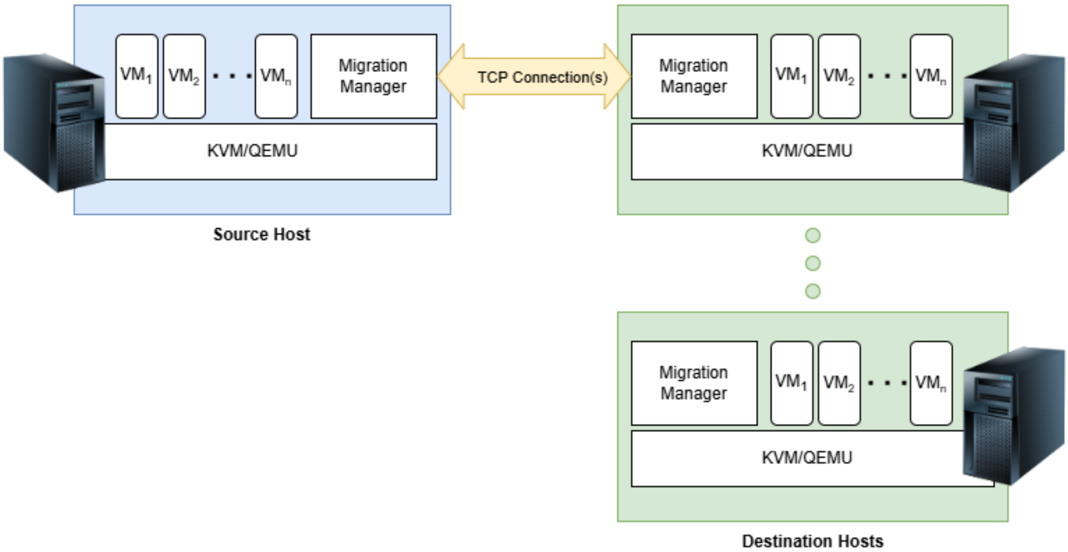


Figure 4: System Architecture

7 Project Timeline

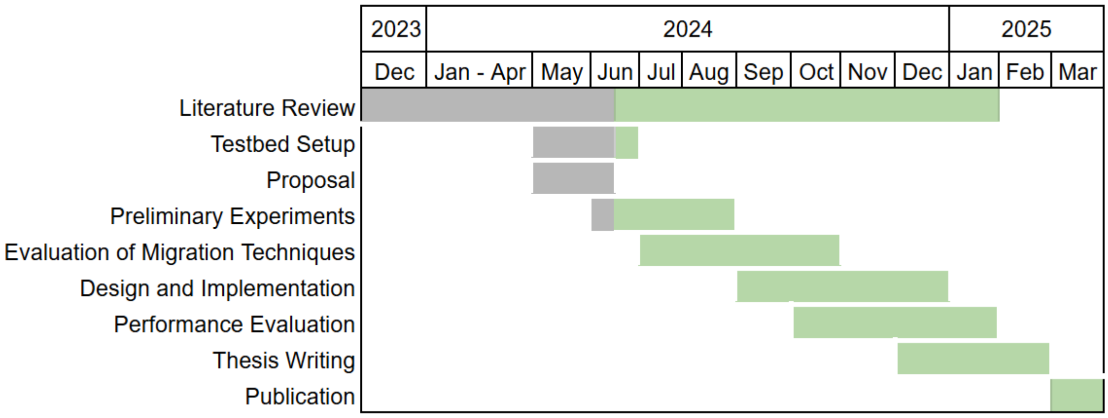


Figure 5: Project Timeline

References

Altahat, Mohammad A. et al. (2020). "Dynamic Hybrid-copy Live Virtual Machine Migration: Analysis and Comparison". In: vol. 171. DOI: 10.1016/j.procs.2020.04.156.

Bahrami, Marziyeh, Abolfazl Toorooghi Haghighat, and Majid Gholipoor (n.d.). *A Review of Virtual Machine Migration Techniques in Data Center*. URL: https://www.researchgate.net/publication/372409749.

Callegati, Franco and Walter Cerroni (2013). "Live migration of virtualized edge networks: Analytical modeling and performance evaluation". In: *2013 IEEE SDN for future networks and services (SDN4FNS)*. IEEE, pp. 1–6.

Chang, Victor, Robert John Walters, and Gary Wills (2013). "Cloud Storage and Bioinformatics in a private cloud deployment: Lessons for Data Intensive research". In: *Cloud Computing and Services Science: Second International Conference, CLOSER 2012, Porto, Portugal, April 18-21, 2012. Revised Selected Papers 2*. Springer, pp. 245–264.

Clark, Christopher et al. (2005). "Live migration of virtual machines". In: *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pp. 273–286.

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.

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.

Dalvandi, Aissan, Mohan Gurusamy, and Kee Chaing Chua (2015). "Time-aware vmflow placement, routing, and migration for power efficiency in data centers". In: *IEEE Transactions on Network and Service Management* 12.3, pp. 349–362.

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.

Deshpande, Umesh, Xiaoshuang Wang, and Kartik Gopalan (2011). "Live gang migration of virtual machines". In: *Proceedings of the 20th international symposium on High performance distributed computing*, pp. 135–146.

Devi, L Yamuna, P Aruna, N Priya, et al. (2011). "Security in virtual machine live migration for KVM". In: *2011 International Conference on Process Automation, Control and Computing*. IEEE, pp. 1–6.

Fernando, Dinuni, Jonathan Terner, et al. (2019). "Live migration ate my vm: Recovering a virtual machine after failure of post-copy live migration". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, pp. 343–351.

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.

Haidri, Raza A et al. (2022). "A deadline aware load balancing strategy for cloud computing". In: *Concurrency and Computation: Practice and Experience* 34.1, e6496.

Hines, Michael R, Umesh Deshpande, and Kartik Gopalan (n.d.). *Post-Copy Live Migration of Virtual Machines*.

Hines, Michael R and Kartik Gopalan (2009). "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning". In: *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pp. 51–60.

Hu, Liting et al. (2008). “Magnet: A novel scheduling policy for power reduction in cluster with virtual machines”. In: *2008 IEEE International Conference on Cluster Computing*. IEEE, pp. 13–22.

Huang, Wei et al. (2007). “Virtual machine aware communication libraries for high performance computing”. In: *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, pp. 1–12.

iPerf (n.d.). URL: https://iperf.fr/. (accessed: 06.06.2024).

Jo, Changyeon et al. (2013). “Efficient live migration of virtual machines using shared storage”. In: *ACM Sigplan Notices* 48.7, pp. 41–50.

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.

Kernel Virtual Machine (n.d.). URL: https://linux-kvm.org/page/Main_Page. (accessed: 06.06.2024).

Kuno, Yosuke, Kenichi Nii, and Saneyasu Yamaguchi (2011). “A study on performance of processes in migrating virtual machines”. In: *2011 Tenth International Symposium on Autonomous Decentralized Systems*. IEEE, pp. 567–572.

Le, Tuan (Nov. 2020). *A survey of live Virtual Machine migration techniques*. DOI: 10.1016/j.cosrev.2020.100304.

Liu, Jiaqiang et al. (2015). “Traffic aware cross-site virtual machine migration in future mobile cloud computing”. In: *Mobile Networks and Applications* 20, pp. 62–71.

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.

Memcached (n.d.). URL: https://memcached.org/. (accessed: 06.06.2024).

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.

Nagarajan, Arun Babu et al. (2007). "Proactive fault tolerance for HPC with Xen virtualization". In: *Proceedings of the 21st annual international conference on Supercomputing*, pp. 23–32.

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.

Nathuji, Ripal and Karsten Schwan (2007). "Virtualpower: coordinated power management in virtualized enterprise systems". In: *ACM SIGOPS operating systems review* 41.6, pp. 265–278.

Nelson, Michael, Beng-Hong Lim, Greg Hutchins, et al. (2005). "Fast Transparent Migration for Virtual Machines." In: *USENIX Annual technical conference, general track*, pp. 391–394.

Padala, Pradeep et al. (2007). "Adaptive control of virtualized resources in utility computing environments". In: *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, pp. 289–302.

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.

Soni, Gulshan and Mala Kalra (Dec. 2013). "Comparative Study of Live Virtual Machine Migration Techniques in Cloud". In: *International Journal of Computer Applications* 84 (14), pp. 19–25. DOI: 10.5120/14643-2919.

Sun, Gang, Dan Liao, Vishal Anand, et al. (2016). "A new technique for efficient live migration of multiple virtual machines". In: *Future Generation Computer Systems* 55, pp. 74–86.

Sun, Gang, Dan Liao, Dongcheng Zhao, et al. (2015). "Live migration for multiple correlated virtual machines in cloud-based data centers". In: *IEEE Transactions on Services Computing* 11.2, pp. 279–291.

Sysbench (n.d.). URL: `https://github.com/akopytov/sysbench`. (accessed: 06.06.2024).

Tsakalozos, Konstantinos et al. (2017). "Live VM migration under time-constraints in share-nothing IaaS-clouds". In: *IEEE Transactions on Parallel and Distributed Systems* 28.8, pp. 2285–2298.

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.

Voorsluys, William et al. (2009). *Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation*. URL: `http://www.cloudbus.org`.

Wood, Timothy, KK Ramakrishnan, et al. (2014). "CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines". In: *IEEE/ACM Transactions On Networking* 23.5, pp. 1568–1583.

Wood, Timothy, Prashant J Shenoy, et al. (2007). "Black-box and Gray-box Strategies for Virtual Machine Migration." In: *NSDI*. Vol. 7, pp. 17–17.