IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 23, 2018, accepted October 29, 2018, date of publication November 23, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2883149

Future Cloud Systems Design: Challenges and Research Directions

AMIR TAHERKORDI[1], FEROZ ZAHID[2], YIANNIS VERGINADIS[3], and GEIR HORN[1]
[1]Department of Informatics, University of Oslo, 0373 Oslo, Norway
[2]Simula Research Laboratory, 1364 Fornebu, Norway
[3]Institute of Communications and Computer Systems, National Technical University of Athens, 15773 Zografou, Greece

Corresponding author: Amir Taherkordi (amirhost@ifi.uio.no)

This work was supported in part by the European Commission's H2020 Research Programme under Grant 731664 (EU MELODIC Project) and Grant 645197 (EU HOLA Cloud Project) and in part by the Norwegian Research Council under Grant 262854/F20 (DILUTE Project).

ABSTRACT Cloud computing has been recognized as the de facto utility computing standard for hosting and delivering services over the Internet. Cloud platforms are being rapidly adopted by business owners and end-users thanks to its many benefits to traditional computing models such as cost saving, scalability, unlimited storage, anytime anywhere access, better security, and high fault-tolerance capability. However, despite the fact that clouds offer huge opportunities and services to the industry, the landscape of cloud computing research is evolving for several reasons, such as emerging data-intensive applications, multicloud deployment models, and more strict non-functional requirements on cloud-based services. In this paper, we develop a comprehensive taxonomy of main cloud computing research areas, discuss state-of-the-art approaches for each area and the associated sub-areas, and highlight the challenges and future directions per research area. The survey framework, presented in this paper, provides useful insights and outlook for the cloud computing research and development, allows broader understanding of the design challenges of cloud computing, and sheds light on the future of this fast-growing utility computing paradigm.

INDEX TERMS Cloud computing, future directions, research challenges.

I. INTRODUCTION

Computing resources have been transformed more and more to a model inspired by traditional utilities such as water, electricity and telephony. In such commodity models, the end-user is offered services based on his or her requirements without having to be aware of where the services are located and how they are delivered. This on-demand delivery of computing as a utility has been realized by technologies such as cluster computing, grid computing and more notably *cloud computing*. Considering the latter, it is defined as an umbrella term to cover a category of on-demand computing services initially offered by reputable IT vendors, such as Amazon, Google, and Microsoft. The main principle behind the cloud computing model is offering computing, storage, and software ''as a service''.

Among several definitions of cloud computing, one of most comprehensive definitions is proposed by Buyya *et al.* [1]. They have defined the cloud as follows: ''Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on a Service-Level Agreement (SLA) established through negotiation between the service provider and consumers''. From this definition and other similar definitions, a set of common characteristics of a cloud platform can be extracted [2], including *i)* pay-per-use; *ii)* elastic capacity and the illusion of infinite resources; *iii)* self-service interface; and *iv)* abstracted or virtualized resources.

Since the inception of the concept of cloud computing, a large and growing body of research has been carried out to address diverse challenges in the design, development and management of cloud computing platforms. As a very broad and rapidly evolving subject, the cloud research encompasses a wide spectrum of basic challenges including the cloud network architecture, network virtulization, cloud resource management, load balancing, cloud application engineering and management, the security and privacy of cloud platforms, and interoperability and openness.

Besides the above primary challenges, the landscape of cloud research is changing and expanding for several reasons, such as the emergence of novel application areas such as

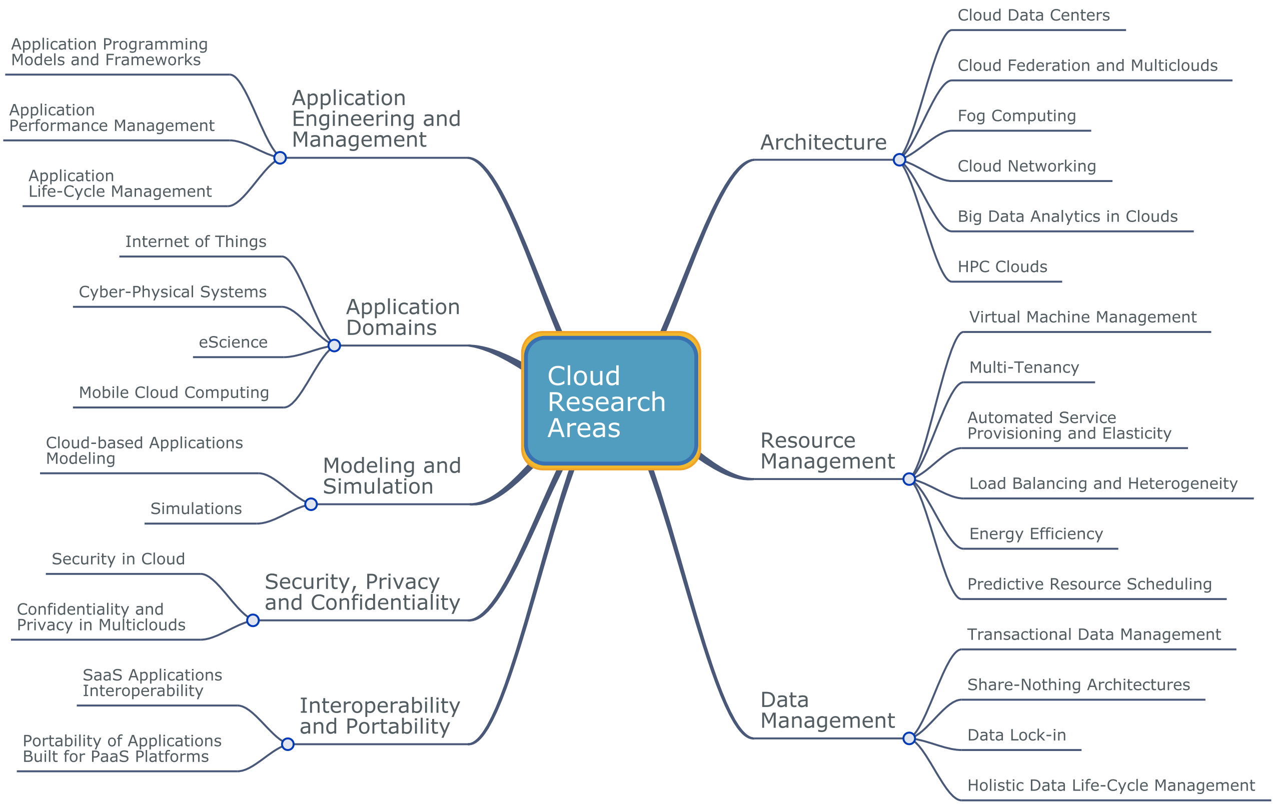
Cloud Research Areas
Application Engineering and Management
Application Programming Models and Frameworks
Application Performance Management
Application Life-Cycle Management
Application Domains
Internet of Things
Cyber-Physical Systems
eScience
Mobile Cloud Computing
Modeling and Simulation
Cloud-based Applications Modeling
Simulations
Security, Privacy and Confidentiality
Security in Cloud
Confidentiality and Privacy in Multiclouds
Interoperability and Portability
SaaS Applications Interoperability
Portability of Applications Built for PaaS Platforms
Architecture
Cloud Data Centers
Cloud Federation and Multiclouds
Fog Computing
Cloud Networking
Big Data Analytics in Clouds
HPC Clouds
Resource Management
Virtual Machine Management
Multi-Tenancy
Automated Service Provisioning and Elasticity
Load Balancing and Heterogeneity
Energy Efficiency
Predictive Resource Scheduling
Data Management
Transactional Data Management
Share-Nothing Architectures
Data Lock-in
Holistic Data Life-Cycle Management

FIGURE 1. The taxonomy of cloud computing research areas.

Internet of Things (IoT), the shift from single provider data centers to multiple ones, emerging data-intensive applications, and more strict quality of service requirements on cloud-based services, such as low latency. These have triggered new areas of cloud research, such as cloud federation and multicloud, data management in the cloud, modeling and simulation of cloud systems, and new low-latency and context-aware cloud architectures, such as Fog. These recent and growing cloud research directions imply the need for a thorough study of ongoing cloud-related research activities and a broad outlook to the future of this computing discipline. The main contribution of this paper is to provide a comprehensive view on the future research directions in cloud computing. To this end, we provide a complete taxonomy of main cloud computing research areas, discuss briefly state-of-the-art approaches for each area and the associated sub-areas, and present the challenges and directions per research area.

Related Surveys: Existing survey works on cloud computing have either provided a brief overview of the research efforts made so far in developing cloud systems or focused only on a specific issue of cloud design, such as security [3]–[5]. The only recent survey on the future of cloud computing is presented by Varghese and Buyya [6], followed by an extended version in [7]. Below, we summarize how this paper differs from the above papers:

- This paper studies more carefully the following topics in cloud computing: *architecture*, *big data*, *application domains and engineering*, *modeling and simulation*, *security and privacy*, and *interoperability and portability*.
- With respect to *resource management* and *networking*, this paper covers more detailed challenges, and more recent work and directions, while *data management* is discussed differently in this paper as compared to the above surveys.
- In the above surveys, the non-functional aspects of cloud research are explored more in detail than in this paper, such as reliability, sustainability, scalability, and usability.
- From a complementary viewpoint, the other surveys have touched some recent topics such as software-defined networks, blockchain, and machine learning.

Therefore, we argue that our paper and the aforementioned surveys are complementary, providing insights from different perspectives for the future. The taxonomy in Figure 1 shows the topics and the associated sub-topics studied in this paper.

The methodology we adopted for the framework of this study consists of the following steps. *First*, we extracted the list of main cloud research areas from the relevant call for papers of reputable journals, conferences and workshops, *e.g.* IEEE International Conference on Cloud Engineering (IC2E), IEEE Cloud, IEEE Access, IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE/ACM International Conference on Utility and Cloud (UCC), IEEE Cloud Computing magazine, and IEEE Transactions on Cloud Computing, to name the most important ones. *Second*, under each main area, we searched carefully for the reported research contributions in the literature and relevant research projects. We selected those contributions with high citation counts or published in top ranking venues or journals. This guided us to the second level of challenges after studying research contributions under each main topic, called cloud research sub-areas, *e.g.* Energy Efficiency as a sub-area of Resource Management. *Finally*, we read carefully the compiled list of relevant research works, grouped them, and linked them based on the chain of citations, resembling a mind map for the entire cloud research (as shown in Figure 1). This, indeed, serves as the framework for discussing state-of-the-art per sub-area and presenting the future directions based on our analysis on efforts made so far as well as the reported potential future work per sub-area.

The paper is structured based on the aforementioned eight main topics, discussed from Section II to Section IX, respectively. In Section X, we present the summary of cloud research challenges and future directions, and make the concluding remarks.

II. CLOUD ARCHITECTURE

In this section, we discuss state-of-the-art and outstanding research challenges at the architectural level in a modern cloud computing system. We classify architectural challenges into several distinct categories. First are the challenges associated with the data center architectures, which fundamentally stem from the unique demands of cloud computing systems unmet by traditional data center architectures. Second, we outline research challenges associated with the use of hybrid and heterogeneous cloud platforms, such as in federated clouds and multicloud setups. Then, we study the *fog computing model*—the new architectural concept in cloud computing. The fourth area we explore relates to the challenges in cloud networking, including the data center level networking issues and those imposed by the federated cloud architectures. Finally, we discuss challenges related to the specific cloud services that can benefit from a modular and holistic approach at the cloud architectural level. In this connection, Big Data analytics and High Performance Computing (HPC) applications in clouds are briefly outlined.

A. CLOUD DATA CENTERS

Many architecture level challenges for clouds can be traced back to the basic building blocks of data centers which are used to realize cloud services. Studies show that about 40% of the costs of a data center go directly to the data center infrastructure like power distribution, cooling, and network equipment [8]. Data Center Networks (DCNs), in particular, are of critical importance in improving the overall performance and utilization of the costly data center resources. Besides contemporary data center requirements, such as performance efficiency, flexibility, and easy management, cloud DCNs also impose unique challenges stemming from the very nature of the shared model of cloud computing. We identify some challenges for the future work in the following.

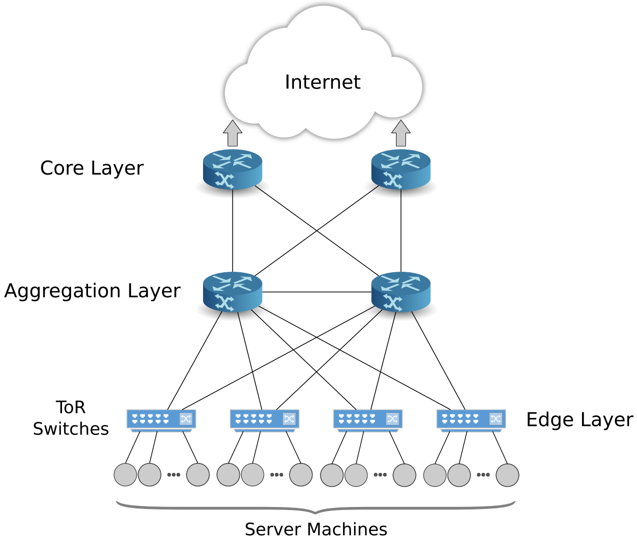

FIGURE 2. Traditional three-layer data center architecture.

1) HIGHER INTER-SERVER BANDWIDTH IN DATA CENTERS

Unlike traditional enterprise data center workloads, a significant part of the total communication in a cloud data center occurs within the server machines in the data center, requiring higher bandwidth between server machines [8]–[10]. Traditional data center architectures are typically comprised of a layered approach consisting of an *edge* layer with Top of Rack (ToR) switches connected at an *aggregation* layer, which in turn are unified at the *core layer* switches (cf. Figure 2). A problem with such an approach is that the links in the network core are often oversubscribed leading to lower bandwidth between server machines connected to different aggregation layers [11]. Various DCN architectures have been proposed in the literature for both traditional enterprise data centers as well as for the needs of modern data centers like clouds. Following the approach taken by Liu *et al.* [12], DCNs can be broadly classified into two categories: architectures with fixed topology, and architectures with flexible topology. Fixed-topology DCNs contain both popular tree-like topologies, such as fat-trees [11], Portland [13], Monsoon [14], and VL2 [15], and recursive topologies such as DCell [16], BCube [17] and FiConn [18]. The DCN architectures based on recursive topologies can also be considered as *server-centric* compared to *switch-centric* fixed topologies [9]. The fixed topology architecture usually employs a single interconnection network throughout the data center, predominately Ethernet. To cater with high bandwidth demands at the network core, flexible topology network architectures, such as Hedera [19] and Helios [20] have been presented.

2) AUTOMATION AND FAST NETWORK RECONFIGURATION

Another important distinction between enterprise data centers and cloud data centers is the automation required for scaling, which is fundamental in cloud computing. Tenant workload in clouds consists of VMs provisioned over data center resources. The VMs often need to migrate between server machines for improving server consolidation, decreasing *fragmentation* [8], and tolerating faults. Current network architectures employ static network reconfigurations, lacking the fast reconfiguration methodologies required for dynamic cloud networks. New DCN architectures are emerging that specifically target cloud data centers, such as DCNet [10]. However, practical evolution of such novel architectures in large-scale clouds is still very limited. More detailed surveys of data center architectures, and DCNs in cloud computing, are provided respectively by Chen *et al.* [9] and Wang *et al.* [21].

B. CLOUD FEDERATION AND MULTICLOUDS

Modern enterprises increasingly rely on hybrid cloud solutions to meet their computational demands by acquiring additional resources from public clouds. Cloud federation [22] enables end users to integrate segregated resources from different cloud systems. The federated clouds offer more freedom to the cloud users, and increase the granularity of choices in application deployment.

1) RESOURCE MANAGEMENT AND CONTEXT-AWARENESS

Popular open-source cloud orchestration solutions, like *OpenStack* [23] and *OpenNebula* [24], provide mechanisms to complement private cloud infrastructures with dynamically acquired resources from public clouds. Nevertheless, resource management is not well integrated with state-of-the-art federated cloud solutions. Further, none of the cloud platforms supports cloud context-awareness, which is needed to optimize application deployment in multicloud environments. Furthermore, multicloud application deployments are subjected to various resource abstraction models offered through different cloud providers, thereby a unified approach is needed for interoperability.

2) LACK OF DATA-AWARENESS

Recent efforts [25]–[27] have targeted model-based approaches for the design, development, deployment, and self-adaptation of multicloud applications. In particular, several cloud modeling frameworks are in active development to equip application developers with capabilities to define a rich set of design-time and run-time attributes like application requirements, Quality of Service (QoS) constraints, and security considerations for multicloud deployments. However, a large number of challenges are still not addressed. In particular, support of data-aware deployments in multicloud environments is still very restricted. Techniques like latency-aware job placement, data-aware scheduling, and data prefetching are well-known in the literature for single-network environments, and should be incorporated by the application deployment model in multiclouds.

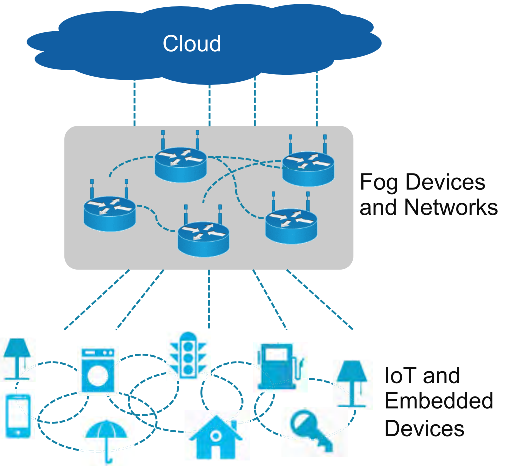

FIGURE 3. **A general architectural model for Fog computing.**

C. FOG COMPUTING

Fog computing is a new computing paradigm that acts as an intermediate architectural layer residing between cloud platforms and devices, in particular IoT smart devices and sensors, as shown in Figure 3. Fog computing is mainly proposed for IoT applications that are geospatially distributed, large-scale, and latency sensitive, while latency-tolerant and large-scale data processing tasks can still be efficiently executed in the cloud. An early fog-based architectural model was introduced by Tang *et al.* [28], proposing a hierarchical fog computing architecture for big data analysis in smart cities. Its goal is to support quick response at neighborhood-wide, community-wide and city-wide levels. Another recent approach is to employ a Software Defined Network (SDN) [29] and Network Function Virtualization (NFV) [30] in order to increase network scalability and reduce costs, in many aspects of fog computing, such as resource allocation, Virtual Machine (VM) migration, traffic monitoring, application-aware control, and programmable interfaces. To maintain a collaborative execution environment, fog nodes can be formed as clusters, either based on the homogeneity of the fog nodes [31] or their location [32]. Cluster based collaboration is effective in exploiting capabilities of several fog nodes simultaneously, but time and dynamic formation of clusters largely depend on the existing load and the availability of fog nodes. As a similar approach, Peer-to-Peer (P2P) collaboration among the fog nodes is very common. P2P collaboration can be realized in hierarchical [33]) and flat order [34]. However, reliability and access control related issues are challenging in P2P collaboration models. The following challenges are worth special attention in the future of cloud fog integration.

1) DYNAMIC FOG-BASED ARCHITECTURES

In order to support on-demand fog service provisioning and orchestration, architectural service and task distribution models are required for orchestrating the segmented functionality of applications deployed over a Fog hierarchy. With massive and diverse IoT services in next-generation IoT cloud computing platforms, providing such on-demand and personalized services becomes challenging because of the lack of scalable and adaptable design solutions for creation, composition and management of fog services. Such architectural models should efficiently and dynamically support migration of computing tasks [35] and services across things, fogs, and cloud platforms, in particular in the case of mobile IoT and fog devices. Considering mobility, the state-of-the-art approaches do not support context-aware and QoS-aware placement of mobile processes [33], [36]. This implies that the dynamics of fog architecture models should be enhanced with supporting and processing contextual and QoS parameters, such as processing power or location of a fog node.

2) PROCESSING DYNAMIC DATA-CENTRIC TASKS

Utilizing fog devices in big data processing applications will introduce new challenges in terms of dynamic task processing and data flow architectures when fog nodes are part of the data processing platform. Dataflow programming models are suitable for fog applications [35], [37], [38]. A dataflow program is basically represented as a directed graph, where tasks are depicted by vertices (nodes) and data dependencies are denoted by edges between nodes. The ability to orchestrate dataflow programs running on things, fog, and cloud platforms is an interesting future research direction. This also applies to the data communication flow between devices that spans multiple networks and domains. In addition, mechanisms are needed to enable the automation in terms of which elements of the initial data flow should be transformed and when, *e.g.* in the presence of bottlenecks, the processing flow should move some computations from the cloud to the fog and vise versa.

3) DATA CONSISTENCY

It can be achieved by coordinating the cloud servers in the data centers controlled by the cloud. However, in fog computing, when data objects on a fog device is updated, it is necessary not only to coordinate the cloud servers, but also invalidate the cached data on the fog node and client devices if strong data consistency is required [39]. This may reduce write performance, not justifying the use of fog nodes as the write cache servers. On the other hand, by transferring the the data object's ownership from the cloud to fog nodes, better write performance can be promised than with cloud computing, as the fog nodes resides at the edge of the network. Therefore, fog computing has the potential to achieve data consistency more efficiently than cloud computing. This indicates that achieving data consistency on fog nodes introduces a number of challenges, including data caching mechanisms at the edge level, consistency preservation for mobile fog nodes, and coordination models for fog-cloud data consistency support.

4) AUTOMATIC SERVICE DISCOVERY

Users and IoT devices should be able to discover fog services based on their current location. Indie Fog [40] relies on a global federated registry for fog services discovery. F2C-Aware [41] proposes a discovery approach that allows devices in WiFi-powered fog-to-cloud systems to become aware of each other and the associated services. Implementing service discovery protocols in fog computing can be quite challenging due to the unknown or dynamic architecture of fog networks. Beyond that, service provisioning in fogs is usually realized dynamically, *i.e.* new virtual machines are orchestrated on the spot when a particular service is needed.

5) CONTEXT-AWARE EDGE RESOURCES DISCOVERY

This is an even greater challenge for augmenting fog computing capabilities. This challenge mainly refers to processing closer to the extreme edge of the network. This implies the need for acquiring and understanding the relevant contextual information derived from any edge device registered in a system. Therefore, the main challenge is to develop advanced context analysis capabilities in order to be able to define the most appropriate devices at the extreme edge of the network that are able to undertake parts of the processing effort, *e.g.* adequate battery level. The limited body of work, in this area, has mainly focused only on user demands as a contextual parameter. CARDAP [42] is a component-based platform that can be used in fog networks for developing complex distributed mobile analytics applications using situation context information captured from the user and his or her environment.

D. CLOUD NETWORKING

The effectiveness of a cloud data center directly depends on *i)* the ability of its network architecture to intelligently provision resources; *ii)* quickly adapting with the irregular demand patterns; and *iii)* its ability to work predictability in a multi-tenant shared environment. In the following, we briefly discuss the state-of-the-art and open challenges for different areas related to cloud networking.

1) COST-EFFECTIVE RELIABLE COMMUNICATION

Reliable communication is an important challenge in cloud data centers. In general, clouds employing traditional network architectures may not prove efficient for the applications which require certain performance guarantees, for example, big data and HPC applications [43]. To provide efficient support for such applications loss-less interconnection networks can be used [44], [45]. Loss-less Ethernet covers various networking technologies, based on classical Ethernet, implementing link-level flow control to avoid packet loss inside the network fabric. Data Centre Bridging (DCB) is one set of standard enhancements to Ethernet aiming to provide a loss-less transport layer. DCB enables a converged unified fabric in data center environments where the Local Area Network (LAN), the System Area Network (SAN), the cluster inter-process communication, and the management traffic all share the same underlying networking infrastructure. With the help of DCB enhancements, high-performance interconnection networking using Ethernet has become a feasible option [46]. However, an important challenge is the cost of such networks in cloud data centers.

2) NETWORK VIRTUALIZATION CHALLENGES

Network virtualization technologies facilitate creation of multiple Virtual Networks (VNs) on top of a shared physical network infrastructure [47]. Most of the work in the field of network virtualization is motivated by the needs of traditional Internet Service Provider (ISP) networks [48]. It has its shortcomings when applied to cloud data center networks—the most important challenges related to scalability, performance isolation, and heterogeneity [49]. As the cloud data centers often rely on economies of scale, packet forwarding schemes require minimum possible forwarding states in the switching elements for scalability. In addition, it is desirable that each tenant receives predictable network performance unaffected by the workload of the other tenants. Both hypervisor level rate-limits and QoS features can be used to provide such bandwidth guarantees. Oktopus [50] provides VN abstractions in the form of virtual clusters, and uses rate-limiting at hypervisor level to enforce per VM bandwidth guarantees. Multiprotocol Label Switching (MPLS) or Resource Reservation Protocol (RSVP) based schemes can also be used to reserve bandwidth along the communication path, but it requires switching components to have traffic engineering capabilities. The problem with the static bandwidth limits is that the network is poorly utilized. To cater this challenge, some solutions like Gatekeeper [51] and SecondNet [52] implement predefined service levels with both soft and hard bandwidth guarantees. Gatekeeper, for instance, implements two bandwidth parameters, minimum guaranteed bandwidth and maximum allowed rate for each VM pair. Such solutions typically suffer from the issues related to the fairness in low load conditions. Seawall [53] assigns proportional weights for bandwidth allocation to the VMs or processes on the links to achieve improved network utilization. It uses congestion-controlled tunnels between VMs to enforce bandwidth sharing policies. A Network Driver Interface Specification (NDIS) packet filter intercepts and limits the rate of transmitted packets. While Seawall focuses on the performance guarantees, a notable drawback is that it does not provide full-address space virtualization.

3) LACK OF INTEGRATION BETWEEN CLOUD AND THE SDN CONTROLLER

Many cloud networks are prone to configuration issues due to the complexity of the architecture. This is largely due to the aggregation of control and forwarding logic in the switching elements. The SDNs enable decoupling of control and data plane to provide a simpler, flexible, and programmable network infrastructure. The control policies are managed centrally using the SDN controller. The Forwarding Devices (FDs) such as switches and routers interact with the SDN controller using a well-defined interface like OpenFlow [54] or ForCES [55]. The research focus in the literature has been mostly on the robust controller architecture [56], [57] and efficient switch design to enable low-overhead programming of the network components. However, from the perspective of a data center, as the management and virtualization layers do not interact with the SDN controller, the benefits of SDNs are limited. For instance, even though techniques and mechanisms of load balancing have been developed in the literature for SDNs [58], the impact on the dynamic cloud networks is limited. This is because sub-optimal bandwidth utilization is often obtained due to not considering network resource requirements for the dynamically changing workload in the cloud. Similarly, the data center is not able to efficiently support applications that require strong QoS guarantees, *e.g.* multimedia services.

4) FEDERATED CLOUD NETWORKING CHALLENGES

The growing popularity of hybrid and multicloud setups requires robust and scalable solutions to provide connectivity between applications across data centers. In particular, the cross-cloud deployments, where application components are simultaneously deployed at different distributed and federated data centers, need mechanisms for secure and efficient communication among application components. In connection with the cloud federation, several cloud network management tools are currently available based on the concept of software-defined data center networks, such as OpenNaaS [59] and CloudNaaS [60]. OpenNaaS, in particular, offers dynamic provisioning and automatic configuration of network resources and defines provider-independent interfaces for access to resources. However, there are still many open challenges in this area for potential future work. Both existing network management solutions and the current cloud platforms have still limited support of uniform network service presentation. In addition, standards must be developed to enable uniform composition of networked resources across domains. Further, dynamic provisioning of overlay networks over heterogeneous interconnection networks is still not addressed.

E. BIG DATA ANALYTICS IN CLOUDS

Big Data is one of the major current trends in computing. In the areas of social media, business intelligence, information security, Internet of Things (IoT), and scientific research, a tremendous amount of data exists or is generated. The data can be both structured and unstructured, and created and collected at a speed surpassing what we can handle using traditional techniques. Users create content, behavior is recorded, sensor data is collected, and experiments run, to mention just a few potential producers and sources of big data. Within the large amount of data produced, the great potential lies

TABLE 1. Cloud architecture research challenges and future directions.

Sub-areas	Challenges and Future Directions	References
Cloud Data Centers	• Higher inter-server bandwidth in data centers • Automation and fast network reconfiguration	[8]–[21]
Cloud Federation and Multiclouds	• Resource management and context-awareness • Lack of data-awareness	[22]–[27]
Fog Computing	• Dynamic fog-based architectures • Processing dynamic data-centric tasks • Data consistency • Automatic service discovery • Context-aware edge resources discovery	[33], [35]–[42]
Cloud Networking	• Cost-effective reliable communication • Network virtualization challenges • Lack of integration between cloud and SDN controller • Federated cloud networking challenges	[43]–[60]
Big Data Analytics in Clouds	• Cross-cloud data processing	[61]–[64], [67], [68]
HPC Clouds	• Performance unpredictability • Lack of flexibility in loss-less interconnection networks	[44], [45], [69]–[77]

in the form of undiscovered values, structures, and relations. To facilitate realizing this potential, which turns out to be the new competitive advantage to the businesses [61], many commercial cloud providers offer specialized big data services in the form of Platform as a Service (PaaS) solutions. These solutions, such as Amazon Elastic Map Reduce (EMR) [62], IBM BigInsights [63], and Microsoft Azure HDInsight [64] are typically implemented by allowing easy use of large-scale data processing frameworks, namely Apache Hadoop [65] and Apache Spark [66]. One main future direction in this area is cross-cloud computing.

Cross-Cloud Data Processing: Many of the above solutions impose *lock-in* as there is no standard and unified way of defining and accessing platform-level services for the end users. Moreover, data migrations between different cloud platforms is inherently costly [67]. In addition, and more importantly, trust has remained a major issue hindering the broader adoption of cloud services by enterprises for data analysis [68]. It is a common perception that, due to lack of control and transparency, data stored in the cloud is prone to theft, misuse, and unauthorized access. The use of multiple cloud providers, *e.g.* as in federated cloud setups for deploying application components and associated data repositories, further escalates the trust issue. In such compositions, it is critically important that access control mechanisms are context-aware and established over the dissociate administrative domains that control storage and inter-cloud data communication during the application life-cycle.

F. HPC Clouds

Traditionally, HPC resources were almost exclusively deployed and committed by large research institutes, universities, national laboratories, and governmental bodies. The engineers and scientists, as HPC users, normally had to wait long before getting access to the highly sought-after HPC resources for their applications. With the emergence of big data workloads as the new HPC *killer application*[1] it arises the need for extending HPC resources to a much wider audience in a flexible and cost-effective way.

Arguably, through HPC clouds, a large number of enterprises, as well as research institutes and academic organizations, could benefit from feature-rich cloud offerings. This potentially saves them substantial capital expenditure while providing *instant* and *elastic* resource capacity for their applications. However, in practice, effective use of cloud computing for HPC systems still remains questionable due to the following challenges.

1) PERFORMANCE UNPREDICTABILITY

Applications running on shared cloud networks are vulnerable to performance unpredictability and violations of service level agreements [69]–[73]. On the contrary, HPC applications typically require predictable network performance from

[1] Coined by PC Week in 1987, the term ‘killer application’ is used to refer to a software application so important for customers that it drives popularity of some larger technology, such as the computer hardware or platform.

TABLE 2. **Cloud architecture research challenges and future directions.**

Sub-areas	Challenges and Future Directions	References
Virtual Machine Management	• Container-based virtualization	[78]–[81]
Multi-tenancy	• Trade-off between system utilization and performance isolation	[50], [52], [53], [90]–[93]
Automated Service Provisioning and Elasticity	• Automated service provisioning for QoS guarantees	[94]–[97]
Load Balancing and Heterogeneity	• Multi-factors for load balancing • Energy saving • Predictive measures of load • Meta-heuristic algorithms	[105]–[111]
Energy Efficiency	• Energy efficient resource management • Energy proportional networking	[114], [116], [118]–[128]
Predictive Resource Scheduling	• Predictive resource scheduling in multiclouds	[134]–[136]

the infrastructure. This shortcoming of shared clouds is also reflected in the market uptake of cloud computing for HPC workloads. A recent market study published by Intersect360 Research [74] shows a lack of market growth for HPC in public clouds, despite mentioning machine learning as a key new trend. The report suggests that the market remains selective with respect to the jobs it offloads to the cloud platforms. The performance unpredictability in a multi-tenant cloud computing system typically arises from server virtualization and network sharing. While the former can easily be addressed by allocating only a single tenant per physical machine, the sharing of network resources still remains a major performance variability issue [70].

2) LACK OF FLEXIBILITY IN LOSS-LESS INTERCONNECTION NETWORKS

Over the last decade, we have seen an incredible growth in the popularity of loss-less interconnection networks, such as InfiniBand (IB) [75], in the HPC systems and data centers. Recently, the use of loss-less interconnection networks in cloud computing has also gained interest in the HPC community [44], [45], [76], [77]. Thanks to the high-throughput and low-latency communication such interconnect solutions offers, cloud systems built on an loss-less HPC interconnects promise high potential of bringing HPC and other performance-demanding applications to clouds [71]. Furthermore, many such interconnects provide sufficient security mechanisms to complement in typical non-trusted data center environments. However, clouds using HPC interconnects have not still matured. Challenges related to load-balancing, low-overhead virtualization, and performance isolation hinder full potential utilization of the underlying interconnect when clouds are deployed on loss-less interconnection networks. Moreover, clouds are characterized by dynamic environments resulting in frequent network reconfiguration, for which efficient mechanisms are yet to be supported in current loss-less interconnection technologies.

Table 2 shows a summary of challenges and future directions related to the cloud architecture.

III. RESOURCE MANAGEMENT

In large-scale cloud environments, efficient resource management is an important challenge affecting both the delivered application performance and the costs of maintaining reliable services for the end users. The key resource management task is to find the optimum allocation of available resources for sustainable application delivery and performance, together with achieving cost-effectiveness and energy-efficiency in a cloud data center. Resource optimization, however, imposes a diverse set of challenges for each research type, and at each cloud layer. Further, resources themselves are either physical objects such as Central Processing Units (CPUs), memory, and disk storage, or virtualized objects encompassing complex high-level services, such as *virtual machines*, or *containers*. In the following, we outline key challenges associated with the resource management and briefly identify research areas where the state-of-the-art still needs improvements to cope with the needs of next generation cloud computing platforms.

A. VIRTUAL MACHINE MANAGEMENT

Server virtualization is the key technology behind modern data centers; it provides abstraction of the hardware and system resources, like CPU and memory, to achieve improved sharing and utilization. Hypervisors such as Xen [78] and KVM [79] allow multiple VM to be co-hosted on a single physical machine to enable server virtualization. Cloud data centers typically rely on server virtualization technologies for providing on-demand provisioning of VMs. However, virtualization also has its own costs. Studies show that the overhead incurred by the use of virtualization could have a substantial negative impact on the overall performance of data centers [80]. Moreover, in a dynamic cloud environment, where VMs are allocated and destroyed often, resource fragmentation may occur, which can result in lower server utilization. To cater for fragmentation, VM migration techniques are employed, which enable moving a running VM from one server hardware to another. VM migrations make a powerful tool to optimize VM allocations in a cloud data center based on the cloud provider's optimization strategies. For instance, VM migration can be employed to allow for power saving by shutting down unused server machines. It will also improve network performance and application performance by co-locating communicating VMs, and avoiding co-locating CPU-intensive workloads. VM migration across geographically distributed data centers or among different cloud providers in a multicloud environment, however, remains challenging to the day.

Container-Based Virtualization: To reduce VM overhead, an interesting alternative candidate for the server virtualization is the recent developments around Linux Containers [81]. Containers offer isolation as close as to VMs without the overhead of running a separate kernel and simulating all the hardware. Containers, however, are not fully cloud-ready, and the support for effective container management in clouds is yet to be realized.

B. MULTI-TENANCY

Multi-tenancy is a salient feature of cloud computing, defined as a scheme where applications belonging to different users are co-located in a shared data center infrastructure [82]. Multi-tenancy promises high utilization of system resources and helps maintaining cost-effective operation for service providers. However, multi-tenant infrastructures also introduce several security and performance challenges [83], [84]. The most critical one is associated with providing performance isolation to the tenants [85], [86]. Previous research has shown that the sharing of resources with other tenants in a shared cloud incurs unpredictable application performance [87]–[89].

Network and performance isolation is a much discussed topic in the literature. Both hypervisor level rate-limits and QoS features have been used to provide appropriate bandwidth to the tenants. SeaWall [53] provides a fair network sharing policy among competing VMs. However, as the sharing policy applies to the VMs instead of tenants, a tenant can practically increase its share of the bandwidth by launching additional source VMs. Other solutions, like Netshare [90], Oktopus [50], and SecondNet [52] work on per tenant bandwidth share basis. However, they require some kind of centralized control plane resulting in reaction time overhead. A more recent approach, EyeQ [91] uses congestion control to provide predictable bandwidth guarantees to the tenant VMs. The isolation system works by enforcing admission control on traffic, thus pushing bandwidth contention to the network edge.

Trade-Off Between System Utilization and Performance Isolation: The most important challenge with respect to multi-tenancy in a shared cloud system is to find the optimum trade-off between high system utilization and performance isolation among tenants [92]. Moreover, performance isolation requirements are application-specific and cloud systems should be equipped with capabilities to monitor and detect contention points between tenants [93]. They also need to resolve the contention points where necessary to use service migrations and other contention-avoidance techniques. In this context, current work is very limited and mostly academic in nature, requiring further research and development in this direction.

C. AUTOMATED SERVICE PROVISIONING AND ELASTICITY

Cloud services are defined as software services that use cloud resources. A cloud service consists of appropriately configured software components deployed into a set of dynamically allocated infrastructure resources. One of the key features of cloud platforms is the capability of acquiring and releasing resources on-demand. The objective of automated service provisioning is to enable automatic allocation and de-allocation of resources from the cloud to satisfy service level objectives, while minimizing operational costs.

Automated Service Provisioning For QoS Guarantees: In order to enable the automated provisioning and management of cloud services, Kirschnick *et al.* proposed an architecture to allow the declarative definition of services using a Template Description Language for the topology design [94]. In addition, a Component Description Language is defined to specify the individual configuration and deployment behavior of software components. Templates enable requirements-driven rapid service provisioning, while hiding the configuration complexities from the end user. Topology and Orchestration Specification for Cloud Applications (TOSCA) is a standard to enable automated deployment, termination, and further management functionality, such as scaling or backing up applications through the two TOSCA main concepts: *i)* application topology; and *ii)* management plans. The former provides a structural description of the application, the components it consists of, the relationships among them, and components' management capabilities. The latter combines these management capabilities to create higher-level management tasks, which can then be executed fully automated to deploy, configure, manage, and operate the application. Wettinger *et al.* presented a generic methodical framework to transform DevOps artifacts into standards-based TOSCA models that can be orchestrated arbitrarily to model and deploy cloud applications [96]. As another recent approach, Naseri and Navimipour proposed a new hybrid agent-based method for efficient service composition in the cloud [97]. They compose services by identifying the QoS parameters and exploit the swarm optimization algorithm to select the best services based on a fitness function. As future work, using autonomic service computing features to provide self-management capabilities such as fault-tolerant cloud services and comprehensive quality of service assurance is an important future direction.

D. LOAD BALANCING AND HETEROGENEITY

Load balancing in cloud platforms is a mechanism that distributes the excess dynamic local workload evenly across all the cloud nodes. Load balancing is aimed at achieving high user satisfaction and resource utilization ratio, by ensuring that no single node is overwhelmed, thereby improving the overall performance of the system. There are mainly two types of load balancing algorithms: *static* and *dynamic* algorithms.

Static Algorithms: In static algorithms, the traffic is divided evenly among the cloud servers. They require a prior knowledge of system resources, so that the decision of shifting of the load does not depend on the current state of the system, making them suitable for systems with low variation in load. The early static algorithms are based on the well-known load balancing technique called *Round Robin* in which all processes are divided among all available processors [98], which may make some nodes heavily loaded and other may be idle. This problem is solved by Radojević and Žagar [99] by introducing a static load balancing algorithm called the Central Load Balancing Decision Model (CLBDM)—an enhancement of the Round Robin technique using session switching at the application layer. Hu *et al.* [100] introduced a static scheduling load balancing approach on virtual machine resources whose technique considers the historical data and the current state of the system. Min-Min and Max-Min algorithms have also been proposed for cloud resource management. In Min-Min algorithms [101], the task with minimum completion time is selected and assigned to the corresponding server. After this assignment, calculated completion time of remaining tasks is updated on the server hosting a task. The Max-Min algorithm [102] works in the same way, but it selects a task with maximum completion time. The Opportunistic Load Balancing Algorithm attempts to keep each node busy. This algorithm deals with the unexecuted tasks faster and in random order to current node, where every task is randomly assigned to the node.

Dynamic Algorithms: In dynamic algorithms, the server with lightest load in the whole network or system is searched and preferred for balancing a load. Babu and Venkata Krishna [103] proposed a Honey Bee Behavior inspired Load Balancing technique which helps to achieve even load balancing across virtual machines to maximize throughput. This approach helps other processes to choose their VM, meaning that if a task has high priority, then it selects a VM having minimum number of priority tasks. Ant Colony Optimization is the other dynamic load balancing technique in which an ant starts the movement as the request is initiated. It uses the "Ants" to collect information about the state of the cloud nodes and uses this to assign tasks to a particular node [104].

Ren *et al.* presented a dynamic load balancing algorithm for clouds based on an algorithm called Weighted Least Connection (WLC). The WLC algorithm assigns tasks to a node based on the number of connections that exist for that node.

There are a number of key challenges, in this area, which are worth considering as future directions.

1) MULTI-FACTORS FOR LOAD BALANCING

The first is to consider multiple types of resources in load balancing, such as memory, computing, storage, and bandwidth together. Moreover, multi-objective optimization algorithms are needed to address the VM load balancing as a multi-objective problem includes parameters such as time of migrations, execution, and SLA violations.

2) ENERGY SAVING

This is an important factor to provide economic efficiency where utilization of resources is maximized. Energy-aware load balancing aims at identifying servers operating outside their optimal energy regime and decides if, and when, they should be switched to a sleep state or what other actions should be taken to optimize the energy consumption. There are some works on energy-aware load balancing. Paya and Marinescu proposed and algorithm that performs load balancing and application scaling to maximize the number of servers operating in the energy-optimal regime [105]. As another solution, Zhou *et al.* proposed a carbon-aware online control framework to dynamically balance and make decisions on three control decisions, including geographical load balancing, capacity right-sizing, and server speed scaling [106]. Existing work still lacks load balancing techniques that consider together energy consumption, carbon emission and cost of services.

3) PREDICTIVE MEASURES OF LOAD

Cloud resource prediction is challenging due to the very dynamic and fluctuating nature of workloads. In order to predict the resource needs, historical time series data of past workload is usually leveraged. There are few works that propose prediction-based load balancing approaches in the cloud. LSRP [107] is an ensemble approach for resource demands prediction through a two-phase prediction mechanism. Bala and Chana designed a prediction-based approach facilitating proactive load balancing through the prediction of multiple resource utilization parameters in the cloud [108]. In existing solutions, migration time represents the common issue of making decisions in a cloud environment due to incomplete information. Analyzing prior requirements and current process properties makes it possible to predict the future load. In addition, for a virtualized infrastructure, it is necessary to investigate the barter between the efficient utilization of the hardware infrastructure and predictability of resources.

4) META-HEURISTIC ALGORITHMS

Meta-heuristic algorithms are mostly inspired from nature, like genetic algorithms, Ant Colony Optimization (ACO), and honeybee foraging algorithms. Wen *et al.* citeAcoSchedCloud proposed a distributed VM migration strategy based on ACO. ACO and Particle Swarm Optimization (PSO) can be combined to deal with virtual machine load balancing [110]. Pang *et al.* [111] proposed a task-oriented resource allocation method (LET-ACO) to optimize the energy consumption by scheduling tasks, using an improved ACO. Due to their large solution space, meta-heuristic algorithms need more time to run and find the final solution, as compared to heuristic algorithms. These types of algorithms, used for load balancing, also need to be improved with respect to their time cost. In addition, evaluating their performance on popular platforms, like OpenStack, is desirable.

E. ENERGY EFFICIENCY

Given the strong correlation between climate change patterns and CO_2 emissions [112], energy efficient systems with low carbon footprints have become a natural topic of recent developments. Modern data centers boost very large infrastructures with high energy requirements, and are subject of large amount of research in improving their power-efficiency [113]–[115]. The power consumption analysis in a data center is complex as power requirements come from different areas such as cooling infrastructure, servers, networking equipment, and operations [116]. In cloud computing, energy efficient resource management can be tackled at different levels, from the hardware infrastructure to virtualized resources and applications, and the distributed workload management across data centers [117]. In the following, we categorize relevant research areas and highlight challenges for the future work.

1) ENERGY EFFICIENT RESOURCE MANAGEMENT

Prominent energy efficient resource management techniques in a single cloud data center include energy-aware VM placements and migrations [118]–[120], server consolidation to save power by shutting down unused machines, Green SLA-aware computing [121], [122], and prediction-based algorithms [123]. Distributed data centers bring additional challenges, mostly related to the global workload management across data centers taking data center characteristics and request proximity into account. A good overview of the energy efficiency in cloud computing and state-of-the-art analysis is provided by Khosravi and Buyya [124]. The trade-off between high energy efficiency and SLA violations, as well as the study of the workload characteristics for better predictions are still topics of active research as existing solutions do not address these areas sufficiently [125]. In addition, for federated and geographically-distributed data centers, VM migrations across geographically-distributed data center sites (for energy saving) is still not explored [124].

2) ENERGY PROPORTIONAL NETWORKING

Network is the most critical resource to provision and manage in a data center, albeit less studied as a resource management problem in the context of clouds. The effectiveness of a cloud data center directly depends on the ability of its network architecture to intelligently provision resources, quickly adapting with irregular demand patterns, and its ability to work energy-efficiently. Networks account for a good proportion of power-consumption in a data center with studies suggesting their share as high as 20% of the total power consumption [114], [126]. Currently, both hardware-level enhancements and network-specific features have been studied for realizing power-efficient networks [127], [128]. However, energy efficiency challenges arising from new technologies, such as federated cloud networking and unified DCN fabrics are still not well-studied. Moreover, challenges related to energy-aware network provisioning mechanisms for cloud systems, considering smart routing, congestion control, and load balancing algorithms, remain to be addressed. Furthermore, as discussed above, the trade-off between power-efficiency and network performance needs to be considered by the future work. In addition, mechanisms of high granularity service differentiation that can work efficiently under different traffic conditions are to be devised for real-world cloud networks [116].

F. PREDICTIVE RESOURCE SCHEDULING

The main aim of predictive resource scheduling is to allocate resources in advance to improve system performance and scheduling quality. Many of the basic principles and techniques of the predictive resource scheduling in the cloud date back to the grid computing days. Chapman *et al.* presented a predictive resource scheduling framework for grid computing infrastructures [129]. The authors have employed a Kalman filter theory [130] based implementation to predict the expected future load on the grid to reduce the job waiting time. Lately, a variety of methods have been used in the literature to predict application performance and job completion time in the grid, ranging from meta-heuristic techniques to genetic algorithms [131] and ant colony optimizations [132], [133].

Predictive Resource Scheduling in Multiclouds: In the context of cloud computing, a problem with predicting cloud resource performance is the unavailability of the data needed at the user end. In particular, with the growing popularity of multiclouds, the end users are faced with the challenge of predicting cloud performance with very limited information about the cloud infrastructure itself. As clouds by definition employs a shared resource model with multiple tenants and heterogeneous workloads, understanding and predicting application performance turns out to be difficult. Islam *et al.* used neural networks and linear regression models to predict upcoming resource demands for adaptive resource provisioning in the cloud [134]. However, the focus remains on dynamic resource scaling and elasticity issues. Efficient auto-scaling has also remained an important topic of research [135], [136]. With application deployments spanning over multiple cloud providers and geographical locations, resource requirement predictions, and auto-scaling techniques need to be revisited to cater for the needs of multicloud users. Table 2 shows a summary of challenges and future directions related to the cloud resource management.

IV. DATA MANAGEMENT IN THE CLOUD

Clouds offer significant advantages over traditional cluster computing architectures including flexibility, ease of deployment, and rapid elasticity—all packed up in an economically attractive pay-as-you-go business model. Thanks to these advantages, enterprises are increasingly moving their business applications together with their back-end data management systems to clouds. Traditionally, distributed file systems, like Global File System (GFS) and Hadoop Distributed File System (HDFS), and Massive Parallel Processing (MPP) analytical database systems were deployed in the cloud, as they support rapid elasticity by scaling-out. However, such systems impose significant challenges in application portability and migration to clouds. More recently, cloud deployments using *non-relational* data management systems are also growing.

In the following, we provide an overview of key data management issues, briefly describe the state-of-the-art, and highlight challenges for the future work.

1) TRANSACTIONAL DATA MANAGEMENT

Traditionally, database transactions are needed to conform to the ACID guarantees: *Atomicity*, *Consistency*, *Isolation*, and *Durability*. However, maintaining ACID transactional guarantees over large distributed infrastructures have proven to be hard [139]. For instance, when data is replicated over geographically-dispersed locations, maintaining consistency can have substantial negative affect on the availability of the system [140], [141]. Although alternative guarantees such as BASE (*Basically Available*, *Soft State*, *Eventual consistency*) are practically sufficient for most cloud applications [142], they make cloud transition harder for some enterprise applications that heavily rely on transactions. A partitioned database system, such as ElasTraS [143], can be used for distributed synchronization, but still consistency across partitions is compromised. Supporting transactional databases in clouds, thus, is subject to challenges which are physically bounded by the latency between distributed cloud data centers.

2) SHARE-NOTHING ARCHITECTURES

In a shared-nothing architecture, each node in a distributed system is independent and self-sufficient. Shared-nothing databases have become very popular due to the scalability features they offer. However, the problem with the shared-nothing architecture is that they are not very suitable for the applications that are transaction-based, as described in the previous section. With the recent advancements in data sharding techniques and NoSQL storage systems, like BigTable [144], HBase [145], and *Cassandra*, some of the challenges associated with non-relational data management systems have been addressed. However, cross-cloud data access in shared-nothing architectures requires context-aware data access control mechanisms and appropriate policy models. However, these issues are still in early stage of development.

3) DATA LOCK-IN

Lock-in is an economic condition in which a customer is dependent on the vendor-specific technology, products, or services as it becomes unfeasible or fairly costly to switch to another competitor [146]. Data lock-in in cloud computing systems can be attributed to two important factors. First, the data migrations across different cloud providers are costly, and the trade-off between *moving data near applications* and *moving applications near the data* often go in favor of the second option [147], [148]. Second, different cloud providers support different data storage technologies, formats, and data access protocols, forcing cloud users into a lock-in due to incompatibilities between cloud systems. In this connection, active standardization efforts are needed to ensure data portability among heterogeneous cloud systems [149].

4) HOLISTIC DATA LIFE-CYCLE MANAGEMENT

Holistic management of the complete data life cycle is an important challenge for efficient data management in clouds. The data life cycle includes distinct phases covering data acquisition, preparation, analysis, integration, aggregation, and final representation of the data [150]. In large-scale distributed environments, optimization of the complete data life cycle is generally application specific, and involves careful planning and proactive management strategies [151]. Automated data placement in the cloud has attracted interest of the researchers in the recent years [152], [153]. However, for efficient data management in the cloud, data life-cycle management needs to be integrated with both application modeling and data storage frameworks. The current approaches [154]–[156] tend to exploit data locality by using data-aware job scheduling, but they do not address combined concerns related to data acquisition, data outsourcing to the cloud, privacy, and confidentiality. With respect to data placement, acquisition or generation in the cloud, long-term storage decisions are desired due to high costs associated with the data migration. Furthermore, the initial selection of the data placement can also potentially affect subsequent application deployments due to *data gravity*, thereby a holistic approach towards the data life cycle management is strongly needed.

Table 3 shows a summary of challenges and future directions related to the cloud data management.

V. APPLICATION ENGINEERING AND MANAGEMENT

In this section, we discuss programming models and frameworks for cloud-based applications, techniques for managing application performance and application life cycle.

TABLE 3. Data management research challenges and future directions.

Sub-areas	Challenges and Future Directions	References
Transactional Data Management	• Supporting transactional databases in clouds	[139]–[143]
Share-Nothing Architectures	• Cross-cloud data access in shared-nothing architectures	[144], [145]
Data Lock-in	• Standardization for data portability	[146]–[149]
Holistic Data Life-Cycle Management	• Holistic approaches towards data life cycle management	[150]–[156]

A. APPLICATION PROGRAMMING MODELS AND FRAMEWORKS

Jin and Buyya developed a MapReduce-based programming framework called Aneka, for the .NET platform, called MapReduce.NET [157]. Its main goal is to support programming data-intensive applications in the cloud and facilitate also development of compute-intensive applications, such as Genetic Algorithm (GA) applications. The heart of the framework is the Aneka Container which is the minimum unit of deployment for Aneka Clouds, and also the run-time environment for distributed applications. The container hosts a collection of general services that perform all the operations required to create an execution environment for applications, including resource reservation, storage and file management, persistence, scheduling, and execution.

Cloud vendors like Amazon and Google have introduced the *serverless* programming models to simplify the development of cloud-native code [158], [159]. The serverless models basically abstract away most of the DevOps related concerns so that developers create actions that load on-demand and are triggered to execute by system generated events or end users. OpenWhisk [160] is a recent serverless programming model that supports multiple programming languages and composition of services using action sequences.

CodeCloud [161] is an architecture and a platform to support the execution of scientific applications in the cloud under different programming models. It features a declarative language of the requirements of applications and virtual infrastructures with an emphasis on software deployment and customization at run-time. It encompasses virtual containers to orchestrate the virtual infrastructure deployment and configurations for different programming models. ServiceSs [162] is a framework for the development, deployment, and execution of parallel applications, business and scientific work flows and compositions of services in the cloud. It provides users with a simple sequential programming model that does not require the use of Application Programming Interfaces (APIs) and enables the execution of the same code on different cloud providers. In the mOSAIC project [163], a reference layered API is provided which focuses on achieving interoperability between clouds. The developer has to provide application requirements through the API while the multi-agent brokering mechanism of mOSAIC searches for services matching the requirements.

Cloud Ready Programming Models: The main future direction, in this area, includes a brand new programming paradigm to develop *clouds ready* applications [164], in which one would write only the business logic. Then, the cloud infrastructure, the framework, and the middleware should be able to take care of all concerns of deployment, monitoring, and self-scaling. This also needs a reasonably sophisticated Infrastructure as a service (IaaS) or PaaS level API, exposed by the cloud infrastructure, supporting application developer's needs, such as generic application life-cycle management.

B. APPLICATION PERFORMANCE MANAGEMENT

In computing, Application Performance Management (APM) is an area that deals with techniques for efficient monitoring, performance optimization, and high-availability of software applications [165]. The main job of an efficient APM mechanism is to monitor the application, detect performance issues, and maintain required level of service. In cloud computing, APM relates to two main areas: efficient cloud monitoring, and dealing with performance unpredictability in clouds, as discussed in the following.

1) CLOUD MONITORING

Clouds are complex, multi-layered infrastructures, often spanning multitude of hardware and software domains. As cloud-based services are increasingly becoming popular, there is a definite need for monitoring both the behavior of the cloud infrastructure and the achieved performance for the applications deployed on them under different conditions and time periods. Enterprise-level continuous cloud monitoring tools have been created to tackle this issue. However, most popular cloud monitoring tools, such as CloudWatch [166] and AzureWatch [167], are proprietary and cloud provider-specific [168]. Thus, there is still a need for providing open

TABLE 4. Application engineering and management research challenges and future directions.

Sub-areas	Challenges and Future Directions	References
Application Programming Models and Frameworks	• Cloud-ready programming models	[157]–[164], [175]
Application Performance Management	• Platform-independent monitoring tools • Performance unpredictability in the cloud	[88], [166]–[170]
Application Life-Cycle Management	• Cross-layer and cross-platform monitoring and adaptation	[95], [171]–[174]

platform-independent monitoring tools, as well as uniform monitoring interfaces for different cloud providers.

2) PERFORMANCE UNPREDICTABILITY IN THE CLOUD

When using resources acquired from public cloud providers, the users have limited control over resource provisioning onto the actual hardware. The lack of knowledge about the infrastructure makes it difficult to schedule application consisting of distributed components efficiently. Even repeated deployments of the same application on the same cloud platform might result in different performance metrics due to the hardware selection in the cloud, uncontrollable by the cloud user. In addition, clouds by definition provide a shared resource model where multiple tenants are served from the same data center infrastructure. Therefore, applications running for one tenant in the shared cloud are affected by the interference from other concurrent workloads, resulting in performance unpredictability, as previously shown in the literature [88], [169]. To cater for this dynamical challenge, applications deployed on clouds need to be continuously monitored and adapted, if needed. This makes sure that the current deployments correspond to the best possible configurations according to the current cloud resource performance, user requirements, constraints, and the execution context. In this context, PaaSage [170] takes a mode-driven feedback controlled approach where the current deployments are continuously monitored, and if they do not fulfill the current model's constraints and goals, a new deployment is proposed and deployed in the cloud. However, PaaSage does not consider the data-awareness needed for many enterprise cloud applications, and data-intensive workload. Therefore, further research is needed to devise mechanisms so that application deployments and reconfigurations in clouds follow the data locality requirements.

C. APPLICATION LIFE-CYCLE MANAGEMENT

The Cloud Application Management Framework (CAMF) is a recently established open source technology project that facilitates life-cycle management operations for cloud applications, in a vendor-neutral manner. To achieve this, CAMF focuses on three core operations: application description, application deployment, and application monitoring. In large development teams, group members can effortlessly share application descriptions, deployment information, artifacts, or even complete cloud projects. Sefraoui *et al.* propsed a cloud Integration and Management Platform (CIMP), which serves as an intermediate between users and cloud solutions and offers additional components that enhance their functionality [171]. TOSCA [95] proposes the concept of ServiceTemplate to capture the structure and the life-cycle operations of cloud-based applications. The plan part of ServiceTemplate defines how the cloud application is managed and deployed. The management plans and management operations trigger transitions between the states of life-cycle: *i.e.* starting, running, stopping, and error. Baryannis *et al.* proposed a research road map for managing the lifecycle of service-based applications on multiclouds [172]. It encompasses extensions to use TOSCA to tie several cloud-specific models together and to connect with work flows and offer a logic-based requirements engineering model.

Cross-Platform Life-Cycle Management: The main research challenge in life-cycle management is *cross-platform and cross-layer monitoring and adaptation* of applications. Existing frameworks have paid little attention to this challenge. The SmartFrog framework [173] is an early solution to manage life-cycle of distributed applications in multiclouds through provisioning, deployment, change management, and termination. To this end, this framework takes into account relationships between software components, virtual infrastructures, and the underlying physical infrastructure. CloudMF [174] leverages model-driven engineering and provides a domain-specific language for specifying the provisioning and deployment of multicloud applications, and uses models for managing the full application life-cycle. Both frameworks above lacks a comprehensive life-cycle management solution which covers all stages from monitoring to reconfiguration.

Table 4 shows a summary of challenges and future directions related to the cloud-based application engineering and management.

VI. APPLICATION DOMAINS

In this section, we discuss specific challenges associated with cloud applications in different domains.

A. INTERNET OF THINGS

In recent years, several attempts have been made to integrate smart things to web-based platforms, such as proprietary Representational State Transfer (REST) based application servers and later to cloud-based platforms. While early work in this area was mainly focused on technological integration challenges for making such an integration happen efficiently and easily, recent cloud-based solutions aim to facilitate wide-scale adoption and integration of IoT systems, exploiting the cloud's benefits in terms of performance, scalability, etc. In particular, IoT can benefit from the unlimited capabilities and resources of the cloud to compensate its technological constraints, *e.g.* storage and processing. On the other hand, the cloud can benefit from IoT by extending its scope to make services of real world things accessible at the cloud level in a more distributed manner. This integration will impact the development of future IoT applications, where information gathering, processing, and transmission will introduce new challenges and requirements, such as real-time and context-aware data processing.

One category of exiting work has mainly focused on exploiting the cloud service delivery models to enable efficient and scalable IoT service delivery [176] using, *e.g.* domain-independent PaaS frameworks for efficient IoT service delivery. Cloud-based 'Hubs' is the other category of results proposed for developing large-scale IoT applications such as smart cities. Under this model, the core IoT infrastructure is exposed as a data hub via a PaaS framework, addressing some of the core technical issues in building cloud-based IoT frameworks. The main advantage of this approach is that multiple hubs can be connected, or federated, to build up a system of systems that can represent significant parts of the IoT ecosystem, *e.g.* the components of a smart city [177]. Considering cloud-based middleware solutions for IoT services delivery, in OpenIoT project, a middleware framework is proposed to enable the dynamic, self-organizing formulation of optimized cloud environments for IoT applications and IoT services delivery [178].

Cloud-based IoT data analysis frameworks are another category of applications in this domain. Vögler *et al.* proposed a generic, scalable, and fault-tolerant data processing framework based on the cloud to allow operators to perform on-line and off-line analyses on gathered data to better understand and optimize the behavior of the available smart city infrastructure [179]. The proposed framework is able to autonomously optimize the application deployment topology by distributing processing load over available infrastructure resources when necessary based on both on-line analysis of the current state of the environment and patterns learned from historical data.

Cloud-based programming for IoT applications defines high-level programming constructs and operators to encapsulate domain-specific knowledge, *i.e.* domain model and behavior. This raises the level of programming abstraction and enable the developer to implement applications without worrying about the complexity of low-level device services and raw sensory data streams publication and processing [180], [181].

Although there exist many industrial products and academic initiatives on cloud-based applications and integration of IoT systems, a number of critical challenges should be carefully addressed.

1) SECURITY

Handling security concerns of cloud-based IoT services is challenging due to the differences in the security mechanisms between IoT devices and cloud computing platforms.

2) REAL-TIME DATA PROCESSING

Real-time provisioning of IoT services is an important design aspect of IoT cloud applications, such as smart cities. Addressing real-time requirements of diverse IoT services in large-scale cloud-based deployments is very important. Recently, some work has been carried out on modeling the IoT data hosted in the cloud with real-time processing support [182]. As mentioned before, fog computing is a recent category of approaches for supporting real-time and low-latency processing of IoT data. There are several frameworks proposing the use of fog nodes for IoT data processing and application programming over fog [35], [37], however they do not specifically meet the real-time processing requirement.

3) STANDARDIZATION

Standardizing cloud computing also presents a significant challenge for IoT cloud-based services due to having to interoperate with various vendors.

4) DYNAMIC DEPLOYMENT IN CLOUD-FOG

There a few works reported recently on dynamic Fogs. They either are focused on using container-based platforms on fogs and studying their feasibility [183], or propose multi-tenant cloud-fog SDNs or NFV orchestration for fog services [184], [185]. To allow for flexible provisioning of applications whose deployment topology evolves over time, we need approaches to clearly separate application components that are independently executable. Moreover, integrating non-functional elasticity and quality-of-service dimensions, *e.g.* context and costs, is very critical in order to further optimize the deployment topology and enable local coordination of topology changes between edge devices.

5) CROSS-CLOUD IoT SERVICES

New cloud platforms are now more frequently used with increasing number of new services, *e.g.* smart city cloud offerings. It will be necessary to ensure that cloud based IoT systems are able to accommodate a number of peer PaaS services. In addition, application developers can be offered a framework that allows them to exploit cloud services and functionality resides in different clouds [177].

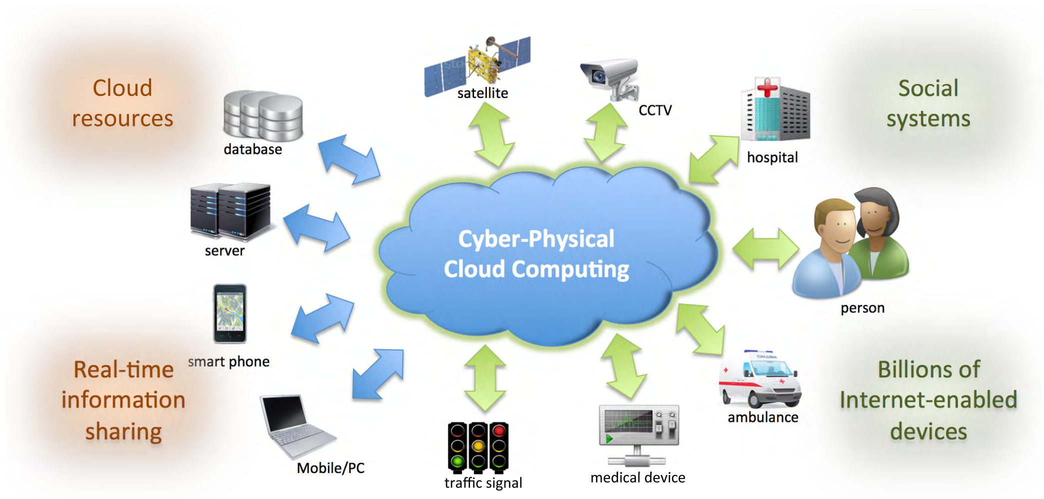

FIGURE 4. The conceptual model of cyber-physical cloud computing systems [188].

6) APPLICATION-ORIENTED RESOURCE MODELING

It refers to evaluating and modeling the resource consumption of IoT applications in order to effectively allocate computing resources on a multi-tenant IoT service platform. The application-oriented resource model will consider device behavior and constraints, physical context of applications, data processing requirements, and usage patterns. Li *et al.* proposed a multi-tenant PaaS model to enable the concept of virtual verticals, as opposed to physically isolated vertical solutions [186].

B. CYBER-PHYSICAL SYSTEMS

Cyber-Physical Cloud Computing (CPCC) is increasingly gaining attention thanks to its scalable, on-demand, and reliable provision of physical services. This has triggered several lines of research to address challenges, such as architectural patterns for integration, virtualization of physical components in the cloud, security, privacy, and cloud-assisted situation-awareness and decision support [187]. The first and the most significant issue, in this context, is the software architectural model for CPCC. As a generic view to the integration of Cyber-Physical Systems (CPS) and the cloud, Simmon *et al.* discuss integration aspects of CPS in the cloud from a high-level viewpoint, such as CPS virtualization, interconnectivity between CPS services in the cloud, reliability and privacy [188]. The paper investigates a diverse set of requirements for CPCC, *e.g.* timeliness, reliability, and security; and proposes a conceptual architecture framework for CPCC, as shown in Figure 4.

The IMC-AESOP research project [189] presented main architectural services for cloud-based industrial applications, *i.e.* a type of CPCC, such as monitoring, management, data handling, and integration. Additionally, there are some results that address particular research challenges in the integration of manufacturing and industrial systems into the cloud, *e.g.* vehicles and robots. In [190], the V-Cloud architecture is presented to enable cloud computing system of vehicles in order to meet safety and comfort requirements for the driver. Jiafu *et al.* designed a vehicle CPS and mobile cloud computing integration architecture to provide mobile services for potential users such as drivers and passengers to access mobile traffic cloud [191]. The future research directions include:

1) NEED FOR STANDARDS

The current status of CPS clouds shows that most of existing solutions are proprietary although some of them rely on open-source platforms, *e.g.* Apache Hadoop with its Map-Reduce framework [138].

2) PRIVACY AND SECURITY

Existing work on the security of CPCC is somewhat limited. There are some solutions on security of Industrial CPS. For example, in [192], the security challenges of industrial Supervisory Control and Data Acquisition (SCADA) systems are discussed when introduced for IoT-cloud environments. SCADA systems are mostly utilized in industrial CPS. Within the same industrial context, Lopez and Rubio proposed an industrial architecture where multiple access control models are assessed when cloud technologies are integrated, with special emphasis on their adaptability and achieving a trade-off between security and efficiency [193]. Although some challenges are common with traditional cloud computing security, CPCC would be subject to additional threats that must be carefully addressed. With respect to the fact that CPCC systems are highly distributed involving different heterogeneous components, attacks can occur at different layers, either at the CPS layer or at the cloud layer. Therefore, comprehensive end-to-end security mechanisms must be provided to ensure the integrity of any transaction that takes place in CPCC systems.

3) PROGRAMMING ABSTRACTIONS

It is important to provide developers with new APIs that allow them to easily interact with CPCC systems, in the same way as traditional clouds, like Amazon Web Services, Google Apps Engine, etc. The recent contributions in this area have mostly revolved around general cloud-level service provisioning model. Activity as a Service [194] is a full-fledged cyber-physical framework to support community, on-line and off-line human activity recognition. This framework is built atop the BodyCloud platform to enable high performance computing of collected sensor data and data storing on the cloud. Wu *et al.* present a smart delivery drone as a Cloud-Based Design Manufacturing (CBDM) service and a corresponding CBDM system architecture is proposed which incorporates CBDM-based design processes [195].

4) REAL-TIME REQUIREMENTS

CPCC systems often include time-sensitive requirements that require real-time guarantees to deliver time-critical data, in particular for automation and control applications, such as vehicle applications. Therefore, cloud-hosted CPS services should be enhanced with the support of real-time requirements. The state-of-the-art has mostly focused on real-time information delivery in vehicular CPS [196], [197].

C. ESCIENCE

eScience is defined as a combination of information technology and science to address challenges related to storing, interpreting, analyzing, and distributing large-scale scientific data. eScience has been applied in various fields such as biology, chemistry, physics and sociology. The earlier eScience applications were mostly deployed to computing grids thanks to their aggregated computational power and storage capacity. Recently, many eScience projects from various scientific disciplines have been shifting to cloud platforms by introducing *eScience as a service*—an emerging and promising direction for science computing.

Cloud-based eScience applications can be generally classified to the following categories [198]: life sciences [199], physical sciences [200], climate and earth sciences [201], and social sciences and humanities. The tasks in these applications are typically grouped into stages that are connected by producer-consumer data sharing relationships. In general, seven common data flow patterns can be envisioned including pipeline, broadcast, scatter, gather, reduce, all-gather, and all-to-all [202]. eScience applications, like typical *many-task applications*, can be viewed as stages of independent tasks that are linked by these data flow patterns, *e.g.* using map-reduce model to schedule jobs.

A dominant category of eScience applications is medical research. Bioinformatics is confronted with increasingly large data sets, *e.g.* in gene sequencing systems. Genome sequencing aims to provide a deep understanding of sequence variations as a foundation for investigating the relationship between genotype and phenotype. It provides important insights into the study of population genetics, *e.g.* causal variants of genes for syndromes such as Freeman-Sheldon syndrome [203]. The other application area is medical image processing. The growth in the volume of medical images produced on a daily basis in modern hospitals has introduced new image processing use cases [204], such as *i)* parameter optimization for lung texture classification using support vector machines; *ii)* content-based medical image indexing and retrieval; and *iii)* dimensional directional wavelet analysis for solid texture classification.

Although there are many eScience applications today hosted by cloud platforms, the development of next generation eScience applications poses new challenges due to problems rooted at the interplay between eScience requirements and cloud computing features. We explain them below.

1) DATA LOCK-IN

Since the eScience projects usually involve a large amount of data provided by different research institutes, such as the genome sequence data, it will become crucial to support standard cross-cloud APIs for faster and cheaper processing of such data. Unfortunately, most existing cloud infrastructures provide very limited capabilities for data, application, and service interoperability. This makes it difficult for the cloud user to migrate data and services from one provider to another. As mentioned earlier in this paper, there are two main factors related to this challenge: *i)* data migration across different cloud providers is costly [147], [148], and *ii)* different cloud providers support different data storage technologies, formats, and data access protocols. With respect to the latter, standardization is needed to ensure data portability among heterogeneous clouds [149].

2) eScience COMMON DEVELOPMENT INFRASTRUCTURE

The efforts in implementing eScience projects in the cloud are rather ad-hoc today, resulting in the lack of reusability of one solution in other eScience applications [205]–[207]. In order to provide a more efficient development cycle and better exploit running eScience systems, we need generic and reusable platforms on which applications from various research fields can be built along with components specific to each application type. For example, crowdsensing frameworks [208], [209] are a generic solution for many applications based on large-scale community data gathering.

3) REAL-TIME PROCESSING OF SCIENTIFIC DATA

Most of existing technologies and tools for scientific data processing, *e.g.* Hadoop [138], are not a catch all technology but rather they are best suited to batch processing applications, as opposed to real-time queries which are issued in new discoveries in scientific applications. Apache Storm [210] provides distributed, real-time stream processing; however using native scheduler and resource management features in particular, become bottlenecks in this framework.

D. MOBILE CLOUD COMPUTING

Mobile Cloud Computing (MCC) refers to an infrastructure where both the data storage and data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and MCC to not just smartphone users but a much broader range of mobile subscribers. The advantages of MCC include extending battery lifetime, improving data storage capacity and processing power, and improving reliability. The integration of two different fields of cloud and mobile computing introduces many technical challenges as discussed below.

1) LOW BANDWIDTH

This is a big issue in MCC as the radio resource for wireless networks is much scarce as compared to traditional wired networks. Sharing the limited bandwidth among mobile users is a common solution to this issue. Jin and Kwok [211] propose a solution to share the bandwidth among users who are located in the same area. Jung *et al.* [212] adopt a data distribution policy which determines when and how much portions of available bandwidth are shared among users from which networks, *e.g.* WiFi and WiMAX.

2) AVAILABILITY

This is another important issue in MCC as service availability is dependent to the availability of wireless networks. One solution is, instead of having a link directly to the cloud, a mobile user can connect to the cloud through neighboring nodes in an ad-hoc manner like the approach proposed in [213]. Zhang *et al.* [214] proposed a WiFi based multi-hop networking system called MoNet, which includes a distributed content sharing protocol for infrastructure-less settings.

3) HETEROGENEITY

MCC is basically used in highly heterogeneous networks in terms of wireless network interfaces. Intelligent Radio Network Access (IRNA) is an effective model to deal with the dynamics and heterogeneity of available access networks. Klein *et al.* [215] proposed an architecture, based on IRNA, to provide an intelligent network access strategy for mobile users to meet the application requirements.

4) OFFLOADING IN THE STATIC ENVIRONMENT

This is basically related to the estimation of performance parameters before the program execution, *e.g.* estimating energy consumption of the code [216]. Several solutions are proposed to find the optimal decision for partitioning applications before offloading, such as the solution proposed in [217].

5) OFFLOADING IN THE DYNAMIC ENVIRONMENT

Approaches in this category deal with offloading in a dynamic network environment; *e.g.* changing connection status and bandwidth. A common technique for dynamic offloading is application partitioning. For example, MAUI [218] uses code portability to create two versions of a mobile application, one for the local execution on devices and the other for the remote execution in the cloud. Another technique is to evaluate the circumstances of executing an application and estimating the efficiency of offloading, *e.g.* Ou *et al.* [219] take into account computations performed locally, ideal offloading without failures, and increased performance using offloading and failure recoveries.

6) ENHANCING THE EFFICIENCY OF DATA ACCESS

Handling the data resources in clouds is not a trivial problem due to the low bandwidth, mobility, and the limitation of resource capacity of mobile devices. Shen *et al.* present the E-Recall framework to address the data access issue. Approaches like [220] propose data access infrastructures to manage, search, share, and archive the rich media resources based on the coordination of mobile search, cloud computing, and multimodality integration. Another type of solutions to increase the efficiency of accessing data on the cloud is using a local storage cache, *e.g.* as done by [221].

7) CONTEXT-AWARE MOBILE CLOUD SERVICES

These types of services fulfill mobile users' needs by monitoring their preferences and provide appropriate services to each of the users. For example, Samimi *et al.* [222] propose a model, called Mobile Service Clouds (MSCs), which contains a gateway choosing an appropriate primary proxy to meet the user requirements, *e.g.* the shortest path and minimum round-trip time, and then sends the result to the user. The VOLARE middleware [223], embedded on a mobile device, is another approach that monitors the resources and contexts of the mobile device, and dynamically adjusts the requirements of the user at run-time.

Table 5 shows a summary of challenges and future directions related to the cloud application domains.

VII. MODELING AND SIMULATIONS OF CLOUD SYSTEMS

In this section, we discuss techniques, tools, and challenges related to the modeling and simulation of cloud systems and applications.

A. CLOUD-BASED APPLICATIONS MODELING

Cloud modeling has recently received the attention of the research community. Cloud modeling approaches are aimed at addressing the diversity of cloud environments by introducing a set of modeling concepts through novel domain-specific languages. Model Driven Engineering (MDE) related techniques are proposed in some recent works focusing on the models, languages, model transformations, and software processes for the model-driven development of cloud-based Software as a service (SaaS).

In addition, general-purpose modeling languages, such as Unified Modeling Language (UML), provide modeling concepts to represent software, platform, and infrastructure

TABLE 5. Application domains research challenges and future directions.

Sub-areas	Challenges and Future Directions	References
Internet of Things	• Security • Real-time data processing • Standardization • Dynamic deployment in cloud-fog • Cross-cloud IoT services • Application-oriented resource modeling	[35], [37], [177], [182]–[186]
Cyber-Physical Systems	• Need for standards • Privacy and security • Programming abstractions • Real-time requirements	[138], [192]–[197]
eScience	• Data lock-in • eScience common development infrastructures • Real-time processing of scientific data	[147]–[149], [205]–[207], [210]
Mobile Cloud Computing	• Low bandwidth, availability, heterogeneity, offloading in the static environment, offloading in the dynamic environment, enhancing the efficiency of data access, and context-aware mobile cloud services	[211]–[223]

artifacts from different viewpoints. Out of these, the deployment view is more relevant and useful for cloud-based applications as they can specify the distribution of application components on the targeted cloud computing platforms. Beyond that, contributions based on UML provide cloud-specific extensions to capture the extensive features of cloud providers at the modeling level, in addition to the generic modeling capabilities of the UML deployment language.

Blueprint [224] is an early contribution to cloud applications modeling. In this approach, applications are described as coarse-grained deployment artifacts providing a uniform representation of an application connected with the required cloud service offerings. Blueprints are described in Extensible Markup Language (XML) and typically represented in terms of a Virtual Architecture Topology (VAT). The idea is to publish such blueprints in a public repository to create a service marketplace.

Cloud Application Modeling Language (CAML) is a well-known modeling approach which enables describing cloud-based deployment topologies directly in UML and refining them with cloud offerings captured by dedicated UML profiles. In CAML, a clear separation is achieved between cloud-provider independent and cloud-provider specific deployment models. MULTICLAPP [226] proposes a UML profile in order to represent components that are expected to be deployed to a cloud platform by applying cloud-provider independent stereotypes to them. CloudML-UFPE proposes modeling concepts to represent cloud offerings connected with the internal resources of a cloud platform. The other line of research on cloud modeling is focused on the use of resources *available in the clouds*. CloudML [227] is perhaps the most well-known Domain Specific Language (DSL) in this area, proposing to define an abstraction layer used to model resources available in clouds. CloudML automatically analyses the user's resource requirements and provisions resources in clouds. CloudMF [174] leverages upon models@run-time and combines it with recent cloud solutions. It consists of a cloud modeling language and a models@run-time environment for enacting the provisioning, deployment, and adaptation of these systems.

The TOSCA aims at proposing portable cloud applications that are described in terms of so-called service templates, based on XML. Service templates in TOSCA can be operational with management plans from which operations can be called to initiate, for instance, the provisioning of applications. We discuss TOSCA in Sections III-C and V-C.

Heat Orchestration Template (HOT) [228] provides a template based orchestration for describing and running a cloud application on OpenStack [23]. A template describes the infrastructure for a cloud application specifying the relationships between resources, *e.g.* a volume and a server, and enabling creation of the infrastructure for launching the application.

The Cloud Application Modelling and Execution Language (CAMEL) is a domain-specific models@run-time DSL extending the CloudML and CloudMF enabling users to specify different aspects of multicloud applications, such as utility functions to drive the deployment and adaptation, metrics for monitoring the application and context, scalability rules for platform level scaling, providers, organizations, users, roles, and security controls.

Cloud Service Description Model (CSDM) [230] is an extension of the Unified Service Description Language (USDL). It splits service information into several modules that support different specification aspects, such as facilitating evaluation of the services with respect to the qualities based on the interactions; supporting both syntactic and semantic service description, enabling the description of various cloud services with different delivery or deployment models.

Cloud-based applications modeling is still under development and research due to the complexities in abstracting various cloud resource types, as well as the discrepancies between cloud vendors in cloud services provisioning. We list below the main future directions in this context.

1) UNIFIED MODELING TECHNIQUES

One important challenge is how to align cloud modeling languages with state-of-the-art software modeling languages which is referred as *unified modeling support*. As an example, we mentioned above that MULTICLAPP [226] proposes cloud modeling based on UML. However, it does not support refining components towards cloud services provided by a certain cloud vendor.

2) MODELING QoS REQUIREMENTS

The developers and vendors may not always be able to *model QoS requirements* at design-time in a cloud-agnostic way. For example, the cloud provider may not be able to specify the location of systems without knowing the location of the consumers. There are a few modeling frameworks that promise support of QoS aspects, such as CloudSim [231]. However, CloudSim is at the simulation level, modeling limited QoS aspects such as response time and budget for virtual machines in the cloud. Guerout *et al.* [232] proposed a cloud architecture modeling concept that includes trade-off analysis between different cloud QoS parameters, such as performance and energy-efficiency.

3) MODELING DYNAMIC ASPECTS

Molding the dynamics of cloud applications will become a crucial need for future adaptive cloud-based applications. Modeling techniques need to be extended in such a way that any run-time changes to the application should be reflected in the model. Moreover, the planned changes can be analyzed and verified using model-based verification techniques, prior to being implemented. The CAMEL modeling framework [229] has covered many aspects of dynamic application behavior, such as dynamic loading of an application form one cloud platform to another cloud platform in a multicloud setting because of performance requirements in data-intensive applications.

4) SIMULATION SUPPORT FOR DEPLOYMENT MODELS

This is another important modeling challenge in this context to make prediction about non-functional properties such as costs and performance before the actual application deployment. We discuss this issue more in detail in the next subsection.

B. SIMULATIONS

As for any other emerging computational domain, cloud computing is also full of unsolved research challenges. To make clouds more efficient, challenges such as achieving workload optimization, higher predictability of services, and energy efficiency, to name a few, must be tackled. The research undertakings often require designing new algorithms, methods, and technologies. However, as the real strength of the proposed methods and algorithms may only be shown on a very large scale infrastructure with hundreds of thousands of virtual machines in place, it is extremely costly to conduct repeated experiments for evaluation. Thus, large-scale modeling and simulation is vital for the evaluation of new research in the domain of cloud computing. Several good simulation tools are already available for modeling and simulation of workloads running on cloud computing data centers, such as CloudSim [231], CloudAnalyst [233], and GreenCloud [234]. CloudSim, in particular, has gained significant popularity and also been extended to include various new features. For instance ContainerCloudSim [235] adds support for containers in cloud data centers. NetworkCloudSim [236] extended CloudSim to include real network simulations. However, some critical challenges still remain in cloud simulation as summarized in the following.

1) SUPPORT FOR COMMUNICATION MODELS

Most of the available cloud simulation tools have limited support of simulating the communication model in the cloud [237]. Furthermore, even the simulators which provide full support for the communication model, such as NetworkCloudSim and GreenCloud are very restrictive in their support of state-of-the-art communication technologies. As new network technologies, DCN topology, storage architectures, and switching fabrics are emerging, the support of state-of-the-art technologies in communication models is very important for realistic cloud simulations. For instance, technologies based on *loss-less Ethernet*, like DCB, enable a converged unified fabric in data center environments. These are interesting topics of current research but their support in cloud communication model simulations is very limited. Moreover, as communication model simulation often requires *flit-level* simulations, the simulation efficiency and support for parallel simulations are also necessary to warrant timely results.

2) SPECIALIZED WORKLOAD SIMULATIONS

As clouds are increasingly being used to run specialized data center workloads, such as big data analytics and

TABLE 6. Modeling and simulation research challenges and future directions.

Sub-areas	Challenges and Future Directions	References
Cloud-based Applications Modeling	• Unified modeling techniques • Modeling QoS requirements • Modeling dynamic aspects • Simulation support for deployment models	[226], [229], [231], [232]
Simulations	• Support for communication models • Specialized workload simulations • Multicloud simulations • Availability of recent production traces	[234], [236]–[241]

machine learning, future simulation tools need to support such workloads for enhanced simulation results. For instance, already available simulators for MapReduce [238], such as HSim [239] and Yarn Scheduler Load Simulator (SLS) [240], can greatly improve specialized workload simulations in the cloud if they are integrated with today's cloud simulation models. It also applies to the specialized PaaS services, that are readily available from prominent cloud providers, but are hard to be added to current simulation models.

3) MULTICLOUD SIMULATIONS

From the cloud user's perspective, support of multicloud simulations, where application components are allowed to be deployed simultaneously on different cloud platforms, can be valuable to select appropriate cloud services for their applications. Even though cloud federation and hybrid cloud simulations are explored in the literature [241], multicloud simulation is still largely an unpaved territory.

4) CLOUD DATASETS AND PRODUCTION TRACES

Open cloud datasets and traces recorded from production systems are paramount to the cloud research. These traces make it possible for the researchers to evaluate newly proposed algorithms, techniques, and tools for large-scale systems without actual implementation in production. For instance, a new workload scheduling algorithm can be comprehensively tested and compared with the state-of-the-art scheduling algorithms using real-world production workload traces to assess its performance and usability in a production environment. Some of the most popular cloud datasets are specified in the following.

Google Compute Traces provides traces of workload running on Google compute cells of a 12.5k-machine cluster for about a month-long period. Google also provides execution traces for their exploratory testing architecture. *Yahoo datasets* consists of several traces including a dataset with a series of traces for the hardware resource utilization such as CPU load, memory utilization, and network traffic during the operation of Sherpa database on a production system. Another Yahoo dataset provides statistical information about file access patterns on an HDFS cluster. Several large companies and research organizations provide traces for the Hadoop workloads. *Facebook Hadoop traces* include a one day duration of historical traces on 600-machine Facebook Hadoop cluster containing around one million jobs in total. *OpenCloud Hadoop logs* are Hadoop logs containing job configuration and execution history files on a production OpenCloud cluster. *Eucalyptus dataset* contains traces of the VM start and stop events together with some anomalies added for the research and analysis. Having pointed out some available datasets, it is imperative to state that more open datasets and traces, specially from large production clusters, are needed for future cloud research. In particular, with the changing cloud workloads, it is eminent that traces from recent day executions are made available regularly.

Table 6 shows a summary of challenges and future directions related to cloud modeling and simulation.

VIII. SECURITY, PRIVACY, AND CONFIDENTIALITY

The valuable transformation of services that exploit the benefits of virtualizing and consuming IT resources in the cloud is accompanied by several security threats and significant challenges to consider [242]. Computing nodes and storage volumes that may respectively host critical applications and persist sensitive data often reside next to potentially hostile virtual environments, leaving sensitive information at risk to theft, unauthorized access, or malicious manipulation [243]. In this section, we focus on the most important security, privacy, and confidentiality challenges detected from an extensive state-of-the-art analysis.

Recently, the Cloud Security Alliance [242] revisited the list of the top security related threats and identified the most critical ones. Among the security concerns that remain high on such lists for several years, are the information disclosure and data loss due to data breaches and account hijacking or insufficient identity and access management. The attack vectors that may result in a data leakage, are inherently increased once an enterprise shifts to the cloud computing paradigm. Moreover, security issues with respect to malicious insiders of either the enterprise that uses cloud resources or even the cloud providers, constitute a constant and significant risk. Current or former employees may intentionally exceed or misuse their access privileges in a manner that can negatively affect the confidentiality, integrity, or availability of the organization's data [242]. Such concerns render cloud security realized through data protection and access control, privacy and confidentiality as top challenges to be addressed.

A. SECURITY IN THE CLOUD

While several attack vectors may be exposed on a SaaS level, mainly due to administrator's misconfigurations, the database takeover along with the post-exploitation of breached data is under the sole responsibility of the application developer [243]. Thus, the protection of the persistent layer of a modern cloud application becomes a necessity, and one of the biggest challenges to efficiently address. The application developer faces significant challenges in the cloud, since she is responsible, firstly, for sanitizing all HTTP-input parameters that could be used as attack vectors by adversaries, and secondly for guarantying that compromised data will become unusable under the existing brute-forcing and reversing techniques [243]. In addition, the mere utilization of IaaS or PaaS providers in order to host or develop a cloud application, may by itself spawn a multitude of inherent vulnerabilities that cannot be tackled effectively without appropriate and transparent mitigation and protection mechanisms, *e.g.* with respect to Distributed Denial of Service (DDoS) attacks). One of the most powerful tools for alleviating such concerns is the development and enforcement of efficient and dynamic access control mechanisms that should be capable of managing all authorization decisions without neglecting to consider the inherent attack vectors that may be met at any level of the cloud stack. Below, we discuss the most significant security challenges.

1) DYNAMIC ACCESS CONTROL IN THE CLOUD

Out of the basic access control models [244], namely Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC) and Attribute Based Access Control (ABAC), only the latter two are considered flexible enough to cope with increased security challenges posed by cloud applications. Furthermore, a clear challenge has been defined in most advanced RBAC [245]–[247] and ABAC efforts [248]–[251] which is to fuse with context-awareness any access control decision that may permit the manipulation of sensitive data, persisted on cloud resources. Specifically, this refers to the need for efficient and flexible access control approaches capable of taking into account a number of contextual parameters that characterize data access requests in the cloud and fusing with advanced security policies, cloud applications in order to restrict access to sensitive data. This still remains only a partially addressed challenge even by the most recent research efforts [243]. It is still partially addressed mainly because there is an implication in implementing a well advanced context-aware authorization engine. The implication comes from the fact that numerous software handlers need to be developed as well, for feeding the policy decision points (PDPs) with raw (*e.g.* 37.9838 °N, 23.7275 °E) or higher level context (*e.g.* Athens, Greece) that should not be corrupted or disputed. In this regard, the most promising approaches will continue towards alleviating the security concerns associated with the adoption of cloud computing by introducing innovative security-by-design frameworks. This will facilitate infusing appropriate context-aware access control policies into cloud applications.

2) DDoS MITIGATION IN THE CLOUD

DDoS attacks is a class of perimeter security attacks that are often launched by a remotely controlled network of botnets sending malformed packets or service requests in order to flood sensitive systems. Their utmost goal is to exhaust the network bandwidth and server resources intended for legitimate users [252]. The implications of such cybersecurity attacks are severe for cloud-based applications since they deteriorate or even completely interrupt any service provisioning of the target system, but they may even affect any co-hosted applications in multi-tenant scenarios by flooding their virtualized and physical resources. Therefore, intensive research efforts towards mitigation approaches in several levels of the cloud stack are currently present in the literature. Specifically, there have been several noteworthy efforts on DDoS mitigation that employ traditional firewall tactics such as IP trace back, anomaly detection, ingress and egress filtering, network self-similarity, etc. [253]–[256]. These approaches, however, present significant limitations [253], especially in the domain of cloud computing, where vendor agnostic and software defined solutions should be put in place for being able to cope with both single and multiple IaaS provider scenarios. Nowadays, many research efforts clearly indicate a strong focus and a promising potential on the emerging SDN paradigm in addressing DDoS flooding attacks. In addition, there is also a clear uptake of lower level approaches that are promising especially for the fog computing domain. These are built on technologies like the in-kernel packet filter known as Berkeley Packet Filter (BPF) and the extended Berkeley Packet Filter (eBPF) [257] that are proposed by researchers for improving cloud security.

B. CONFIDENTIALITY AND PRIVACY IN MULTICLOUDS

As mentioned before, it is evident that the most critical part of a modern cloud application is the data persistency layer and the database itself [242]. As all sensitive information resides in this layer, the database-takeover constitutes the ultimate goal for every external or internal adversary and the utmost fear of any data owner that uses cloud resources. Thus, additional major challenges are the confidentiality and privacy concerns that dictate the need to enforce safe-guarding mechanism for protecting users' records that may reside on cloud resources. To this end, any cloud-based deployment must first ensure that especially the sensitive data is stored in an encrypted form. Nevertheless, the cryptographic protection of sensitive data still remains a very active and challenging

TABLE 7. Security and privacy research challenges and future directions.

Sub-areas	Challenges and Future Directions	References
Security in the Cloud	• Dynamic access control in the cloud • DDoS mitigation mechanisms in the cloud	[243]–[253], [253]–[257]
Confidentiality and Privacy in Multiclouds	• Functionality-preserving encryption in multiclouds • Secure key management in multiclouds • Fragmentation and distribution of sensitive data in multiclouds	[242], [243], [258]–[268]

research domain, since the use of efficient cryptographic algorithms present certain security trade-offs. Data encryption offers indeed robust security, but at the cost of reducing the efficiency of the service and limiting the functionality that can be applied over the encrypted data stored on cloud premises [258]. For example, in a data leakage incident, the post-exploitation risk, can be increased in cases where a simple symmetric encryption algorithm has been employed for protecting the cloud application data [243]. Several modern cracking tool kits like oclHashcat that utilize Graphics Processing Unit (GPU) power are able to crack ciphers using brute-force techniques with an attack rate that may reach billion attempts per second. Based on this, three concrete challenges are formed: functionality-preserving encryption, secure key management, and fragmentation and distribution of sensitive data in multiclouds, that are further discussed below.

1) FUNCTIONALITY-PRESERVING ENCRYPTION IN MULTICLOUDS

There is a new wave of functionality-preserving algorithms [259], [260] that have recently been emerged attempting to provide a better balance between confidentiality and usability. This can be valuable especially in the context of cloud applications. Specifically, these new approaches do not use probabilistic encryption for sensitive data, even though it may be highly secure because of the ability to prevent statistical attacks. Instead, they propose searchable encryption schemes [261] and their variations like Format [262] and Order Preserving Encryption [263] for increasing the efficiency and speed of the respective querying and data exploitation. The reason is that the probabilistic encryption constitutes data completely unusable, obfuscating it to both adversaries and legitimate users. Thus, any kind of processing by a legitimate user is impossible without first decrypting all the data (*e.g.* finding the average values out of several entries in a database). These approaches [261]–[263] aim to efficiently address this issue by allowing legitimate users to receive the result of a function without being exposed to any details of the individual data artefacts used for calculating the function output. The most recent approaches [264], [265] take a step further in order to ease the security compromises that take place against efficiency, by creating separate encrypted indexes for the data, whose functionality needs to be preserved. Nevertheless, much more work is needed in order to overcome domain specific constraints and additionally cope with the distributed nature of unstructured data persisted in multiclouds [258].

2) SECURE KEY MANAGEMENT IN MULTICLOUDS

This challenge is also important for ensuring the design and development of appropriate mechanisms in order to assure that encryption keys cannot be revealed to malicious users. Any cryptographic keys used, must not be embedded in source code or be distributed in an unprotected manner, since there is a significant chance of discovery and misuse. Keys need to be appropriately secured through a public key infrastructure (PKI) that ensures safe key creation, propagation and revocation control [242]. This still remains a challenge in the cloud computing domain where the honest-but-curious adversarial model is usually considered [266] for interacting entities, *e.g.* cloud providers. To alleviate this challenge there is a clear direction of work towards distributed key management approaches [267].

3) FRAGMENTATION AND DISTRIBUTION OF SENSITIVE DATA IN MULTICLOUDS

There are privacy issues which stem from the plethora of available automated exploitation tools, like SQLMap, and the widely spread sophisticated techniques that try to evade intrusion detection systems (IDS) and intrusion prevention systems (IPS) [268]. The existence of such techniques highlights that the risk of database compromise is greater than ever before. A promising direction of work with respect to addressing such privacy concerns constitutes novel approaches that focus on fragmentation and distribution of sensitive data artefacts in a way that, even if the encryption key is somehow intercepted by an adversary, the sensitive information is still protected [243]. Nevertheless, several advancements over the current state-of-the art are still required in order to deal with unstructured data that may have already been distributed over multiclouds for processing efficiency and fail-over purposes.

Table 7 shows a summary of challenges and future directions related to cloud security, privacy and confidentiality.

TABLE 8. Interoperability research challenges and future directions.

Challenges and Future Directions	References
• SaaS applications interoperability • Portability of applications built for PaaS platforms	[271]–[275]

IX. INTEROPERABILITY AND PORTABILITY

The spectrum of cloud computing products and services is very diverse ranging from IaaS, to PaaS, and SaaS, in addition to the recent novel and more specific services such as Big Data as a Service. The variety of cloud services and platforms has led to heterogeneous and vendor-specific cloud architectures and technologies, increasing the risk of vendor lock-in for customers. Vendor lock-in causes a user being tied to a particular cloud service provider due to the technical difficulties and costs of migrating to equivalent cloud services from other providers. To address this concern, the portability and interoperability of cloud services should be carefully considered by cloud vendors and application developers.

Interoperability, in cloud computing, is defined as *the capability of public clouds, private clouds, and other software systems hosted within the enterprise to communicate each other, and understand service interfaces, configuration, forms of authentication and authorization, data formats, etc. in order to cooperate and interoperate with each other.* The most common type of communication happens between components on the cloud service customer side and the components of the cloud service provider on the other side. The highest degree of interoperability is that the interfaces are standardized so that the customer can switch to another cloud service provider with minimal impact on the customer's components.

Portability in cloud computing is threefold: cloud *data* portability, *system* portability and *application* portability. The former refers to ability to transfer data from one cloud service to another cloud service, without having to re-enter the data [269]. This includes portability of the syntax and semantics of the transferred data. System portability represents the ability to migrate virtual machine instances, machine images, applications or even services, and their relative contents from one cloud provider to another [270]. Application portability indicates the ability to transfer an application or application components from one cloud service to a equivalent cloud service and run the application in the target cloud service, without having to make significant changes to the application code.

There are many interoperability challenges in cloud computing, raising from the lack of standardized interfaces and API. There are many cloud standardization initiatives. Some of them focus on standardizing parts of a cloud computing service such as authentication and data access. The other type aims to standardize how different elements of a cloud service should work together as a solution. Nevertheless, cloud standards are in development, both for IaaS and PaaS offerings. While PaaS cloud services have lower levels of interoperability, the greatest level of interoperability is found for IaaS cloud services, where functionality is rather equivalent and there are a number of standard interfaces, such as Cloud Data Management Interface (CDMI) [271]. Leading cloud vendors are greatly influencing the development of new standards, even imposing their own standards in the market. The European Telecommunications Standards Institute has created a cloud group to consider cloud standardization needs and conformity with interoperability standards. The Cloud Standards Customer Council (CSCC) is dedicated to accelerating the successful adoption of cloud computing and identifying related standards. The Open Cloud Computing Interface (OCCI) is a REST based protocol and API, published by the Open Grid Forum (OGF) to define standards for a shareable and homogeneous interface to support all kinds of management tasks in the cloud environment [272]. In spite of the above initiatives, there are still open challenges in cloud interoperability and portability, presented below and summarized in Table 8.

4) SaaS APPLICATIONS INTEROPERABILITY

This issue presents the greatest challenge in this context. There are very few standard APIs for SaaS applications and switching from one SaaS application to another SaaS application with comparable functionality may require interface changes [271], [272]. Interface mapping layers and Enterprise Service Bus (ESB) are the attempts to address this issue, like the interoperable cloud-computing-based platform [273] for the management of administrative processes of public administrations. However, more generic approaches for SaaS interoperability are missing.

5) PORTABILITY OF APPLICATIONS BUILT FOR PaaS PLATFORMS

The differences between PaaS platforms can lead to heavy re-engineering of customer code when the code is moved between those platforms. Common open source PaaS platforms such as Cloud Foundry and containerization technologies [274] such as Docker (allowing subdivision and independent deployment of parts of an application) are two promising approaches in this context. Besides these, there are solutions that promise semantically interconnect heterogeneous PaaS offerings across different cloud providers, *e.g.* when they share the same technology [275].

X. CONCLUSIONS

Cloud computing is about to realize the dream of computing as a utility. It is widely used by small- and large-scale IT services providers in order to make software and hardware services delivery less costly, and more secure, more reliable, and more scalable. Despite the significant development in cloud computing, the current technologies are not yet mature enough to realize fully the potential of true utility computing. Many key solutions in this domain are still in their infancy, such as automatic resource provisioning, cross-cloud services, novel fog- and IoT-based cloud services, and cloud modeling. This implies that there is still tremendous opportunities for researchers to make fundamental contributions in this field, and make significant impact on the advancements of cloud computing.

In this paper, we have provided a survey on state-of-the-art solutions for various cloud research areas and a broader understanding of the design challenges of cloud computing. Our analysis has identified the potential of future research directions for cloud-based systems. Tables 1-8 summarize the areas we studied in this paper, including their associated sub-areas. For each sub-area of cloud research, those tables list the topics that should be further researched with the potential of high impact results in the future.

REFERENCES

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generat. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.

[2] W. Voorsluys, J. Broberg, and R. Buyya, *Introduction to Cloud Computing*. Hoboken, NJ, USA: Wiley, 2011, pp. 1–41.

[3] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. Inácio, "Security issues in cloud environments: A survey," *Int. J. Inf. Secur.*, vol. 13, no. 2, pp. 113–170, 2014.

[4] S.-Y. Jing, S. Ali, K. She, and Y. Zhong, "State-of-the-art research study for green cloud computing," *J. Supercomput.*, vol. 65, no. 1, pp. 445–468, Jul. 2013.

[5] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," in *Proc. 6th Int. Conf. Semantics Knowl. Grid (SKG)*, Nov. 2010, pp. 105–112.

[6] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Gener. Comput. Syst.*, vol. 79, pp. 849–861, Feb. 2018.

[7] R. Buyya *et al.*. (2018). "A manifesto for future generation cloud computing: Research directions for the next decade." [Online]. Available: https://arxiv.org/abs/1711.09123

[8] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2009.

[9] T. Chen, X. Gao, and G. Chen, "The features, hardware, and architectures of data center networks: A survey," *J. Parallel Distrib. Comput.*, vol. 96, pp. 45–74, Oct. 2016.

[10] D. Comer, R. H. Karandikar, and A. Rastegarnia, "DCnet: A new data center network architecture," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 1–6.

[11] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.

[12] Y. Liu, J. K. Muppala, and M. Veeraraghavan. (2013). *A Survey of Data Center Network Architectures*. [Online]. Available: https://www.semanticscholar.org/paper/A-Survey-of-Data-Center-Network-Architectures-Liu-Muppala/676bf0c711107389f0452553ed0c3c59921db4e5

[13] R. N. Mysore *et al.*, "PortLand: A scalable fault-tolerant layer 2 data center network fabric," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 39–50, 2009.

[14] A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Towards a next generation data center architecture: Scalability and commoditization," in *Proc. ACM Workshop Program. Routers Extensible Services Tomorrow*, 2008, pp. 57–62.

[15] A. Greenberg *et al.*, "VL2: A scalable and flexible data center network," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 51–62, 2009.

[16] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: A scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 75–86, 2008.

[17] C. Guo *et al.*, "BCube: A high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 63–74, 2009.

[18] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu, "FiConn: Using backup port for server interconnection in data centers," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2276–2285.

[19] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. NSDI*, 2010, p. 19.

[20] N. Farrington *et al.*, "Helios: A hybrid electrical/optical switch architecture for modular data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 339–350, 2010.

[21] B. Wang, Z. Qi, R. Ma, H. Guan, and A. V. Vasilakos, "A survey on data center networking for cloud computing," *Comput. Netw.*, vol. 91, pp. 528–547, Nov. 2015.

[22] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," in *Proc. Int. Conf. Cloud Comput.*, 2011, pp. 32–38.

[23] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "OpenStack: Toward an open-source solution for cloud computing," *Int. J. Comput. Appl.*, vol. 55, no. 3, pp. 38–42, 2012.

[24] D. Milojičić, I. M. Llorente, and R. S. Montero, "OpenNebula: A cloud management tool," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 11–14, Mar./Apr. 2011.

[25] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems," in *Proc. IEEE 6th Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2013, pp. 887–894.

[26] E. Di Nitto *et al.*, "Supporting the development and operation of multi-cloud applications: The modaclouds approach," in *Proc. 15th Int. Symp. Symbolic Numeric Algorithms Sci. Comput. (SYNASC)*, Sep. 2013, pp. 417–423.

[27] G. Sousa, W. Rudametkin, and L. Duchien, "Automated setup of multi-cloud environments for microservices applications," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2016, pp. 327–334.

[28] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proc. ASE BigData SocialInform. (ASE BD&SI)*, 2015, pp. 28:1–28:6.

[29] I. Ku, Y. Lu, and M. Gerla, "Software-defined mobile cloud: Architecture, services and use cases," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2014, pp. 1–6.

[30] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[31] V. Cardellini, V. Grassi, F. L. Presti, and M. Nardelli, "On QoS-aware scheduling of data stream applications over fog computing infrastructures," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 271–276.

[32] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[33] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the Internet of Things," in *Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput. (MCC)*, 2013, pp. 15–20.

[34] H. Shi, N. Chen, and R. Deters, "Combining mobile and fog computing: Using coap to link mobile device clouds with fog computing," in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, Dec. 2015, pp. 564–571.

[35] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "*FogFlow*: Easy programming of IoT services over cloud and edges for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 696–707, Apr. 2018.

[36] E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwälder, "Incremental deployment and migration of geo-distributed situation awareness applications in the fog," in *Proc. 10th ACM Int. Conf. Distrib. Event-Based Syst. (DEBS)*, 2016, pp. 258–269.

[37] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing IoT applications in the fog: A distributed dataflow approach," in *Proc. 5th Int. Conf. Internet Things (IOT)*, Oct. 2015, pp. 155–162.

[38] D. Nguyen, Z. Shen, J. Jin, and A. Tagami, "ICN-fog: An information-centric fog-to-fog architecture for data communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[39] Z. Hao, E. Novak, S. Yi, and Q. Li, "Challenges and software architecture for fog computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 44–53, Mar./Apr. 2017.

[40] C. Chang, S. N. Srirama, and R. Buyya, "Indie fog: An efficient fog-computing infrastructure for the Internet of Things," *Computer*, vol. 50, no. 9, pp. 92–98, 2017.

[41] Z. Rejiba, X. Masip-Bruin, A. Jurnet, E. Marin-Tordera, and G. Ren, "F2C-aware: Enabling discovery in Wi-Fi-powered fog-to-cloud (F2C) systems," in *Proc. 6th IEEE Int. Conf. Mobile Cloud Comput.*, Mar. 2018, pp. 113–116.

[42] P. P. Jayaraman, J. B. Gomes, H. L. Nguyen, Z. S. Abdallah, S. Krishnaswamy, and A. Zaslavsky, "CARDAP: A scalable energy-efficient context aware distributed mobile data analytics platform for the fog," in *Advances in Databases and Information Systems*, Y. Manolopoulos, G. Trajcevski, M. Kon-Popovska, Eds. Cham, Switzerland: Springer, 2014, pp. 192–206.

[43] R. R. Expósito, G. L. Taboada, S. Ramos, J. Touriño, and R. Doallo, "Performance analysis of HPC applications in the cloud," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 218–229, 2013.

[44] V. Mauch, M. Kunze, and M. Hillenbrand, "High performance cloud computing," *Future Gener. Comput. Syst.*, vol. 29, no. 6, pp. 1408–1416, 2013.

[45] J. Zhang, X. Lu, M. Arnold, and D. K. Panda, "MVAPICH2 over openstack with SR-IOV: An efficient approach to build HPC clouds," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 71–80.

[46] D. Géhberger, D. Balla, M. Maliosz, and C. Simon, "Performance evaluation of low latency communication alternatives in a containerized cloud environment," in *Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2018, pp. 9–16.

[47] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, no. 5, pp. 862–876, Apr. 2010.

[48] M. F. Bari *et al.*, "Data center network virtualization: A survey," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 2, pp. 909–928, 2nd Quart., 2013.

[49] T. Koponen *et al.*, "Network virtualization in multi-tenant datacenters," in *Proc. 11th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2014, pp. 203–216.

[50] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 242–253, Aug. 2011.

[51] H. Rodrigues, J. R. Santos, Y. Turner, P. Soares, and D. Guedes, "Gatekeeper: Supporting bandwidth guarantees for multi-tenant datacenter networks," in *Proc. WIOV*, 2011, p. 6.

[52] C. Guo *et al.*, "SecondNet: A data center network virtualization architecture with bandwidth guarantees," in *Proc. 6th Int. Conf. Co-NEXT*, 2010, Art. no. 15.

[53] A. Shieh, S. Kandula, A. G. Greenberg, and C. Kim, "Seawall: Performance isolation for cloud datacenter networks," in *Proc. HotCloud*, 2010, p. 1.

[54] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[55] L. Yang, R. Dantu, T. Anderson, and R. Gopal, "Forwarding and control element separation (ForCES) framework," Internet Eng. Task Force, Fremont, CA, USA, Tech. Rep. 3746, 2004.

[56] D. Erickson, "The beacon openflow controller," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp. 13–18.

[57] E. Ng, "Maestro: A system for scalable openflow control," Rice Univ., Houston, TX, USA, Tech. Rep. TR10-11, 2010.

[58] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load balancing mechanisms in the software defined networks: A systematic and comprehensive review of the literature," *IEEE Access*, vol. 6, pp. 14159–14178, 2018.

[59] *OpenNaaS: Open Platform for Network-as-a-Service*. Accessed: Nov. 2, 2017. [Online]. Availabe: http://opennaas.org/

[60] T. Benson, A. Akella, A. Shaikh, and S. Sahu, "CloudNaaS: A cloud networking platform for enterprise applications," in *Proc. 2nd ACM Symp. Cloud Comput.*, 2011, Art. no. 8.

[61] T. McGuire, J. Manyika, and M. Chui, "Why big data is the new competitive advantage," *Ivey Bus. J.*, vol. 76, no. 4, pp. 1–4, 2012.

[62] *Amazon Elastic MapReduce*. Accessed: Nov. 30, 2016. [Online]. Availabe: https://aws.amazon.com/emr/

[63] *IBM BigInsights for Apache Hadoop*. Accessed: Nov. 30, 2016. [Online]. Availabe: http://www-03.ibm.com/software/products/en/ibm-biginsights-for-apache-hadoop

[64] *Microsoft Azure HDInsight*. Accessed: Nov. 30, 2016. [Online]. Availabe: https://azure.microsoft.com/en-us/services/hdinsight/

[65] *Apache Hadoop*. Accessed: Nov. 20, 2018. [Online]. Availabe: https://hadoop.apache.org/

[66] *Apache Spark: Lightning-Fast Unified Analytics Engine*. Accessed: Nov. 20, 2018. [Online]. Availabe: https://spark.apache.org/

[67] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost optimization for dynamic replication and migration of data in cloud data centers," *IEEE Trans. Cloud Comput.*, to be published.

[68] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proc. HotCloud*, 2009, Art. no. 3.

[69] T. Sterling and D. Stark, "A high-performance computing forecast: Partly cloudy," *Comput. Sci. Eng.*, vol. 11, no. 4, pp. 42–49, Jul. 2009.

[70] Q. He, S. Zhou, B. Kobler, D. Duffy, and T. McGlynn, "Case study for running HPC applications in public clouds," in *Proc. 19th ACM Int. Symp. High Perform. Distrib. Comput.*, 2010, pp. 395–401.

[71] A. Gupta and D. Milojicic, "Evaluation of HPC applications on cloud," in *Proc. 6th Open Cirrus Summit (OCS)*, 2011, pp. 22–26.

[72] P. Mehrotra *et al.*, "Performance evaluation of Amazon EC2 for NASA HPC applications," in *Proc. 3rd Workshop Sci. Cloud Comput.*, 2012, pp. 41–50.

[73] K. R. Jackson *et al.*, "Performance analysis of high performance computing applications on the Amazon Web services cloud," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Nov./Dec. 2010, pp. 159–168.

[74] *Intersect360 Research*. Accessed: Jul. 1, 2017. [Online]. Availabe: http://www.intersect360.com/

[75] (2015). *InfiniBand Architecture Specification: Release 1.3*. [Online]. Available: http://www.infinibandta.com/

[76] M. Hillenbrand, V. Mauch, J. Stoess, K. Miller, and F. Bellosa, "Virtual InfiniBand clusters for HPC clouds," in *Proc. 2nd Int. Workshop Cloud Comput. Platforms*, 2012, p. 9.

[77] P. Rad, R. V. Boppana, P. Lama, G. Berman, and M. Jamshidi, "Low-latency software defined network for high performance clouds," in *Proc. 10th Syst. Syst. Eng. Conf. (SoSE)*, May 2015, pp. 486–491.

[78] P. Barham *et al.*, "Xen and the art of virtualization," *ACM SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 164–177, Dec. 2003.

[79] A. K. Qumranet, Y. K. Qumranet, D. L. Qumranet, U. L. Qumranet, and A. Liguori, "KVM: The Linux virtual machine monitor," in *Proc. Linux Symp.*, vol. 1, 2007, pp. 225–230.

[80] U. Drepper, "The cost of virtualization," *Queue*, vol. 6, no. 1, pp. 28–35, 2008.

[81] *The Linux Containers*. Accessed: Nov. 30, 2016. [Online]. Availabe: https://linuxcontainers.org/

[82] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[83] H. Takabi, J. B. D. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security Privacy*, vol. 8, no. 6, pp. 24–31, Nov./Dec. 2010.

[84] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan./Feb. 2012.

[85] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Apr. 2010, pp. 27–33.

[86] C. J. Guo, W. Sun, Y. Huang, Z. H. Wang, and B. Gao, "A framework for native multi-tenancy application development and management," in *Proc. 9th IEEE Int. Conf. E-Commerce Technol.*, Jul. 2007, pp. 551–558.

[87] A. Gupta and D. Milojicic, "Evaluation of HPC applications on cloud," in *Proc. 6th Open Cirrus Summit (OCS)*, Oct. 2011, pp. 22–26.

[88] P. Bientinesi, R. Iakymchuk, and J. Napper, "HPC on competitive cloud resources," in *Handbook of Cloud Computing*. Boston, MA, USA: Springer, 2010, pp. 493–516.

[89] A. Iosup, N. Yigitbasi, and D. Epema, "On the performance variability of production cloud services," in *Proc. 11th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGrid)*, May 2011, pp. 104–113.

[90] S. Radhakrishnan, R. Pan, A. Vahdat, and G. Varghese, "Netshare and stochastic netshare: Predictable bandwidth allocation for data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 5–11, 2012.

[91] V. Jeyakumar, M. Alizadeh, D. Mazières, B. Prabhakar, C. Kim, and A. Greenberg, "EyeQ: Practical network performance isolation at the edge," in *Proc. 10th USENIX Conf. Netw. Syst. Design Implement.*, 2013, pp. 297–312.

[92] B. P. Rimal and M. Maier, "Workflow scheduling in multi-tenant cloud computing environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 1, pp. 290–304, Jan. 2017.

[93] F. Zahid, E. G. Gran, B. Bogdański, B. D. Johnsen, and T. Skeie, "Efficient network isolation and load balancing in multi-tenant HPC clusters," *Future Gener. Comput. Syst.*, vol. 72, pp. 145–162, Jul. 2017.

[94] J. Kirschnick, J. M. A. Calero, L. Wilcock, and N. Edwards, "Toward an architecture for the automated provisioning of cloud services," *IEEE Commun. Mag.*, vol. 48, no. 12, pp. 124–131, Dec. 2010.

[95] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, *TOSCA: Portable Automated Deployment and Management of Cloud Applications*. New York, NY, USA: Springer, 2014.

[96] J. Wettinger, U. Breitenbücher, and F. Leymann, "Standards-based DevOps automation and integration using TOSCA," in *Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2014, pp. 59–68.

[97] A. Naseri and N. J. Navimipour, "A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm," *J. Ambient Intell. Humanized Comput.*, to be published.

[98] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Comput.*, vol. 13, no. 5, pp. 14–22, Oct. 2009.

[99] B. Radojević and M. Žagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments," in *Proc. 34th Int. Conv. MIPRO*, May 2011, pp. 416–420.

[100] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *Proc. 3rd Int. Symp. Parallel Archit., Algorithms Program.*, 2010, pp. 89–96.

[101] T. Kokilavani and D. G. Amalarethinam, "Load balanced min-min algorithm for static meta-task scheduling in grid computing," *Int. J. Comput. Appl.*, vol. 20, no. 3, pp. 42–48, 2011.

[102] X. Ren, R. Lin, and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast," in *Proc. IEEE Int. Conf. Cloud Comput. Intell. Syst.*, Sep. 2011, pp. 220–224.

[103] L. D. D. Babu and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2292–2303, May 2013.

[104] K. Nishant *et al.*, "Load balancing of nodes in cloud using ant colony optimization," in *Proc. 14th Int. Conf. Comput. Modelling Simulation (UKSim)*, Mar. 2012, pp. 3–8.

[105] A. Paya and D. C. Marinescu, "Energy-aware load balancing and application scaling for the cloud ecosystem," *IEEE Trans. Cloud Comput.*, vol. 5, no. 1, pp. 15–27, Mar. 2017.

[106] Z. Zhou *et al.*, "Carbon-aware load balancing for geo-distributed cloud services," in *Proc. IEEE 21st Int. Symp. Modelling, Anal. Simulation Comput. Telecommun. Syst.*, Aug. 2013, pp. 232–241.

[107] S. Liao, H. Zhang, G. Shu, and J. Li, "Adaptive resource prediction in the cloud using linear stacking model," in *Proc. 5th Int. Conf. Adv. Cloud Big Data (CBD)*, Aug. 2017, pp. 33–38.

[108] A. Bala and I. Chana, "Prediction-based proactive load balancing approach through VM migration," *Eng. Comput.*, vol. 32, no. 4, pp. 581–592, 2016.

[109] W.-T. Wen, C.-D. Wang, D.-S. Wu, and Y.-Y. Xie, "An ACO-based scheduling strategy on load balancing in cloud computing environment," in *Proc. 9th Int. Conf. Frontier Comput. Sci. Technol.*, Aug. 2015, pp. 364–369.

[110] K.-M. Cho, P.-W. Tsai, C.-W. Tsai, and C.-S. Yang, "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing," *Neural Comput. Appl.*, vol. 26, no. 6, pp. 1297–1309, 2015.

[111] S. Pang, W. Zhang, T. Ma, and Q. Gao, "Ant colony optimization algorithm to dynamic energy management in cloud data center," *Math. Problems Eng.*, vol. 2017, Dec. 2017, Art. no. 4810514.

[112] N. Oreskes, "The scientific consensus on climate change," *Science*, vol. 306, no. 5702, p. 1686, 2004.

[113] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[114] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," *ACM SIGARCH Comput. Archit. News*, vol. 38, pp. 338–347, Jun. 2010.

[115] J. Shuja, S. A. Madani, K. Bilal, K. Hayat, S. U. Khan, and S. Sarwar, "Energy-efficient data centers," *Computing*, vol. 94, no. 12, pp. 973–994, 2012.

[116] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos, "Cloud computing: Survey on energy efficiency," *ACM Comput. Surv.*, vol. 47, no. 2, 2015, Art. no. 33.

[117] Y. Sharma, B. Javadi, W. Si, and D. Sun, "Reliability and energy efficiency in cloud computing systems: Survey and taxonomy," *J. Netw. Comput. Appl.*, vol. 74, pp. 66–85, Oct. 2016.

[118] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proc. 10th IEEE/ACM Int. Conf. Cluster, Cloud Grid Comput.*, May 2010, pp. 826–831.

[119] C. Ghribi, M. Hadji, and D. Zeghlache, "Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms," in *Proc. 13th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGrid)*, May 2013, pp. 671–678.

[120] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Proc. IEEE 5th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2012, pp. 750–757.

[121] M. E. Haque, K. Le, Í. Goiri, R. Bianchini, and T. D. Nguyen, "Providing green SLAs in high performance computing clouds," in *Proc. Int. Green Comput. Conf. (IGCC)*, Jun. 2013, pp. 1–11.

[122] Í. Goiri *et al.*, "Greenslot: Scheduling energy consumption in green datacenters," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2011, p. 20.

[123] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing, "Utilizing green energy prediction to schedule mixed batch and service jobs in data centers," *ACM SIGOPS Operating Syst. Rev.*, vol. 45, no. 3, pp. 53–57, 2012.

[124] A. Khosravi and R. Buyya, "Energy and carbon footprint-aware management of geo-distributed cloud data centers: A taxonomy, state of the art, and future directions," in *Advancing Cloud Database Systems and Capacity Planning With Dynamic Applications*. Hershey, PA, USA: IGI Publishing, 2017, p. 27.

[125] A. Hameed *et al.*, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, vol. 98, no. 7, pp. 751–774, Jul. 2016.

[126] F. Zahn, P. Yebenes, S. Lammel, P. J. Garcia, and H. Fröning, "Analyzing the energy (dis-) proportionality of scalable interconnection networks," in *Proc. 2nd IEEE Int. Workshop High-Perform. Interconnection Netw. Exascale Big-Data Era (HiPINEB)*, Mar. 2016, pp. 25–32.

[127] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 223–244, 2nd Quart., 2011.

[128] A. Berl *et al.*, "Energy-efficient cloud computing," *Comput. J.*, vol. 53, no. 7, pp. 1045–1051, 2010.

[129] C. Chapman, M. Musolesi, W. Emmerich, and C. Mascolo, "Predictive resource scheduling in computational grids," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, Mar. 2007, pp. 1–10.

[130] C. K. Chui and G. Chen, *Kalman Filtering*. Cham, Switzerland: Springer, 2017.

[131] Y. Gao, H. Rong, and J. Z. Huang, "Adaptive grid job scheduling with genetic algorithms," *Future Gener. Comput. Syst.*, vol. 21, no. 1, pp. 151–161, 2005.

[132] Z. Xu, X. Hou, and J. Sun, "Ant algorithm-based task scheduling in grid computing," in *Proc. Can. Conf. Elect. Comput. Eng. (IEEE CCECE)*, vol. 2, May 2003, pp. 1107–1110.

[133] R.-S. Chang, J.-S. Chang, and P.-S. Lin, "An ant algorithm for balanced job scheduling in grids," *Future Gener. Comput. Syst.*, vol. 25, no. 1, pp. 20–27, 2009.

[134] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generat. Comput. Syst.*, vol. 28, no. 1, pp. 155–162, 2012.

[135] Z. Gong, X. Gu, and J. Wilkes, "PRESS: Predictive elastic resource scaling for cloud systems," in *Proc. CNSM*, Oct. 2010, pp. 9–16.

[136] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2011, pp. 500–507.

[137] S. Ghemawat, H. Gobioff, and S. Leung, "The Google file system," *ACM SIGOPS Operating Syst. Rev.*, vol. 37, no. 5, pp. 29–43, 2003.

[138] D. Borthakur *et al. HDFS Architecture Guide, Hadoop Apache Project 53*. Accessed: Apr. 2018. [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[139] S. Sakr, A. Liu, D. M. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 311–336, 3rd Quart., 2011.

[140] D. J. Abadi, "Data management in the cloud: Limitations and opportunities," *IEEE Data Eng. Bull.*, vol. 32, no. 1, pp. 3–12, Mar. 2009.

[141] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant Web services," *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, 2002.

[142] D. Pritchett, "BASE: An ACID alternative," *Queue*, vol. 6, no. 3, pp. 48–55, 2008.

[143] S. Das, D. Agrawal, and A. El Abbadi, "ElasTraS: An elastic, scalable, and self-managing transactional database for the cloud," *ACM Trans. Database Syst.*, vol. 38, no. 1, 2013, Art. no. 5.

[144] F. Chang *et al.*, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, 2008, Art. no. 4.

[145] M. N. Vora, "Hadoop-HBase for large-scale data," in *Proc. Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, vol. 1, Dec. 2011, pp. 601–605.

[146] J. McKendrick, "Cloud computing's vendor lock-in problem: Why the industry is taking a step backward," Forbes, New York, NY, USA. Accessed: Aug. 2018.

[147] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated data placement for geo-distributed cloud services," in *Proc. 7th USENIX Conf. Networked Syst. Design Implement.*, 2010, p. 2.

[148] G. Zhang, L. Chiu, and L. Liu, "Adaptive data migration in multi-tiered storage based cloud environment," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2010, pp. 148–155.

[149] S. Ortiz, "The problem with cloud-computing standardization," *Computer*, vol. 44, no. 7, pp. 13–16, Jul. 2011.

[150] H. Jagadish *et al.*, "Big data and its technical challenges," *Commun. ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[151] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big data: Issues and challenges moving forward," in *Proc. 46th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2013, pp. 995–1004.

[152] D. Yuan, Y. Yang, X. Liu, and J. Chen, "A data placement strategy in scientific cloud workflows," *Future Gener. Comput. Syst.*, vol. 26, no. 8, pp. 1200–1214, 2010.

[153] T. Wang, S. Yao, Z. Xu, and S. Jia, "DCCP: An effective data placement strategy for data-intensive computations in distributed cloud computing systems," *J. Supercomput.*, vol. 72, no. 7, pp. 2537–2564, Jul. 2016.

[154] L. Wang *et al.*, "G-Hadoop: MapReduce across distributed data centers for data-intensive computing," *Future Generat. Comput. Syst.*, vol. 29, no. 3, pp. 739–750, Mar. 2013.

[155] K. Wang, X. Zhou, T. Li, D. Zhao, M. Lang, and I. Raicu, "Optimizing load balancing and data-locality with data-aware scheduling," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2014, pp. 119–128.

[156] P. Kondikoppa, C.-H. Chiu, C. Cui, L. Xue, and S.-J. Park, "Network-aware scheduling of mapreduce framework ondistributed clusters over high speed networks," in *Proc. Workshop Cloud Services, Fed., 8th Open Cirrus Summit*, 2012, pp. 39–44.

[157] C. Vecchiola, X. Chu, and R. Buyya, "Aneka: A software platform for.NET based cloud computing," in *Proc. High Perform. Comput. Workshop*, vol. 18. Amsterdam, The Netherlands: IOS Press, 2008, pp. 267–295.

[158] *Amazon Lambda Service*. Accessed: Oct. 2018. [Online]. Availabe: https://aws.amazon.com/lambda/

[159] *Google Cloud Functions*. Accessed: Oct. 2018. [Online]. Availabe: https://cloud.google.com/functions/docs

[160] I. Baldini *et al.*, "Cloud-native, event-based programming for mobile applications," in *Proc. Int. Conf. Mobile Softw. Eng. Syst. (MOBILESoft)*, 2016, pp. 287–288.

[161] M. Caballer, C. De Alfonso, G. Moltó, E. Romero, I. Blanquer, and A. García, "CodeCloud: A platform to enable execution of programming models on the Clouds," *J. Syst. Softw.*, vol. 93, pp. 187–198, Jul. 2014.

[162] F. Lordan *et al.*, "ServiceSs: An interoperable programming framework for the cloud," *J. Grid Comput.*, vol. 12, no. 1, pp. 67–91, 2014.

[163] B. Di Martino, *Building a Mosaic of Clouds*. Berlin, Germany: Springer, 2011, pp. 571–578.

[164] (2016). *Research Challenges in Cloud Computing*. [Online]. Availabe: https://ec.europa.eu/newsroom/document.cfm?doc_id=19453

[165] *The Anatomy of APM*. Accessed: Oct. 25, 2017. [Online]. Availabe: http://www.apmdigest.com/

[166] *Amazon CloudWatch*. Accessed: Oct. 26, 2017. [Online]. Availabe: http://aws.amazon.com/cloudwatch

[167] *CloudMonix AzureWatch*. Accessed: Oct. 26, 2017. [Online]. Availabe: http://www.cloudmonix.com/aw/

[168] K. Fatema, V. C. Emeakaroha, P. D. Healy, J. P. Morrison, and T. Lynn, "A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives," *J. Parallel Distrib. Comput.*, vol. 74, no. 10, pp. 2918–2933, 2014.

[169] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," in *Cloud computing*. Berlin, Germany: Springer, 2009, pp. 115–131.

[170] L. Schubert, J. Domaschka, and P. Guisset. PaaSage-making cloud usage easy, CloudScape. Accessed: Apr. 2018. [Online]. Available: https://paasage.ercim.eu/

[171] O. Sefraoui, M. Aissaoui, and M. Eleuldj, *CIMP: Cloud Integration and Management Platform*. Cham, Switzerland: Springer, 2017, pp. 391–400.

[172] G. Baryannis *et al.*, "Lifecycle management of service-based applications on multi-clouds: A research roadmap," in *Proc. Int. Workshop Multi-Cloud Appl. Federated Clouds (MultiCloud)*, 2013, pp. 13–20.

[173] A. Papaioannou, D. Metallidis, and K. Magoutis, "Cross-layer management of distributed applications on multi-clouds," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 552–558.

[174] N. Ferry, F. Chauvel, H. Song, A. Rossini, M. Lushpenko, and A. Solberg, "CloudMF: Model-driven management of multi-cloud applications," *ACM Trans. Internet Technol.*, vol. 18, no. 2, pp. 16:1–16:24, 2018.

[175] C. Jin and R. Buyya, "MapReduce programming model for .NET-based cloud computing," in *Proc. Eur. Conf. Parallel Process.* Berlin, Germany: Springer, 2009, pp. 417–428.

[176] A. Taherkordi, F. Eliassen, and G. Horn, "From IoT big data to IoT big services," in *Proc. Symp. Appl. Comput. (SAC)*, 2017, pp. 485–491.

[177] R. Lea and M. Blackstock, "City hub: A cloud-based IoT platform for smart cities," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2014, pp. 799–804.

[178] J. Soldatos, M. Serrano, and M. Hauswirth, "Convergence of utility computing with the Internet-of-Things," in *Proc. 6th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput. (IMIS)*, Jul. 2012, pp. 874–879.

[179] M. Vögler, J. M. Schleicher, C. Inzinger, and S. Dustdar, "Ahab: A cloud-based distributed big data analytics framework for the Internet of Things," *Softw. Pract. Exper.*, vol. 47, no. 3, pp. 443–454, 2017.

[180] M. Sharifi, M. A. Taleghan, and A. Taherkordi, "A publish-subscribe middleware for real-time wireless sensor networks," in *Computational Science–ICCS*. Berlin, Germany: Springer, 2006.

[181] S. Nastic, S. Sehic, M. Vögler, H.-L. Truong, and S. Dustdar, "PatRICIA—A novel programming model for IoT applications on cloud platforms," in *Proc. IEEE 6th Int. Conf. Service-Oriented Comput. Appl.*, Dec. 2013, pp. 53–60.

[182] A. Taherkordi and F. Eliassen, "Scalable modeling of cloud-based IoT services for smart cities," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2016, pp. 1–6.

[183] P. Bellavista and A. Zanni, "Feasibility of fog computing deployment based on docker containerization over raspberrypi," in *Proc. 18th Int. Conf. Distrib. Comput. Netw. (ICDCN)*, New York, NY, USA, 2017, pp. 16:1–16:10.

[184] R. Vilalta, A. Mayoral, R. Casellas, R. Martínez, and R. Muñoz, "Experimental demonstration of distributed multi-tenant cloud/fog and heterogeneous sdn/nfv orchestration for 5G services," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2016, pp. 52–56.

[185] R. Vilalta *et al.*, "TelcoFog: A unified flexible fog and cloud computing architecture for 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 36–43, Aug. 2017.

[186] F. Li, M. Voegler, M. Claessens, and S. Dustdar, "Efficient and scalable IoT service delivery on cloud," in *Proc. IEEE 6th Int. Conf. Cloud Comput.*, Jun./Jul. 2013, pp. 740–747.

[187] A. Taherkordi and F. Eliassen, "Towards independent in-cloud evolution of cyber-physical systems," in *Proc. IEEE Int. Conf. Cyber-Phys. Syst., Netw., Appl.*, Aug. 2014, pp. 19–24.

[188] E. D. Simmon *et al.*, "A vision of cyber-physical cloud computing for smart networked systems," NIST, Gaithersburg, MD, USA, Tech. Rep. 7951, 2013.

[189] S. Karnouskos *et al.*, "A SOA-based architecture for empowering future collaborative cloud-based industrial automation," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 5766–5772.

[190] H. Abid, L. T. T. Phuong, J. Wang, S. Lee, and S. Qaisar, "V-cloud: Vehicular cyber-physical systems and cloud computing," in *Proc. 4th Symp. Appl. Sci. Biomed. Commun. Technol.*, 2011, Art. no. 165.

[191] J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou, and H. Cai, "VCMIA: A novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing," *Mobile Netw. Appl.*, vol. 19, no. 2,, pp. 153–160, 2014.

[192] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.

[193] J. Lopez and J. E. Rubio, "Access control for cyber-physical systems interconnected to the cloud," *Comput. Netw.*, vol. 134, pp. 46–54, Apr. 2018.

[194] R. Gravina *et al.*, "Cloud-based activity-aaservice cyber-physical framework for human activity monitoring in mobility," *Future Gener. Comput. Syst.*, vol. 75, pp. 158–171, Oct. 2017.

[195] D. Wu, D. W. Rosen, L. Wang, and D. Schaefer, "Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation," *Comput. Aided Des.*, vol. 59, pp. 1–14, Feb. 2015.

[196] J. Wan, D. Zhang, S. Zhao, L. T. Yang, and J. Lloret, "Context-aware vehicular cyber-physical systems with cloud support: Architecture, challenges, and solutions," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 106–113, Aug. 2014.

[197] J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou, and H. Cai, "VCMIA: A novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 153–160, 2014.

[198] A. C. Zhou, B. He, and S. Ibrahim. (2014). "A taxonomy and survey on eScience as a service in the cloud." [Online]. Available: https://arxiv.org/abs/1407.7360

[199] A. Matsunaga, M. Tsugawa, and J. Fortes, "CloudBLAST: Combining MapReduce and virtualization on distributed resources for bioinformatics applications," in *Proc. IEEE 4th Int. Conf. eSci.*, Dec. 2008, pp. 222–229.

[200] J. Li, M. Humphrey, C. van Ingen, D. Agarwal, K. Jackson, and Y. Ryu, "eScience in the cloud: A MODIS satellite data reprojection and reduction pipeline in the windows azure platform," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. (IPDPS)*, Apr. 2010, pp. 1–10.

[201] C. Evangelinos and C. N. Hill, "Cloud computing for parallel scientific HPC applications: Feasibility of running coupled atmosphere-ocean climate models on Amazon's EC2," in *Proc. 1st Workshop Cloud Comput. Appl. (CCA)*, 2008, pp. 2–34.

[202] Z. Zhang *et al.*, "Scientific computing meets big data technology: An astronomy use case," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct./Nov. 2015, pp. 918–927.

[203] T. Kwon, W. G. Yoo, W.-J. Lee, W. Kim, and D.-W. Kim, "Next-generation sequencing data analysis on cloud computing," *Genes Genomics*, vol. 37, no. 6, pp. 489–501, 2015.

[204] E. A. Mohammed, B. H. Far, and C. Naugler, "Applications of the mapreduce programming framework to clinical big data analysis: Current landscape and future trends," *BioData Mining*, vol. 7, no. 1, p. 22, 2014.

[205] Y. Demchenko *et al.*, "CYCLONE: A platform for data intensive scientific applications in heterogeneous multi-cloud/multi-provider environment," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW)*, Apr. 2016, pp. 154–159.

[206] X. Li, J. Song, and B. Huang, "A scientific workflow management system architecture and its scheduling based on cloud service platform for manufacturing big data analytics," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 1–4, pp. 119–131, 2016.

[207] E. Afgan *et al.*, "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update," *Nucleic Acids Res.*, vol. 46, no. W1, pp. W537–W544, 2018.

[208] G. Cardone, A. Corradi, L. Foschini, and R. Ianniello, "Participact: A large-scale crowdsensing platform," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 21–32, Jan./Mar. 2016.

[209] H. Xiong, Y. Huang, L. E. Barnes, and M. S. Gerber, "Sensus: A cross-platform, general-purpose system for mobile crowdsensing in human-subject studies," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. (UbiComp)*, 2016, pp. 415–426.

[210] *Apache Storm*. Accessed: Mar. 2018. [Online]. Availabe: http://storm.apache.org/

[211] X. Jin and Y.-K. Kwok, "Cloud assisted P2P media streaming for bandwidth constrained mobile subscribers," in *Proc. IEEE 16th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Washington, DC, USA, Dec. 2010, pp. 800–805.

[212] E. Jung, Y. Wang, I. Prilepov, F. Maker, X. Liu, and V. Akella, "User-profile-driven collaborative bandwidth sharing on mobile phones," in *Proc. 1st ACM Workshop Mobile Cloud Comput., Services, Social Netw. Beyond (MCS)*, 2010, pp. 2:1–2:9.

[213] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. 1st ACM Workshop Mobile Cloud Comput., Services, Social Netw. Beyond (MCS)*, 2010, pp. 6:1–6:5.

[214] L. Zhang, X. Ding, Z. Wan, M. Gu, and X.-Y. Li, "WiFace: A secure geosocial networking system using WiFi-based multi-hop MANET," in *Proc. 1st ACM Workshop Mobile Cloud Comput., Services, Social Netw. Beyond (MCS)*, 2010, pp. 3:1–3:8.

[215] A. Klein, C. Mannweiler, J. Schneider, and H. D. Schotten, "Access schemes for mobile cloud computing," in *Proc. 11th Int. Conf. Mobile Data Manage.*, 2010, pp. 387–392.

[216] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[217] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: A partition scheme," in *Proc. Int. Conf. Compil., Archit., Synth. Embedded Syst. (CASES)*, 2001, pp. 238–246.

[218] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 49–62.

[219] S. Ou, K. Yang, A. Liotta, and L. Hu, "Performance analysis of offloading systems in mobile wireless environments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Glasgow, Scotland, Jun. 2007, pp. 1821–1826.

[220] L. Kennedy, S. F. Chang, and A. Natsev, "Query-adaptive fusion for multimodal search," *Proc. IEEE*, vol. 96, no. 4, pp. 567–588, Apr. 2008.

[221] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, "Pocket cloudlets," in *Proc. Int. Conf. Archit. Support Program. Lang. Operating Syst. (ASPLOS)*, 2011, pp. 171–184.

[222] F. A. Samimi, P. K. McKinley, and S. M. Sadjadi, *Mobile Service Clouds: A Self-Managing Infrastructure for Autonomic Mobile Computing Services*. Berlin, Germany: Springer, 2006.

[223] P. Papakos, L. Capra, and D. S. Rosenblum, "VOLARE: Context-aware adaptive cloud service discovery for mobile systems," in *Proc. 9th Int. Workshop Adapt. Reflective Middleware (ARM)*, 2010, pp. 32–38.

[224] D. K. Nguyen, F. Lelli, Y. Taher, M. Parkin, M. P. Papazoglou, and W.-J. van den Heuvel, *Blueprint Template Support for Engineering Cloud-Based Services*. Berlin, Germany: Springer, 2011.

[225] A. Bergmayr, J. Troya, P. Neubauer, M. Wimmer, and G. Kappel, "UML-based cloud application modeling with libraries, profiles, and templates," in *Proc. 2nd Int. Workshop Model-Driven Eng. Cloud*, vol. 1242, 2014, pp. 56–65.

[226] J. Guillén, J. Miranda, J. M. Murillo, and C. Canal, *A UML Profile for Modeling Multicloud Applications*. Berlin, Germany: Springer, 2013, pp. 180–187.

[227] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems," in *Proc. IEEE 6th Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2013, pp. 887–894.

[228] *Heat Orchestration Template (HOT)*. Accessed: Apr. 2018. [Online]. Availabe: https://docs.openstack.org/heat/

[229] N. Nikolov, A. Rossini, and K. Kritikos, "Integration of DSLs and migration of models: A case study in the cloud computing domain," *Procedia Comput. Sci.*, vol. 68, pp. 53–66, Sep. 2015.

[230] L. Sun, J. Ma, H. Wang, and Y. Zhang, "Cloud service description model: An extension of USDL for cloud services," *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 354–368, Mar./Apr. 2018.

[231] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.

[232] T. Guerout, S. Medjiah, G. D. Costa, and T. Monteil, "Quality of service modeling for green scheduling in clouds," *Sustain. Comput., Inform. Syst.*, vol. 4, no. 4, pp. 225–240, 2014.

[233] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Apr. 2010, pp. 446–452.

[234] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: A packet-level simulator of energy-aware cloud computing data centers," *J. Supercomput.*, vol. 62, no. 3, pp. 1263–1283, 2012.

[235] S. F. Piraghaj, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, "ContainerCloudSim: An environment for modeling and simulation of containers in cloud data centers," *Softw., Pract. Exper.*, vol. 47, no. 4, pp. 505–521, Apr. 2017.

[236] S. K. Garg and R. Buyya, "NetworkCloudSim: Modelling parallel applications in cloud simulations," in *Proc. 4th IEEE Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2011, pp. 105–113.

[237] A. Ahmed and A. S. Sabyasachi, "Cloud computing simulators: A detailed survey and future direction," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Feb. 2014, pp. 866–872.

[238] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[239] Y. Liu, M. Li, N. K. Alham, and S. Hammoud, "HSim: A MapReduce simulator in enabling cloud computing," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 300–308, 2013.

[240] (2017). *Yarn Scheduler Load Simulator (SLS)*. [Online]. Available: https://hadoop.apache.org/docs/stable/hadoop-sls/SchedulerLoadSimulator.html

[241] W. Zhao, Y. Peng, F. Xie, and Z. Dai, "Modeling and simulation of cloud computing: A review," in *Proc. IEEE Asia–Pacific Cloud Comput. Congr. (APCloudCC)*, Nov. 2012, pp. 20–24.

[242] CSA. (2016). *The Treacherous 12—Cloud Computing Top Threats*. [Online]. Availabe: https://cloudsecurityalliance.org/group/top-threats

[243] Y. Verginadis, A. Michalas, P. Gouvas, G. Schiefer, G. Hübsch, and I. Paraskakis, "PaaSword: A holistic data privacy and security by design framework for cloud services, *J. Grid Comput.*, vol. 15, no. 2, pp. 219–234, Jun. 2017.

[244] M. Decker, "Modelling of location-aware access control rules," in *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*. Hershey, PA, USA: Information Science Reference, 2011, Ch. 57, pp. 912–929.

[245] G. Zhang and M. Parashar, "Context-aware dynamic access control for pervasive applications," in *Proc. Commun. Netw. Distrib. Syst. Modeling Simulation Conf.*, 2004, pp. 21–30.

[246] S. M. Chandran, LoT-RBAC*: A Location and Time-Based RBAC Model*. Berlin, Germany: Springer, 2005.

[247] D. Kulkarni and A. Tripathi, "Context-aware role-based access control in pervasive computing systems," in *Proc. 13th ACM Symp. Access Control Models Technol. (SACMAT)*, New York, NY, USA, 2008, pp. 113–122.

[248] A. Corrad, R. Montanari, and D. Tibaldi, "Context-based access control management in ubiquitous environments," in *Proc. 3rd IEEE Int. Symp. Netw. Comput. Appl. (NCA)*, Sep. 2004, pp. 253–260.

[249] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma, "Context sensitive access control," in *Proc. 10th ACM Symp. Access Control Models Technol. (SACMAT)*, New York, NY, USA, 2005, pp. 111–119.

[250] A. van Cleeff, W. Pieters, and R. Wieringa, "Benefits of location-based access control: A literature study," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun. (GreenCom), Int. Conf. Cyber, Phys. Social Comput. (CPSCom)*, Dec. 2010, pp. 739–746.

[251] S. Veloudis, I. Paraskakis, and C. Petsos, "Ontological framework for ensuring correctness of security policies in cloud environments," in *Proc. BCI*, 2017, Pp. 23:1–23:8.

[252] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2046–2069, 4th Quart., 2013.

[253] P. Chaignon, D. Adjavon, K. Lazri, J. François, and O. Festor, "Offloading security services to the cloud infrastructure," in *Proc. Workshop Secur. Softwarized Netw., Prospects Challenges (SecSoN)*, New York, NY, USA, 2018, pp. 27–32.

[254] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 408–415.

[255] J.-H. Jun, C.-W. Ahn, and S.-H. Kim, "DDoS attack detection by using packet sampling and flow features," in *Proc. 29th Annu. ACM Symp. Appl. Comput. (SAC)*, New York, NY, USA, 2014, pp. 711–712.

[256] V. Jyothi, X. Wang, S. K. Addepalli, and R. Karri, "Brain: Behavior based adaptive intrusion detection in networks: Using hardware performance counters to detect DDoS attacks," in *Proc. 29th Int. Conf. VLSI Design, 15th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2016, pp. 587–588.

[257] S. Baidya, Y. Chen, and M. Levorato, "eBPF-based content and computation-aware communication for real-time edge computing," in *Proc. INFOCOM*, Honolulu, HI, USA, Apr. 2018, pp. 865–870.

[258] D. Sánchez and M. Batet, "Privacy-preserving data outsourcing in the cloud via semantic data splitting," *Comput. Commun.*, vol. 110, pp. 187–201, Sep. 2017.

[259] F. Baldimtsi, A. Kiayias, and K. Samari, "Watermarking public-key cryptographic functionalities and implementations," in *Proc. Int. Conf. Inf. Secur.*, 2017, pp. 173–191.

[260] M. Azraoui, K. Elkhiyaoui, M. Önen, and R. Molva, "Publicly verifiable conjunctive keyword search in outsourced databases," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Sep. 2015, pp. 619–627.

[261] I. Demertzis and C. Papamanthou, "Fast searchable encryption with tunable locality," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, New York, NY, USA, 2017, pp. 1053–1067.

[262] M. Bellare and V. T. Hoang, "Identity-based format-preserving encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2017, pp. 1515–1532.

[263] D. Movshovitz, "Method and computer program product for order preserving symbol based encryption," U.S. Patent 9 734 350, Aug. 15, 2017. [Online]. Availabe: https://www.google.com/patents/US9734350

[264] M. Egorov and M. Wilkison. (2016). "ZeroDB white paper." [Online]. Available: https://arxiv.org/abs/1602.07168

[265] G. Ouffoué *et al.*, "Intrusion detection and attack tolerance for cloud environments: The CLARUS approach," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, Jun. 2016, pp. 61–66.

[266] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proc. Conf. Hot Topics Cloud Comput. (HotCloud)*, Berkeley, CA, USA, 2009, pp. 1–5.

[267] R. Dowsley, M. Gabel, G. Hubsch, G. Schiefer, and A. Schwichtenberg, "A distributed key management approach," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2016, pp. 509–514.

[268] A. Michalas, N. Komninos, N. R. Prasad, and V. A. Oleshchuk, "New client puzzle approach for DoS resistance in ad hoc networks," in *Proc. IEEE Int. Conf. Inf. Theory Inf. Secur.*, Dec. 2010, pp. 568–573.

[269] (2014). *Cloud Standards Customer Council, Interoperability and Portability for Cloud Computing: A Guide*. [Online]. Availabe: http://www.cloud-council.org/deliverables/CSCC-Interoperability-and-Portability-for-Cloud-Computing-A-Guide.pdf

[270] B. Di Martino, G. Cretella, and A. Esposito, *Cloud Portability and Interoperability*. Springer, 2015.

[271] *Cloud Data Management Interface by SNIA*. Accessed: Jan. 2018. [Online]. Availabe: http://snia.org/cdmi

[272] B. Di Martino, G. Cretella, and A. Esposito, "Advances in applications portability and services interoperability among multiple clouds," *IEEE Cloud Comput.*, vol. 2, no. 2, pp. 22–28, Mar./Apr. 2015.

[273] D. R. Recupero *et al.*, "An innovative, open, interoperable citizen engagement cloud platform for smart government and users' interaction," *J. Knowl. Economy*, vol. 7, no. 2, pp. 388–412, 2016.

[274] C. Pahl, "Containerization and the PaaS cloud," *IEEE Cloud Comput.*, vol. 2, no. 3, pp. 24–31, May/Jun. 2015.

[275] E. Kamateri *et al.*, "Cloud4SOA: A semantic-interoperability PaaS solution for multi-cloud platform management and portability," in *Service-Oriented and Cloud Computing*. Berlin, Germany: Springer, 2013, pp. 64–78.

AMIR TAHERKORDI received the Ph.D. degree from the Department of Informatics, University of Oslo, in 2011. He is currently a Researcher with the Networks and Distributed Systems Group, Department of informatics, University of Oslo.

His research has been focused on distributed computing and software engineering aspects of emerging technologies, such as Internet of Things (IoT), Clouds and Fogs/Edges, Cyber-Physical Systems, and Smart Grids.

His current work is focused on software architectures, programming abstractions, service distribution, and middleware systems for IoT, as well as adaptation middleware solutions for multicloud applications (EU H2020 MELODIC project).

He has experience from several Norwegian and EU research projects and published several articles in high-ranked conferences and journals. He was selected as a Young Talented Researcher by the Norwegian Research Council in 2017 to work on a novel IoT service computing model for future Fog-Cloud computing systems (NFR DILUTE project).

FEROZ ZAHID received the Ph.D. degree from the University of Oslo in 2017. He has several years of industrial experience in software development, IT consultancy, and system design. He is currently a Research Scientist at the Simula Research Laboratory. His research interests include interconnection networks, distributed systems, cloud computing, energy-efficient systems, network security, machine learning, and big data. Several of his research results have been patented.

YIANNIS VERGINADIS He received the Diploma and Ph.D. degrees in electrical and computer engineering from the National Technical University of Athens, Greece, in 2001 and 2006, respectively. His Ph.D. thesis was on Inter-organizational Workflow Management Systems in e-Government. He has more than 10 years of experience in research areas, such as management of information systems, software engineering, workflow management, e-Government, and cloud computing. He was an Adjunct Lecturer at the Department of Mechanical and Industrial Engineering, University of Thessaly, and at the Technological Educational Institute of Kalamata. He is a Senior Researcher at the Institute of Communication and Computer Systems, National Technical University of Athens.

GEIR HORN received the Cand. Scient. degree in cybernetics and the Ph.D. degree in computer science and mathematical learning from the University of Oslo. He has been a Research Scientist at the Centre for Industrial Research and the SIMULA Research Laboratory, the Research Director at SINTEF Electronics and Cybernetics, and has been responsible for the IT-sector at the Norwegian Industrial AttachÃľ in Paris. He is currently the Head of the European ICT Projects at the Faculty of Mathematics and Natural Sciences, University of Oslo. He has been the Principal Investigator and the Project Leader of more than 17 European collaborative projects, and has also been evaluating research proposals and been a peer reviewer of other running projects on behalf of the European Commission since 1997. He is also serving on multiple conference programme committees, and has published more than 50 scientific papers.

Dr. Horn started working with high performance computing and data center interconnect technologies. This interest was naturally expanded over the years into Grid and Cloud computing with a special focus on distributed systems and context aware self-adaptive applications and autonomic computing. He has developed several solutions using stochastic combinatorial optimization based on reinforcement learning and learning automata. His current research interests are on how to handle complexity and services choreography for large-scale distributed applications through adaptation, autonomic decisions, self-awareness, and emergence..