

A Survey of Data Center Consolidation in Cloud Computing Systems

Ruslan Malikov^{1*}, Nigora Abdiyeva¹, Jurabek Abdiyev²

¹Samarkand State Institute of Foreign Languages. Samarkand, 140104, street Bustonsaroy 93

²Physical-technical Institute of NPO "Physics – Sun" of Uzbekistan Academy of Sciences Uzbekistan, Tashkent, Chingiz Aitmatov street 2B.

Author: ruslan.malikov.1987@gmail.com (R. Malikov)

Abstract: *Virtualization technology is the backbone of cloud systems, the fastest-growing energy consumers, globally. It holds an imperative position in the resource management area by providing solutions for many related critical problems and challenges. Resource consolidation remains one of the techniques most used to develop new and more efficient resource management policies. Virtual machine packing, which is mainly addressed, allows cloud data centers to move from a particular state to a more optimized one. Currently, the unprecedented advancements in container-based cloud and containerized workloads have created new consolidation opportunities. In this article, we study the problem of consolidating data centers within distributed cloud systems. We give a generic view on IT consolidation at different levels of cloud services. Firstly, we present an overview of virtualized data centers and consolidation. Next, we present a brief thematic taxonomy and an illustration of some consolidation solutions from the literature. This presentation will then be followed by a discussion of certain research questions as well as a proposal for a certain number of future orientations in this field which we assume essential in order to contribute to the resolution of the problematic dealt with in this article.*

Keywords: Cloud computing, Virtualization, Container, Consolidation, Resource management, Migration, CRIU.

1. INTRODUCTION.

Outsourcing and dematerialization efforts as well as the popularity of virtualization technologies have given rise to the cloud computing paradigm. This business model is generally used to describe the large-scale distributed infrastructure or IaaS (Infrastructure as a Service), platform as a service or PaaS and software services or SaaS (Software as a Service). The main purpose is to conduct computing in distributed installations operated by third parties IT services. Recent years, the cloud computing industry is growing exponentially and deploying gigantic data centers that can hold more than a million servers to meet the growing demands of customers and satisfying IT requirements. Amongst the main causes behind the growth of cloud and its wider adoption, we cite its capacity to scale down or up computing resources according to the current demand, elastically. Thanks to virtualization technology customers can access services in a pay-as-you-use model based on their usage. Virtualization enables the same physical machine to host and run multiple virtual servers for better resource utilization and aggregate power consumption reduction [1–4]. Indeed, cloud infrastructures face a number of challenges that should be overcome. The most critical are related to resource management, power consumption, cost and performance. The major inefficiencies in data center (DC) deployments are caused by the poor management of resources [5]. Most of the time, 30% of cloud servers exploit 10%–15% of resource capacity on average [6]. Energy consumption represents another big challenge and still a hot topic despite optimization efforts. In 2010, data center consumption was worth approximately 1.1%–1.5% of the world's energy [7] which has evolved to 2% actually [8]. In 2014, DCs in the U.S. consumed about 1.8% of total U.S. electricity, which was expected to increase by about 4% between 2014 and 2020 [9]. Moreover, energy consumption in cloud systems raises many issues in terms of CO₂ emissions with more than 43 million tons of CO₂ per year [7]. Besides the environmental impact, power consumption in virtualized data centers has drastic effects on operational expenditure (OPEX) caused by electricity costs. From a cloud provider perspective, OPEX represents a key issue to maximize their profits. To deal with the aforementioned challenges, data center consolidation (DCC) is a powerful strategy adopted. It is an organizational strategy using more efficient technologies to reduce IT assets. The main objective is to maximize resource exploitation by keeping the minimum number of active physical machines (PM) and multiplexing several objects (virtual machines, containers, etc.) in the same PM. Data centers are facing a challenge to decide when, how, and which objects have to be consolidated into a single PM [10]. The consolidation may involve a migration process that allows the achievement of objectives and requirements in terms of resource and power management and the associated costs, system maintenance, fault tolerance, etc. However, migration is resource intensive and may have drastic effects on the whole performance.

As mentioned earlier, the consolidation process uses virtualization technology that creates isolated environments in

the same machine. This guarantees portability and acceptable security level and allows to run workloads in isolation. If an application fails, it does not compromise the execution of another application. Previous consolidation works have focused on infrastructure level. Virtual machine (VM) consolidation problem was deeply studied and we can consider that is resolved to a satisfactory level. However, this approach was not enough to solve underutilized or overutilized VM problems along with underutilized or overutilized PMs. Moreover, memory overhead of VMs limits the consolidation possibilities. For example, in a real cloud located in France, Tchana et al. observed, over a four month period, that 805 VMs used less than 12% of the active PMs CPU [11]. Although the focus was initially on VM consolidation, literature work did not exclude any service level. With the proliferation of containers which represent a lighter and more agile alternative than VMs, and the popularity and wider use of container-based virtualization technology, researchers interest is now turning to this new model. The same for software and data level consolidation that have started to emerge to offer more flexibility to reconfigure and optimize data center configurations with finer granularity. However, these new research areas have not been deeply and extensively studied [12]. Even the existing research articles that we have explored [13–23] studied only the consolidation problem at the IaaS level. The majority of this work focuses on migration [15–17] and no work summarizes the consolidation problem in cloud data centers in general by taking into account the different service level. In addition, a lot of work is limited to presenting and comparing existing approaches from a technical point of view, focusing, mainly, on the algorithmic approach adopted such as [14,18].

The proposed study will position the consolidation problem in large scale cloud environments, in line with the evolution of virtualization technologies and the elastic and efficient resource management. It aims to provide an overview of IT consolidation works in cloud systems. The research objectives relating to this article thereby are To present a summary on consolidation issue according to the different levels of cloud service,

- To develop a fairly generic taxonomy that reflects the different aspects relating to the consolidation problem,
- To conduct a literature review on consolidation research in cloud data centers to better understand the current works in this area. All of the presented examples from literature work find their classification in the proposed taxonomy regardless of the service level,
- Finally, to determine the limits of current approaches and identify some appropriate research gaps that have not been addressed in scientific literature.

The rest of the paper is organized as follows. After introducing the work and presenting the review methodology adopted in Section 2, we present cloud computing briefly by focusing on the service models and the two main virtualization technologies in Section 3. They represent the core of our classification adopted in the rest of the paper. Then, in Section 4, we present an overview of consolidation in cloud data centers (CDC). Section 5 is devoted to present our taxonomy of CDC consolidation and to synthesize the literature work according to the proposed taxonomy. In Section 6, we present a taxonomy and main aspects of the evaluation methods used in IT consolidation. Section 7 presents an illustration of the results of some relevant work that describes how consolidation performs. In Section 8, we give some findings and discuss some research issues and future orientations. Section 9 describes the major drawbacks of using the consolidation of workloads in cloud data centers. While Section 10 concludes this work.

2. REVIEW METHODOLOGY.

2.1. RESEARCH QUESTIONS.

The research carried out within this article aims to address some research questions. This part is devoted to the identification of survey questions which are in the number of seven with a fifth question divided into four sub-questions. Our goal is to answer these questions by conducting a review of existing research.

- RQ1 : What are the main reasons and motivations behind DC consolidation in the cloud and what is the process?
- RQ2 : How can the performance of cloud data centers be affected through virtualization architectures and consolidation process and how can consolidation service ensure application performance?
- RQ3 : How should CDC consolidation be characterized?
- RQ4 : At which level is CDC consolidation performed?
- RQ5 : What methods have been adopted in CDC consolidation?
- RQ5-1 : What kind of approaches have been used?
- RQ5-2 : How should the optimization process be described?
- RQ5-3 : What resources and metrics have been considered in the optimization process?
- RQ5-4 : What algorithms are being exploited?
- RQ6 : What are the evaluation methods used in existing CDC approaches?
- RQ7 : What should be the future research challenges and orientations behind DC consolidation in the cloud?

2.2. SEARCH PROCEDURE AND SELECTION PROCESS.

To elaborate the current work on IT consolidation in cloud systems a systematic methodology has been adopted. We will describe this review methodology in the following. The review process contains a series of steps and is documented as a review protocol. We synthesize the main steps followed to elaborate this work in a simplified way without going into details. The idea of our research article is not to elaborate an exhaustive list of consolidation approaches, but to introduce the IT consolidation in whole in the different service level and present some relevant works in each level. Inspired by Kitchen ham [24], the main steps we have taken relate to, first, the initial research based on search terms which are in line with prominent and recent works on consolidation, and strings formed by combining terms and Boolean operators (AND and OR operators). We focused on works from the relevant publishers like IEEE, ACM, Elsevier, Springer and Wiley online library. In case of duplication due to the use of multiple digital libraries, we have analyzed the search results and identified, then removed the duplicates based on our subjective criteria. After the initial search, the results obtained were filtered. For example, if the same approach is presented in more than a paper, one of these papers is selected according to a few criteria such as the publication date (often we keep the most recent), the expertise of the authors who participate, etc. The research results provided by this step were then selected based on abstract, title and full text screening to obtain a list of primary studies. For the final stage, which concerns the quality assessment procedure, we randomly selected some papers from selection results of the previous phase. Regarding the time distribution of consolidation works, we recall that VM consolidation is the oldest and dates back several years with the largest amount of work identified in the literature. Since containerization is an emerging field and container-based virtualization became a popular and wider used virtualization technology, container consolidation has started to emerge in recent years. This is also the case with software consolidation where several software applications can be collocated in the same container or VM, and data consolidation with the proliferation of Hybrid Storage Systems (HSS).

3. BACKGROUND OF CLOUD DATA CENTER.

In this section, we will present some preliminary concepts relating to cloud computing and virtualized data centers. The cloud highlights an outsourced IT infrastructure available via the internet where processing and storage services are performed at data center level. The vision is that IT will not be performed on local computers, but in distributed installations operated by third party IT services. The cloud is growing exponentially and deploying gigantic data centers that can host more than a million servers in order to meet the growing demands of customers. Cloud architecture relies on monolithic or geographically distributed data centers. A data center contains a set of servers grouped in racks. Through virtualization, a server is divided into virtualization elements (VMs and/or containers).

3.1. CLOUD SERVICE MODELS.

Several service models have emerged with cloud computing, the paradigm which proposes the on demand use of resources provided and maintained at the hardware and software level. The three models recognized by NIST (National Institute of Standards and Technology) are SaaS, PaaS and IaaS. Moreover, there are many other known services such as Container as a service (CaaS) which is a recent trend that lies between PaaS and IaaS [12,25]. Also, Database as a service (DBaaS) is a cloud-based database beating conventional databases such as Mongo DB, IBM and Oracle.

3.2. ISOLATION AND VIRTUALIZATION.

Virtualization is one of the most important technologies that have given birth to cloud computing. It allows resource sharing in cloud systems, giving the possibility of multiple applications to run on different platforms. Through this promising technology, hardware resources can be divided by simulating partial or complete machines with multiple execution environments that can act as isolated systems. Contrariwise, in traditional architecture, only one natively installed OS is supported by each physical machine for stability and uniformity [26]. A native environment is faster than a virtualized one [27]. In the latter, CPU utilization can increase by 40%–60% [28]. However, resources allocated in a virtualized environment are closer to the real needs of actual tasks compared to traditional architecture where an entire host is dedicated to run the same tasks [26]. There are many types and levels of virtualization in cloud computing like network virtualization, storage virtualization, etc. Here we are interested in hardware and operating system (OS) level virtualization. In this area, as shown in Fig. 1, there are two main types of virtualization, namely, VM-based virtualization and container-based

virtualization.

3.2.1. VM-BASED VIRTUALIZATION.

Virtual machines are the backbone at the infrastructure level in the cloud [26]. A VM is an illusion of a computer device created by emulation software and executes an application. VM-based virtualization is the most used technique in the cloud environment in which physical resources are virtually distributed at the hardware level through a hypervisor [29]. A Hypervisor or Virtual Machine Monitor (VMM) allows multiple operating systems to share one single PM through virtual machines. It is the management layer which manages and controls all instantiated VMs that run an independent OS [30]. There are two main types of hypervisors. A type 1 hypervisor also called “bare metal” or native which is a software running directly on a hardware platform. The second type of hypervisor also called hosted, is a software that runs inside another operating system. At third level, above the hardware, a guest operating system will run. The bare metal hypervisor runs directly on the hardware of the host machine and does not require a host OS to run which requires additional re- sources to operate. By avoiding the extra software layer between the host hardware and VMs [26], bare metal hypervisor performs better than type 2 [29]. Type 1 hypervisor is more secure [31] and resource efficient, but it needs a special configuration in the host machine to use it [26]. The majority of worldwide used hypervisors like VMware ESX, KVM, Xen, and Hyper-V are bare metal [32].

3.2.2. CONTAINER-BASED VIRTUALIZATION.

The second level of virtualization, which we are interested in here is lightweight operating system level virtualization based on the concept of containers. Containers can be considered as a new revolution in the cloud era [33]. They hold packaged, self-contained, ready-to-deploy parts of applications [34] and all of their dependencies [32]. The containerized environment is composed of a container engine and any number of isolated user-space [27] or container instances [12]. All are hosted on the same machine and share the OS kernel, including root file system, libraries and common files [35]. This avoids code duplication, among others. Nowadays, many providers are adopting containers and expose container services such as Google Container Engine, Docker Datacenter and Amazon Elastic Container Service (ECS). The new cloud service model named CaaS appeared with the containers. It frees applications, making them completely independent of the PaaS and eliminates dependencies [30]. As examples of CaaS providers, we cite Google and Amazon [12]. Note that the entire application can be run in a container or each component of the distributed application is executed inside a container and the containers are deployed on PMs. Docker is the most used container implementations, which is an open source management tool for containers that auto- mates the deployment of applications [36]. Another emerging new container technology competitor of Docker named Rocket was designed to be a more secure, interoperable, and an open container solution [32].

3.2.3. ISOLATION AND PERFORMANCE DISCUSSION.

Table 1 presents a general comparison between VM-based and container-based virtualization according to some relevant criteria. Both virtualization techniques provide isolation which offers an isolated space in the machine. Hypervisor adds an extra layer of virtualization, but containers provide an isolated virtual environment at OS level. These isolate and control resources, but also processes on the core-level of the OS what makes them much more lightweight and resource efficient. This process isolation is done using Namespaces [26]. In Linux OS, Linux control groups (c groups) isolate resource usage, such as memory, CPU, block I/O and provide resource management [32]. However, it is hard to provide the same level of isolation between containers as VMs do.

Several works [26–28,35,38] have compared the two virtualization methods in terms of performance by adopting several metrics. For example, in [38], the authors prove that Docker container outperforms VMs in almost all metrics considered such as throughput, average response time, etc. In [27], the authors compared the docker container and KVM virtual machine and conducted a lot of experiments in terms of memory, CPU and I/O performance. They were based on several parameters such as the number of VMs and containers running on machines, copying large files and configuring a web server. It was revealed that Docker is more resource (CPU, Hard Disk Drive (HDD) and RAM) efficient and faster than KVM even if it is the same hardware [27]. For CPU usage, the difference is not too important. However, in terms of memory, KVM wastes additional resources with a difference of 3.6–4.6 times for the OS even though no operation performed [27].

A deep comparison of traditional architecture, virtualized one, containerized and containerization through virtualization in the case of game-based interactive simulation applications in [26] reveals some observations that

we have tried to synthesize in Fig. 2. As we mount in the virtualization stack (Fig. 2(b) the right narrow), the CPU and memory usage increase. This is caused by the required overhead to support each technique and to host additional OSs [26]. The difference between containerized and traditional architectures is due to supplementary software used for simulation software containerization. This containerization layer added a minimal overhead to realize isolation and abstraction rules (c groups and namespaces) and to run docker engine and its daemon [26]. On the contrary, The addition of hypervisors to deployment methods is the most intensive in memory and CPU. Hosting a VM comes with 11%–17% [26] increase in CPU usage compared to traditional and containerized methods to deploy interactive simulation software.

Mavridis et al. affirmed that, containers, when running natively on a PM (bare metal), perform better than running on top of the VM, except security considerations. Bare metal containers deliver more performance per server [35] and are more resource, storage and energy efficient and cost-effective [29,35]. Due to the missing virtual hardware layer, containers can be created [35] and migrated faster [26], and start quickly than VMs [29] which takes a long time to be created [27]. Using the host OS, containers do not demand to reboot any operating system to restart [27]. This virtualization method enables a dense deployment with a greater number of containers per machine than VMs, since the containers are more light-weight [35]. Containers running natively are more efficient in terms of system resources [39], and require less hardware by the fact that they reduce the number of layers to manage [35]. They also demand less space to keep multiple copies of a container and then provide optimized storage use with minimum footprint [29]. According to Mavridis et al. Bare metal containers consumed less time to perform the same tasks than running containers on top of VMs. On the contrary, increased

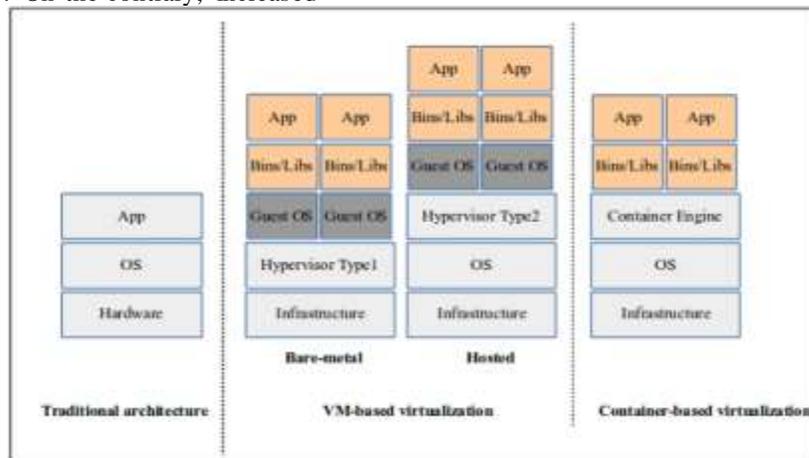


Fig. 1. Virtualization architectures: traditional server architecture [with applications and one OS installed on physical hardware] vs virtualized infrastructure with virtual machines [the bare metal hypervisor runs directly on the hardware; this avoids the use of an extra software layer, the hosted hypervisor runs inside another OS] vs containerized architecture [the container-based architecture can be designed natively, i.e. the containers run in the host OS, or through virtualization when the containers run in VMs].

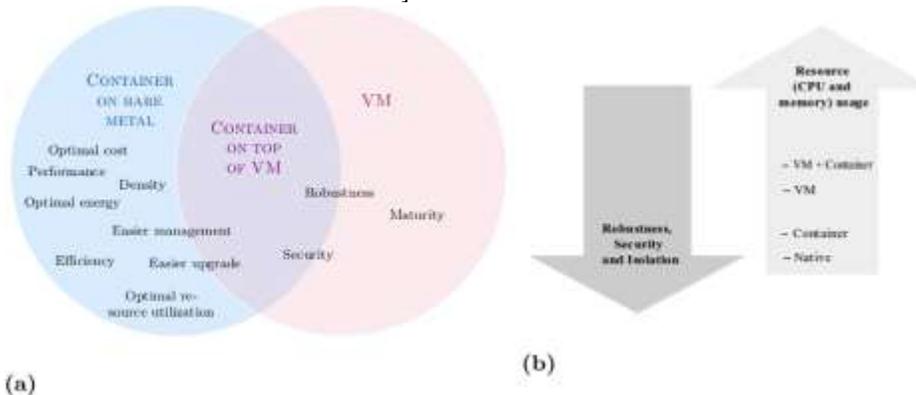


Fig. 2. Main advantages of each virtualization architecture, i.e. VM, container on bare metal and container on top of VM (a) – comparison of deployment architectures, i.e. the inverse relation between resource usage and robustness, security and isolation through the virtualization stack (b)

The time needed by Guest OS to install in terms of minutes affects rapid deployment, the hypervisor-based deployment requires different operating systems, compromising performance and cost [35]

Functionalities Since development is recent, it lacks many functionalities. More mature cost of operation, prolonged execution time and resource wastage result from VMs that perform a single function in most cases that need parts of the operating system for execution [29].

Moreover, container-based virtualization has a weak isolation [40]. With containers the processes are isolated on the shared OS. This poses serious security issues. Therefore, the risk that viruses, bugs or other intrusions could carry over from one container to the other increases [40] since user data is untrustable and not secure. This is not the case with the VMs which are more secure.

In addition, the system calls represent a significant vulnerability. Also, collocated instances of containers may affect each other if one is compromised, which is not the case with VMs thanks to the high level of performance isolation they provide [29]. Moreover, a compromised kernel compromise containers [28]. Using containers on top of VMs represents a good solution where the VM plays the role of boundary of security [35]. On the other hand, the majority of older systems use VMs. In this case, running containers on top of VMs allows us to handle the already existing infrastructures and to host container-based services [28]. It guarantees the easiest management and upgrade, an optimal energy and resource management, and allows to overcome the incompatibility between software and hardware of the PM [35]. Further, it increases security, robustness, functionality and scalability of applications by adding an extra layer of control and protection of software in cloud environments [26]. In conclusion, this configuration allows to enhance security and improve the weak isolation of containers and extends the functionalities of VMs [28]. On the other hand, this affects the performance of the container. Several factors can affect the total performance of containers when running on top of VMs, like the number of containers and VMs, the application type, the storage mechanism, the different types of workloads and the guest OS [28]. To improve the overall performance in this configuration, running several containers on VMs could be a better approach [28].

To summarize, all indications denote that containers outperforms VMs in terms of upgraded performance, scalability, energy consumption, provider profit, etc. But as mentioned earlier, containerization suffers from a weak isolation that may create significant security problems. These issues can be resolved by running containers on top of VMs. While VM presents a strong isolation, the main advantage of a container is the low performance overhead [28] due to the fact that containers share the same operating system kernel [29] that increases server consolidation percentage [25]. Despite that the performance overhead cost is important in VM and VM + container architectures, authors in [26] considered it as a trade off for virtualization's OS robustness. Adding a container to a hypervisor-based virtualization environment has no significant impact on performance overhead in terms of resources [26]. Also, the container + VM architecture provides best software isolation, security and robust environment [26] (Fig. 2(a)). Authors in [41] proved that with this configuration, the extra overhead is negligible. In terms of security, CaaS providers, like Google and Amazon, consider that containers supply suitable environments for semi-trusted workloads, while VMs offer extra security for untrusted workloads [12]. In addition, the system load optimization implementation is more easy in VMs (than containers).

The selection of technique should depend on the availability of resources, the number of competing software jobs and the type of software to be scheduled [26]. The organization applications requirements define the most suitable virtualization technique to adopt. It depends on the available infrastructure and the type of the application [35]. For the need of security and easy management, a container on top of the VM is a good alternative. If the system has to fast with low overhead, a container on bare metal is recommended. Although, if several OSs must be executed in the same PM, virtual machines represent the best alternative. In conclusion, therefore, VM-container virtualization is considered the best configuration at least until now.

Finally, containers, being more interoperable and portable [26, 39], represent a new trend, especially, for the emerging cloud native infrastructure [39]. This is due to their positive impacts, not only on development, but also on the easiest and consistent [28] deployment aspects [42]. In cloud-native platforms, based on DevOps approaches and micro-service architectures, containers can support continuous development and deployment [42], and are considered the preferred methods for the deployment of micro-services [29,43].

4. OVERVIEW OF CONSOLIDATION.

After presenting the two virtualization technologies related to our work, the aim of this section is to respond to RQ1 and present consolidation briefly by recalling certain useful definitions relating to CDC consolidation and the main reasons and motivations behind it. Further, we will present the migration process and CRIU technology in a nutshell.

4.1. WHAT IS CDC CONSOLIDATION?

The purpose of consolidation is to optimize resource utilization of data center elements while minimizing active ones through object placement and/or migration. The idea is to find the initial placement sequence or the mapping plan

after migration. This placement is an important approach to improve energy efficiency and resource usage in cloud infrastructures, among others. More specifically, when a set of applications request a cloud data center to allocate computing resources (memory, CPU, etc.), they are mapped to a set of VMs or containers. Then, containers are allocated to VMs or PMs and VMs are allocated to PMs depending on the underlying virtualization technology. Also, when it comes to storing data, placement strategies are employed [44]. During the migration process the placement is changed to improve DCs configurations. Fig. 3 shows examples of VM (Fig. 3(a)) and software (Fig. 3(b)) consolidation.

To sum up, consolidation in cloud data centers consists of grouping objects to fill unused resources [11]. These objects can be VMs, containers, software or data. Consolidation can be limited to placement which consists of mapping objects to elements that can be a PM, VM or container, or object migration which consists of moving objects from one element to another. It can be performed at the VM, container, software (application) or data level by placing or moving these objects between the PMs (VMs) regardless of the level of consolidation. Although most existing work has focused on the problem of scheduling VMs on PMs, several consolidation scenarios can take place in cloud data centers. According to the literature, the different consolidation scenarios that we have identified are:

- VM-PM [10,45-49]: to consolidate VMs on PMs,
- Software-VM [11]: to collocate several software applications on the same VM,
- Container-VM [33,49,50]: to consolidate multiple containers to a set of VMs,
- Container-VM/PM [51]: to consolidate multiple containers to a set of VMs or PMs
- Container-PM [52]: to consolidate containers on PMs under the VM-Container configuration,
- Container-VM and VM-PM [12,30]: to consolidate containers on VMs and VMs on PMs,
- Application-PM [49]: to consolidate applications (VM + container) on PMs,
- Data-PM [44]: to consolidate data on storage systems in PMs.

Several hybridizations can take place. For example, container consolidation can be coupled with VM consolidation as in [12,30] in container-VM-based virtualization architectures and the same for software and VM [53] as shown in Fig. 3(b). This allows, to maximize utilization, minimize application response time and optimize, not only the number of PMs used, but also the number of VMs and/or containers, optimizing, thus, the billing costs and power consumption and improving the overall placement performance of the executed applications. Objects and data consolidation represents similarities regardless of the level in which the consolidation is carried out because they all try to exploit the flexibility available in determining the appropriate placement.

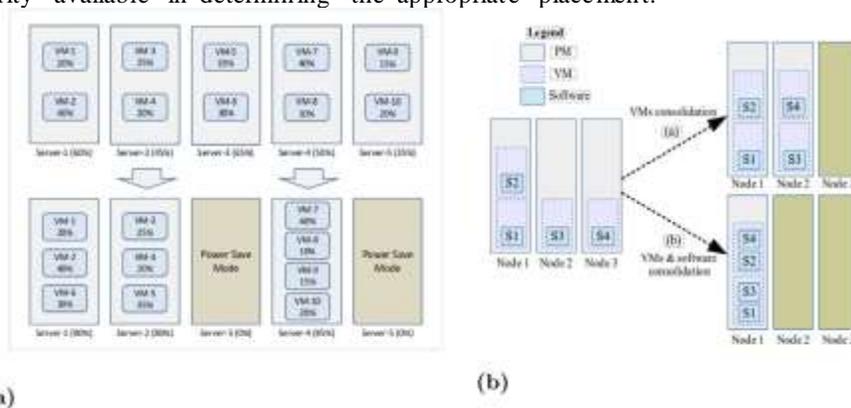


Fig. 3. Examples of consolidation: only VM consolidation, which freed up two servers (3 and 5) and allowed better utilization of the other servers (1, 2 and 4) (a)

– VM consolidation with and without software consolidation, in the second case, two machines being freed against only one machine in the first case [11] (b).

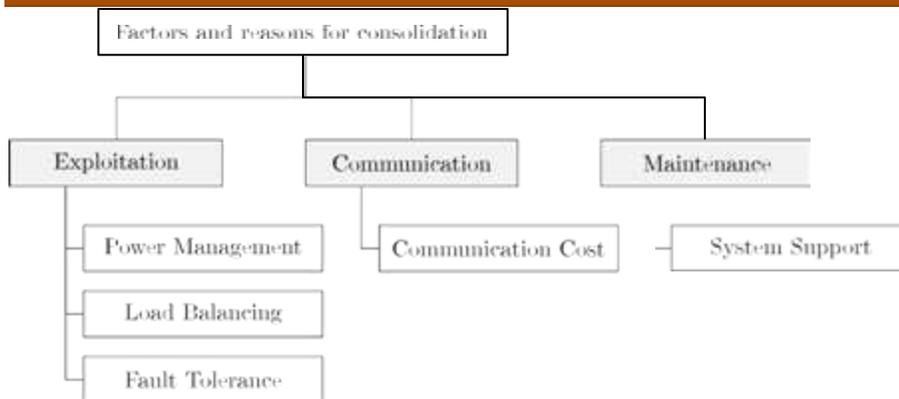


Fig. 4. Main reasons for consolidation classified in three factors.

4.2. FACTORS, REASONS AND REQUIREMENTS FOR CONSOLIDATION.

To answer the question why to consolidate in the CDC environment, this section is devoted to presenting the reasons and requirements for consolidation in the cloud. Reasons are classified in three factors related to exploitation, communication and maintenance as depicted in Fig. 4.

In terms of exploitation, consolidation is carried out for three main reasons: (i) power management, (ii) load balancing [54] and (iii) fault tolerance [55]. For the communication factor, managing and optimizing communication costs [56] is the main reason behind CDC consolidation while the system support allows maintenance through consolidation.

In addition to the reasons presented above, consolidation is performed in cloud DCs to meet several needs. As shown in Fig. 5, the main needs which aim to improve the quality process in the consolidation process are :

- Performance: to avoid performance degradation and service level agreement (SLA) violations or maximize application performance,
- Scalability: characterizes the capacity of the system to support the increasing workloads by using additional resources [57],
- Efficiency: is associated with elasticity and depicts how cloud resource can be efficiently employed as it scales up or down [32],
- Reliability: to maximize service reliability and hardware reliability [58],
- Availability: to maximize service availability that can be escaped by hardware failures.

4.3. CONSOLIDATION AND MIGRATION PROCESSES.

How to enable consolidation in CDC systems? In order to answer this question and to unify the consolidation approach, we describe a generic process by recalling the consolidation steps. This process can be defined as a chronological sequence of tasks intended to produce better configurations in the cloud data centers. The main steps in the consolidation process can be summarized as follows:

- Resource monitoring to get resource states on the basis of which the consolidation decision is made.
- Identification of situations for which consolidation must be triggered, often, based on a static threshold to specify the status of the machines and decide if they are over/under-utilized.
- The selection of objects (VM, container, software/application, or data) to migrate in order to resolve the situation in case of migration, which corresponds to new objects in the case of a new placement. Many criteria can be used in the selection, such as the correlation coefficient in [33] which allows to select the containers with the most correlated workloads with the servers that host them or those with high CPU usage. In [50], authors select the containers that may recover their migration cost.
- Destination selection for the selected objects after requirements analysis, mainly, in resources to respect PMs and objects capacities. In this phase, the host destination is selected according to several strategies. For example, in [52] two strategies were tested, one based on the workload history between container and PM to select the most irrelevant

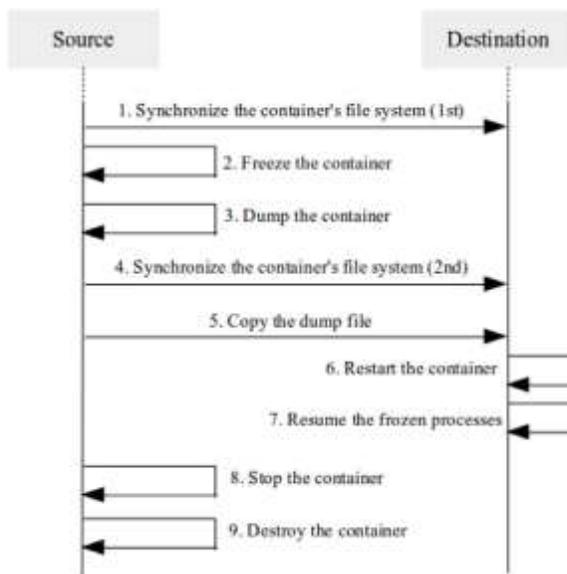


Fig. 5. Requirements for consolidation.

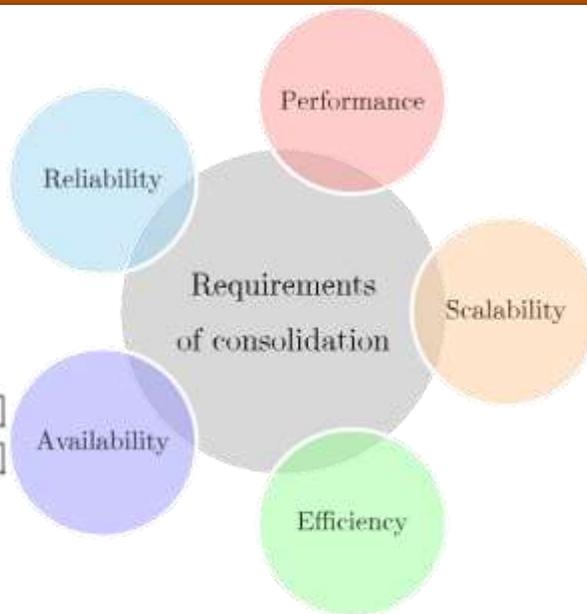


Fig. 6. Container live migration with CRIU.

PM and the other based on the CPU usage to select the PM with the least utilization. In addition, in the case of container consolidation, the destination VM must also be selected. In the consolidation of software, the PM, VM and/or the destination container is selected according to the adopted virtualization architecture.

Execution of consolidation by placing the new objects or migrating objects already placed according to the appropriate reconfiguration plan chosen. This consists of object mapping onto elements in order to implement the new placement which improves the previous data center configuration. A prediction phase can take place to obtain additional estimates on the state of resources, workloads, etc., in order to further improve the consolidation results.

Consolidation can involve a migration process. To perform the latter, and depending on how the memory state is transferred between source and destination host, we have identified several migration patterns (Table 2) as will be presented later (Section 5 Type).

These patterns are widely used in virtual machine migration. For container migration, the most used technique according to the literature is based on checkpoint restore (C/R) mechanism. This technique consists of taking a snapshot of the state of the application and then restarting it later at that saved point. Checkpoint/restore can be performed at application, user, OS or hypervisor level [60]. One of the most well-known tools that uses this technique is the Linux project Checkpoint/Restore In Userspace (CRIU) [61]. In container context, it is implemented as loadable modules and uses the available kernel interfaces such as *ptrace* system [62] allowing a process to control the execution and changes the registers and memory of another process. For non-live migration, according to [59], the C/R is implemented by: (i) check pointing the program at the source (generally as files),

(ii) copying the check pointed data from the source machine to the destination, and (iii) restarting the program at destination. Although the CRIU technique supports both migration models (live and non-live) [59], it is more widely used in the live migration of container service. This involves detaching and reattaching a set of processes running in a container to a new OS kernel after transferring them to a remote machine [62] as shown in Fig. 6. This container instance moving is done transparently i.e. while maintaining open network connections and preserving the state of the running containerized applications, and resources (CPU, network, disk and memory states). CRIU adopts an incremental pre-dump to support pre-copy by using a feature named "soft-dirty bit" which is implemented in the kernel of the Linux system and allowing to track the memory changes [62]. As in post-copy schema the state of the CPU is transferred first and then the memory pages, the migrated application may access missing memory pages. To handle this page fault, CRIU performs on-demand paging by injecting the required pages in the memory address space of running tasks after transferring them from the source machine. This user-space paging is implemented by the new mechanism added to the Linux kernel named *user faultfd* [62]. A new project named P. Haul which is based on the CRIU pre-dump action, works similar to VM pre-copy manner. It allows us to conduct a series of incremental dumps [63] while maintaining the running of the container. This resolves the halting of containers caused by the stop-and-copy technique proceeding to the one time dump of the state of the container, which seems insufficient for today's latency-sensitive complex service containers [63].

4.4. DISCUSSION ON MIGRATION COST.

After briefly presenting some migration patterns and technologies, in this paragraph, we will discuss the main criteria that make the migration process costly. Migration costs were not adequately addressed in research [64]. The fact that the cost of migration is sometimes higher than the energy savings makes it a very important factor that should no longer be ignored in consolidation work [64]. Generally, to characterize migration, a migration cost (MC) and a migration duration are considered [65] although in some works like [66] the migration costs are reduced to the migration time. The cost of migration tends to estimate the impact of the migration process on the applications and the resulting loss of revenue [65]. Several factors can affect these costs so they can vary considerably.

The different migration orders cause different migration costs for the same migrated objects combination [66]. Live migration incurs additional overhead [67] and causes performance degradation of migrated objects and can interrupt their services. It aggravates the overhead of data transmissions and produces additional energy consumption in CDC [64] as discussed above. In pre-copy schema, the downtime is optimal [59] but for memory intensive applications, this time is important along with the network traffic and migration cost because pre-copy synchronizes the state of migrated objects at the source and destination. For this type of application, the post-copy outperforms and minimizes the TMT [59]. To optimize the application performance and increase its reliability [62], hybrid migration represents the best choice [59]. The migration time in non-live checkpoint based migration is more optimal compared to live migration because the former, after saving everything in persistent storage during the check pointing phase, can kill the check pointed application at the source [59]. For live container migration, in a pre-copy pattern, the migrated container continues its execution in the source node and its memory is transferred iteratively. Then, when the dirtied memory is small, the container is posed while copying the remaining states to the destination [68]. This schema allows to minimize the downtime. In Voyager [70] for example, the total downtime, which reflects the time needed to transfer the in-memory state via CRIU is between 2–3 s. However, in terms of TMT, migration lasts longer, especially in case of very high pages dirtying rate because of iterative memory copying and on-demand pages fetching [68]. Thus, the resulting massive network traffic and costly page fault memory with its related latency impact negatively the application performance during TMT [59]. A solution proposed in [68] to solve this consists in migrating containers intra-machine to avoid the costly inter-machine network communication and guarantees a fast state transfer. In post-copy container technique, pages are migrated only when they refer to the destination machine.

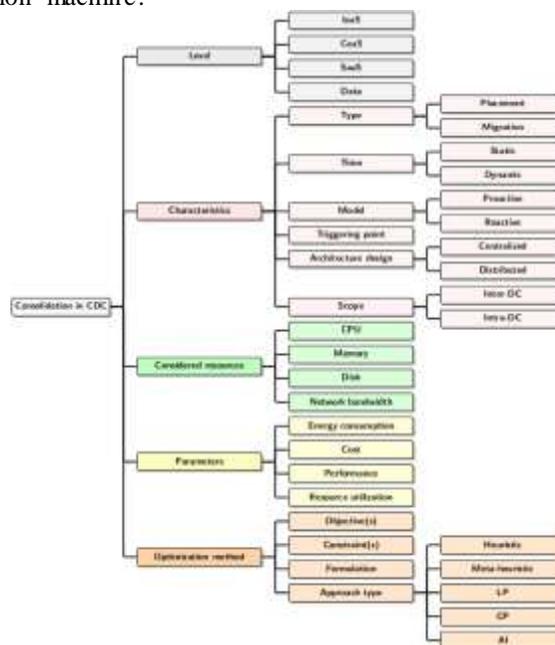


Fig. 7. Consolidation taxonomy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

On demand page migration, although it reduces the initial cost of migration, it increases the TMT [68]. In [50], the authors considered an additional migration cost component which they called marginal cost. This cost is due to the power switch on/off of hosts and the cost needed to start a new VM or terminate an existing one if necessary during migration as they have adopted a VM-container configuration. We can conclude that the migration cost increases with increasing memory size and decreasing network bandwidth [66]. Increasing the bandwidth optimizes the migration

time and thus the migration cost, but it also increases the energy consumed during migration. Thereby, a trade-off between power cost and migration cost is necessary [65]. If available network bandwidth is considered as a fixed quantity during migration, the cost of migration greatly depends on the number of migrated objects and the size of each object. Authors in [66] consider that the factors affecting the migration cost can only be known in the probabilistic sense and proposed a probabilistic approach to quantify it.

5. CONSOLIDATION TAXONOMY AND LITERATURE SYNTHESIS.

Consolidation works in cloud systems touch on several aspects. Thus, presentation and classification can be done according to many parameters and considerations. The taxonomy given in Fig. 7 below allows us to synthesize several aspects relating to consolidation problems in cloud data centers.

We have grouped the aspects used to describe the consolidation work under the characteristics heading (the part of taxonomy colored in pink). The level of service in which the consolidation is performed is detailed in the level subsection (the part of taxonomy colored in light gray). Optimization method (the part of taxonomy colored in orange) designates the method adopted in the consolidation process in order to optimize one or more objective according to well-defined metrics. In addition to these metrics, other metrics are used in the evaluation. The all are grouped in the parameters subsection (the part of taxonomy colored in yellow). The considered resources are also presented which correspond to the part of taxonomy colored in green.

5.1. LEVEL.

As we all know, there are three standardized service models in cloud computing, namely SaaS, PaaS and IaaS. Virtual machines represent hardware virtualization and belong to the IaaS level. As the main objective of consolidation is to keep the minimum number of active physical machines to minimize energy consumption and costs and improve performance, among others, consolidation works were initially oriented towards infrastructure by multiplexing several VMs in the same PM. Many works have addressed the importance of virtual machine placement appropriately. However, containers represent a major trend in virtualized cloud environments as mentioned in Section 3. Thus, container-based virtualization is another way that allows the evolution of consolidation towards the platform and CaaS levels. We were able to identify some container consolidation work [12,30,33,49–52] which are recent and whose objective is to consolidate containers on the smallest number of VMs and/or PMs [33]. Ambitions in the academic environment are constantly evolving in this area and have enabled the implementation of consolidation at SaaS and data levels with software and data consolidation, respectively. Just recently, depending on the underlying virtualization technology, multiple consolidation has become possible since containers and virtual machines can be migrated alternately or concurrently [49]. In our article, we adopt the service level in which consolidation is performed as the basic criterion for classification and synthesis of some representative examples of literature work. Table 3 presents a non-exhaustive list of these works with a brief description of the purpose of each one. Regardless of the level of service, consolidation works involve similar aspects.

5.2. PARAMETERS.

Several variables and metrics were considered in the consolidation process as well as for the evaluation of the proposed approaches. We have grouped these metrics and variables in the parameter section which are presented together because there are parameters which are used at the same time in consolidation and evaluation. In Table 5, we present a non-exhaustive list of parameters considered in surveyed consolidation work. Metrics are important aspects of decision making in the consolidation process. They are variables relating to measurements i.e. the measurements carried out within the DC cloud to trigger the consolidation process or estimate the optimization variables. Often, system metrics that are related to resources are the most used in literature work, but there are high level application metrics that can be used to further improve consolidation results. The parameters used in the algorithms of literature approaches such as the concentration of pheromone [45] in ACO-based approaches are not exposed here. For the control of the consolidation process, as mentioned above, the majority of works use the utilization rate of system resources (memory, CPU, Disk, I/O, etc.) to start the consolidation. However, this rate does not always represent the best sign for decision making. The best indicator is the behavior of the application. If it works well even with a high resource utilization rate, then the consolidation process is costly without the need. In this case, high-level application metrics like error rate, latency, throughput, etc., are to be considered as primary motivators for making scaling decisions and no longer system metrics.

In some cases, advanced consolidation features like affinity [12] are used which complicates the decision process.

Concerning the optimization variables, they are generally based on metrics. For example, the energy consumption

of a server can be described by a linear relationship between the CPU utilization and the electric power consumption [45] of PM when its utilization is 0% (piddle) and 100% (pbusy). A nonlinear model of energy consumption was used in [78] which is not limited to the CPU resource, but also considers the influence of memory utilization on the PM's energy consumption. Power consumption is calculated according to the PM utilization. In this case, the utilization and the values of the electric power (piddle and pbusy) are the measurements carried out i.e. the metrics, and the electric energy is the objective of optimization (i.e. decision variable). Power utilization of CPU is the widely used to estimate the energy consumption of servers since it occupies the most important part of the power. The power of the processor is characterized by the number of instructions it is able to process per second in million instructions per second (MIPS).

In general, the power is determined based on other measures such as CPU occupation, memory occupation, the overhead of VMs, overhead of memory, number of containers assigned to each VM, etc. These latter variables are low-level metrics. In fact, the main optimization variables treated in the consolidation work are energy consumption, and costs. The cloud is a high energy consumer. In 2012, this energy monster consumed 2% of the global energy in the world and continues to grow [19]. This rate of energy consumption increases operating costs and poses environmental challenges with a significant emission of carbon dioxide (CO₂).

Energy consumption in cloud data centers is formed by consumption at the server, network equipment, storage and infrastructure level [9]. The physical servers occupy the most important part of this consumption, which represents approximately 60% of the total consumption [7], as shown in Fig. 8.

For the remaining consumption, cooling systems and network elements are the two main power consumers in a data center, after the servers which have the biggest share [33].

The consumption at the server level is composed by dynamic contribution and static contribution. The latter represents around 70% and is strongly correlated with temperature due to the leakage currents which increases as the technology decreases [48]. Researchers, mainly, focus on static consumption with the aim of minimizing the number of active servers and thus the energy consumed by the latter through consolidation. However, many researchers have treated dynamic contributions based on strategies like DVFS. This technique balances the power, according to the workload in the system by reducing the operating voltage and the frequency. The principle is the dynamic scaling of the voltage and the frequency by modifying the frequency according to the variations in the utilization made by the dynamic workload. These policies help to reduce data center power consumption under dynamic workload conditions, and are frequency dependent. DVFS is an architecture level technique and it is by far one of the most effective common methods to achieve energy savings but also degrades system performance.

Economically speaking, the cloud optimizes costs for the client compared to legacy systems. Clients pay their consumption without incurring installation, update, maintenance costs, etc. However, for the provider, cost management represents a great challenge. From a provider point of view, the key issue is to maximize profits by minimizing operational costs.

6. EVALUATION METHOD.

With the cloud computing service and deployment models, we can imagine several execution scenarios in cloud data centers. Thus, to evaluate the performance of consolidation work, several aspects can be considered. The evaluation method is closely related to the research methodology adopted (fundamental/applied, quantitative/qualitative, descriptive/explanatory, etc.). In the field of cloud computing, to evaluate the performance of the work carried out on consolidation, the evaluation method tends to specify, mainly, the evaluation platform used, the datasets, the different configurations as well as the evaluation metrics used. In addition, the majority of studies present a comparison with other consolidation approaches to judge the performance of the research carried out. In Fig. 9, we represent the main dimensions of the evaluation method.

Regarding the platform, the experiments can be carried out either in real cloud environments or in the form of a simulation. In the latter case, several simulators were used and we will cite some of the best known simulators used in surveyed papers. The most popular simulator is CloudSim [81], an open source simulator written in Java language used for example in [33,46,48–50]. CloudSim, unlike other simulators, provides the management of on-demand resource provisioning which represents accurately the models of virtualized DCs [48]. New versions from version 2.1 used by Rajkumar et al. supports the energy consumption accounting as well as the execution of service applications with workloads that vary over time [82]. Cloud Analyst [83] is another simulator developed at the University of Melbourne and based on the Java CloudSim simulator. It is suitable for consolidation work treating geographic distance as [47] because it presents evaluation tools according to geographic distribution of data centers and users [83]. CloudSim was extended to model a CaaS provider. Container cloudsim [84] is a simulator which extends the cloudsim and which is a containerized cloud toolkit for simulation that applies VM-Container

configuration as a virtual environment. It was used in the evaluation of container consolidation work such as [49,51,52]. Otherwise, an in-depth study of simulators in cloud computing is provided here [85,86] to which readers can refer.

As mentioned earlier, several configurations can take place to experiment and evaluate consolidation work which defines problem instances, and size (ie. number of PMs, VMs, containers and the storage system in data consolidation). All these elements can be homogeneous, which means that they have identical capacities of resource (the considered resources in general are CPU and memory) or heterogeneous otherwise [12]. Finally, several datasets and benchmarks were used to measure the performance of the consolidation work. Some works have been based on real workloads i.e. traces from real cloud as the real workload traces provided by CloudSim, and others on a synthetic workload like TPC-C (Transaction Processing Performance Council — Type C) and TPC-H (Transaction Processing Council Ad-hoc) used in [44]. Below a list of the most known and used datasets in surveyed papers, rather than exhaustive.

- PlanetLab Workload : is composed by data traces obtained from the monitoring system for PlanetLab [87], the CoMon project [88]. PlanetLab contains more than a thousand computers located at more than 600 sites scattered around the world and available as a testbed for distributed systems and computer networking research. The workload consists of data collected every five minutes with different resources.

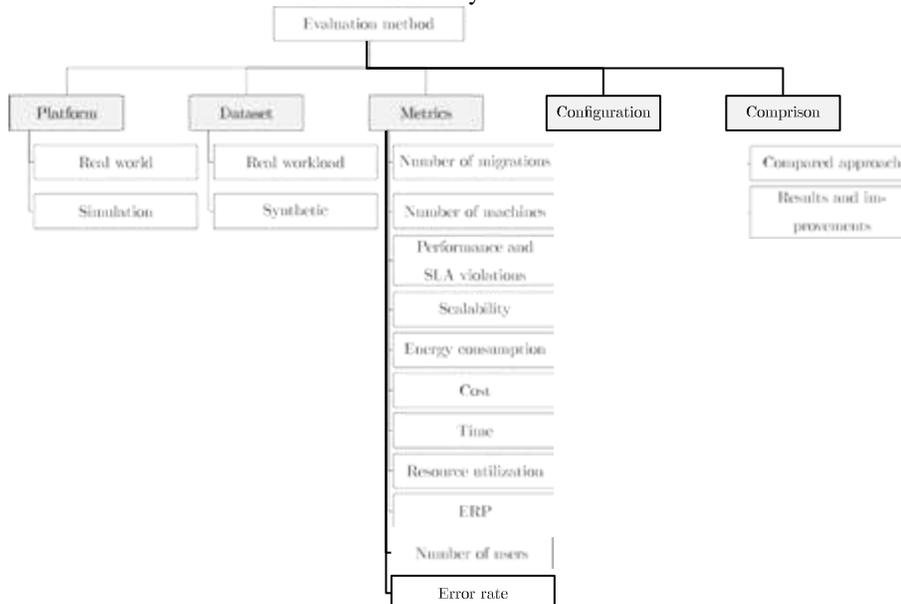


Fig. 9. Evaluation method taxonomy — Metrics: Number of migrations (VMs/containers), Number of machines (PMs/VMs), Energy consumption (booting energy, number of power events, peak power, cumulative energy), Cost (total cost, electricity bill (in the U.S\$)), Time (latency, execution time, migration time, algorithm execution time), Resource utilization (resource demand, global utilization, resource usage of PMs and VMs, remaining resource waste of PMs and VMs).

- Google Workload : is composed of thousands of jobs from Google servers with one or more task each. The workload contains samples of memory and CPU utilization for jobs per task [89].
- TU-Berlin workload : is composed of server traces of the Technical University of Berlin (TU-Berlin) and contains periodic samples of memory and CPU utilization for servers per hour in a week period [10].
- SPECjms2007 : is the industry-standard benchmark for evaluating the performance of enterprise MOM (Message Oriented Middleware) servers based on JMS (Java Message Service). SPECjms2007 models the supply chain for a chain of supermarkets [11]. It was used to bench the Joram servers [90] in [11].

Table 8 summarizes surveyed papers according to the evaluation method. In this table, we briefly present some improvements of the approaches proposed in the surveyed articles against some compared approaches. Still, to better illustrate the behavior of the consolidation service in different cloud service levels, we discuss the results of some relevant consolidation work in general in the next section.

7. ANALYSIS AND DISCUSSION OF THE RESULTS OF SOME RELEVANT WORKS STUDIED ON CONSOLIDATION.

After synthesizing some reference articles and presenting the evaluation method, this section is devoted to

illustrating and discuss the results of some relevant work to show how consolidation performs in different levels according to the proposed taxonomy (Section 5). Indeed, several factors can affect the consolidation results such as the heterogeneity of hardware platforms and resources where similar applications may perform differently on different machines [50]. Other factors concern multi-modality, mainly related to the architecture of the CPU and the performance of the workload [49] as well as the different heterogeneous service levels. In addition, the different types of workload and consolidation policies and algorithms that take into account different goals or inputs often make their comparison difficult or even impossible [91].

Fig. 10(a) corresponds to the work of Tchana et al. [11]. The main idea was to resolve the under-utilization of virtual machines by dynamically consolidating several software in a few number of VMs to reduce the number of VMs used and conclude free VMs. This, in combination with VM consolidation, allows to further minimize the number of active PMs. To test their solution, Tchana et al. have used a cloud hosting an internet server and JMS messaging using the enterprise LAMP and SPECjms2007 benchmarks on private cloud (Vmware) and public cloud (Amazon EC2). They considered a SaaS offering two applications; Joram server (for JMS messaging) and LAMP server (for web application). In addition, two situations are presented; With Software Consolidation (WSC) where both software and virtual machine consolidation are enabled, and WithOut Software Consolidation (WOSC) where only VM consolidation is enabled. Running 6 LAMP and 15 SPECJms2007 scenarios over 30 h providing a varied workload, simulation results showed that software consolidation SC accelerates the consolidation of virtual machines with 2 freed PMs more as shown in Fig. 10(b) compared to only VM consolidation that freed only one PM. Around 37 VMs (M1 medium instances) on Amazon EC2 are used, configured to run for an hour and charged at \$0.120 per VM per hour. As shown in Fig. 10(a), During the 25 h, the number of VMs suffered a drastic decrement. This saves around \$500 i.e. 40.5% of cost saving in public cloud as shown in Fig. 10(b). Minimizing the VM number allows more flexibility and offers more possibilities to minimize the number of PMs thus minimizing the overall energy consumed for a specific workload. With the suggested approach, applied in a private cloud, approximately 40% of power being saved with the particular used workload. In [49], Khan et al. have dealt with consolidation in different levels of service by making the best decision of migrating VM, containers or applications (Vms + containers) which optimizes energy consumption while preserving performance. The heterogeneity of applications is modeled based on the priority of tasks representing the type of workload. The execution times of containers were used as a performance metric by the fact that workload runtime and performance are inversely proportional.

[52] Simulation using container Cloud Sim Energy consumption and the amount of container migrations 300, 500 and 2500 heterogeneous PMs, VMs and containers respectively Real-world workload traces from PlanetLab CorHS, LFHS

It outperforms the other two algorithms over time consuming about 10% less energy during a one-day simulation, while the container migrations are also about 15% less on average

[30] Simulation Resource usage and remaining resource waste of PMs and VMs CloudSim Total number of container migration, total energy consumption, execution time (in second), ERP, electricity bill (in U.S \$) 12,583 heterogeneous PMs, around one million tasks (3 different kinds of workload), 3,800 VMs of 6 types (Amazon's instance types) Google workload traces Scheduling: RR, FF, R, FU and EPC-FU, various migration possibilities: no migration, all possible migrations and CPER, CPER was also compared to CMCR (VMs

migration) EPC-FU allows energy savings by 1.27% (no migration) and 0.93% (dynamic migration) respectively, compared to RR, and 1.19% compared to R, it also allows 0.71 more cost efficient than FF (with CPER), in term of performance, FF, FU and EPC-FU are most efficient and produces optimal results with CPER, by combining CPER and EPC-FU 61.89% of migratable containers were able to recover their migration cost

Naive algorithm, i) both software and VM consolidation enabled, ii) only VM consolidation

For small instances: G-COPS and H-COPS reduce the cost of OPA by

50% and 44% and the cost of DOT by 11% and 6% respectively,

For large instances: H-COPS outperforms OPA by about 46% in term of cost and executed in a reasonable time (<1s), it gave an error rate less than 10% compared to the exact solution

Multiple [49] Simulation using Cloud Sim and container Cloud Sim N overall cost, the authors considered that services are charged at \$0.095 per hour and a tariff plan of 1 min with an energy cost of \$0.08 per kWh. In order to evaluate their work, Khan et al. have considered several consolidation possibilities depending on the migration of VM, container, application (VM + container) or no migration.

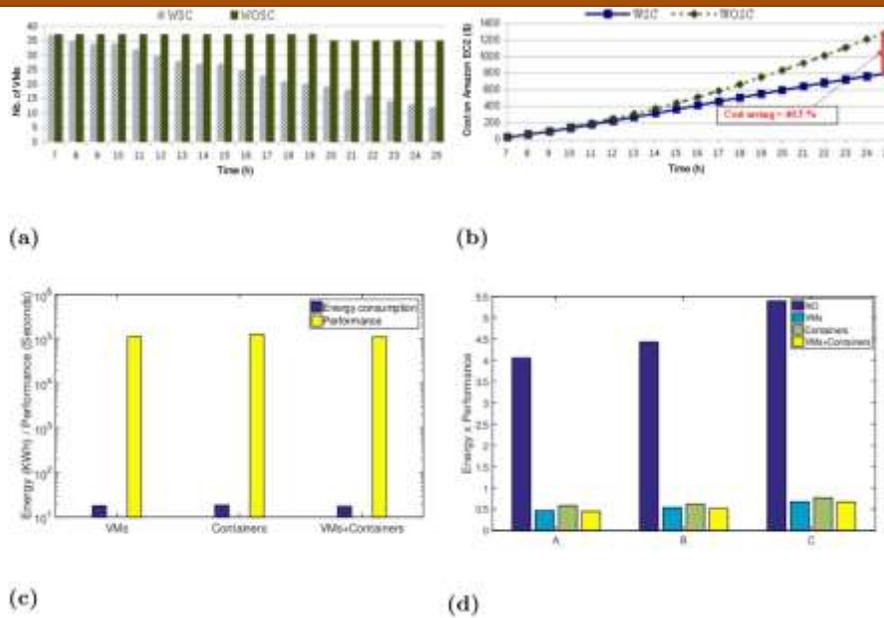


Fig. 10. Cost saving on Amazon EC2: nb. of VMs per hour (a), VMs charged (b) [11] — Average energy consumption and workload performance when VMs, containers or applications (VMs + containers) are migrated (c), Energy Runtime (Performance) Product (d) [49].

A mixed workload was used based on PlanetLab and Google datasets. The data (CPU utilization) provided by PlanetLab were recorded at 5 min regular intervals from more than 1000 VMs whose extend is over 10 days and which are available in CloudSim. From Google dataset, the runtimes and priorities of tasks and VM runtimes were selected from the containerized Google’s platform. With the suggested approach, the authors concluded that migrating containers allow significant savings in terms of energy, while VM migration is more efficient in terms of performance. The authors have shown how application migration can balance between energy and performance. Fig. 10(c) shows that VM migration surpasses container migration in terms of energy (around 3.2% more efficient) and performance (around 11.2% more efficient). But it is less efficient than application migration (both container and VM migration) which provided around 11.9% more in terms of performance and 5.5% in terms of energy. To illustrate the energy optimization results under performance constraints (Fig. 10(d)), the product of energy and performance (ERP) was used. Results affirmed that consolidating VMs + container for three different types of applications have the lower ERP. Hence, it achieves the best result in terms of performance and energy. When migrating applications, around 707.54 kWh energy can be saved, and 4.16 kWh and 10.15 kWh more compared to VM and container migration respectively. For \$0.08 per hour energy price, this can translate to almost \$0.6m annual savings. Khan et al. observed that increasing the number of container migrations due to their size can increase energy consumption and migrating the same container multiple times can create performance loss. As a proposal to address this, they considered the migration of containers running inside VMs which, according to their experiments, allows for a trade-off between energy and performance. Nevertheless, if we consider a whole cost that integrates the energy and performance in terms of virtualization architectures (and resulted overhead) along with the migration costs, these results remain to be approved. The number of containers in experimentation is more important than the VMs one. If a VM is migrated several times or a large number of VMs are migrated, certainly this could produce different findings. The authors discussed the effect of the repeated migration of containers that may affect negatively the performance of work- load, which was not addressed in SC. In SC, the authors have shown how consolidation can improve power consumption with- out worrying about how migration impacts performance which was addressed in [49]. In the latter, the authors raised the trade- off involved between migrating VMs and containers. They showed that migrating containers can be more energy efficient than VMs which confirms the results in [11]. On the other hand, migrating VMs is more performance efficient. To this end, we believe that performance considerations are important to validate the gain of migrating software (running in containers) because the performance losses can exceed the energy gains, for example, if the migration policies are very expensive. Authors in [49] suggest that the migration of VM + containers can achieve a balance. Thus, EPC [49] compared to SC [11] allows further savings of 14.6% energy and 7.9% performance.

8. MAJOR FINDINGS AND RESEARCH CHALLENGES.

8.1. MAJOR FINDINGS.

We carefully selected 14 papers from a number of works reviewed regarding consolidation in cloud data centers, and presented a deep-diving analysis of these articles. This work has been summarized in tables after discussing the main aspects relating to consolidation as illustrated in the proposed taxonomies.

On the other hand, it is worth noting that containers have enabled new use cases in cloud computing, such as just in time deployment, microservices management and orchestration (ex- ample by Kubernetes). They are encapsulated in virtual machines to enhance security and isolation [68]. To migrate containers, the CRIU technology is the most used. It allows to freezing all or parts of a running application and checkpoint it in the form of files. Then, by restoring those files at the destination node, applications can resume execution at frozen point. Container memory copying can be performed in a pre-copy or post-copy manner depending on the order of transfer of the processor and the memory. In the pre-copy pattern, the memory is transferred iteratively before the processor state and the opposite in post-copy [50]. As we discussed in Section 3, containers are lighter and more agile than VMs with a relatively short lifetimes and a high density deployment. This allows consolidation to be performed at a finer granularity and allows a more flexible and optimized migration service. However, container service and application migration can create a huge traffic with very high migration cost. Therefore, it will be necessary to balance between the fact of “reducing” the migration problem and the gains of the consolidation service.

In fact, consolidation brings obvious benefits. For the client, he demands low prices and optimized QoS. From the provider side, the major concerns are to make profit and offer an acceptable quality of service. With the increasing adoption of cloud and the panoply of service models offered by the cloud actors, allowing to leasing virtualized hardware (IaaS) or running software (SaaS) or platform (PaaS), the task is more and more challenging to cloud providers to make a profit, and at the same time satisfy the customer. In Fig. 11, we list the main benefits of service consolidation from cloud provider’s perspective.

Cloud providers aim to guarantee their profitability while choosing the best configuration of resources which also guarantees the requirements of clients as specified in the SLA. For IaaS providers, the main impacts are those related to DC utilization and VM performance that directly impacts provider revenue [50]. In SaaS/PaaS cases, in order to decrease the monetary costs, providers, mainly, tend to reduce the entire capacity of provisioned resources by minimizing the number of virtual machines [95]. The SaaS provider can play the role of a client of IaaS provider by leasing VM instances, for example on peak demand periods, to be able to accept and satisfy their customer requests when its reserved infrastructure (and/or its computational resources) are totally busy, and thus maximize profit [96]. It can also use its own private virtualized clusters. In the first case, two levels of SLA are to be considered: (i) customer QoS parameters such as the response time, and infrastructure level parameters such as the service initiation time [95]. Although in both cases, minimizing the number of VMs may improve the provider profit, the first case allows more important saving since no capital expenditure (CAPEX) required.

8.2. RESEARCH CHALLENGES.

Instead of fully studied fields, of which we have presented some relevant ones, we conclude that there are other important fields not yet explored or thoroughly studied such as service continuity while migrating objects. Some interesting aspects which do not only concern consolidation parameters have not been explored in the literature on which we will focus on the following.

Interference implications. Interference and interdependence issues may result by the fact that performance isolation between co-located objects cannot be guaranteed by virtualization technologies [97]. In literature, some works have addressed the implications of co-located interference effects such as [79,97,98], and the correlation of resource utilization between co-located VMs [99]. However, interference parameter remains very little explored and the works that addressed it have only dealt with the infrastructure-level interference and have omitted communications and interferences between containers [51], software applications and data interdependencies.

Energy consumption in DC level. Energy consumption has aroused the interest of several researchers. Most of them concentrate on server level energy consumption, fewer are the work that treated the consumption of storage [44] and network [100,101] equipments, and energy consumption at the infrastructure level (i.e. lighting, cooling, controls) is neglected. As mentioned above, DC level energy consumption, which depends on the consumption of IT equipment and that at the network architecture level represents 40% of total consumption, which is enormous. Unfortunately, this consumption has not been sufficiently addressed. Interest focused mainly on consumption at the server level, which represents 60% of total consumption, focusing on the static contribution which represents 70% of this consumption. The objective was to minimize the number of active servers and thus the energy consumed by them but this increases their use. The dynamic consumption at server level has been the subject of some work by

applying the DVFS technique [48,102] which, applied to the server CPU, can save up to 34% of this consumption [103].

However, this remains insufficient compared to DC level energy consumption. To optimize this consumption, in [104] a network architecture for DCs called Exchanged Cube-Connected Cycles (ExCCC) was proposed. By adopting this strategy, we must review the entire network architecture which is not practical for several cloud providers. Reviewing the energy consumption at DC level allows to improve the return on investment of cloud providers saving more energy in CDCs. Also, optimizing this consumption helps combat pollution and global warming by decreasing the CO₂ emission.

Uncertainty in cloud environment. Cloud computing is a dynamic and highly changing environment. These large-scale virtualized distributed systems differ from previous environments in the way that they introduce continuous uncertainty into the calculation process [105]. Uncertainty represents a main difficulty in cloud adoption. This creates additional challenges in terms of resource management and associated costs. Indeed, the sources of uncertainty in these environments are multiple and depend on several factors. The elasticity related problems, the energy consumption and the dynamic pricing are the important sources of uncertainty according to Tchemykh et al. [106]. In addition to dynamic elasticity, different service levels and real-time constraints, we cite:

- Resource and service cost: These costs are based on many factors, mainly involving the technical characteristics of the equipment. However, it is impossible to obtain exact knowledge about the system. Parameters such as the effective processor speed, the number of available processors and the actual bandwidth change over time [105]. In addition, the demands for workload vary over time. Providers may not know the amount of data and the computation required by users, which makes it very difficult, if not impossible, to have correct knowledge about users and their actual needs. Thus, the workload in this environment is unpredictable [107]. It is unstable and varies randomly over time dramatically and depends, in large part, on user behavior.
- Virtualization [108]: Virtualization makes it possible to run multiple operating systems and multiple applications on the same server and at the same time. With the rapid proliferation of virtual machines and containers, administrators and authorized users can create a large number of VMs or containers in a matter of minutes. Often, these elements are transient and are removed after a few weeks. Thus, the volatility of VMs and containers as well as the ease of their handling make resources fluctuating and difficult to estimate accurately.
- Decoupling between applications and infrastructures: The dematerialization and disconnection between hardware and software in the cloud makes tracking virtualized applications that are not traditionally installed, and thus the estimation of its real resource needs and relative revenues a tedious task.

To sum up, the set of dynamically virtualized and scalable IT resources, storage, software and services have made it impossible

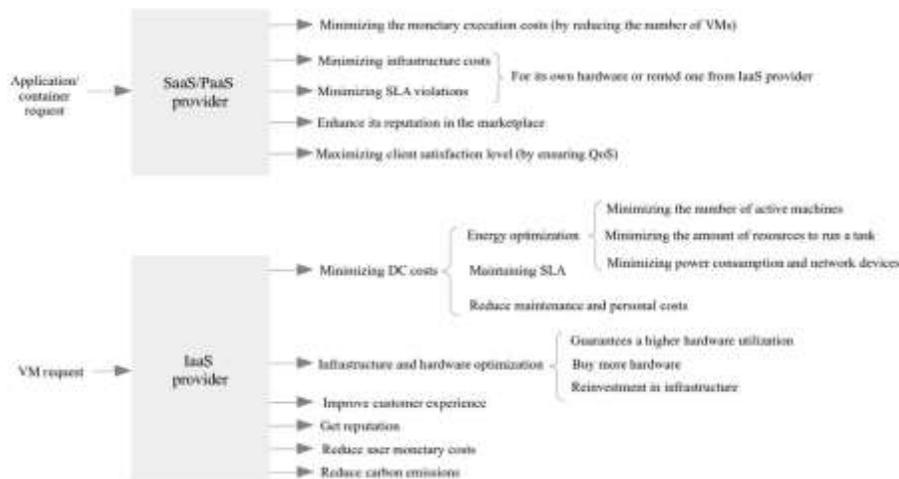


Fig. 11. Main benefits of consolidation service from providers perspective.

to have exact knowledge of the real needs of users and their applications which are not predictable and which are difficult to estimate. This incompleteness due to the lack of information on applications and resources creates a dimension of uncertainty which greatly depends on the users who share resources and their behavior. However, other than the sources of uncertainty that can affect the consolidation of resources, the migration and consolidation policies themselves represent sources of uncertainty affecting, mainly, processing time, performance, bandwidth, etc. [108]. Like the elasticity policies, the total accuracy of consolidation policies remains subjective [107]. This makes the effect of any optimization rule through the consolidation process prone to uncertainty. Few are the works that have addressed the uncertainty in the field of the consolidation of cloud resources, only at IaaS level, which we quote for

example [109] and [110]. Despite extensive research on uncertainty issues in computational biology, decision making in economics, etc. [111], a study of the uncertainty for cloud computing and in terms of resource management is limited [106]. Thus, designing consolidation strategies in the presence of uncertainties due to the dynamic behavior of the execution context and resources, to the properties of services relative to the characteristics of the cloud environment and to user behavior, represents a challenge.

Software license consolidation. Software services represent the most important part of the evolution in the cloud computing and do not cease increasing. This is due to the popularity of the cloud and the importance of software which are in the center of the concerns of any company. According to forecasts between 2016 and 2021, it was proven that, in the corporate cloud, the SaaS market will remain dominant even in 2021, representing \$97 billion in global spending [112]. With the increasing popularity of cloud and SaaS, the focus today is not just on data but on development. Unfortunately, software license consolidation has not been investigated yet. Literature work focused on the technical characteristics of equipment and the optimization of standard resources (network, CPU, memory, etc.) and have omitted software licenses that are typically an even more precious and expensive resource than compute-power [113]. Thus, reviewing the management and optimization of software licenses in the commercial cloud has become a priority. Licenses must be provisioned and released “on the fly” like other resources, elastically, to avoid problems of under/over provisioning and overpaying. In other words, they must be dynamically adjusted according to needs and actual use. The aim is to further minimize the number of PMs, VMs [12] and/or containers in the commercial cloud and optimize the overall cost including the cost of licenses in addition to energy cost.

Security and privacy. The pervasiveness and openness of cloud environments along with the accessibility of cloud services via the internet cause severe security issues through the consolidation process. Many risks and attacks can take place along this process. For example, while migrating objects from one PM to another, if the links between these two hosts are not safe, the identity of the destination host can be usurped and migrated objects can be redirected to another host, or migration can be initiated by a malicious host. Unfortunately, consolidation works did not give much importance to security considerations [114] which is one of the crucial factors that should be considered in the future research. In the literature we have identified very few works that address security as [115] which only deals with the case of VM consolidation. Many of the security issues, that should be addressed, are related to the characteristics of cloud systems and virtualization type. We specified in Section 3 that a container-based architecture which is an OS level virtualization, although it is a good virtualization alternative thanks to its agility, lightweight and scalability among others, suffer from a weak isolation level [40] and presents enormous risks. Unlike VMs that provide security for untrusted workloads, containers, as CaaS providers such as Google and Amazon consider, provide suitable environments for semi-trusted workloads [12]. Practically, all security properties are involved through the consolidation process. We identify some research questions and needs in terms of security below from a consolidation perspective.

Authentication and access control: nothing can prevent a malicious host to initiate a migration if the authentication step was not performed to establish a trust between two hosts implied in consolidation. So, how to guarantee that the migration process is performed by an authorized person? And what policies should be used to control his access to reduce risks of attacks?

Confidentiality, integrity and non-repudiation: how to guarantee that data and applications moved from source to destination PM are the same and have not undergone modifications? How to provide a high level of confidence so that these data and applications are not disclosed? With the dematerialization and the multiplication of the different levels of service and the different scenarios and actors involved in cloud computing how to guarantee that the data emanate from source hosts who claim to be it? This concerns the better monitoring to avoid the non-repudiation property.

Availability: how to guarantee the availability of consolidated objects and service continuity during the migration process, especially for a critical applications like real time ones?

Communication: like communication costs, times and inter-dependency issues through communication links between PMs during the migration process, how to secure these links to provide a trust channel is another question that has to be addressed in future works.

Privacy : how to guarantee that the client’s data and files are not lost or compromised?

For further reading on security requirements in the context of cloud computing, readers may refer to [116].

9. PRACTICAL REQUIREMENTS AND LIMITATIONS OF WORKLOAD CONSOLIDATION IN CDCS.

We were able to conclude that the consolidation service, without adding new hardware, allows to expand the capacity of the data center. With less hardware it allows to run more workloads with lowered power demands and lower OPEX. Despite the benefits of consolidation service whose the main ones are discussed in the previous section, however, there are still many limitations to be resolved. In Section 3, we discussed some negative side effects of the

migration process that can be very important, mostly, in terms of resource wastage, computational overhead, performance and costs if we do not foresee a balance between the contradictory objectives. This is notable especially if the consolidation strategy and migration process are continuously performed in short-term intervals so that can cause an over-consolidation that hampers the capabilities of the workload migration and the performance of the application. In addition to the required expertise and technical know how from the service providers to monitor and manage their DC systems, there are many other aspects related to implementation that represents some downsides and need to be investigated. The way the objects (containers, software, VMs or data) are mapped onto machines (VMs or PMs) and the way the consolidation solution is designed present many limitations and requirements. Here, we focus on the implementation related aspects of consolidation solutions and the major requirements and limitations. As we discussed in Section 5, work-loads consolidation solutions may be performed either statically or dynamically. The dynamic consolidation is enabled by live migration. For practical perspective, Tuan Le [117] discussed some live migration solutions in commercial virtualization platforms and cloud platforms. The static allocation of objects to machines is a source of power wastage especially when heuristic methods are used which tend to seek the local solutions instead of global ones. This can cause the unnecessary usage of resources and increase the migration overhead [118]. As earliest open source software implementation to overcome this, we cite Entropy [118] that implements the dynamic consolidation. It is based on CSP to search the globally optimal solutions taking into account the migration overhead. Since the used Choco solver to resolve the CSP problem can consider only one objective, the authors designed their solution in two phases. They are executed in sequential order to treat the two considered objectives of minimizing the number of used machines and minimizing the number of migrations. The major drawback of this solution is due to the sequentially of the two phases that increases the computation time and limits the scalability of the system. The use of a central controller that is responsible in all aspects of virtual machine placement optimization also limited the scalability of Entropy. So, another difficulty related to the design and implementation of consolidation solutions, that can face the service providers, concerns the distribution of consolidation. Designing distributed solutions enables the natural scaling and remedies to such a centralized approach that can be very costly, especially, if it is implemented in large-scale cloud systems and geo-distributed data centers, as discussed in Section 5. A second implementation framework of dynamic VM consolidation and hierarchical distributed management of VMs for private clouds called Snooze is given in [119]. The management hierarchy is composed of three layers: local controller on each physical node that manages virtual machines it hosts, an intermediate layer constituted by group managers (GMs) and group leader that uses data about GMs and centralizes information about the whole cluster [119]. Although it partially solves the scalability problem, the proposed system is not able to optimize the placement of VMs across all the nodes of the cluster [120]. A more completely scalable distributed VM consolidation implementation solution named Open Stack Neat was proposed by Beloglazov et al. [120]. It is an open-source software for dynamic VM consolidation based on the Open Stack platform. Unlike Snooze, this system allows to treat the full set of hosts. Yet, this solution presents some scalability limitations of virtual machine placement decisions. In fact, the distribution of VM consolidation algorithms does not guarantee the total scalability of the system since, in each controller host, only one instance of the global manager is deployed. This also creates a single point of failure. The authors discussed the idea of replicating the global manager to resolve this issue, but there still a problem of synchronization between them. In their work [49], the authors suggested a distributed implementation of the consolidation technique, separating the parts of VM selection and under-load/over-load detection from the VM allocation part. They planned to run the first part (consolidator) in every compute PM and the second one on a separate controller PM. The cited systems concern virtual machines consolidation and are detailed in [121]. However, the discussed dynamic, distribution and scalability aspects concern not only the VM consolidation solutions. Another important challenge that faces service providers to be able to implement their consolidation solutions is related to a functional test-bed to which Khan et al. drew attention in their work [49]. Based on the Open Stack platform and inspired by its modular architecture and the multiple components that offer, the authors suggested to replace the scheduling approach of Nova scheduler which is the responsible for the placement of virtual machines to PMs in Open Stack, by their proposed scheduling approach. Nova offers many metrics that can be used to monitor the cluster and get data about consolidation parameters. For example, Khan et al. proposed to use it to gather data about energy consumption, utilization and performance. They considered that a local manager, being installed in each VM, by analogy with [11], can do the same job as for each PM. They also suggested to install a docker container on each VM for precedent discussed systems. This allows to implement their consolidation solutions in containerized virtual platforms with a minor modification, since hypervisor has the access to VMs and containers so that it is responsible to consolidate the workload. Moreover, Sareh et al. have adopted a container-VM architecture (a CaaS environment), where applications are executed on containers and containers runs in VMs, to design their proposed container consolidation system in [33]. The proposed system was based on two main modules. The first is installed on each active machine and gives information about their load status, and the second is installed on a separate node and is responsible of consolidation algorithms. Like Open Stack

Neat, in this work, the central unit responsible for consolidation limits the scalability of the system and creates one point of failure. For software consolidation, a similar approach of separating the local manager from consolidator was proposed. The main difference with the precedent solution was by adding another component of monitoring in the same machine as a consolidator. Although the reconfiguration plan (migrations) ran through the local manager (installed on each machine), it is calculated by the consolidation manager (central unit) deployed in a separate host. This proposed architecture improves the scalability of the proposed system with regard to the execution of the migration plan, but creates a single point of failure with regard to the compute of the software assignments and the reconfiguration map. In the last solution, each software is packed in a virtual container that runs in a VM hosted in a PM. Nevertheless, the question of real and functional test-bed remains for bare metal containerized platforms when containers run directly in hardware. Intel introduced the notion of generic workloads that is application agnostic in Cloud Integrated Advances Orchestrator (CIAO)¹ project. Through the same system, the proposed orchestrator handles VMs, containers, and bare metal applications in CIAO cluster. However, only a generic view of the architecture and its component's description was given and no official publication that provides a description about the cluster optimization process and how it is performed and evaluated was found. A brief description of CIAO's architecture is given here [49]. To sum up, there is a large volume of works published in the field of workload consolidation some of which presents software implementation solutions and are publicly available. In terms of implementation, the providers face, especially, the dynamic, distribution, and scalability aspects along with the heterogeneity of workloads and platforms. Many works are based on the Open Stack platform. For real needs in the cloud with the diversification of virtualization technologies, the heterogeneity of platforms and the multiple opportunities of consolidation, bring into question the applicability of certain practical systems whose generalization of their results should require significant efforts. Finally, we believe that, with the emergence of cloud native architectures, the micro-services, being independent and interoperable, are not constrained to technological uniformity. Thus, managing the heterogeneity of platforms along with the aforementioned issues related to implementation of consolidation solutions becomes a simpler question to address.

10. CONCLUSION AND FUTUR WORKS.

The cloud paradigm is very promising compared to traditional IT systems. It is based on virtualization, which gives rise to virtualized data centers. One of the major challenges facing cloud data centers is related to resource management that represents a primary concern by industry and academics. In literature studies, it was revealed that less than 15% on average of full capacity are used by hosts [122]. This incurs other problems, the most remarkable are those relating to energy consumption, costs and performance. Virtualization makes it possible to reduce energy budget, computational cost and enhance performance. It has given rise to several techniques enabling an efficient resource management, such as consolidation, which is the most common one. Consolidation process allows us to reach various objectives such as power management, system maintenance, fault tolerance, etc. Consolidation works address the importance of placing objects initially or after migration in an appropriate manner with the aim of improving energy efficiency and resource utilization in cloud infrastructures [45] In this paper, the notions of virtualized data centers and consolidation in cloud computing systems are presented in a holistic manner. A comprehensive survey on IT consolidation in the cloud and thematic taxonomy are proposed to synthesize and categorize the reported literature. Some relevant works from the state-of-the-art are investigated and analyzed according to several aspects related to the characteristics of consolidation, parameters, optimization and evaluation methods in accordance with proposed taxonomy. In addition, an illustration and a discussion of the results of some recent work has been proposed. Finally, we have summarized the article while synthesizing the CRIU technology as well as the benefits of workload consolidation on cloud providers before highlighting some research challenges and trends in IT consolidation. We ended up presenting main barriers of consolidation solutions from an implementation point of view.

Various lines of actions may extend the current IT consolidation work on several axes as mentioned in the last section in terms of interference and interdependence objectives, security, DC level energy consumption, uncertainty of cloud environment and software license consolidation.

But also other aspects need to be addressed. For example, from an architectural point of view, revising the archiving of resource management in the cloud has become a necessity especially with the emergence of decentralized architectures which have proven their advantages in resource management in the cloud as in [123]. Regarding algorithms and techniques, we think that extending the algorithms to consider other resources than the CPU allows to further improve performance in CDC and creates similarities between these algorithms which makes their comparison more realistic [91]. For the evaluation method, developing simulators for the CaaS and SaaS levels is another critical need today. Finally, given the different consolidation scenarios that can take place in cloud data centers in addition to the

flexibility available in these environments, we believe that further exploring hybridization possibilities through different service models can improve DC configurations and optimize resource management.

REFERENCES

- [1] X. Fu, C. Zhou, Virtual machine selection and placement for dynamic consolidation in cloud computing environment, *Front. Comput. Sci.* 9 (2015) 322–330, <http://dx.doi.org/10.1007/s11704-015-4286-8>.
- [2] S. Ismaeel, A. Miri, Using ELM techniques to predict data centre VM requests, in: *Proceedings of the 2nd IEEE International Conference on Cyber Security and Cloud Computing*, IEEE, 2015, pp. 80–86.
- [3] S. Ismaeel, A. Miri, Multivariate time series ELM for cloud data centre workload prediction, in: *Proceedings, Part I, of the 18th International Conference on Human-Computer Interaction. Theory, Design, Development and Practice*, Vol. 9731, 2016, pp. 565–576.
- [4] F. Kong, X. Liu, A survey on green-energy-aware power management for datacenters, *ACM Comput. Surv.* 47 (2014).
- [5] Y.C. Lee, A.Y. Zomaya, Energy efficient utilization of resources in cloud computing systems, *J. Supercomput.* 60 (2012) 268–280.
- [6] M. Uddin, A. Shah, R. Alsaqour, J. Memon, Measuring efficiency of tier level data centers to implement green energy efficient data centers, *Middle East J. Sci. Res.* 15 (2013) 200–207.
- [7] J. Koomey, Growth in Data Center Electricity Use 2005 to 2010, Technical Report, Analytics Press, completed at the request of The New York Times, 2011.
- [8] N. Engbers, E. Taen, Green Data Net, Technical Report, European Commission, 2014.
- [9] A. Shehabi, S. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, E. Masanet, D. Sartor, M. Herrlin, W. Lintner, United States Data Center Energy Usage Report, Technical Report, Lawrence Berkeley National Laboratory, Berkeley, California, 2016.
- [10] T.C. Ferreto, M.A.S. Netto, R.N. Calheiros, C.A.F. De Rose, Server consolidation with migration control for virtualized data centers, *Future Gener. Comput. Syst.* 27 (2011) 1027–1034.
- [11] A. Tchana, N.D. Palma, I. Safieddine, D. Hagimont, Software consolidation as an efficient energy and cost saving solution, *Future Gener. Comput. Syst.* 58 (2016) 1–12.
- [12] T. Shi, H. Ma, G. Chen, Energy-aware container consolidation based on PSO in cloud data centers, in: *2018 IEEE Congress on Evolutionary Computation, CEC, 2018*, pp. 1–8.
- [13] M. Xu, W. Tian, R. Buyya, A survey on load balancing algorithms for virtual machines placement in cloud computing, *Concurr. Comput.: Pract. Exper.* 29 (2016).
- [14] A. Varasteh, M. Goudarzi, Server consolidation techniques in virtualized data centers: A survey, *IEEE Syst. J.* 11 (2017) 772–783.
- [15] R.W. Ahmad, A. Gani, S.H. Ab. Hamid, M. Shiraz, A. Yousafzai, F. Xia, A survey on virtual machine migration and server consolidation frameworks for cloud data centers, *J. Netw. Comput. Appl.* 52 (2015) 11–25.
- [16] R.W. Ahmad, A. Gani, S.H. Ab. Hamid, M. Shiraz, F. Xia, S.A. Madani, Virtual machine migration in cloud data centers: A review, taxonomy, and open research issues, *J. Supercomput.* 71 (2015) 2473–2515.
- [17] A. Choudhary, M.C. Govil, G. Singh, L.K. Awasthi, E.S. Pilli, D. Kapil, A critical survey of live virtual machine migration techniques, *J. Cloud Comput.* 6 23 (2017).
- [18] M.H. Shirvani, A.M. Rahmani, A. Sahafi, A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: Taxonomy and challenges, *J. King Saud Univ.–Comput. Inf. Sci.* (2018).
- [19] S. Ismaeel, R. Karim, A. Miri, Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres, *J. Cloud Comp.* 7 (2018).
- [20] B. Bermejo, C. Juiz, C. Guerrero, Virtualization and consolidation : a systematic review of the past 10 years of research on energy and performance, *J. Supercomput.* 75 (2019) 808–836.
- [21] A. Ashraf, B. Byholm, I. Porres, Distributed virtual machine consolidation: A systematic mapping study, *Comp. Sci. Rev.* 28 (2018) 118–130.
- [22] R.M.B. Abadi, A.M. Rahmani, S.H. Alizadeh, Server consolidation techniques in virtualized data centers of cloud environments: A systematic literature review, *Softw. - Pract. Exp.* 48 (2018) 1688–1726.
- [23] R.M.B. Abadi, A.M. Rahmani, S.H. Alizadeh, Challenges of server consolidation in virtualized data centers and open research issues: a systematic literature review, *J. Supercomput.* 76 (2020) 2876–2927.
- [24] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering (version 2.3), Technical Report, Keele University and University of Durham, 2007.
- [25] W.A. Hanafy, A.E. Mohamed, S.A. Salem, A new infrastructure elasticity control algorithm for containerized cloud, *IEEE Access* 7 (2019) 39731–39741.
- [26] S.C. Mondesire, A. Angelopoulou, S. Sirigampola, B. Goldiez, Combining virtualization and containerization to support interactive games and simulations on the cloud, *Simul. Model. Pract. Theory* 93 (2018) 233–244.
- [27] M. Chae, H. Lee, K. Lee, A performance comparison of linux containers and virtual machines using docker and KVM,

Cluster Comput. 22 (2019) 1765–1775.

- [28] I. Mavridis, H. Karatza, Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing, *Future Gener. Comput. Syst.* 94 (2019) 674–696.
- [29] J. Watada, A. Roy, R. Kadikar, H. Pham, B. Xu, Emerging trends, techniques and open issues of containerization: A review, *IEEE Access* 7 (2019) 152443–152472.
- [30] M.K. Hussein, M.H. Mousa, M.A. Alqarni, A placement architecture for a container as service (CaaS) in a cloud environment, *J. Cloud Comp.* (2019).
- [31] B. Bermejo, C. Juiz, Virtual machine consolidation: a systematic review of its overhead influencing factors, *J. Supercomput.* 76 (2020) 324–361.
- [32] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, P. Merle, Elasticity in cloud computing: State of the art and research challenges, *IEEE Trans. Serv. Comput.* 11 (2017) 430–447.
- [33] S.F. Piraghaj, A.V. Dastjerdi, R.N. Calheiros, R. Buyya, A framework and algorithm for energy efficient container consolidation in cloud data centers, in: 2015 IEEE International Conference on Data Science and Data Intensive Systems, 2015, pp. 368–375.
- [34] S. Soltesz, H. Pözl, M.E. Fiuczynski, A. Bavier, L. Peterson, Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors, *Oper. Syst. Rev.* 41 (2007) 275–287.
- [35] I. Mavridis, H.D. Karatza, Performance and overhead study of containers running on top of virtual machines, in: 2017 IEEE 19th Conference on Business Informatics, Vol. 2, CBI, IEEE, 2017, pp. 32–38.
- [36] D. Merkel, Docker: Lightweight linux containers for consistent development and deployment, *Linux J.* 2014 (2014).
- [37] W. Felter, A. Ferreira, R. Rajamony, J. Rubio, An updated performance comparison of virtual machines and linux containers, in: 2015 IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS, IEEE, 2015, pp. 171–172.
- [38] B. Kavitha, P. Varalakshmi, Performance analysis of virtual machines and docker containers, in: S.M. R S. (Ed.), *Data Science Analytics and Applications. DaSAA 2017. Communications in Computer and Information Science*, Vol. 804, Springer, 2018, pp. 99–113.
- [39] C. Pahl, Containerization and the PaaS cloud, *IEEE Cloud Comput.* 2 (2015) 24–31.
- [40] I. Alobaidan, M. Mackay, P. Tso, Build trust in the cloud computing - isolation in container based virtualisation, in: 2016 9th International Conference on Developments in ESystems Engineering, DeSE, IEEE, 2016, pp. 143–148.
- [41] Q. Ali, Scaling web 2.0 applications using docker containers on vsphere 6.0, 2016, <https://blogs.vmware.com/performance/2015/04/scaling-web-2-0-applications-using-docker-containers-vsphere-6-0.html>.
- [42] C. Pahl, A. Brogi, J. Soldani, P. Jamshidi, Cloud container technologies: A state-of-the-art review, *IEEE Trans. Cloud Comput.* 7 (2019) 677–692.
- [43] N.D. Keni, A. Kak, Adaptive containerization for microservices in distributed cloud systems, in: 2020 IEEE 17th Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2020, pp. 1–6.
- [44] D. Boukhelef, J. Boukhobza, K. Boukhalifa, H. Ouarnoughi, L. Lemarchand, Optimizing the cost of dbaaS object placement in hybrid storage systems, *Future Gener. Comput. Syst.* 93 (2019) 176–187.
- [45] Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, *J. Comput. System Sci.* 79 (2013) 1230–1242.
- [46] R. Shaw, E. Howley, E. Barrett, An advanced reinforcement learning approach for energy-aware virtual machine consolidation in cloud data centers, in: 2017 12th International Conference for Internet Technology and Secured Transactions, ICITST, IEEE, 2017, pp. 61–66.
- [47] E. Dhib, K. Boussetta, N. Zangar, N. Tabbane, Cost-aware virtual machines placement problem under constraints over a distributed cloud infrastructure, in: 2017 Sixth International Conference on Communications and Networking, ComNet, IEEE, 2017, pp. 1–5.
- [48] P. Arroba, J. Moya, J. Ayala, R. Buyya, Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers, *Concurr. Comput.: Pract. Exper.* 29 (2017).
- [49] A.A. Khan, M. Zakarya, R. Khan, I.U. Rahman, M. Khan, A.u.R. Khan, An energy, performance efficient resource consolidation scheme for heterogeneous cloud datacenters, *J. Netw. Comput. Appl.* 150 (2020).
- [50] A. Khan, M. Zakarya, R. Buyya, R. Khan, M. Khan, O. Rana, An energy and performance aware consolidation technique for containerized datacenters, *IEEE Trans. Cloud Comput.* (2019).
- [51] J. Liu, S. Wang, A. Zhou, J. Xu, F. Yang, SLA-driven container consolidation with usage prediction for green cloud computing, *Front. Comput. Sci* 14 (2020) 42–52.
- [52] T. Shi, H. Ma, G. Chen, Multi-objective container consolidation in cloud data centers, in: T. Mitrovic, B. Xue, X. Li (Eds.), *AI 2018: Advances in Artificial Intelligence, AI 2018*, in: *Lecture Notes in Computer Science*, vol. 11320, Springer, Cham, 2018, pp. 783–795.
- [53] A. Sen, A. Garg, A. Verma, T. Nayak, Cloudbridge: On integrated hardware-software consolidation, *ACM SIGMETRICS*

Perform. Eval. Rev. 39 (2011).

- [54] U.K. Jena, P.K. Das, M.R. Kabat, Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment, *J. King Saud Univ. – Comput. Inf. Sci.* (2020).
- [55] W. Zhang, X. Chen, J. Jiang, A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems, *Tsinghua Sci. Technol.* 26 (2021) 95–111.
- [56] L. Lv, Y. Zhang, Y. Li, K. Xu, D. Wang, W. Wang, M. Li, X. Cao, Q. Liang, Communication-aware container placement and reassignment in large-scale internet data centers, *IEEE J. Sel. Areas Commun.* 37 (2019) 540–555.
- [57] N.R. Herbst, S. Kounev, R. Reussner, Elasticity in cloud computing: What it is, and what it is not, in: *Proceedings of the 10th International Conference on Autonomic Computing, ICAC 13, 2013*, pp. 23–27.
- [58] Y. Sharma, W. Si, D. Sun, B. Javadi, Failure-aware energy-efficient VM consolidation in cloud computing systems, *Future Gener. Comput. Syst.* 94 (2019) 620–633.
- [59] C.C. Chou, Y. Chen, D. Milojicic, N. Reddy, P. Gratz, Optimizing post-copy live migration with system-level checkpoint using fabric-attached memory, in: *2019 IEEE/ACM Workshop on Memory Centric High Performance Computing, MCHPC, IEEE, 2019*.
- [60] J. Adam, M. Kermarquer, J.B. Besnard, L. Bautista-Gomez, M. Pérache, P. Carribault, J. Jaeger, A. D.Malony, S. Shended, Checkpoint/restart approaches for a thread-based MPI runtime, *Parallel Comput.* 85 (2019) 204–219.
- [61] R.K. Mehta, J. Chandy, Leveraging checkpoint/restore to optimize utilization of cloud compute resources, in: *2015 IEEE 40th Local Computer Networks Conference Workshops, LCN Workshops, IEEE, 2015*.
- [62] R. Stoyanov, M. Kollingbaum, Efficient live migration of linux containers, in: *International Conference on High Performance Computing ISC High Performance 2018, Vol. 11203, Springer, Cham, 2018*, pp. 184–193.
- [63] Y. Qiu, C. Lung, S. Ajila, P. Srivastava, LXC container migration in cloudlets under multipath TCP, in: *2017 IEEE 41st Annual Computer Software and Applications Conference, COMPSAC, IEEE, 2017*, pp. 31–36.
- [64] H. Xu, Y. Liu, W. Wei, Y. Xue, Migration cost and energy-aware virtual machine consolidation under cloud environments considering remaining runtime, *Int. J. Parallel Programm.* 47 (2019) 481–501.
- [65] A. Verma, P. Ahuja, A. Neogi, pMapper: Power and migration cost aware application placement in virtualized systems, in: *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, Springer, Berlin, Heidelberg, 2008*, pp. 243–264.
- [66] W. Dargie, Estimation of the cost of VM migration, in: *2014 23rd International Conference on Computer Communication and Networks, ICCCN, IEEE, 2014*, pp. 1–8.
- [67] Q. Wu, F. Ishikawa, Q. Zhu, Y. Xia, Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters, *IEEE Trans. Serv. Comput.* 12 (2019) 550–563.
- [68] P.K. Sinha, S.S. Doddamani, H. Lu, K. Gopalan, Mwar: Accelerating intra-host live container migration via memory warping, in: *IEEE INFO-COM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2019*, pp. 508–513.
- [69] N.M. TYJ, V. G, Adaptive deduplication of virtual machine images using AKKA stream to accelerate live migration process in cloud environment, *J. Cloud Comput.* 8 (2019).
- [70] S. Nadgowda, S. Suneja, N. Bila, C. Isci, Voyager: Complete container state migration, in: *2017 IEEE 37th International Conference on Distributed Computing Systems, ICDCS, IEEE, 2017*, pp. 2137–2142.
- [71] Z. Gong, X. Gu, J. Wilkes, PRESS: Predictive elastic resource scaling for cloud systems, in: *2010 International Conference on Network and Service Management, IEEE, 2010*, pp. 9–16.
- [72] F. Farahnakian, P. Liljeberg, J. Plosila, LiRCUP: linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers, in: *Proceedings of the 39th EUROMICRO Conference on Software Engineering and Advanced Applications, 2013*, pp. 357–364.
- [73] J.E. Hajlaoui, M.N. Omri, D. Benslimane, A QoS-aware approach for discovering and selecting configurable IaaS cloud services, *Comput. Syst.: Sci. Eng.* 32 (2017).
- [74] Z. Li, X. Yu, L. Zhao, A strategy game system for QoS-efficient dynamic virtual machine consolidation in data centers, *IEEE Access* 7 (2019) 104315–104329.
- [75] S.K. Addya, A.K. Turuk, B. Sahoo, A. Satpathy, M. Sarkar, A game theoretic approach to estimate fair cost of VM placement in cloud data center, *IEEE Syst. J.* 12 (2018) 3509–3518.
- [76] L. Guo, G. Hu, Y. Dong, Y. Luo, Y. Zhu, A game based consolidation method of virtual machines in cloud data centers with energy and load constraints, *IEEE Access* 6 (2018) 4664–4676.
- [77] Z.A. Mann, Resource optimization across the cloud stack, *IEEE Trans. Parallel Distrib. Syst.* 29 (2018) 169–182.
- [78] R. Zhang, Y. Chen, B. Dong, F. Tian, Q. Zheng, A genetic algorithm-based energy-efficient container placement strategy

in caas, *IEEE Access* 7 (2019) 121360–121373.

- [79] L.C. Jersak, T. Ferreto, Performance-aware server consolidation with adjustable interference levels, in: *SAC '16: Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016, pp. 420–425.
- [80] H. Liu, H. Jin, C.Z. Xu, X. Liao, Performance and energy modeling for live migration of virtual machines, *Cluster Comput.* 16 (2013) 249–264.
- [81] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, R. Buyya, Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Softw. - Pract. Exp.* 41 (2011) 23–50.
- [82] A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, *Concurr. Comput. Pract. Exp.* 24 (2012) 1397–1420.
- [83] B. Wickremasinghe, R.N. Calheiros, R. Buyya, Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications, in: *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, IEEE, 2010, pp. 446–452.
- [84] S.F. Piraghaj, A. Dastjerdi, R.N. Calheiros, R. Buyya, Container-cloudsim: an environment for modeling and simulation of containers in cloud data centers, *Softw.-Pract. Exp.* 47 (2017) 505–521.
- [85] N. Mansouri, R. Ghafari, B.M.H. Zade, Cloud computing simulators: A comprehensive review, *Simul. Model. Pract. Theory* (2020).
- [86] A. Ismail, Energy-driven cloud simulation: existing surveys, simulation supports, impacts and challenges, *Cluster Comput.* (2020).
- [87] L. Peterson, A. Bavier, M.E. Fiuczynski, S. Muir, Experiences building PlanetLab, in: *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, 2016, pp. 351–366.
- [88] K. Park, V.S. Pai, Comon: a mostly-scalable monitoring system for planetlab, *ACM SIGOPS Oper. Syst. Rev.* 40 (2006) 65–74.
- [89] H.J. Reiss C, Google Clusterusage Traces: Format+ Schema, Technical Report, Google Inc, Mountain View, CA, USA, 2011.
- [90] SPECjms2007: industry-standard benchmark for evaluating the performance of enterprise message-oriented middleware servers based on JMS, 2007, <http://www.spec.org/jms2007/>.
- [91] S. Di, D. Kondo, W. Cime, Host load prediction in a google compute cloud with a bayesian model, in: *SC '12: Proceedings of the International Conference on High Performance Computing*, 2012, pp. 1–11.