A Taxonomy of Live Migration Management in Cloud Computing

TIANZHANG HE and RAJKUMAR BUYYA[*], Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia

Cloud Data Centers have become the key infrastructure for providing services. Instance migration across different computing nodes in edge and cloud computing is essential to guarantee the quality of service in dynamic environments. Many studies have been conducted on dynamic resource management involving migrating Virtual Machines to achieve various objectives, such as load balancing, consolidation, performance, energy-saving, and disaster recovery. Some have investigated to improve and predict the performance of single live migration. Recently, several research studies service migration in edge-centric computing paradigms. However, there is a lack of taxonomy and survey that focuses on the management of live migration in edge and cloud computing environments. In this paper, we examine the characteristics of each field and propose a migration management-centric taxonomy to provide a holistic framework and guideline for researchers on the topic, including the performance and cost model, migration generations in resource management algorithms, migration planning and scheduling, and migration lifecycle management and orchestration. We also identify research gaps and opportunities to improve the performance of resource management with live migrations.

CCS Concepts: • **Computer systems organization** → **Distributed architectures**; **Dependable and fault-tolerant systems and networks**; **Reliability**; **Availability**; • **Networks** → **Network management**; **Cloud computing**; **Network monitoring**.

Additional Key Words and Phrases: Live migration management, service management, resource management, edge computing, cloud computing

ACM Reference Format:
TianZhang He and Rajkumar Buyya. 2023. A Taxonomy of Live Migration Management in Cloud Computing. *ACM Comput. Surv.* 00, JA, Article 00 (August 2023), 33 pages. https://doi.org/10.1145/3615353

1 INTRODUCTION

The emergence of cloud computing has facilitated the dynamic provision of computing, networking, and storage resources to support the services on an on-demand basis. Traditionally, the process directly running on the operating systems is the foundational element to host the service by utilizing the resources. With the development of virtualization, Virtual Machines (VM), as one of the major virtualization technologies to host cloud services, can share computing, networking, and storage resources from the physical machines. In addition, the container is the emerging virtualization instance to support a more elastic services framework due to its flexibility and small footprint [56]. Application providers can lease virtualized instances (VMs or containers) from cloud providers

[*] The corresponding author.

Authors' address: TianZhang He, hetianzhang91@gmail.com; Rajkumar Buyya, rbuyya@unimelb.edu.au, Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Parkville, VIC, Australia, 3010.

0360-0300/2023/8-ART00 $15.00
https://doi.org/10.1145/3615353

with various flavors under different Service Level Agreements (SLAs). Then, the VM or container managers initialize the instances and the cloud broker or orchestrator selects the feasible placement based on the available resources and the allocation policy.

Under highly dynamic environments, cloud providers need to prevent the violation of the Service Level Agreement (SLA) and guarantee the Quality of Service (QoS), such as end-to-end delay, task processing time, etc. Therefore, there have been extensive works [108, 120] focusing on dynamic resource management in performance, accessibility, energy, and economy in order to benefit both cloud computing subscribers and providers. Live migration of process, VM, container, or storage is the key feature to support dynamic resource management in both edge and cloud computing environments. It can migrate and synchronize the running state of the instance, such as VM or container, from one host to another without disrupting the services [16]. Live migration provides a generic approach without any application-specific configuration and management. Many works have been focused on the different objectives of resource management through live migration in both cloud [120] and edge computing environments [47, 73, 86, 108], such as load balancing, over-subscription, consolidation, service response time, networking, energy, disaster recovery, and maintenance for hardware and software updates.

Commercial cloud infrastructure and services providers, such as AWS, Azure, Google, IBM, RedHat, etc, have been integrating live VM and container migration [4, 5, 75, 87]. For example, to make the compute infrastructure cost-effective, reliable, and performant, Google Cloud Engine introduced dynamic resource management for E2 VMs through performance-aware live migration algorithms [5]. Google has adopted live VM and container migration into its cluster manager [75, 105] for the purposes, such as higher priority task preemption, kernel, and firmware software updates, hardware updates, and reallocation for performance and availability. It manages all computing tasks and container clusters with up to tens of thousands of physical machines. A lower bound of 1,000,000 migrations monthly has been performed with 50 ms average downtime during the migration [87]. AWS regularly performs routine hardware, software, power, and network maintenance with minimal disruption via live migrations [2]. Amazon EC2 uses live migration when running instances that need to be moved from one server to another to dynamically manage CPU resources, optimize the placement of instances, or maintain hardware. Microsoft Azure also utilizes live migration to improve virtual machine resiliency with predictive ML [3]. ML model predicts disk failures and computing node failures [116], and proactively transfers instances to new hosts via live migrations. The IBM Cloud Infrastructure Center provides the IaaS management of non-containerized and containerized workloads. It supports live VM migration with shared storage or block live migration only within the same availability zone [1]. However, it lacks migration management for IaaS users, such as fine-tuning of live migration mechanisms and network management (e.g. traffic engineering).

From cloud to edge computing, the processing resources and intelligence have been pushed to the edge of the network to facilitate time-critical services with higher bandwidth and lower latency [50, 94]. With the combination of different paradigms, live migration can be performed between edge servers, physical hosts in the LAN network, or different data centers through the WAN [41]. For example, consolidation through live migrations between different physical hosts within a cloud data center can reduce the overall network communication latency and energy cost of both hosts and network devices [11]. Live migrations between data centers through WAN aim to optimize performance [37, 71, 82, 111], such as delay, jitter, packet loss rate, response time, as well as energy cost, running cost, regulation adoption, and evacuation before disasters [104]. The mobility-induced migration [86, 108, 120] in edge computing is based on the user position and the coverage of each edge server and its base stations. When the position of the end-user

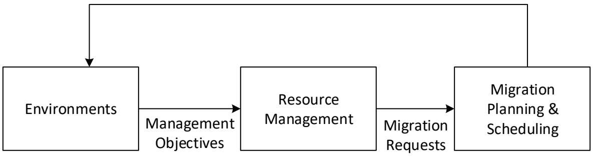


Fig. 1. A general migration management framework

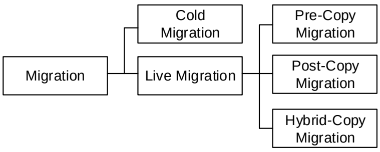


Fig. 2. Categories of migration types

changes dramatically, the end-to-end latency will be suffered. As a result, the service may need to be migrated from the previous edge servers to the adjacent ones.

As the state transmission and synchronization are through the network, the performance of live migration heavily relies on the network resource, such as bandwidth and delay. The edge and cloud data center network has been established to provide data transmission for both live migration and service connectivity. However, with the expansion of edge and cloud computing, tens of thousands of nodes connect with each other, which makes it difficult to manage and configure the networking resource at scale. To overcome the network topology complexity, Software-Defined Networking (SDN) [112] is introduced to provide centralized networking management by separating the data and control plane in the traditional network devices. The SDN controllers can dynamically update the knowledge of the whole network topology through the southbound interfaces based on the OpenFlow protocol. It also provides northbound interfaces for high-level networking resource management such as flow routing, network separation, and bandwidth allocation. As a result, it provides fine-grained network resource management for applications and resource management policies [44], and migration scheduling [42, 43, 107].

This article focuses on the research of migration management during dynamic resource management in edge and cloud computing environments. Figure 1 illustrates the general migration management workflow. Based on the various objectives, the resource management algorithms find the optimal placement by generating multiple live migrations. With the generated multiple migration requests, the migration planning and scheduling algorithms optimize the performance of multiple migrations, such as total and individual migration time and downtime, while minimizing the migration cost and overheads, such as migration influence on application QoS. On the other hand, the computing and networking resources are reallocated and affected by multiple migrations.

For migration management, it is essential to minimize migration costs and maximize migration performance, while achieving the objectives of dynamic resource management. Since dynamic resource management requires multiple instance migrations to achieve the objectives, we investigate migration management in the context of multiple migrations solutions and challenges. Based on the proposed taxonomy, we also review related state-of-art works in each category and identify the gaps and future research opportunities.

Although many surveys [63, 64, 78, 83, 86, 91, 98, 108, 114, 117, 120] of live migration have been presented in the contexts of performance, mechanism, optimization of single live migration, and general migration-based dynamic resource management, they only focus on the specific migration aspects and neglect the aspects of migration management including migration request generation in resource policies and migration planning and scheduling algorithms. Furthermore, there is no literature focusing on the taxonomy of migration management in a systematic way. The purpose of the proposed taxonomy is to fill the gaps between live migration techniques, dynamic resource management, and migration scheduling and planning, and to propose a framework for future research on resource management through live migration in edge and cloud computing

Table 1. Summary of related surveys in live migration

reference	Migration Issues				Resource Type				Environment		Granularity	
	cost	mechanism	application	management	VM	container	network	storage	cloud	edge	single	multiple
[98]	✓	✓			✓				✓		✓	
[91]		✓			✓		✓				✓	
[114]	✓	✓			✓				✓		✓	
[78]		✓	∂		✓	∂	∂	∂	∂		✓	
[64]		✓			✓	✓					✓	
[117]		✓			✓		∂	∂			✓	
[120]	✓	✓	✓		✓		✓	✓	✓		✓	∂
[108]		∂	✓	∂	✓	✓	∂			✓	✓	
[83]		✓			✓				✓		✓	∂
[86]	✓		✓		∂	∂	✓			✓		
[63]	✓	✓			✓				✓		✓	∂
Our work	✓	∂	✓	✓	✓	✓	✓	∂	✓	✓	✓	✓

✓ denotes broad discussion or the main scope of the respective issue.
∂ denotes partial discussion or the secondary scope on the respective issue.

environments. In this paper, we identify the following five aspects of migration management in both edge and cloud computing environments:

- migration performance and cost model
- resource management policy and migration generation
- migration planning and scheduling
- scheduling lifecycle management and orchestration
- evaluation method and platforms

The rest of the paper is organized as follows: We compare and summarize related surveys in Section 2. Section 3 introduces critical concepts of live migration techniques and migration running environments. Section 4 presents the proposed taxonomy of migration management. Section 5 describes the details of essential aspects of migration planning and scheduling algorithms. We review the work related to migration management in Section 6, including migration generation in dynamic resource management and migration planning and scheduling. We analyze gaps within current studies and identify future directions in Section 7. We conclude the paper in Section 8.

2 RELATED SURVEYS

In this section, we introduce the details of related surveys in the context of live migration. Several surveys are conducted to investigate and summarize the works on various aspects of live migration, including migration elements, migration types, migration overheads, optimization mechanisms, motivations and objectives, robustness and security, networking continuity, etc. We summarize the related surveys and illustrate the level of detail covered on the respective issue in Table 1. The cost category includes migration performance and overhead costs. The mechanism category includes migration type and optimization of performance or overhead. The application category includes migration motivations and use cases for resource management, and the migration management aspects include migration generations with different resource policies, policy-level migration networking management, as well as planning and scheduling of single and multiple migrations. The migration granularity consists of single and multiple migrations for heterogeneous and homogeneous migration types, co-located VMs in the same source and destination, and VMs from and to arbitrary hosts. As migration research has evolved in various ways, the main topics discussed extensively in related surveys cover only a limited range of existing works at the time of publication.

Strunk [98] investigated and reviewed the works on the single VM live migration parameters in terms of the physical machine's CPU and net utilization, VM's CPU and net utilization, memory size, and dirty page rate, a taxonomy of migration cost parameters, performance prediction modeling, migration overheads such as service performance loss and migration energy consumption. Similarly, Xu et al. [114] also reviewed and summarized the performance overheads of live VM migration in the Infrastructure-as-a-Service (IaaS) cloud. The survey investigated various causes and scenarios for VM performance overheads during migrations and the performance and overhead modeling methods and compared the complexity and effectiveness of various overhead mitigation techniques. Medina et al. [78] reviewed the works focusing on the mechanism of VM migration and process replication for the purpose of high availability. They also reviewed the mechanism and implementation of the hypervisor (Xen, VirtualBox, KVM, and VMWare) that supports live VM migration, live VM and storage migration across the WAN, and live migration use cases of load balancing, overloaded host management, and energy efficiency. They also partially covered a few works on trace/replay techniques and container migration by OpenVZ.

Yamada et al. [117] and Noshy et al. [83] reviewed the live migration mechanisms for pre-copy, post-copy, and hybrid migration and corresponding optimizations, such as memory compression, deduplication, checkpoint/restore or trace/replay, pipeline and multicore parallelization. Noshy et al. [83] also mentioned the research directions on live migration of multiple VMs in total migration time and impacts on co-located VMs. Zhang et al. [120] presented a comprehensive live VM migration survey mainly talking about migration motivations, migration types, and optimization techniques, network layer-2 to layer-4, and SDN solutions for the network continuity issue over the WAN. They also reviewed the papers on multiple migrations on optimizing the total migration time for co-located VMs in the same host and VMs with network connections. The authors also partially reviewed the works on migration overheads on other services and had limited coverage of the difference between single and multiple migrations. Le [63] also presented a comprehensive survey on live VM migration mechanisms and optimizations. Co-located VM migrations are reviewed in the context of migration optimization. The author compared the performance between pre-copy and post-copy migrations and reviewed the works on single migration models and investigated migration applications in commercial cloud platforms. The survey also identified some directions for live migration optimization without proper reference and citation which had already been covered by other papers, such as adaptive stopping conditions for pre-copy [42], and ML-based working set prediction [119].

For live container migration, Milojičić et al. [79] summarized the early efforts for live process migration and indicated the challenges that process migration need to be solved. From the High Availability (HA) perspective, Li et al. [64] compared VM and container in virtualization mechanisms and implementation differences between the hypervisor and container-based platform for live migration. The authors also reviewed the works of failure detection based on CRIU and OpenVZ and pointed out the HA and optimization features missed in container-based platforms.

For the survey of mobility-induced service migration focusing on service and user mobility, Machen et al. [73] reviewed the works of live service migration in Mobile Edge Computing (MEC). They investigated the layered framework, live container migration, and the performance evaluation of various applications, including gamer servers, RAM simulation, video streaming, and face detection (OpenCV). Wang et al. [108] reviewed the works of dynamic resource management in the name of service migration in MEC. The authors mainly focus framework of service migration, data transmission optimization, and strategies for service migration decisions. Specifically, the authors reviewed the strategies for service migration (dynamic service placement) in edge computing in (1) following the mobile users, (2) MDP-based service migration with one and two-dimensional optimization, and (3) time window-based service migration. In addition, Rejiba et al. [86] also focused on mobility-induced service migration in edge computing. They proposed a taxonomy and review of dynamic resource management algorithms with mobility-induced migration generation based on different objectives, such as cost-related migration tradeoffs and avoidance, and migration success rate due to user mobility. They mixed the system-level single migration optimizations into the orchestration-level taxonomy of resource management. However, only focusing on single migration generation, these surveys neglect migration management in terms of migration scheduling models in dynamic resource management and multiple migration planning and scheduling.

For security and robustness, Shetty et al. [91] focused on live migration security, including secure live migration, control policies (DoS, Internal, Guest VM, false resource advertise), transmission channel (insecure and unprotected), and migration module (stack, heap, integer overflow). Zhang et al. [120] reviewed the robustness aspect of different migration types. Kokkinos et al. [62] focused on live migration in long-distance networks (WAN) for disaster recovery.

In summary, most surveys only investigated the single live migration at the OS system level and dynamic resource management through live migration. Therefore, we summarize and proposed taxonomy and review the representatives of live migration in the context of migration management in edge and cloud computing, including the performance and cost model, migration generations in resource management algorithms, migration planning and scheduling, and migration lifecycle management and orchestration.

3 LIVE MIGRATION BACKGROUND

As there are notable surveys covering live migration techniques and optimization mechanisms, this section only reviews the critical concepts and running environments of live migration.

3.1 Migration Runtime Environment

For *VM migration*, libvirt, as an open-source toolkit for hypervisor management, is widely used in the development of cloud-based orchestration layer solutions. Integrating with hypervisors, such as KVM/QEMU, one can track the details of live migration through management user interface command *virsh domjobinfo* including dirty page rate, expected downtime, iteration rounds, memory bandwidth, remaining memory, total memory size, etc. CRIU [17] is the de-facto software for *live container migration*. It relies on the *ptrace* to seize the processes and injects the parasite code to dump the memory pages of the process into the image file from the address space of the process. Additional states, such as task states, register, opened files, and credentials, are also gathered and stored in the dump files. P.Haul is an extension to CRIU that makes live migration with CRIU possible by implementing management of each live migration phase. Baruah et. al [10] discusses the challenges of scaling the core count on a single Graphics processing unit (GPU) chip and proposes a solution called Griffin, which introduces modifications to the Input-Output Memory Management Unit (IOMMU) and GPU architecture to improve the performance of multi-GPU systems. Griffin enables efficient runtime page migration based on locality information, leading to better load balancing and up to a 2.9× speedup on multi-GPU systems with low implementation overhead. Shi et al. [93] propose a system called Memory/disk operation aware Lightweight VM Live Migration (MLLM) that improves the performance of cross data-center migration by reducing dirty data. MLLM predicts disk read and memory write working sets to optimize migration models and data transfer sequences. They showcase the potential use of machine learning and proposes a hierarchical network model. They experimental results demonstrate that MLLM reduces migration time by 62.9% and service downtime by 36.0% compared to existing methods, and the improved working set estimation algorithm reduces memory pre-copy time by 9.32% on average.

3.2 Migration Types

Figure 2 illustrates the categories of migration types. Generally, instance and storage migration can be categorized as cold migration and live migration. For the performance trade-off analysis,

Table 2. Performance comparison of different migration types

mig. type	mig. time	downtime	data	stability	disruption	complexity
cold	long	large	small	high	large	low
pre-copy	long	tiny-small	mid-large	mid	tiny-small	mid
post-copy	short-mid	tiny-large	small-mid	low	tiny-mid	high
hybrid-copy	short-mid	tiny-mid	small-mid	mid	small-mid	high

memory and storage transmission can be categorized into *Push*, *Stop-and-Copy*, and *Pull* phases. Live migration can be further categorized into *pre-copy* [16], *post-copy* [46], and *hybrid migration* [48, 89]. The design and continuous optimization and improvement of live migration mechanisms are striving to minimize downtime and live migration time. **Downtime** refers to the time interval during a migration when a service is unavailable due to the need for state synchronization and network rerouting. For the single migration, **migration time** refers to the time interval between the start of the pre-migration phase to the finish of post-migration phases that instance is running at the destination host. For multiple migrations, **total migration time** is the time interval between the start of the first migration and the completion of the last migration. Table 2 illustrates the performance comparison of different migration types in migration time, downtime, transferred data size, migration stability and robustness, service disruption level, and operational complexity.

Cold Migration: Specifically, although provides simplicity over the live migration solution, it bears the disadvantage that both migration time and downtime are proportional to the amount of physical memory allocated to the VM. It suffers significant VM downtime and service disruption during the migration process. It is usually utilized for scenarios where service downtime is acceptable, such as during scheduled maintenance or when migrating VMs between data centers.

Pre-copy migration: The majority of instance memory pages are transferred and dirty pages are later iteratively copied to the destination host while the instance still running on the source host. The migration time of pre-copy migration depends on the allocated bandwidth, dirty page rate, data compression ratio, memory size, maximum iteration threshold, and downtime threshold [42]. Compared to other migration types, minimal downtime and low disruption can be guaranteed given sufficient bandwidth. The instance at the source server is alive until the copying instance at the destination is ready, making it more robust to be restored after a migration failure. It is typically a better choice for mission-critical workloads and scenarios where minimizing downtime is crucial. As cloud services have become a critical capability for modern businesses, it is important to minimize their service downtime. Therefore, it is critical to find an optimal time to migrate a virtual machine in the pre-copy approach. To address the issue, Haris et. al [40] propose a machine learning-based method to optimize pre-copy migration. Their work is organised in three stages: feature selection, model generation and application of the proposed model in pre-copy migration. Their experimental results show that their approach is able significantly reduce the downtime or service unavailability during the migration process.

Post-copy migration: Only a subset of instance memory pages are initially migrated to the destination host. The remaining pages are transferred in the background while the instance continues to run on the destination host. Depending on workload characteristics [42], it could potentially result in a shorter migration time and less overall downtime. However, it may lead to performance degradation and instability of the migrating instance, and disruption to other instances running on the same host. The complexity of post-copy is generally considered to be higher than pre-copy migration due to the additional coordination and management required to ensure a successful migration. It may be a good choice in scenarios where minimizing migration time is more important than downtime under a stable network environment.

Hybrid post-copy, as an optimization technique based on pre-copy and post-copy migrations, aims to reach the balance point by leveraging all three phases. It starts with pre-copy and then activates the post-copy mode when the memory copy iteration does not achieve a certain percentage increase compared with the last iteration. It could reduce the migration time with slightly increased downtime in certain situations depending on the application workloads. However, it bears the same disadvantages of post-copy migration that pulling faulted pages slow down the processing speed which may degrade the QoS, and VM reboot may occur when the network is unstable.

From the **instance storage** perspective, live migration can be categorized into *shared storage-based* live migration, the instance has ephemeral disks that are located on storage shared between the source and destination hosts; *block live migration* (block migration), the instance has ephemeral disks that are not shared between the source and destination hosts; and *volume-backed* live migration. Shared-storage and volume-backed live migration does not copy disks. Block live migration requires copying disks from the source to the destination host. As a result, the migration takes more time to converge and puts more load on the network. Most of the research neglect storage migration costs with the assumption that the storage is shared between source and destination hosts.

In general, when selecting a migration technique, it is important to consider factors such as the size of the migrating instance, the workload of the application, the available network bandwidth between the source and destination hosts, and the required downtime and disruption to the migrating instance and other instances running on the same host.

3.3 Migration Network Environment

Migration Span indicates the geographic environment where live migrations are performed. It is critical to analyze the migration span since various computing and networking settings and configurations directly affect migration management. We categorize live migration based on the migration span into LAN (Layer-2), such as intra-data center, and WAN (Layer-3) environment, such as inter-data center and edge-cloud migrations.

Intra-Cloud: The source and destination hosts of intra-cloud migration are in the same LAN environment. Hosts often share the data via Network-Accessed Storage (NAS), for example, NFS, Ceph, shared LUNs, and GlusterFS. It excludes the need for storage transmission or block live migration. In addition, for a share-nothing data center architecture, live migration flows are separated from the tenant data network via a dedicated management network to alleviate the network overheads of migrations on other services.

Inter-Clouds & Edge-Cloud: Compared with live migration in LAN, migration via WAN environments faces more challenges in network continuity, connectivity and security, and data transmission due to bandwidth limitation and storage migration [62, 120]. For networking continuity, connections to migrating instances should be kept alive. IP address attached to the migrating instance should also be migrated to keep connectivity (e.g. *keepalived*). For network security, the live migration stream should be secured via public key encryption (e.g. QEMU-native TLS). For data transmission issues, the network bandwidth and paths of migration flows can not be guaranteed when source and destination servers are connected via the Internet. Providing a dedicated backbone network between data centers, the allocated bandwidth of live migrations can be managed. There is often no shared storage and dedicated migration network between the data center sites in WAN environments. Therefore, live storage migration is necessary while live instance migration focuses on memory synchronization. It also applies to the architecture without shared storage in LAN.

Edge Computing: It includes both LAN and WAN architecture. In the edge WAN solutions, edge data centers are connected through WAN links as the traditional inter-data center architectures. With the emerging cloud-based 5G solution [53], edge data centers can be connected through hybrid wireless and dedicated backbone links and shared with the regional cloud data center and network storage. The motivations and use cases of dynamic resource management in edge computing are similar to those in cloud computing environments. On the other hand, live migration at edges is often referred to as service migration focusing on the mobility-induced migrations in Mobile Edge Computing (MEC) [86, 108]. When the end user moves away from the edge server where the service is allocated, service migrations are generated to guarantee the service end-to-end delay. However, compared to the periodical resource management in cloud data centers, it lacks proper management for multiple migration scheduling with stochastic arrival patterns in edge computing.

3.4 SDN-enabled Solutions

By decoupling the networking software and hardware, SDN can simplify traffic management and improve the scheduling performance of live migrations in intra-data centers (SDN-enabled data centers) [43, 44, 96] and inter-data centers (SD-WAN) [43, 54, 107]. In general, SDN-enabled solutions can reduce the migration overheads on the system and other applications when generating migration requests during dynamic resource management [44]. Integrating SDN into computing management can also improve the performance of concurrent multiple migration planning and scheduling in dynamic and complex networking environments [43, 107]. Specifically, SDN can provide several benefits for live migration management:

Centralized Control: In an SDN-enabled environment, network control is centralized in a software controller, which can provide a unified view of the network topology and resources, which is critical to migration orchestration and resource allocation optimization.

Network Visibility: It provides better network visibility, such as the network status of individual nodes and the flows of network traffic of migrations and other instances. It can also discover and monitor application-level communication (virtual links). Based on current network conditions, more intelligent strategies about when, where, and how to perform live migrations can be performed.

Dynamic Resource Allocation: With SDN, the migration orchestrator can dynamically configure routes and allocate bandwidth to live migration and other application flows. It reduces the network overheads of migrations on other networked services and ensures sufficient resources are available for migrations.

Cost Savings: SDN can reduce the operational cost of live migration, by optimizing the use of network resources and reducing the need for dedicated migration infrastructure. This can help organizations optimize their resource utilization and reduce costs associated with live migration in their fog, edge, and cloud computing.

Automation and Resilience: The intent framework provided by SDN allows the migration orchestrator to specify high-level user intent for resource allocation and automate network configuration and optimization. It can also improve network resilience by utilizing real-time monitoring and network provisioning to agilely detect and respond to network issues, ensuring that the live migration continues to operate even in the face of unexpected network disruptions.

4 TAXONOMY OF MIGRATION MANAGEMENT

Based on the existing literature, we categorize migration management into five aspects, as shown in Fig. 3, namely: (1) performance and cost model; (2) policy and migration generation; (3) migration planning and scheduling; (4) migration lifecycle and orchestration and (5) evaluation method.

4.1 Migration Performance and Cost Modeling

The migration performance and cost modeling is the fundamental element for migration management to evaluate and predict the total cost and scheduling performance of multiple migration requests. Based on the related literature and our observation, this section identifies the *parameters*, *metrics*, and *modeling methods* involved in the performance and cost model of live migrations.

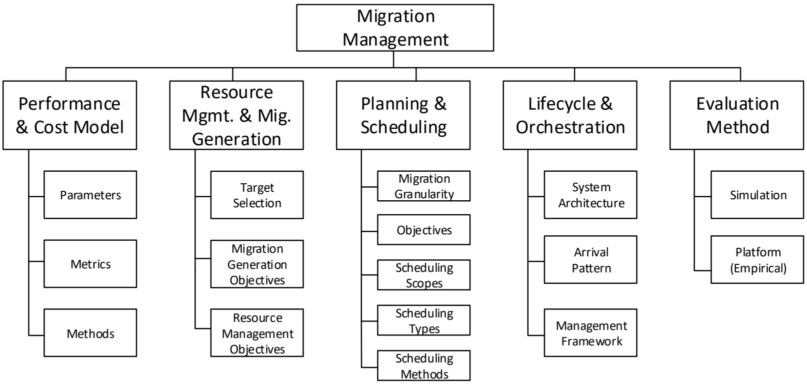


Fig. 3. Taxonomy of migration management

Table 3. Parameters of the live migration model

Category	Parameters				
Computing	CPU utilization	memory utilization	I/O interface	WSS size	
Networking	bandwidth	interfaces	routing	delay/distance	layers/hops
Storage	sharing	data size	storage type	read/write speed	
Single migration	dirty page rate	iterations threshold	downtime threshold	configuration delay	priority
Multiple migration	migration impact	concurrency	migration time	routing	scheduling window

4.1.1 Parameters. Table 3 illustrates the parameters involved in live migration under three categories: computing, networking, and storage resources. Moreover, we also identify the migration parameters in the granularity aspect: single and multiple migrations. The *computing resource* parameters include CPU load and utilization, memory load and utilization, memory size, dirty page rate, WSS size as frequent updating memory pages for live migration optimization, and I/O interfaces (i.g. cache interface, host network card interface, inter-CPU interfaces). The *networking resource* parameters include the migration routing, available bandwidth (link and routing bandwidth for single and multiple paths), migration distance (the number of network hops), the number of involved network layers, network delay (link delay and end-to-end latency). In addition, the *storage-related* parameters include writing and reading speed and storage data size.

The parameters of single live migration include dirty page rate, the threshold of pre-copy iterations, downtime threshold for pre-copy migration, configuration delay in pre and post-migration processes, and the priority of the migration request. We also need to consider several parameters of multiple migration scheduling, such as The migration impact on other services, the running and subsequent migrations in computing and networking aspects, the concurrency for multiple migration scheduling as resource contention among migrations, the single migration time in multiple migration scheduling, the migration routing considering the traffic of other services and migrations, and the scheduling window of each migration with various priorities and levels of urgency.

4.1.2 Metrics. Many works have investigated the metrics of single migration performance. However, few works are focusing on the performance metrics of multiple migrations. Therefore, we also extend the investigation of the single live migration metrics to the multiple migrations in these categories. As shown in Table 4, we categorize these metrics into different categories, namely time-related, data, QoS, energy, and SLA.

Table 4. Metrics of live migration performance

Category	Metric		
Time	migration time	downtime	deadline/ violation time
Data	dirty memory	storage	stop-and-copy size
QoS	response time	network delay	available resource
Energy	physical host	network device	cooling
SLA	service availability	success ratio	policy performance

Migration Time: is the key parameter used to evaluate the single migration performance and overhead. A large migration time normally results in a large overhead on both computing and networking resources for the migration VMs and other services.

Downtime: is one of the two main parameters used to evaluate the single migration performance. During the downtime caused by migration, the service is not available to the users.

Iteration Number: For migration types utilizing the pre-copy strategy, the number of iteration rounds directly affects the migration converging hence the migration time and downtime.

Data Transmission Size: is the key parameter to judge the network overhead during the migration across the network. For pre-copy migration, it is highly positively correlated to migration time. The total amount of data transmission is the sum of the data amount of each instance. It can be divided into two aspects: memory data and storage data.

Total Migration Time: of multiple migrations is the time interval between the start of the first migration and the completion of the last migration. This is the key parameter to evaluate the multiple migration performance and overheads.

Average Migration Time: is the average value of the sum of the migration time of all instances within the time interval. With the continuous arrival migration requests, the average migration time is preferable to the total migration time. The total migration time of a bunch of instances is only a suitable parameter for the periodically triggered source management strategies.

Average Downtime: Similar to the average migration time, the average downtime is the mean value of the sum of downtime of all instance migrations within a time interval. Time unit, such as millisecond (ms) and second (s), is used for migration time and downtime.

Energy Consumption: consists of the electricity cost, green energy cost, cooling cost, physical host, and networking devices cost. It is a critical metric of live migration overheads used for green energy algorithms and data centers. Joule (J) is used as a unit of energy and Wh or KWh is used in electrical devices.

Deadline Violation: Migration request or task is the key element for multiple migration scheduling. Migrations with different time requirements will have corresponding deadlines and scheduling windows. Therefore, with different priorities and urgency, the number of deadline violations is a key metric for evaluating deadline-aware or priority-aware migration scheduling algorithms.

Resource Availability: The remaining computing, networking, and storage resources during and after the single migration and multiple migrations. It is critical for migration success as there should be sufficient resources for the new instance in the destination. Furthermore, resource availability during the migration affects the performance of the subsequent migrations. Resource availability after the migration is also a metric for the resource reconfiguration evaluation for various policies.

Quality of Service: Response time, end-to-end delay, network delay, and network bandwidth during and after the single migration or each migration during multiple migration scheduling for the migrating service and other services (co-located VMs, shared-resource VMs, connected VMs).

Service Level Agreement: Migration may cause service unavailable due to migration-related issues, such as downtime, network continuity, network robustness, and migration robustness. Therefore, cloud providers provide the SLA to subscribers and tenants as the availability rate for the services with and without migrations. Therefore, SLA violations are another critical metric.

4.1.3 Modeling Methods. This section presents different modeling approaches on migration performance and overhead, including *theoretical* and *experimental modeling (profiling and prediction).*

Theoretical: In theoretical modeling, the system behaviors are described mathematically using formulas and equations based on the correlation of each identified parameters [7, 8, 12, 13, 52, 60, 60, 68, 69, 109, 115]. Some works only model the migration costs and performance based on the correlation of the parameters. Other works follow the behaviors of live migration, such as iterative copying dirty page rate to model, the performance, and overheads in finer granularity.

Profiling: Experimental modeling methods are based on measurements with controlled parameters. Empirical running analysis, such as Monte Carlo, are relied on repeated random sampling and time-series monitoring and recording for migration performance, overheads, and energy consumption profiling [7, 49, 100]. For both overhead and performance modeling, empirical experiment profiling can also derive the coefficient parameters in the model equations [51, 52, 69, 99]. Regression algorithms are also used to model the cost and performance based on the measurement [8, 29].

Prediction: Generally, mathematic cost models can be used to estimate migration performance and overheads. Performance estimation and prediction algorithms [7, 8, 66, 69, 98, 99] are proposed to simulate migrations processes to minimize the prediction error. Furthermore, Machine Learning (ML)-based modeling [28, 55] are adopted to generalize parameters in various resources to obtain a more comprehensive cost and performance prediction model.

4.1.4 Cost and Overhead Modeling. The cost and overhead models of live migration are integrated into the modeling and optimization regarding the objectives of dynamic resource management policies. Similar to the migration parameters, the cost and overhead modeling can be categorized into computing, networking, storage, and energy caused by virtualization and live migration (see survey [114]), migration influence on subsequent migrations and co-located services [9, 32, 38, 88, 115], and networking resource competitions among multiple migrations [9, 102, 107].

4.1.5 Performance Modeling. For resource management and migration scheduling algorithms, the performance models of single and multiple migrations are used to maximize migration performance while achieving the objectives of resource management. The category of the performance model is similar to the performance metrics classification. In other words, performance models can be categorized as migration success rate, migration time, downtime, transferred data size, iteration number, and deadline violation [114]. With the complexity of modeling multiple migrations, it is difficult to model the performance of multiple migrations directly on total migration time. Therefore, multiple migration performance and cost models are focusing on the single migration performance based on the currently available resources during multiple migration scheduling [32, 102] or the involved parameters for the sharing resources, such as shared network routing, shared links, shared CPU, memory, network interfaces, and the number of migrations in the same host [9, 38, 90, 107].

4.2 Migration Generation in Resource Management

In this section, we discuss migration request generation of resource management algorithms. We investigate how the migration performance and overhead models integrated with the policy affect the optimization problems and migration generations in two aspects: *migration target selection* and *migration generation objectives.* Figure 4 illustrates the details of each aspect.

4.2.1 Migration Target Selection. The targets of one migration include the selections of source, destination, instance, and network routing. During the migration generation process of the resource management policy, the targets for migrations can be selected simultaneously or individually.

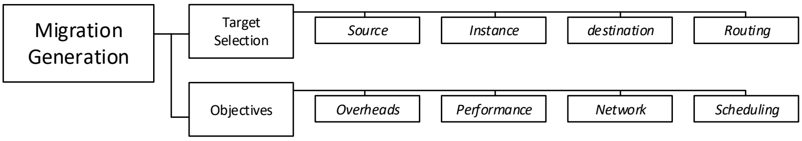


Fig. 4. Categories of migration generation in dynamic resource management

For simultaneous solutions, such as approximation algorithms, migration instances, source or destination hosts, and network routes are generated at the same time. For individual solutions, such as heuristics, migration requests are generated one at a time in each algorithm loop.

Source Selection: The source host or site of one migration request is selected based on the objectives of resource management policies, such as failure, resource utilization, energy consumption, over-subscripted host, and under-subscripted host.

Instance Selection: During the instance selection for migration request, the migration generation algorithm needs to consider the various objectives of resource management policy, the availability of resources in potential destinations, and the overheads of live migration, such as dirty page rate and the number of allowed migrations. For use cases, such as gang migration, disaster recovery, software update, and hardware maintenance, all instances within the source hosts or sites will be selected. In scenarios such as mobility-induced migration, there is no need to select the instance.

Destination Selection: Many works considered the selection of migration destination as a bin packing problem, where items with different volumes need to be packed into a finite number of bins with fixed available volumes in a way to minimize the number of bins used. There are several online algorithms to solve the problems, such as Next-Fit (NF), First-Fit (FF), Best-Fit (BF), Worst-Fit (WF), etc. By considering both objectives of various resource management, such as energy consumption and networking consolidation, and migration overheads and performance, heuristic and approximation algorithms are proposed.

Flow Routing: The available bandwidth and network delay are critical for migration performance, such as migration time and downtime. The migration flows of pre-copy migration are elephant flows and post-copy migrations are sensitive to network delay. Meanwhile, in the network architecture where the migration traffic and service traffic share network links, the migration traffic can significantly affect the QoS of other services. In the SDN-enabled data centers, the allocated bandwidth and routing of both migration and services traffic can be dynamically configured to improve the performance of live migration and minimize the migration network overheads.

4.2.2 Migration Generation Objectives. This section summarizes the different objectives of migration selections in resource management policies from the perspective of live migration management, including *migration overhead*, *performance*, *network*, and *scheduling awareness*. Many works are proposed in dynamic resource management for various objectives, such as performance, networking [21, 45], energy [14, 21, 45], QoS, and disaster recovery [62]. Most works only consider the memory footprint (memory size) and available bandwidth. Without the proper live migration modeling, the selected instance migration requests for reallocation will result in unacceptable scheduling performance and service degradations. In addition, these works with migration performance and energy models [35, 69, 92, 98, 122] only consider the individual migration performance and computing and networking overheads during the migration generation or instance placement selection for the dynamic management policy. The total migration cost, as a result, is a linear model, which can not reflect the concurrency among migrations and resource contentions between migrations and services. As a result, the solution is only optimized for sequential migration scheduling.

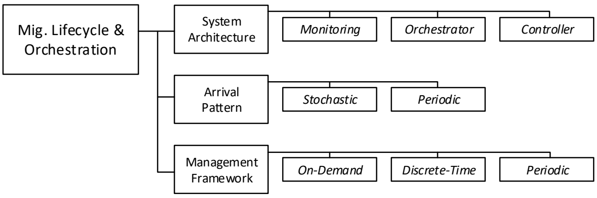


Fig. 5. Categories of migration lifecycle management and orchestration

Overheads-aware: Most works of resource management policies focus on minimizing the cost or overheads of migrations and modeling the total cost as a sum of the individual live migration cost while achieving objectives of resource management, such as energy consumption [11, 14, 110] and network flow consolidation [18, 35]. As shown in Sect 4.1.1, the overheads of the migrating instance can be categorized as computing, networking, and storage overhead, including the total number of migrations, number of co-located instances, memory size, dirty page rate, CPU workloads, data compression rate, bandwidth allocation, and migration flow routing of the migrating instance.

Performance-aware: Some works focus on the migration performance optimization integrating with the objectives of resource management, such as the cost and performance of migrations [18, 19, 35, 90, 115]. However, providing multiple migration requests, existing resource management solutions through live migrations do not consider and guarantee migration scheduling performance.

Network-aware: Apart from network routing and bandwidth allocation, the network contentions among migrations and between migration and applications, as well as instance connectivity need to be considered to optimize the networking throughput and communication cost during or after the migrations [20, 30, 57, 74, 103, 122]. For instance, two migrations may content the same network path which can lead to migration performance degradation, such as larger the total and individual migration time. On the other hand, without a dedicated migration network, migration and application flows also compete for the same network link leading to QoS degradation. For instance connectivity, one migration completion can free up more bandwidth due to flow consolidation. As a result, subsequent migrations can be converged faster with more available bandwidth.

Scheduling-aware: Current works do not consider the migration scheduling performance [9, 43, 107] beyond the linear model of migration cost and interfaces. In the migration generation phase of resource management policies (Fig. 1), we can optimize the performance of single or multiple migration scheduling, and ensure the original resource management policy achieves optimal or near-optimal performance [44]. Compared to policy-aware migration management, it is more adaptive without the need for specific modeling and design. It has less impact on the enormous amounts of existing dynamic resource management policies.

4.3 Migration Lifecycle and Orchestration

Based on various scenarios of dynamic resource management through live migrations, it is critical to investigate the migration lifecycle and orchestration layer including migration arrival patterns and corresponding management framework. Therefore, this section summarizes the migration lifecycle management and orchestration in the several aspects: *arrival pattern*, *orchestration*, *monitoring*, and *management framework*. Figure 5 illustrates the details of each aspect.

4.3.1 Arrival Patterns. Arrival patterns of migration requests based on various paradigms and objectives in fog, edge, and cloud computing environments can be categorized as *periodic* and *stochastic* patterns. For existing dynamic resource management algorithms, the migration generation and arrival patterns are periodic due to the overhead of live migrations. Dynamic resource management, such as regular maintenance and updates, load balancing, and energy-aware consolidation, triggers the reallocation periodically. On the other hand, arrival patterns of event-driven migrations are often stochastic due to the nature of the service. For example, the mobility-induced migration in edge computing is based on user behaviors and movement [86, 108].

4.3.2 Migration Orchestration. The system architecture of edge and cloud data centers consists of an orchestration layer, a controller layer, and a physical resource layer. Several orchestration systems have been proposed to control computing and network resources in fog, edge, and cloud computing environments [22, 33, 76, 77]. The policy orchestrator provides the algorithms and strategies for joint computing and networking resource provisioner, network engineering server, and migration manager (lifecycle manager, migration generator, planner, and scheduler). Network topology discovery, cloud monitoring, and network monitoring are essential for computing and networking provisioning. Combined with the cloud manager and network engineering server, live VM Management software can efficiently control the lifecycle of single migration and schedule migrations in a finer granularity by jointly provisioning computing and networking resources.

The controller layer facilitates the allocation of computing and networking resources. The cloud controller (e.g. OpenStack) and the container control plane (e.g. Kubernetes) are responsible for the autonomous deployment, monitoring, scaling, and management of VMs and containerized applications based on provided strategies in the orchestration layer. On the other hand, the SDN controller manages the OpenFlow switches and flow forwarding entries and collects the device and flow statistics through southbound interfaces. Networking applications through SDN controller northbound interfaces perform topology discovery, network provisioning, and network engineering for both application and migration flows.

4.3.3 Management Framework. Based on the characteristic of resource management policies in various scenarios and use cases, the triggered pattern of migration planning can be categorized into periodic, discrete-time, and on-demand types.

On-demand: In the on-demand framework, the migration will be planned and scheduled whenever the request arrives [86, 108]. The on-demand framework can be applied to scenarios that individuals are the subjects to trigger the migration from one host to another, such as mobility-induced migration and migration requests by public cloud subscribers and tenants.

Periodic: In the periodic planning framework [120], the migration plans are calculated based on the migration requests periodically generated by the dynamic resource management policy. The time interval between each planning can be configured as the value of the resource optimization interval. Within each optimization interval, multiple migration requests generated by the redistribution policy will not be affected by the previous round of migration. The management interval varies from a few hours to a few days. With such large time intervals, all migration instances can be completed before the reallocation time for the planning of new migration requests.

Discrete-Time: In the discrete-time framework, the arrival migration requests are put into queues to regulate the migration arrival speed and processed migration number. The migration planner will periodically read multiple migration requests from the entire or partial schedule-wait queue and read migrations from the schedule-wait queue as input. Then, it calculates the migration schedule based on a configured time interval (e.g., one second). For each input with multiple migration requests, the planner will calculate the migration plan based on the states of computing and networking resources. There is a schedule buffer between the planner and the scheduler. The migration scheduler will schedule these migration requests based on the calculated sequence and current computing and networking states. During the small time interval of each input, some instances of the previous migration plan have not yet been started by the scheduler, which could

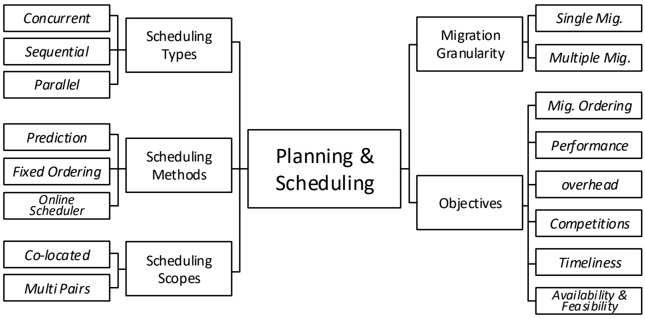


Fig. 6. Categories of mig. planning and scheduling algorithms

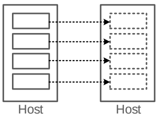


Fig. 7. Co-located migrations

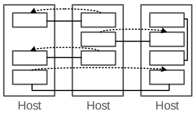


Fig. 8. Multi-pairs migrations

affect the decision of the current planning round. The discrete-time framework is suitable for scenarios where migrations may arrive randomly and much more frequently than traditional dynamic resource management policies (e.g., mobility-induced migrations in edge computing).

5 MIGRATION PLANNING AND SCHEDULING

In this section, we introduce the taxonomy for migration planning and scheduling, including *migration granularity*, *scheduling objectives*, *scopes*, *types*, and *methods*. Figure 6 illustrates the details of each category. Compared to real-time task scheduling, there is enough time for more complex migration scheduling, which further improves multiple migration performance and alleviates migration overheads. When it comes to multiple migrations, based on the objectives of live migration, the migration planning algorithm needs to calculate and optimize the sequence of each migration. In other words, the planning algorithm needs to consider availability, concurrency, correlation, and objective. We review the related state-of-art works in Section 6.2.

5.1 Migration Granularity

The migration granularity in the context of migration planning and scheduling can be categorized into *single migration* and *multiple migrations*.

Single Migration: Only one instance is migrated at the same time. The research scope of single migration is the performance and overhead of individual migration, including migration mechanisms, and optimization techniques in both computing and networking aspects. The key metrics of single migration performance are migration time and downtime. The overheads of single migration include network interfaces, CPU, and memory stress, migration process overheads in the source and destination hosts (i.g. dirty memory tracing overheads), service-level parameters (e.g. response time), and available bandwidth for the migration service and other services in the data center.

Multiple Migrations: In multiple migrations, multiple instances are considered to be migrated simultaneously. Multiple migrations can be divided into various aspects based on the instance and service locations and connectivity, including co-located instances (gang migration) and cluster migrations (same source and destination pair), and related instances (connected services and applications, VNFs in SFC, entire virtual network, VMs in the same virtual data center).

The overheads and performance of multiple migrations should be evaluated and modeled in all migration management phases of the migration generation, planning, and scheduling. The overheads of multiple migrations can be categorized into *service-level* and *system-level* overheads. The service-level overheads include multiple migration influences on the migrating service, subsequent migrations performance, and other services in the data centers, and the system-level overheads include the multiple migration influence on the entire system, such as the availability of networking and computing resources. On the other hand, the performance of multiple migrations can be divided into global and individual performance. Total migration time, downtime, and transferred data size are the critical metrics for the global migration performance of multiple migrations. Performance metrics, such as average migration time, average downtime, and deadline violation, are used for individual performance in multiple migrations.

5.2 Objectives and Scopes

This section summarizes the objectives of migration planning algorithms and scheduling strategies. It can be categorized into *migration ordering*, *migration competitions*, *overhead and cost*, *migration performance*, *migration timeliness*, and *migration availability and feasibility*. We also categorized the scheduling scope into two aspects: co-located scheduling and multiple migration scheduling with multiple source-destination pairs (multi-pairs).

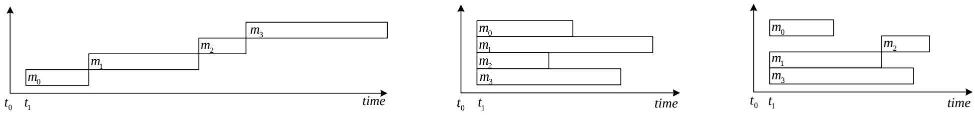
Migration Ordering: The works of migration ordering problems focus on the feasibility of multiple migrations [38, 90] and migration ordering of co-located instances [32, 88] in the one-by-one scheduling solutions. Given a group of migration requests, The feasibility problem addresses whether a given migration can be scheduled, and the scheduling order of these requests due to resource deadlocks. The performance problem in the migration ordering context is finding an optimized order to migrate the co-located instance in order to minimize the overheads and maximize the performance of multiple migrations in a one-by-one scheduling manner.

Migration Competitions: Resource competition problems include the competition among migrations and between migrations and other services during the simultaneous migration scheduling [9, 102, 107]. Since migrations and services are sharing computing and networking resources, it is essential to determine the start sequence of migrations in both sequential and concurrent scheduling manner to minimize resource throttling and maximize resource utilization with respect to both QoS and migration performance. The resource dependencies and competitions among migrations and services need to be considered in both the migration generation phase and the planning and scheduling phase in order to improve the performance of multiple migrations and minimize the overheads of multiple migrations.

Overhead and Cost: It is critical to minimize migration overheads and costs to alleviate the QoS degradations and guarantee the SLA during live migration scheduling. The computing overheads on CPU, memory and I/O interfaces affect the co-located instances negatively. The migrations also share the same network links with other services. It may lead to QoS degradations due to the lower bandwidth allocation. As a result, a longer migration process leads to larger computing (CPU and memory) and networking overheads (network interfaces and available bandwidth). Network management policies and routing algorithms are adopted to dynamically allocate the bandwidth and network routing to migrations. Furthermore, the migration downtime also needs to be managed to avoid unacceptable application response time and SLA violations.

Migration Performance: The performance of multiple migration scheduling is one of the major objectives of migration planning and scheduling algorithms. The total migration time is highly relative to the final management performance. In other words, a shorter migration time leads to a quicker optimization convergence. Furthermore, in green data center solutions, the energy consumptions induced by live migration need to be modeled properly.

Timeliness: The timeliness of the migration schedule is also critical to resource management performance [102, 121]. For migration with various priorities and urgencies, inefficient migration planning or scheduling algorithms may result in migration deadline violations, which leads to QoS degradations and SLA violations. For example, some VNFs need to be migrated as soon as possible



(a) Sequential (b) Parallel (c) Concurrent

Fig. 9. Scheduling types of multiple migration scheduling

to maintain low end-to-end latency requirements. Some migration requests of web services with high latency tolerance and robustness are configured with a much larger scheduling window.

Availability and Feasibility: Migration availability and feasibility problems are also considered in the planning and scheduling algorithms [32, 38, 90]. There should be reserved resources in the destination hosts and sites in order to host the new instance. In the context of multiple migrations, resource deadlock and network inconsistency may also affect the migration success ratio. Intermediate migration hosts and efficient migration ordering algorithms [32, 38, 90] are proposed to solve the migration feasibility issue.

Scheduling Scopes: The multiple migration planning and scheduling algorithms can be divided into co-located and migrations with multiple source-destination pairs. In co-located instance migrations (Fig. 7), such as gang migration [25, 32], the solution only focuses on one source and destination pair. On the other hand, Figure 8 illustrates that multiple instances migration involves various source and destination hosts or sites [9, 43, 107]. For the migrations across dedicated network links, networking contentions between migrations and services are omitted. Without dedicated migration networks, some works consider the virtual network connectivity among applications during the migration schedule.

5.3 Scheduling Types

The migration scheduling types of multiple migrations can be categorized as *sequential* multiple migration, *parallel* multiple migration, and *concurrent* multiple migration.

Sequential: In the sequential multiple migration solution depicted in Fig. 9(a), migration requests are scheduled in the one-by-one manner [32, 38, 58, 70, 71, 88, 118]. In most scenarios of live VM migration, the network bandwidth is not sufficient for the networking requirement of live migration when sharing network links with other migrations. Therefore, sequential migration scheduling for networking resource-dependent migrations is the optimal solution. It is used for migration scheduling in co-located multiple migration scheduling and planning.

Parallel: In the parallel or synchronous multiple migration solution depicted in Fig. 9(b), migration requests start simultaneously [23, 58, 67, 70, 101, 118, 123]. For migrations with network link sharing, parallel migration scheduling is preferred only when the networking overheads induced by dirty pages and the memory footprint of migrating instances are smaller than the migration computing overheads. In other words, in most scenarios, the parallel migration may result in longer total and individual migration time and downtime. On the other hand, for migrations without network links sharing across dedicated migration networks, the minimum total and individual migration time can be achieved. Some solutions are mixing the sequential and parallel migration solutions in that groups of migrations are started at the same time as in parallel solution and these groups of migrations are scheduled sequentially.

Concurrent: Furthermore, concurrent or asynchronous migration planning and scheduling algorithms [9, 43, 107] are proposed to schedule multiple migration requests efficiently by calculating and scheduling the start time of each migration independently (Fig. 9(c)). The migrations contend the network resources with other migrations and services. Furthermore, the service traffic relocation induced by migration completion may affect the subsequent migrations. Therefore, it is essential to manage the dependency and concurrency among migrations during multiple migration scheduling.

5.4 Scheduling Methods

After the phase of multiple migration planning, the calculated migration plan is going to be scheduled in the migration scheduling phase. We categorize scheduling methods into three types, namely *prediction*, *fixed ordering*, and *online scheduler*. In order to schedule a migration, migration managers and schedulers are aware of when the migration ends or needs to begin.

Prediction: Based on the prediction model and current available computing and networking resources, the start time of each migration is configured during the planning phase [107]. Furthermore, the bandwidth allocated to each migration is also configured based on the available bandwidth at the time of migration planning.

Fixed ordering: Multiple migration tasks are scheduled based on the order calculated by migration planning algorithms [9]. In other words, one migration is started as soon as possible when one or several specific migrations are finished. The fixed ordering model of multiple migration requests is similar to the dependent tasks, which can be modeled as a Directed Acyclic Graph (DAG).

Online Scheduler: The states of the networking environment, such as network topology, available links and interfaces, available bandwidth, and network delay, are constantly changed. The computing resources, such as memory, vCPU, storage, destination hosts, and sites, may also become unavailable during the multiple migration scheduling. Integrating with dynamic computing and networking management, an online migration scheduler can dynamically start the migrations based on current states of computing and networking [43]. The resource may not be available based on the predicted scheduling time or orders. The online scheduler can dynamically adjust the migration plan and balance the allocated bandwidth to migration and application traffic. As a result, it can guarantee both QoS and migration performance.

6 CURRENT RESEARCH ON MIGRATION MANAGEMENT

This section reviews and analyzes current research on migration management, focusing on migration modeling during dynamic resource management and migration planning and scheduling algorithms. Each of these works may involve several categories of live migration management. Therefore, we select the main focus categories of the papers to organize and present the reviews in a more direct manner. In the end, we summarize and identify the gaps in these works.

6.1 Migration Modeling in Dynamic Resource Management

We summarize and review selected works on migration request generation during resource management of load balancing, energy-saving, network communication optimization, and migration-aware solutions. Table 5 summarizes the characteristics of migration request generation based on four categories: migration computing parameters, migration network parameters, objectives of migration optimization in the solution, and objectives of resource management.

There are notable works utilizing live migration by generating migration requests during dynamic resource management. However, most of the works neglect migration modeling in computing, networking, energy, or the overheads and performance of migration request scheduling. For example, Witanto et al. [110] propose a machine-learning selector to dynamically choose existing consolidation strategies to manage the trade-off between energy and SLA violation (migration downtime) based on system availability. Li et al. [65] propose a greedy-based VM scheduling algorithm to minimize the total energy consumption, including the cooling and server power

Table 5. Characteristics of Migration Modeling in Dynamic Resource Management

Reference	Mig. Com.			Mig. Net.				Mig. Obj.			Res. Obj.			
	mem	dirty rate	cpu	bw	route	hop	layer	perf.	cost	num	comp.	net.	energy	QoS
Mann et al. [74]	✓	✓		✓					✓			✓		
Forsman et al. [36]	✓	✓		✓					✓		✓			
Xiao et al. [113]	✓								✓		✓		✓	✓
Witanto et al. [110]									✓		✓			
Li et al. [65]													✓	
Tso et al. [103]	✓						✓		✓			✓		
Cao et al. [14]	✓					✓			✓			✓	✓	
Cui et al. [18, 19]	✓	✓		✓					✓			✓		
Xu et al. [115]	✓	✓	✓	✓				✓	✓	✓				✓
Cui et al. [20]	✓				✓	✓			✓			✓		
Flores et al. [35]	✓			✓	✓	✓		✓	✓			✓		✓
He et al. [44]	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓

Mig. Com.: Migration Computing parameters - mem (memory size), cpu (CPU load and utilization); **Mig. Net.**: Migration Networking parameters - bw (bandwidth), route (mig. traffic routing), hop (mig. distance), layer (involved network layers); **Mig. Obj.**: Migration Objectives - perf. (mig. performance), cost (mig. cost and overheads), num (mig. number); **Res. Obj.**: Resource management Objectives - comp. (computing), net. (networking), energy (device and cooling energy cost), and QoS (response time and SLA violations).

consumption models. Chen et al. [15] investigated the problem of service function chain migration to minimize the total network operation cost with a deep reinforcement learning framework to generate migration requests. It selects the VM with the minimum utilization from the hosts with a temperature above the threshold and selects the server with the minimum power increase as the migration destination. However, like many other works, the above papers do not consider any migration cost models or scheduling models, which makes the solution unrealistic. The multiple migration requests are also without the actual migration planning and scheduling.

For the works on networking provisioning, Tso et al. [103] propose a distributed network-aware live VM migration scheme to dynamically reallocate VMs to minimize the overall communication footprint of active traffic flows in multi-tier data centers. The proposed distributed solution generates migration requests iteratively based on the local VM information (one-by-one migration scheduling) to localize VM traffic to the low-tier network links. However, the work lacks a realistic comparison between migration overheads and migration benefits for communication management. With the help of SDN, it may be untenable to argue that a centralized approach to gaining knowledge of global traffic dynamics is too costly. Addya et al. [6] propose the Low Energy Application Workload Migration (LEAWM) model, which aims to minimize the per-bit migration cost in virtual machine (VM) migration over Geo-distributed clouds. Their model utilizes an Ant Colony Optimization (ACO) based bi-objective optimization technique to strike a balance between migration delay and migration power, considering the variation in electricity prices at different Internet Service Providers (ISPs) to determine the most feasible migration path. Their simulation-based evaluation demonstrates that the LEAWM model can achieve a reduction of 25%–30% in migration time and approximately 25% in electricity cost compared to the baseline approach.

Cao et al. [14] investigate the VM consolidation problem considering network optimization and migration cost. Migration overheads are modeled as the host power consumption (the product of power increase and migration time) and traffic cost (the product of VM memory size and the hop distance between source and destination). Cziva et al. [22] propose an SDN-based solution to minimize network communication costs through live migration in a multi-tire data center network.

The communication costs are modeled as the product of the average traffic load per time unit and the weight of the link layer. The solution can be improved by modeling the migration network traffic cost. Similarly, Cui et al. [18, 19] study the joint problem of dynamic policy reconfiguration and migration of VNF network chaining in an SDN environment to find the optimal placement with minimum total communication cost. The authors consider both migration time and traffic data during migration. However, it lacks information on how the communication cost is modeled. The migration cost is considered by comparing the network rate in data per time unit with the total transferred data size of live migration. Modeling the migration cost as transferred data size without considering the networking bandwidth and routing may result in poor migration scheduling performance and QoS degradation.

There are few works in resource management phases that actually consider migration scheduling [20, 35, 44, 115]. Based on VMs and destination candidates provided by existing resource management policies, Xu et al. [115] propose a migration selector (iAware) to minimize the single migration cost in terms of single migration execution time and host co-location interference. It considers the dirty page rate, memory size, and available bandwidth for the single migration time. They argue that co-location interference from a single live migration on other VMs in the host in terms of performance degradation is linear to the number of VMs hosted by a physical machine. However, only one-by-one migration scheduling is considered. Cui et al. [20] propose a new paradigm for migration generation by dynamically constructing adaptive network topologies based on the VM demands. It reduces VM migration costs and increases the communication throughput among VMs. The migration cost is modeled as the product of the number of network hops and memory size. The general VM migration costs are replaced by specific cost metrics, such as migration time and downtime based on allocated bandwidth and measured dirty page rate. Flores et al. [35] propose a placement solution that integrates migration selection with data centers policies to minimize the total cost of migrations and VM communications by considering network topology, VM virtual connections, communication cost, and network hops for live migration cost.

However, these cost models of migration are still linear without considering the concurrent scheduling performance of multiple migrations. These existing research efforts only consider alleviating single migration overheads and total migration costs with a linear model, such as single migration time and co-location interference during the selection of the potential VMs and migration destinations. However, by neglecting the resource dependency among potential migration requests, the existing solutions can result in QoS degradation and SLA violations during the migration schedule. Considering cost models of both single and concurrent multiple migrations, He et al. [44] propose an SDN-based concurrent-aware solution for the migration request generation phase that can be integrated with existing dynamic resource management policies. Based on dependency graphs with the help of the SDN controller, the generated migration requests minimize the potential computing and networking resource competition among migration and between migration requests and applications. During the concurrent scheduling phase, the multiple migration performance is optimized by minimizing the migration interference and the convergence time of reallocation while achieving the objective of dynamic resource management.

6.2 Migration Planning and Scheduling

In this section, we review the state-of-the-art works on migration planning and scheduling algorithms. Studies of migration planning and scheduling focus on various aspects, such as migration feasibility, migration success or failure ratio, migration effects, scheduling deadline, application QoS, scheduling orders, migration scheduler, migration routing, and migration performance in total migration time, average migration time, downtime, and transferred data size. As shown in Table 6, we summarize the characteristics of the reviewed solutions of migration planning and scheduling

Table 6. Comparisons of Solutions on Multiple Migration Planning and Scheduling

Reference	Schedule			Net			Scope		Heter	Mig Obj				
	Seq.	Parl.	Cnrc.	Mig. net.	Net. mgmt.	Connect.	Co-loc.	Multi src-dst		QoS aware	Mig. order	Mig. Feas.	Mig. perf.	Mig. cost
Deshpande et al. [26]		✓					✓							✓
Deshpande et al. [24, 25]		✓					✓							✓
Rybina et al. [88]	✓						✓				✓		✓	✓
Fernando et al. [32]	✓				✓		✓				✓		✓	✓
Ghorbani et al. [38]	✓				✓			✓				✓		
Sun et al. [101]	✓	✓					✓		✓				✓	
Deshpande et al. [23]		✓					✓		✓	✓			✓	✓
Zheng et al. [123]	✓	✓			✓	✓				✓			✓	✓
Liu et al. [67]		✓			✓	✓				✓			✓	✓
Lu et al. [70]	✓			✓	✓	✓				✓			✓	✓
Lu et al. [71]	✓	✓			✓	✓				✓			✓	✓
Kang et al. [58]	✓	✓			✓								✓	✓
Ye et al. [118]	✓	✓											✓	✓
Sarker et al. [90]			✓		✓	✓		✓				✓	✓	✓
Bari et al. [9]			✓			✓		✓		✓	✓		✓	✓
Wang et al. [107]			✓	✓	✓			✓			✓		✓	✓
He et al. [43]			✓	✓	✓	✓		✓		✓	✓		✓	✓

Schedule Type: Seq. (Sequential), Parl. (Parallel), Cnrc. (Concurrent), **Net**: Networking related - Mig. net. (Dedicated migration network), Net. mgmt. (Networking management), Connect. (instance Connectivity), **Scope**: Co-loc. (Co-located instances), **Heter**: Heterogeneous solutions (mixing various migration types), and **Mig Obj**: migration management objectives - Mig. order (migration ordering), Mig. feas. (migration feasibility), Mig. perf (migration performance), Mig. cost (Migration cost and overheads).

into the following categories: scheduling type, migration networking awareness, migration scopes, heterogeneous migration types, and migration scheduling objectives.

6.2.1 Co-located Multiple Migrations. Deshpande et al. [26] consider the migration of multiple co-located VMs in the same host as the live gang migration problem. They optimize the performance of multiple live migrations of co-located VMs based on the memory deduplication and delta compression algorithm to eliminate the duplicated memory copying from the source to the destination host. Since co-located VMs share a large amount of identical memory, only identical pages need to be transferred during the iterative memory copying in pre-copy migration. They also employ delta compression between copied dirty pages to reduce the migration network traffic. The authors [24, 25] further investigate the same problem using cluster-wide global deduplication to improve the technique from the co-located VMs in the same host to the same server rack.

Rybina et al. [88] investigate the co-located resource contentions on the same physical host. The authors evaluate all possible migration orders in a sequential manner in terms of total migration time. They find that it is better to migrate the memory-intensive VM in order to alleviate the resource contentions. Fernando et al. [32] proposed a solution for the ordering of multiple VMs in the same physical host (gang migration) to reduce the resource contentions between the live migration process and the migrating VMs. The objectives of the solution are minimizing the migration performance impact on applications and the total migration time. The migration scheduler decides the order of migrations based on different workload characteristics (CPU, Memory, network-intensive) and resource usage to minimize the total migration time and downtime. Furthermore, an SDN-enabled network bandwidth reservation strategy is proposed that reserves bandwidth on the source and destination hosts based on the migration urgency. When the available bandwidth in the destination can not satisfy the migration requirement, network middle-boxes [27] are used as network intermediaries to temporarily buffer the dirty memory.

6.2.2 Migration Feasibility. For the migration scheduling feasibility, Ghorbani et al. [38] proposed a heuristic one-by-one migration sequence planning for a group of migration requests to solve the problem of transient loop and network availability during the migration. The authors consider an environment in which the requirement of the virtual network can be satisfied through bandwidth over-subscription. Given the bandwidth requirement of virtual links between instances, a random migration sequence will result in migration failure for most instances. With the flow install time of the current SDN controller implementation, the orders of network updates due to migration within the forwarding table may cause the transient loop issue. The authors did not consider concurrent VM migration scheduling with various network paths.

6.2.3 Heterogeneous and Homogeneous Solutions. Multiple migration can be divided into heterogeneous and homogeneous solutions. For the heterogeneous solution, different types of live migration (pre-copy and post-copy migrations) are used simultaneously. For instance, in an environment where all migrations are sharing the same network, Sun et al. [101] consider the sequential and parallel migration performance of multiple VMs. An improved one-by-one migration scheduling was proposed based on the assumption that the migration downtime is large enough. When the first VM is stopped during the downtime of pre-copy migration, the algorithm stops and performs the post-copy on the remaining connected VMs within the same service. Furthermore, an m-mixed migration algorithm was proposed for parallel migrations started at the same time. The algorithm chooses the first m VMs to perform pre-copy migration, while the rest are post-copy migrations.

The networking contentions between migrations and application traffic can increase the migration time and degrade the application QoS. To reduce the network contentions between migrations and applications, solutions for co-located VM migrations with both pre-copy and post-copy migrations are adopted [23]. The intuition is utilizing both inbound and outbound network interfaces. When the co-located instances are migrated using pre-copy or post-copy, the traffic between co-located instances contends with the migration traffic. Thus, the migrating instance with post-copy in the destination communicates to another migration instance with pre-copy in the source, which alleviates the network contentions between the application and pre-copy migration traffic.

6.2.4 Instance Correlation and Connectivity. The instances in the data center are often connected through network virtual links under various application architectures, such as multi-tier web applications and Virtual Network Functions (VNFs) in Service Function Chaining (SFC). Several studies focus on optimizing the multiple migration of multi-tier applications and network-related VMs. Research [61, 106] evaluates the impact of live migration on multi-tier web applications, such as response time. Zheng et al. [123] investigate the multi-tier application migration problem and propose a communication-impact-driven coordinated approach for a sequential and parallel scheduling solution. Liu et al. [67] work on the correlated VM migration problem and propose an adaptive network bandwidth allocation algorithm to minimize migration costs in terms of migration time, downtime, and network traffic. In the context of multi-tier applications, the authors proposed a synchronization technique to coordinate correlated VM migrations entering the stop-and-copy phases at the same time, which reduces the network traffic between correlated applications across the inter-data center network.

Lu et al. [70] also focus on the correlated VMs migration scheduling of multi-tier web applications with a dedicated network for migrations. The authors investigate the sequential and parallel migration strategies for multiple migrations which start at the same time. The proposed heuristic algorithm groups the related VMs based on the monitoring results of communication traffic and sorts the sequential migration order based on migration time and resource utilization. Expanding the concept from multi-tier application connectivity, Lu et al. [71] proposed a separation strategy by partitioning a large group of VMs into traffic-related subgroups for inter-cloud live migration.

Partitioned by a mini-cut algorithm, subgroups of pre-copy migrations are scheduled sequentially and VMs in the same subgroup are scheduled in parallel to minimize the network traffic between applications across inter-data center networks. Filali et al. [34] discusses the design of a switch migration scheduling algorithm for load balancing in distributed architectures of the SDN control plane. The algorithm utilizes a multi-step ARIMA forecasting model to predict long-term controller load, enabling the scheduling of switch migration operations in advance to prevent overload. This work experimentally demonstrated the effectiveness of the their algorithm in improving load balancing and reducing response time of controllers.

6.2.5 Parallel and Concurrent Scheduling. Studies [58, 101] investigate solutions for mixed sequential and parallel migrations. Kang et al. [58] proposed a feedback-based algorithm to optimize the performance of multiple migration in both total and single migration time considering the sequential and parallel migration cost, and available network resources. It adaptively changes the migration number based on the TCP congestion control algorithm. Ye et al. [118] investigate the multiple migration performance in sequential and parallel migrations. The authors conclude that sequential migration is the optimal solution when the available network bandwidth is insufficient. Singh et al. [95] proposed a bio-inspired algorithm to optimise the task allocation to VM to limit the number of VM migration requests. As a result, the performance is improved in terms of load, migration cost, energy, and resource utilization.

For the multiple migration planning and scheduling algorithms of concurrent migrations, the migration planning algorithm and migration scheduler determine when and how one migration of the multiple migration requests should be performed within the time interval of the total migration time. Sarker et al. [90] proposed a naive heuristic algorithm of multiple migration to minimize migration time and downtime. The proposed scheduling algorithm starts all available migrations with minimum migration cost until there is no migration request left. The deadlock problem is solved by temporarily migrating VMs to an intermediate host. Bari et al. [9] proposed a grouping-based multiple migration planning algorithm in an intra-data center environment where migrations and applications share the network links. The authors model the multiple VM migration planning based on a discrete-time model as a MIP problem. The available bandwidth may be changed after each migration due to the reconfiguration of virtual network links. The subsequent migrations are affected by the previous migration result. Considering the influence of each migration group during the scheduling, the proposed algorithm sets the group start time based on the prediction model. Migration in each group can be scheduled simultaneously if the resources occupied by the previous group are released. However, the authors neglect the influence of individual migration in their solution, which can lead to performance degradation of the total migration time. Without considering the connectivity among applications in a WAN environment, Wang et al. [107] simplify the problem by maximizing the network transmission rate but directly minimizing the total migration time. With the help of SDN, the authors introduce multipath transmission for multiple migrations. A fully polynomial-time approximation FPTAS algorithm is proposed to determine the start time of each migration. Considering individual migration impact in terms of migration time, migration bandwidth allocation, networking routing and overheads, available resources after migration, and migration deadline and priority, He et al. [43] propose SLA-aware multiple migration planning algorithms to maximize concurrent migration groups with minimum cost and an SDN-enabled online migration scheduler to manage the migration lifecycle based on the planning groups. The experimental results show that the proposed solution can achieve the minimum total migration time while minimizing individual migration time, migration deadline violations, SLA violations, and energy consumption. Khan et al. [59] highlight the need for an effective VM migration strategy in cloud computing to maintain QoS and reduce energy consumption. Their work proposes a hybrid optimization algorithm combining cuckoo search and particle swarm optimization to achieve objectives such as minimizing energy consumption, computation time, and migration cost, while maximizing resource utilization.

6.3 Summary and Comparison

All studies covered in the survey are summarized in Table 5 and Table 6 based on the taxonomy (Fig. 3), respectively. For migration generation in resource management, many studies optimize at least two of the objectives in computing resources, networking resources, energy consumption, and application QoS. Most researchers do not consider the migration scheduling performance which has no tick in the table. These studies model migration computing costs linearly or individually based on memory size, available bandwidth, or the total migration number. A number of works consider the actual single pre-copy live migration model based on memory size, dirty page rate, and available bandwidth. For the migration network cost modeling and management, most works only consider the available bandwidth or the transferred data volume, while few works consider network routing or migration distance on hops and layers. Integrating with existing resource management algorithms, few works focus on migration interference and scheduling performance. However, the proposed linear models are only suitable for sequential migration scheduling optimization.

For migration planning and scheduling algorithms, researchers actively study the migration scheduling performance and migration cost. Some consider the application QoS during migrations, while others focus on migration availability and scheduling feasibility. For heterogeneous and homogeneous solutions, most works focus on homogeneous solutions with one live migration type such as pre-copy migration, while others consider both pre-copy and post-copy migrations. However, there are no concurrent planning and scheduling algorithms that consider the case of multiple migrations utilizing mixed types. The scheduling scope is varied based on the proposed method, early studies focus on co-located migrations with one source and destination pair, while recent proposals consider multiple migration scheduling with various source and destination pairs.

For networking management and efficiency, some researchers consider the migration scheduling in dedicated migration networks or do not consider the network overheads on other services and applications. Others consider the connectivity of correlation instances with virtual network communication during migrations. Without considering the network over-subscription, the bandwidth requirements of virtual links between instances are guaranteed during migrations. To improve migration performance and reduce migration traffic impacts, networking management algorithms are adopted to optimize the network routing for migration traffic and application traffic.

The scheduling types are varied based on the proposed solution and scheduling scopes, most of the works consider sequential migration scheduling, while others focus on parallel migrations or apply both types. Recently, several researchers focus on concurrent migration planning and scheduling for efficient, optimal, and generic solutions. Few works focus on the timeliness of migration based on optimizations and prediction models (i.e., migration finishes before a given deadline). Thus, the processing time of migration planning and scheduling algorithm also needs to be considered and optimized when the migration targets are real-time applications. Energy consumption is also a critical objective in the resource management of data centers, and migration energy cost models need to be investigated.

7 GAPS ANALYSIS AND FUTURE DIRECTIONS

The taxonomy and review cover several challenges of migration management in cloud and edge data centers. However, cloud and edge computing can be improved by addressing several key issues of live migration and corresponding management. In this section, we analyze gaps and future directions of live migration and migration management in edge and cloud computing.

7.1 Scheduling-Aware Resource Management

As shown in the proposed framework of migration management (Fig. 1), migration requests are generated based on resource management policies. The objectives of resource management can be achieved only when generated migration requests converged at the destination hosts. On the other hand, the objective of migration planning and scheduling algorithms is to maximize the performance of multiple migrations in total migration time, individual migration time, and downtime. There exist notable research efforts in dynamic resource management that alleviate single migration overheads, such as single migration time and co-location interference while selecting the potential VMs and migration destinations. However, by neglecting the resource dependency among potential migration requests, the existing solutions of dynamic resource management can result in the Quality of Service (QoS) degradation and Service Level Agreement (SLA) violations during the migration schedule. Therefore, it is essential to integrate both single and multiple migration overheads into VM reallocation planning [43]. To further optimize system performance, the scheduling performance of multiple migrations should be considered in the resource management phase.

7.2 Flexible Networking Management by Network Slicing

The containerized services allocated in the mobile edge clouds bring up the opportunity for large-scale and real-time applications to have low-latency responses. Meanwhile, live container migration is introduced to support dynamic resource management and users' mobility. The container footprint becomes much smaller compared to the VM. As a result, the proportion of network resource limitations on migration performance is reduced. Instance-level parallelizing for multiple migration can improve the scheduling performance due to the computing cost of migration. Therefore, it is critical to investigate networking slicing algorithms to concurrently schedule both VM and container migrations to improve multiple migration performance and alleviate the impacts on service traffic. Holistic solutions are needed to manage the networking routing and bandwidth allocation based on the various network requirements of live VM migrations, live container migrations, and applications.

7.3 Optimization Modeling, Compatibility, and Combination

There are continuous efforts striving to improve the performance and alleviate the overheads of live migration through system-level optimization, as investigated in survey [120]. The disadvantage of the original pre-copy migration is migration convergence. The optimization works of improving live migration performance mainly focus on reducing the live migration time and downtime, including reducing the memory data required for transmission to the destination, speeding up the migration process in the source host, and increasing the bandwidth between the source and destination hosts. However, there are gaps between current migration cost and performance modeling and the existing migration optimization mechanisms in process parallelizing, compression, de-duplication, memory introspection, checkpoint and recovery, remote direct memory access (RDMA), and application awareness. The existing system-level optimizations need to be modeled carefully and properly to reflect the nature and characteristics of these optimizations. Although extensive optimizations on live VM migration exist, each one claims a certain performance improvement compared to the default live VM migration mechanism. It is unclear the compatibility of each migration optimization technology and the performance improvement of the combination of various mechanisms.

7.4 Live Container Migration Optimization and Scheduling

From the perspective of single migration, there is a research interest and trend of adapting or replicating optimization mechanisms of live VM migration to develop and improve live container migration [64, 73, 80, 81, 97]. For example, Remote Direct Memory Access (RDMA) can be utilized as the high-speed interconnect technique, which has been applied to improve live VM migration. Recently, RDMA is utilized to improve the performance of live container migration [84]. It enables the remote access of memory and disk data without the CPU and cache. Multipath TCP (MPTCP) allows TCP connections to utilize multiple paths to increase the network throughput and redundancy of migrations [85, 107]. The container layered storage can be leveraged to alleviate synchronization overheads of a file system in the architecture without shared storage [72]. The performance and cost modeling of live container migration with these optimizations should be investigated. The corresponding scheduling algorithms of multiple container migration with such optimizations also need to be investigated in both edge and cloud computing environments.

7.5 Management Scalability and Efficiency

With the expansion of edge and cloud computing networks, the complexity of the migration planning and scheduling problem becomes unavoidably large. Meanwhile, the timeliness requirement of the management algorithm is changed from cloud to edge computing. For time-critical applications, the resource and migration management algorithms need to be scalable to suit large-scale computing and networking environments. Therefore, it is crucial to study distributed management frameworks and algorithms to ensure the scalability of migration management algorithms. The problems of edge data center placement and base station clustering need to be further investigated based on user mobility to reduce the migration request number. Each edge manager and SDN controller need to cover a certain area and data centers and cooperate with other controllers. A certain strategy needs to be developed to determine the size of the cluster area and the placement of each manager and controller based on the parameters such as network delay and processing capability.

7.6 Autoscaling and Live Migration

For instance-level scaling, scaling up and down is used for allocation and deallocation of virtualized instances in cloud computing to elastically provision the resource in the same host. In addition, system-level scaling up [39] supports fine-grained scaling on the system resources, such as CPU, memory, and I/O, to reduce the considerable overhead and extra cost. On the other hand, scaling out is used to support application allocation in other compute nodes to increase the processing capacity of the service. The load balancer will distribute the traffic to all running instances of the application. Current resource management strategies of cloud providers perform live migration for both stateful and stateless services to achieve the objectives, such as energy consumption and traffic consolidation. There is no configuration and information to differentiate the instance types (stateful and stateless) across various cloud types (IaaS, PaaS, SaaS, and FaaS). The cloud or service providers can further reduce management overheads by strategically performing autoscaling and live migration to different applications. Therefore, it is critical to integrate the autoscaling and live migration strategies to holistically manage the resources based on the instance types to minimize the management overhead and cost. The advantages and disadvantages of instance-level scaling, resource-level scaling, and live migration need to be investigated. In addition, the specific SLA of migration and scaling is needed for various instance types.

7.7 Robustness and Security

SDN-based network resilience management and strategies, such as traffic shaping, anomaly detection, and traffic classification, need to be investigated to tackle the network robustness and security issue for live migration. Migration stability and robustness also need to be investigated to increase the availability and accessibility of dynamic resource management. Compared to pre-copy, post-copy migration starts the instance at the destination host as soon as the initial memory is copied, which makes it vulnerable to state synchronization failure. Thus, if there is a post-copy failure, the running processes can not be resumed and the migrating instance is shut down. Fernando et al. [31] proposed a failure recovery mechanism for post-copy migration to alleviate the cost of post-copy migration failure. Failure cost models need to be developed based on different migration types and mechanisms and applied to services with various SLA levels accordingly.

8 SUMMARY AND CONCLUSIONS

Efficient management of live migrations is the key to facilitating autonomous and dynamic resource provisioning in edge and cloud computing. In this paper, we present a taxonomy of migration management, which includes performance and cost models, migration generation in resource management policies, planning and scheduling algorithms, management lifecycle and orchestration, and evaluation methods. Each aspect of the taxonomy is explained in detail and corresponding papers are presented. We further review and analyze representative works of dynamic resource management focusing on migration request generation in migration computing parameters, networking parameters, and migration objectives. We also categorize and review the state-of-the-art research of migration management in terms of migration planning and scheduling in migration scheduling types, migration network awareness, scheduling scope, heterogeneous and homogeneous solutions, and scheduling objectives. Various objectives of multiple migration scheduling are investigated. Finally, we analyze gaps and future directions of live migration and migration management in edge and cloud computing.

ACKNOWLEDGMENTS

This work is partially supported by an Australian Research Council (ARC) Discovery Project (ID: DP160102414). We thank editors and anonymous reviewers for their insightful comments and helpful suggestions on improving this manuscript.

REFERENCES

[1] accessed 04 Jan 2023. Planning for live migration in IBM Cloud Infrastructure Center. https://www.ibm.com/docs/en/cic/1.1.6?topic=hypervisors-planning-live-migration

[2] accessed 1 Feb 2023. AWS Live migration for maintenance. https://aws.amazon.com/ec2/features/

[3] accessed 1 Feb 2023. Improving Azure Virtual Machine Resiliency with Predictive ML and Live Migration. https://azure.microsoft.com/en-us/blog/improving-azure-virtual-machine-resiliency-with-predictive-ml-and-live-migration/

[4] accessed 22 Jan 2020. Container migration with Podman on RHEL. https://www.redhat.com/en/blog/container-migration-podman-rhel

[5] accessed 29 June 2021. Dynamic resource management in E2 VMs. https://cloud.google.com/blog/products/compute/understanding-dynamic-resource-management-in-e2-vms

[6] Sourav Kanti Addya, Anurag Satpathy, Bishakh Chandra Ghosh, Sandip Chakraborty, Soumya K Ghosh, and Sajal K Das. 2023. Geo-distributed Multi-tier Workload Migration over Multi-timescale Electricity Markets. *IEEE Transactions on Services Computing* (2023), 1–14. https://doi.org/10.1109/TSC.2023.3270921

[7] Sherif Akoush, Ripduman Sohan, Andrew Rice, Andrew W Moore, and Andy Hopper. 2010. Predicting the performance of virtual machine migration. In *Proceedings of 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE, 37–46.

[8] Mohammad Aldossary and Karim Djemame. 2018. Performance and Energy-based Cost Prediction of Virtual Machines Live Migration in Clouds. In *Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018)*. 384–391.

[9] Md Faizul Bari, Mohamed Faten Zhani, Qi Zhang, Reaz Ahmed, and Raouf Boutaba. 2014. CQNCR: Optimal VM migration planning in cloud data centers. In *Proceedings of 2014 IFIP Networking Conference*. IEEE, 1–9.

[10] Trinayan Baruah, Yifan Sun, Ali Tolga Dinçer, Saiful A Mojumder, José L Abellán, Yash Ukidave, Ajay Joshi, Norman Rubin, John Kim, and David Kaeli. 2020. Griffin: Hardware-software support for efficient page migration in multi-gpu systems. In *Proceedings of 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 596–609.

[11] Anton Beloglazov and Rajkumar Buyya. 2012. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE transactions on parallel and distributed*

systems 24, 7 (2012), 1366–1379.

[12] David Breitgand, Gilad Kutiel, and Danny Raz. 2010. Cost-aware live migration of services in the cloud. *SYSTOR* 10 (2010), 1815695–1815709.

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

[14] Bo Cao, Xiaofeng Gao, Guihai Chen, and Yaohui Jin. 2014. NICE: network-aware VM consolidation scheme for energy conservation in data centers. In *Proceedings of 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 166–173.

[15] Ruoyun Chen, Hancheng Lu, Yujiao Lu, and Jinxue Liu. 2020. MSDF: A Deep Reinforcement Learning Framework for Service Function Chain Migration. In *Proceedings of 2020 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–6.

[16] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. 2005. Live migration of virtual machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. 273–286.

[17] CRIU. 2019. Live migration. https://criu.org/Live_migration

[18] Lin Cui, Richard Cziva, Fung Po Tso, and Dimitrios P Pezaros. 2016. Synergistic policy and virtual machine consolidation in cloud data centers. In *Proceedings of IEEE INFOCOM 2016-IEEE International Conference on Computer Communications*. IEEE, 1–9.

[19] Lin Cui, Fung Po Tso, Dimitrios P. Pezaros, Weijia Jia, and Wei Zhao. 2017. PLAN: Joint Policy- and Network-Aware VM Management for Cloud Data Centers. *IEEE Transactions on Parallel and Distributed Systems* 28, 4 (2017), 1163–1175.

[20] Yong Cui, Zhenjie Yang, Shihan Xiao, Xin Wang, and Shenghui Yan. 2017. Traffic-aware virtual machine migration in topology-adaptive dcn. *IEEE/ACM Transactions on Networking* 25, 6 (2017), 3427–3440.

[21] Richard Cziva, Christos Anagnostopoulos, and Dimitrios P Pezaros. 2018. Dynamic, latency-optimal vNF placement at the network edge. In *Proceedings of IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 693–701.

[22] Richard Cziva, Simon Jouët, David Stapleton, Fung Po Tso, and Dimitrios P Pezaros. 2016. SDN-based virtual machine management for cloud data centers. *IEEE Transactions on Network and Service Management* 13, 2 (2016), 212–225.

[23] Umesh Deshpande and Kate Keahey. 2017. Traffic-sensitive live migration of virtual machines. *Future Generation Computer Systems* 72 (2017), 118–128.

[24] Umesh Deshpande, Unmesh Kulkarni, and Kartik Gopalan. 2012. Inter-rack live migration of multiple virtual machines. In *Proceedings of the 6th international workshop on Virtualization Technologies in Distributed Computing Date*. 19–26.

[25] Umesh Deshpande, Brandon Schlinker, Eitan Adler, and Kartik Gopalan. 2013. Gang migration of virtual machines using cluster-wide deduplication. In *Proceedings of 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. IEEE, 394–401.

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

[27] Umesh Deshpande, Yang You, Danny Chan, Nilton Bila, and Kartik Gopalan. 2014. Fast server deprovisioning through scatter-gather live migration of virtual machines. In *Proceedings of 2014 IEEE 7th International Conference on Cloud Computing*. IEEE, 376–383.

[28] Mohamed Esam Elsaid, Hazem M Abbas, and Christoph Meinel. 2019. Machine Learning Approach for Live Migration Cost Prediction in VMware Environments. In *Proceedings of the 9th International Conference on Cloud Computing and Services Science (CLOSER)*. 456–463.

[29] Mohamed Esam Elsaid and Christoph Meinel. 2014. Live migration impact on virtual datacenter performance: VMware vMotion based study. In *Proceedings of 2014 International Conference on Future Internet of Things and Cloud*. IEEE, 216–221.

[30] Vincenzo Eramo, Emanuele Miucci, Mostafa Ammar, and Francesco Giacinto Lavacca. 2017. An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Transactions on Networking* 25, 4 (2017), 2008–2025.

[31] Dinuni Fernando, Jonathan Terner, Kartik Gopalan, and Ping Yang. 2019. Live migration ate my vm: Recovering a virtual machine after failure of post-copy live migration. In *Proceedings of IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 343–351.

[32] Dinuni Fernando, Ping Yang, and Hui Lu. 2020. SDN-based Order-aware Live Migration of Virtual Machines. In *Proceedings of IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 1818–1827.

[33] Silvia Fichera, Molka Gharbaoui, Piero Castoldi, Barbara Martini, and Antonio Manzalini. 2017. On experimenting 5G: Testbed set-up for SDN orchestration across network cloud and IoT domains. In *Proceedings of 2017 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 1–6.

[34] Abderrahime Filali, Soumaya Cherkaoui, and Abdellatif Kobbane. 2019. Prediction-Based Switch Migration Scheduling for SDN Load Balancing. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–6.

[35] Hugo Flores, Vincent Tran, and Bin Tang. 2020. PAM & PAL: Policy-Aware Virtual Machine Migration and Placement in Dynamic Cloud Data Centers. In *Proceedings of IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2549–2558.

[36] Mattias Forsman, Andreas Glad, Lars Lundberg, and Dragos Ilie. 2015. Algorithms for automated live migration of virtual machines. *Journal of Systems and Software* 101 (2015), 110–126.

[37] Chang Ge, Zhili Sun, Ning Wang, Ke Xu, and Jinsong Wu. 2014. Energy management in cross-domain content delivery networks: A theoretical perspective. *IEEE Transactions on Network and Service Management* 11, 3 (2014), 264–277.

[38] Soudeh Ghorbani and Matthew Caesar. 2012. Walk the line: consistent network updates with bandwidth guarantees. In *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 67–72.

[39] Rui Han, Li Guo, Moustafa M Ghanem, and Yike Guo. 2012. Lightweight resource scaling for cloud applications. In *Proceedings of 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 644–651.

[40] Raseena M Haris, Khaled M Khan, Armstrong Nhlabatsi, and Mahmoud Barhamgi. 2023. A machine learning-based optimization approach for pre-copy live virtual machine migration. *Cluster Computing* (2023), 1–20.

[41] Eric Harney, Sebastien Goasguen, Jim Martin, Mike Murphy, and Mike Westall. 2007. The efficacy of live virtual machine migrations over the internet. In *Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing (VTDC'07)*. IEEE, 1–7.

[42] TianZhang He, Adel N. Toosi, and Rajkumar Buyya. 2019. Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers. *J. Parallel and Distrib. Comput.* 131 (2019), 55–68.

[43] TianZhang He, Adel N. Toosi, and Rajkumar Buyya. 2021. SLA-aware multiple migration planning and scheduling in SDN-NFV-enabled clouds. *Journal of Systems and Software* 176 (2021), 110943.

[44] TianZhang He, Adel N. Toosi, and Rajkumar Buyya. 2022. CAMIG: Concurrency-Aware Live Migration Management of Multiple Virtual Machines in SDN-Enabled Clouds. *IEEE Transactions on Parallel and Distributed Systems* 33, 10 (2022), 2318–2331.

[45] Brandon Heller, Srinivasan Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. 2010. Elastictree: Saving energy in data center networks. In *Proceedings of 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 10)*, Vol. 10. 249–264.

[46] Michael R Hines, Umesh Deshpande, and Kartik Gopalan. 2009. Post-copy live migration of virtual machines. *ACM SIGOPS Operating Systems Review* 43, 3 (2009), 14–26.

[47] Cheol-Ho Hong and Blesson Varghese. 2019. Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. *ACM Computing Surveys (CSUR)* 52, 5 (2019), 1–37.

[48] Liang Hu, Jia Zhao, Gaochao Xu, Yan Ding, and Jianfeng Chu. 2013. HMDC: Live virtual machine migration based on hybrid memory copy and delta compression. *Applied Mathematics & Information Sciences* 7, 2L (2013), 639–646.

[49] Wenjin Hu, Andrew Hicks, Long Zhang, Eli M Dow, Vinay Soni, Hao Jiang, Ronny Bull, and Jeanna N Matthews. 2013. A quantitative study of virtual machine live migration. In *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*. 1–10.

[50] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. 2015. Mobile edge computing A key technology towards 5G. *ETSI white paper* 11, 11 (2015), 1–16.

[51] Qiang Huang, Fengqian Gao, Rui Wang, and Zhengwei Qi. 2011. Power consumption of virtual machine live migration in clouds. In *Proceedings of 2011 Third International Conference on Communications and Mobile Computing*. IEEE, 122–125.

[52] Qingjia Huang, Kai Shuang, Peng Xu, Jian Li, Xu Liu, and Sen Su. 2014. Prediction-based dynamic resource scheduling for virtualized cloud systems. *Journal of Networks* 9, 2 (2014), 375.

[53] Huawei. 2020. White Paper: 5G Network Architecture - A High-Level Perspective. https://www.huawei.com/en/technology-insights/industry-insights/outlook/mobile-broadband/insights-reports/5g-network-architecture

[54] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. 2013. B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 3–14.

[55] Changyeon Jo, Youngsu Cho, and Bernhard Egger. 2017. A machine learning approach to live migration modeling. In *Proceedings of the 2017 Symposium on Cloud Computing*. 351–364.

[56] Ann Mary Joy. 2015. Performance comparison between linux containers and virtual machines. In *Proceedings of 2015 International Conference on Advances in Computer Engineering and Applications*. IEEE, 342–346.

[57] Dharmesh Kakadia, Nandish Kopri, and Vasudeva Varma. 2013. Network-aware virtual machine consolidation for large data centers. In *Proceedings of the Third International Workshop on Network-Aware Data Management*. 1–8.

[58] Tae Seung Kang, Maurício Tsugawa, Andréa Matsunaga, Takahiro Hirofuchi, and José AB Fortes. 2014. Design and implementation of middleware for cloud disaster recovery via virtual machine migration management. In *Proceedings*

of 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. IEEE, 166–175.

[59] Mohd Sha Alam Khan and R Santhosh. 2022. Hybrid optimization algorithm for vm migration in cloud computing. *Computers and Electrical Engineering* 102 (2022), 108152.

[60] Shinji Kikuchi and Yasuhide Matsumoto. 2011. Performance modeling of concurrent live migration operations in cloud computing systems using prism probabilistic model checker. In *Proceedings of 2011 IEEE 4th International Conference on Cloud Computing*. IEEE, 49–56.

[61] Shinji Kikuchi and Yasuhide Matsumoto. 2012. Impact of live migration on multi-tier application performance in clouds. In *Proceedings of 2012 IEEE Fifth International Conference on Cloud Computing*. IEEE, 261–268.

[62] Panagiotis Kokkinos, Dimitris Kalogeras, Anna Levin, and Emmanouel Varvarigos. 2016. Survey: Live migration and disaster recovery over long-distance networks. *ACM Computing Surveys (CSUR)* 49, 2 (2016), 1–36.

[63] Tuan Le. 2020. A survey of live Virtual Machine migration techniques. *Computer Science Review* 38 (2020), 100304.

[64] Wubin Li and Ali Kanso. 2015. Comparing containers versus virtual machines for achieving high availability. In *Proceedings of 2015 IEEE International Conference on Cloud Engineering*. IEEE, 353–358.

[65] Xiang Li, Peter Garraghan, Xiaohong Jiang, Zhaohui Wu, and Jie Xu. 2017. Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. *IEEE Transactions on Parallel and Distributed Systems* 29, 6 (2017), 1317–1331.

[66] Ziyu Li and Gang Wu. 2016. Optimizing VM live migration strategy based on migration time cost modeling. In *Proceedings of the 2016 Symposium on Architectures for Networking and Communications Systems*. 99–109.

[67] Haikun Liu and Bingsheng He. 2014. Vmbuddies: Coordinating live migration of multi-tier applications in cloud environments. *IEEE Transactions on Parallel and Distributed Systems* 26, 4 (2014), 1192–1205.

[68] Haikun Liu, Hai Jin, Xiaofei Liao, Liting Hu, and Chen Yu. 2009. Live migration of virtual machine based on full system trace and replay. In *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*. 101–110.

[69] Haikun Liu, Hai Jin, Cheng-Zhong Xu, and Xiaofei Liao. 2013. Performance and energy modeling for live migration of virtual machines. *Cluster computing* 16, 2 (2013), 249–264.

[70] Hui Lu, Cong Xu, Cheng Cheng, Ramana Kompella, and Dongyan Xu. 2015. vhaul: Towards optimal scheduling of live multi-vm migration for multi-tier applications. In *Proceedings of 2015 IEEE 8th International Conference on Cloud Computing*. IEEE, 453–460.

[71] Tao Lu, Morgan Stuart, Kun Tang, and Xubin He. 2014. Clique migration: Affinity grouping of virtual machines for inter-cloud live migration. In *Proceedings of 2014 9th IEEE International Conference on Networking, Architecture, and Storage*. IEEE, 216–225.

[72] Lele Ma, Shanhe Yi, Nancy Carter, and Qun Li. 2018. Efficient live migration of edge services leveraging container layered storage. *IEEE Transactions on Mobile Computing* 18, 9 (2018), 2020–2033.

[73] Andrew Machen, Shiqiang Wang, Kin K Leung, Bong Jun Ko, and Theodoros Salonidis. 2017. Live service migration in mobile edge clouds. *IEEE Wireless Communications* 25, 1 (2017), 140–147.

[74] Vijay Mann, Akanksha Gupta, Partha Dutta, Anilkumar Vishnoi, Parantapa Bhattacharya, Rishabh Poddar, and Aakash Iyer. 2012. Remedy: Network-aware steady state VM management for data centers. In *Proceedings of International Conference on Research in Networking*. Springer, 190–204.

[75] Victor Marmol and Andy Tucker. 2018. Task Migration at Scale Using CRIU. In *Proceedings of Linux Plumbers Conference*.

[76] Barbara Martini, Davide Adami, Molka Gharbaoui, Piero Castoldi, Lisa Donatini, and Stefano Giordano. 2016. Design and evaluation of SDN-based orchestration system for cloud data centers. In *Proceedings of 2016 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[77] Arturo Mayoral, Ricard Vilalta, Raul Muñoz, Ramon Casellas, and Ricardo Martínez. 2017. SDN orchestration architectures and their integration with cloud computing applications. *Optical Switching and Networking* 26 (2017), 2–13.

[78] Violeta Medina and Juan Manuel García. 2014. A survey of migration mechanisms of virtual machines. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 1–33.

[79] Dejan S Milojičić, Fred Douglis, Yves Paindaveine, Richard Wheeler, and Songnian Zhou. 2000. Process migration. *ACM Computing Surveys (CSUR)* 32, 3 (2000), 241–299.

[80] Andrey Mirkin, Alexey Kuznetsov, and Kir Kolyshkin. 2008. Containers checkpointing and live migration. In *Proceedings of the Linux Symposium*, Vol. 2. 85–90.

[81] Shripad Nadgowda, Sahil Suneja, Nilton Bila, and Canturk Isci. 2017. Voyager: Complete container state migration. In *Proceedings of 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2137–2142.

[82] Kenneth Nagin, David Hadas, Zvi Dubitzky, Alex Glikson, Irit Loy, Benny Rochwerger, and Liran Schour. 2011. Inter-cloud mobility of virtual machines. In *Proceedings of the 4th Annual International Conference on Systems and Storage*. 1–12.

[83] Mostafa Noshy, Abdelhameed Ibrahim, and Hesham Arafat Ali. 2018. Optimization of live virtual machine migration in cloud computing: A survey and future directions. *Journal of Network and Computer Applications* 110 (2018), 1–10.

[84] Maksym Planeta, Jan Bierbaum, Leo Sahaya Daphne Antony, Torsten Hoefler, and Hermann Härtig. 2021. MigrOS: Transparent Live-Migration Support for Containerised RDMA Applications. In *Proceedings of 2021 USENIX Annual Technical Conference (USENIX ATC 21)*. USENIX Association, 47–63.

[85] Yuqing Qiu, Chung-Horng Lung, Samuel Ajila, and Pradeep Srivastava. 2019. Experimental evaluation of LXC container migration for cloudlets using multipath TCP. *Computer Networks* 164 (2019), 106900.

[86] Zeineb Rejiba, Xavier Masip-Bruin, and Eva Marín-Tordera. 2019. A survey on mobility-induced service migration in the fog, edge, and related computing paradigms. *ACM Computing Surveys (CSUR)* 52, 5 (2019), 1–33.

[87] Adam Ruprecht, Danny Jones, Dmitry Shiraev, Greg Harmon, Maya Spivak, Michael Krebs, Miche Baker-Harvey, and Tyler Sanderson. 2018. Vm live migration at scale. *ACM SIGPLAN Notices* 53, 3 (2018), 45–56.

[88] Kateryna Rybina, Abhinandan Patni, and Alexander Schill. 2014. Analysing the Migration Time of Live Migration of Multiple Virtual Machines. *CLOSER 2014: Proceedings of the 4th International Conference on Cloud Computing and Services Science* 14 (2014), 590–597.

[89] Shashank Sahni and Vasudeva Varma. 2012. A hybrid approach to live migration of virtual machines. In *Proceedings of 2012 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. IEEE, 1–5.

[90] Tusher Kumer Sarker and Maolin Tang. 2013. Performance-driven live migration of multiple virtual machines in datacenters. In *Proceedings of 2013 IEEE International Conference on Granular Computing (GrC)*. IEEE, 253–258.

[91] Jyoti Shetty, MR Anala, and G Shobha. 2012. A survey on techniques of secure live migration of virtual machine. *International Journal of Computer Applications* 39, 12 (2012), 34–39.

[92] Bin Shi and Haiying Shen. 2019. Memory/disk operation aware lightweight vm live migration across data-centers with low performance impact. In *Proceedings of IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 334–342.

[93] Bin Shi, Haiying Shen, Bo Dong, and Qinghua Zheng. 2022. Memory/Disk Operation Aware Lightweight VM Live Migration. *IEEE/ACM Transactions on Networking* 30, 4 (2022), 1895–1910.

[94] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE internet of things journal* 3, 5 (2016), 637–646.

[95] Shalu Singh and Dinesh Singh. 2023. A bio-inspired vm migration using re-initialization and decomposition based-whale optimization. *ICT Express* 9, 1 (2023), 92–99.

[96] Jungmin Son and Rajkumar Buyya. 2018. SDCon: Integrated control platform for software-defined clouds. *IEEE Transactions on Parallel and Distributed Systems* 30, 1 (2018), 230–244.

[97] Radostin Stoyanov and Martin J Kollingbaum. 2018. Efficient live migration of linux containers. In *Proceedings of International Conference on High Performance Computing*. Springer, 184–193.

[98] Anja Strunk. 2012. Costs of virtual machine live migration: A survey. In *Proceedings of 2012 IEEE Eighth World Congress on Services*. IEEE, 323–329.

[99] Anja Strunk. 2013. A lightweight model for estimating energy cost of live migration of virtual machines. In *Proceedings of 2013 IEEE Sixth International Conference on Cloud Computing*. IEEE, 510–517.

[100] Anja Strunk and Waltenegus Dargie. 2013. Does live migration of virtual machines cost energy?. In *Proceedings of 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 514–521.

[101] Gang Sun, Dan Liao, Vishal Anand, Dongcheng Zhao, and Hongfang Yu. 2016. A new technique for efficient live migration of multiple virtual machines. *Future Generation Computer Systems* 55 (2016), 74–86.

[102] Konstantinos Tsakalozos, Vasilis Verroios, Mema Roussopoulos, and Alex Delis. 2017. Live VM migration under time-constraints in share-nothing IaaS-clouds. *IEEE Transactions on Parallel and Distributed Systems* 28, 8 (2017), 2285–2298.

[103] Fung Po Tso, Gregg Hamilton, Konstantinos Oikonomou, and Dimitrios P Pezaros. 2013. Implementing scalable, network-aware virtual machine migration for cloud data centers. In *Proceedings of 2013 IEEE Sixth International Conference on Cloud Computing*. IEEE, 557–564.

[104] Mauricio Tsugawa, Renato Figueiredo, Jose Fortes, Takahiro Hirofuchi, Hidemoto Nakada, and Ryousei Takano. 2012. On the use of virtualization technologies to support uninterrupted IT services: A case study with lessons learned from the Great East Japan Earthquake. In *Proceedings of 2012 IEEE International Conference on Communications (ICC)*. IEEE, 6324–6328.

[105] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *Proceedings of the Tenth European Conference on Computer Systems*. 1–17.

[106] William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya. 2009. Cost of virtual machine live migration in clouds: A performance evaluation. In *Proceedings of IEEE International Conference on Cloud Computing*. Springer, 254–265.

[107] H. Wang, Y. Li, Y. Zhang, and D. Jin. 2019. Virtual Machine Migration Planning in Software-Defined Networks. *IEEE Transactions on Cloud Computing* 7, 4 (2019), 1168–1182.

[108] Shangguang Wang, Jinliang Xu, Ning Zhang, and Yujiong Liu. 2018. A survey on service migration in mobile edge computing. *IEEE Access* 6 (2018), 23511–23528.

[109] Bing Wei, Chuang Lin, and Xiangzhen Kong. 2011. Energy optimized modeling for live migration in virtual data center. In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, Vol. 4. IEEE, 2311–2315.

[110] Joseph Nathanael Witanto, Hyotaek Lim, and Mohammed Atiquzzaman. 2018. Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management. *Future generation computer systems* 87 (2018), 35–42.

[111] Timothy Wood, KK Ramakrishnan, Prashant Shenoy, Jacobus Van der Merwe, Jinho Hwang, Guyue Liu, and Lucas Chaufournier. 2014. CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines. *IEEE/ACM Transactions On Networking* 23, 5 (2014), 1568–1583.

[112] Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. 2014. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials* 17, 1 (2014), 27–51.

[113] Zhen Xiao, Weijia Song, and Qi Chen. 2012. Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE transactions on parallel and distributed systems* 24, 6 (2012), 1107–1117.

[114] Fei Xu, Fangming Liu, Hai Jin, and Athanasios V Vasilakos. 2013. Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions. *Proc. IEEE* 102, 1 (2013), 11–31.

[115] Fei Xu, Fangming Liu, Linghui Liu, Hai Jin, Bo Li, and Baochun Li. 2013. iAware: Making live migration of virtual machines interference-aware in the cloud. *IEEE Trans. Comput.* 63, 12 (2013), 3012–3025.

[116] Yong Xu, Kaixin Sui, Randolph Yao, Hongyu Zhang, Qingwei Lin, Yingnong Dang, Peng Li, Keceng Jiang, Wenchi Zhang, Jian-Guang Lou, Murali Chintalapati, and Dongmei Zhang. 2018. Improving Service Availability of Cloud Systems by Predicting Disk Error. In *Proceedings of 2018 USENIX Annual Technical Conference (USENIX ATC 18)*. USENIX Association, Boston, MA, 481–494.

[117] Hiroshi Yamada. 2016. Survey on mechanisms for live virtual machine migration and its improvements. *Information and Media Technologies* 11 (2016), 101–115.

[118] Kejiang Ye, Xiaohong Jiang, Dawei Huang, Jianhai Chen, and Bei Wang. 2011. Live migration of multiple virtual machines with resource reservation in cloud computing environments. In *Proceedings of 2011 IEEE 4th International Conference on Cloud Computing*. IEEE, 267–274.

[119] Ei Phyu Zaw. 2019. Machine Learning Based Live VM Migration for Efficient Cloud Data Center. In *Big Data Analysis and Deep Learning Applications*, Thi Thi Zin and Jerry Chun-Wei Lin (Eds.). Springer Singapore, Singapore, 130–138.

[120] Fei Zhang, Guangming Liu, Xiaoming Fu, and Ramin Yahyapour. 2018. A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials* 20, 2 (2018), 1206–1243.

[121] Jiao Zhang, Fengyuan Ren, and Chuang Lin. 2014. Delay guaranteed live migration of virtual machines. In *Proceedings of IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 574–582.

[122] Weizhe Zhang, Shuo Han, Hui He, and Huixiang Chen. 2017. Network-aware virtual machine migration in an overcommitted cloud. *Future Generation Computer Systems* 76 (2017), 428–442.

[123] Jie Zheng, Tze Sing Eugene Ng, Kunwadee Sripanidkulchai, and Zhaolei Liu. 2014. Comma: Coordinating the migration of multi-tier applications. In *Proceedings of the 10th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*. 153–164.