Security-Aware Energy Management in Clouds

Jianzhou Mao, Ting Cao, Xiaopu Peng, Tathagata Bhattacharya
Wei-Shinn Ku, and Xiao Qin
Department of Computer Science and Software Engineering
Auburn University, Auburn, AL.

***Abstract*—Cloud computing over the Internet reveals a remarkable potential to provide on-demand services to consumers with great flexibility in a cost- effective manner. Security issues coupled with resource allocations in cloud computing remain a challenging problem to be tackled by the industry and academia. While moving towards the concept of on-demand services and resource pooling in a distributed computing environment, security is a major obstacle for this new dreamed vision of computing capability. At the same time, the research field of energy-efficient networking infrastructures is of great importance for service providers, network administrators, and equipment manufacturers. In this paper, we articulate novel energy-aware scheduling policies catered for virtual machines running on clouds, in which service-level agreements (SLAs) are fulfilled. After addressing security concerns in cloud computing, we advocate for a research roadmap towards future security-aware energy management in clouds. We proposes a high-level design for a security- and frequency-ware DVFS model or *SF-DVFS*, which orchestrates security services, security overhead analysis, and DVFS control green cloud computing systems. We delve into the main technical challenges associated with the proposed SF-DVFS model. We expect that this paper will open exciting perspectives for future security research in energy-efficient cloud computing platforms.**

***Index Terms*—cloud computing, Security overhead, Energy optimization, Load balancing, Real-Time Scheduling, DVFS.**

I. Introduction

In this paper, we propose a research roadmap for security-aware energy management for cloud computing infrastructures. This roadmap is inspired by the following three trends.

- Cloud computing is an effective technology that delivers interesting services to customers over the Internet.
- There is a pressing demand to build energy-efficient clouds housed in large-scale datacenters.
- Building trustworthy cloud environments remains a challenging issue.

With the advanced virtualization technologies deployed in data centers, cloud infrastructures become a predominant computing platform (see, for example, Amazon Elastic Compute Cloud (EC2) [1] and Microsoft Azure [2]). Virtual computation environments furnish on-demand and elastic computation and storage capabilities, thereby facilitating large-scale data analytic and big-data applications. In modern virtualization techniques, resources residing in physical machines are partitioned into individual virtual machines (VMs), which isolates one application from the counterparts running on the other VMs. Multiple VMs assigned to one physical machine share resources on the same machine. One or more applications may run on a virtual machine; in contrast, an large-scale application can make use of enormous resources across multiple virtual machines.

In an drastically expanding digital world, big data are changing the way we live, work, and entertain. International Data Corporation or IDC speculates that the aggregated data around the world will grow from 33 zettabytes in 2018 to 175ZB by 2025 at a significant annual grow rate of 61% [3]. To accommodate such a massive amount of data, the scale of data centers is demanded to snowball to reach an unprecedented and unbelievable level. The global data-center market is estimated to exceed $174 billion by 2023, representing an annual rate of approximately 4% during the forecast period. To meet such pressing demands, the largest technology companies such as Facebook, Google, Amazon, and Microsoft are focusing on the development of modular and hyper-scale data-center construction facilities [4].

The energy consumption of these large-scale datacenters is truly tremendous.For example, the global data-center power market size will hit the bar of $10.77 billion by year 2025, expanding at an annual rate of 6.9%, even faster than that of the datacenter market [5]. Globally,Power consumption of datacenters is close to 416 terawatts, representing three percent of all electricity generated on the planet. In other words, data center energy consumption around the world accounts for 40 percent more than all the energy consumed in the United Kingdom [6].Nowadays, over 80% of the worlds energy still being generated by fossil fuels [7],which could lead the CO2 emissions and other global environmental problems like global warming.

Cloud computing offers services with scalable resources in a protected view. Although cloud features are well understood from a business point of view, building trustworthy cloud environments remains a challenging issue. Cloud computing has increasingly gained its popularity among individual users and organizations, but recently raised security issues demand new solutions. For example, organizations have a dire need for secure infrastructures when data are transferred to and managed at remote locations.

It is a conventional wisdom to handle big data in local storage systems, where data processing, movement, and management are carried out in local domains. More often than not, security measures developed by cloud service providers are transparent to the public and; for this reason, some enterprise users hesitate to rely on cloud services and infrastructure to store and process digital assets [8] [9].

The remainder of this paper is organized as follows. Section II outlines the evolution of cloud computing systems from the perspectives of load balancing, energy conservation, and security issues. In Section III, we introduce various of scheduling policies in clouds. We present a research roadmap, where approaches and directions are discussed in section III. Finally, Section V elaborates concluding remarks.

II. Cloud Computing Systems

A. *Virtual Machines and Load Balancing*

The purpose of load balancing is to evenly distribute computing workloads across multiple computing resources to maximize the overall system performance. Load balancing aims to achieve an array of objectives, including (1) optimizing resource usage, (2) maximizing throughput, (3) minimizing response time, and (4) avoiding the overload of any resource. VM-based load balancing is implemented through live VM migrations in data centers, where a primarily concern is to optimize the usage of physical computing resources by migrating virtual machines from heavily loaded PMs to those with least workload. By dynamically adjusting the locations of VMs, one may optimize various objective functions to provide superb cloud services. Sample objective functions include, but not limited to, improving performance, boosting system security, minimizing failure impact, and reducing energy consumption.

Fig. 1 illustrates a classic load balancing architecture in a cloud computing platform. All user requests are submitted to the load balancing module, which is responsible for dispatching requests to virtual machines to optimize resource utilization and energy efficiency.

Load balancing plays a critical role in guaranteeing the service-level agreements (SLAs) of applications in cloud computing. The increasing workload of applications in virtual machines may trigger overloaded utilization in one resource or more (e.g., CPU, memory, I/O and network bandwidth) on physical machines. More often than not, an overly loaded physical machine degrades application performance of all the VMs running on the PM. Consequently, unbalanced load inevitably impose an adverse impact on the finish times of batch applications and the response times of interactive applications. To eliminate the potential bottleneck, one has to migrate excess load from overloaded physical machines to underutilized ones in computing clouds.

It is arguably true that load balancing techniques powered by VM migrations confront the following challenges.

- **Overhead.** It is prudent to quantify the amount of overhead involved in deploying a load balancing system. Load balancing overhead entails VM migration cost and communication cost. For example, load of each physical machines ought to be periodically collected by a load balancing mechanism, which pays the communication cost to monitor load across multiple PMs. A well-designed load balancing algorithm should reduce such an overhead.
- **Prediction.** Due to the dynamic changes of application workload in VMs, it is inefficient to make migration decisions merely based on the current status of the system. An ideal load balancing algorithm should be equipped with a capacity of accurately predicting workload to orchestrate VM management prior to any sharp changes in future load. Such proactive approaches avert making last-minute load-balancing decisions, which are in some cases too late.

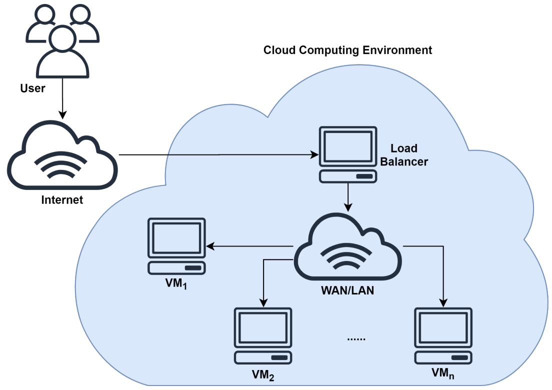

Fig. 1. A load balancing architecture for clouds, where all user requests scheduled and dispatched by the load balancing module to optimize resource utilization and energy efficiency.

- **Performance.** Various performance metrics are introduced to assess the efficiency of cloud computing systems. Performance of a computing cloud can be measured from the perspectives of system throughput as well as user experience and satisfaction. Given performance requirements prescribed by end users, computing clouds are responsible to ensure such requirements defined as quality of service (QoS). Modern load balancing mechanisms seek to boost overall system performance while meeting QoS requirements.

B. *Energy Conservation in Clouds*

To build energy-efficient data centers, one has to pin point the hot spot of energy-burning components as well as deploying energy conservation techniques. Fig. 2 lists ten commonly adopted energy conservation techniques in cloud computing environments.

Intuitively, energy savings should not come at the the cost of performance. The overall objective for modern data centers is to employ energy-saving techniques without violating QoS requirements or downgrading performance. Evidence shows that the waste of power is likely to be originated from the inefficient utilization of provisioned facilities. The VM consolidation [10] and DVFS [11] (Dynamic Voltage and Frequency Scaling) techniques are two practical energy conservation methods predominantly adopted in data centers. One common approach to achieving high energy efficiency is to dynamically scale down the size of running clusters by the virtue of VM management. With the help of virtualization, energy consumed by computing clusters can be curtailed by applying VM migrations and consolidations. Such a technique aims to stack VMs into a minimum number of physical machines (PMs) to cut back energy consumption in data centers. Decisions of turning on or off PMs should fulfill the requirements of maintaining satisfied performance while reducing consumed power.

The dynamic voltage frequency scaling (DVFS) technique [11] has been widely investigated by the research community as a feasible solution to reduce energy consumption of IT equipment. DVFS allows processors to be running at multiple frequencies under different supply voltages, thereby offering ample opportunities to slash energy consumption of cloud computing platforms by scaling back processor supply voltages. The energy consumption of a processor is approximately proportional to processor frequency and the square of the processor voltage. Decreasing the processor voltage and frequency will conserve energy by lowering down processor performance. Nevertheless, such slowed down performance for the purpose of energy saving is acceptable as long as QoS requirements or service-level agreements are met.

Apart from DVFS and VM consolidation, a diversity of power management strategies were constructed to make clouds energy efficient. For example, Gu *et al.* proposed a multi-sleep model, where one active state may transition into multiple sleep states [12]. The multiple sleep state entail different sleep power and transition delays. A growing attention has been paid to energy-efficient disk arrays [13] and storage clusters [14]. Similar to VM consolidation, virtual-machine provisioning [15] offers energy savings for VM-based clouds.

Numerous prior studies are evident that thermal management can boost the energy efficiency in cloud computing. For instance, Amritpal and Kinger applied temperature-aware scheduling to reduce energy consumption [16]. Arroba *et al.* devised novel power and thermal-aware strategies to provide joint cooling and computing optimizations from a local perspective based on the global energy consumption of metaheuristic-based optimizations [17]. Nowadays, machine learning becomes a game changer for the development of green cooling policies in thermal management [18].

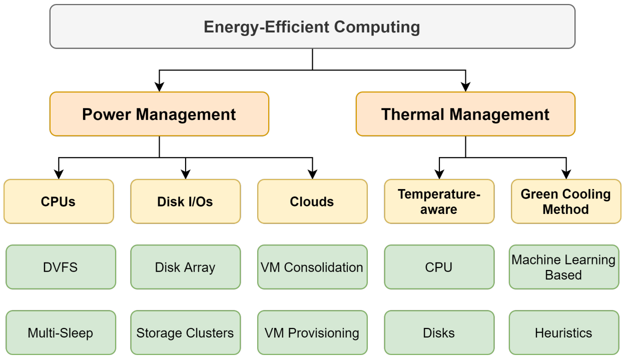

Fig. 2. Ten commonly adopted energy conservation techniques for clouds.

C. Security Issues in Cloud Computing

Fig. 3 summarizes the five major data security issues to be addressed in the arena of cloud computing. A risk of data misuse is likely to occur when resources are shared among multiple organizations. To avert such a risk, it is prudent to secure storage infrastructures along with processed and archived data. Data protection, a vital and challenging feature of cloud computing, keeps any potential security threats at bay. Authentication, authorization, and access control services are devised for to enhance data security in clouds.

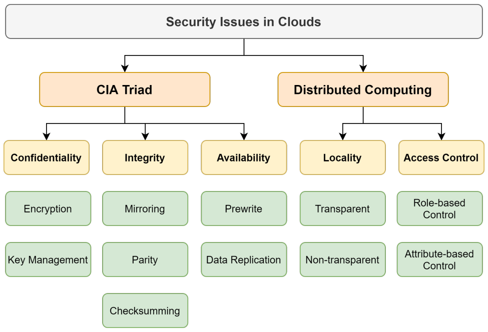

Fig. 3. There are five major data security issues to be addressed in the arena of cloud computing.

Confidentiality, *integrity*, and *availability*, which are known to as the CIA triad, are the three critical properties of data centers. Confidentiality ensures that data owned by cloud service consumers should not be revealed to unauthorized parties under any circumstance [15]. Various encryption techniques [19] and key management [20] mechanisms are deployed to ensure high confidentiality of cloud services. Data integrity entails confidence that data stored in and transferred to/from clouds are not fiddled by unauthorized users. Data integrity can be detected by modern techniques like mirroring, parity, or checksumming at either the file or the block levels [21]. Data availability implies that data should be readily accessed by users without any delay or deny of service when the users issue requests. A handful of leading solutions are available to achieve high data availability. For example, data replication [22] and prewrite operation [23] are two common practices to furnish high data availability to cloud computing systems.

When it comes to distributed computing in clouds, two security challenges to be tackled are *locality* and *access*. Nowadays, data tends to be distributed across multiple regions, where pinpointing the location of data is non-trivial. When data are moved or migrated to from one geographic location into another, the laws and regulations governing on the data may change. Consequently, cloud service providers have to be compliant with data privacy laws according to the geographic locations. This emerging challenge is referred to as *locality* issues of data security in cloud computing environments. Such a data locality issue is handled by clouds in two fashions. On one hand, cloud service providers make data locality transparent to end users. On the other hand, users are in full control of data locations to meet prescribed security requirements. We refer the former one as transparent data locality and the later one as non-transparent data locality. A benefit of the transparent approach is that users can easily access their data without being aware of the locations of data. In contrast, non-transparent location policies enable cloud user to elect desired service locations to safeguard data with respect to locality.

Access control is regarded as a second security issue in distributed computing over clouds. In an organization where computing platforms are outsourced to clouds, members of the organization are authorized manage a portion of data in accordance to access policies. Such data may not be retrieved or modified by the other members of the organization in the distributed computing environments. Most leading-edge access control techniques applied to cloud computing fall into two camps, namely, role-based and attribute-based schemes. For example, Zhou *et al.* designed a role-based encryption scheme to enforce access control policies for encrypted data stored in public clouds [24]. Yang *et al.* developed a time-domain attribute-based access control scheme, which allows a group of users to securely share videos in clouds [25].

III. Scheduling in Clouds

At the heart of a cloud computing platform that orchestrates a diversity of virtualized resources, scheduling mechanisms become a vital component to optimize resource utilization. A client may leverage multiple virtualized computing resources to accomplish tasks submitted to clouds. An overarching goal of task scheduling is to slate tasks running on computing clouds to achieve specific objectives. Sample objectives include minimizing response time, maximizing performance, reducing energy consumption, improving system security, and to name just a few.

Fig. 4 depicts a scheduling architecture designed for cloud computing platforms, in which scheduling mechanisms and security-service optimization modules are fully integrated. Similar architectures can be found in the literature (see, for example, [26] [27]). In the illustrated architecture, a cloud is highlighted in a dotted box. Cloud users dynamically submit a wide range of tasks to the scheduler, which oversees virtualized resources in the cloud. After scheduling decisions are made by the schedule, tasks are dispatched to corresponding virtual machines. As a part of the scheduling mechanism, a monitor periodically keeps track of the utilization of the virtual machines as well as physically machines in the cloud. Apart from scheduling tasks, the scheduler is in charge of launching appropriate security services for input and output data of tasks to fulfill user requirements.

A. Real-Time Scheduling

The timeliness of to real-time applications is a key toward high quality of service (QoS) on clouds. Virtual machines can be handled as tasks from the perspective of real-time scheduling. Therefore, we use terms virtual machines and tasks interchangeable throughout this manuscript.

For hard real-time applications, the timeliness measures the system capability of guaranteeing deadlines specified by users. In the realm of cloud computing, timeliness is referred to as a performance metric that entails the sum of utility or benefits obtained by real-time tasks or services [28].

Real time tasks embrace deadlines, which are specified in the format of QoS requirements. Missing deadlines is treated as a failure or an error for the real-time tasks submitted to clouds. The ability to satisfy deadlines (a.k.a., timing constraints) of real-time tasks is an overarching goal to be achieved by schedulers managing virtualized resources in cloud computing environments.

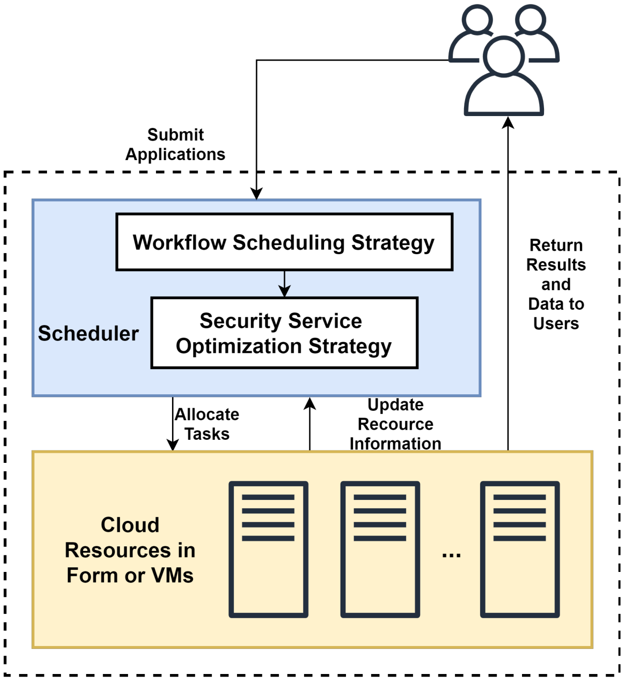
Submit Applications
Workflow Scheduling Strategy
Scheduler
Security Service Optimization Strategy
Return Results and Data to Users
Allocate Tasks
Update Recource Information
Cloud Resources in Form or VMs
...

Fig. 4. The scheduling architecture for cloud computing platforms, which embrace scheduling mechanisms and security-service optimization modules.

As conventional schedulers, real-time schedulers customized for clouds aim to make good trade-off among multiple factors such as scheduling complexity, real-time performance, energy efficiency, and security [29]. Real-time tasks ought to be correctly and timely carried out by clouds. Evidence shows that obtaining a minimal schedule for a set of real-time tasks running in multiprocessor systems is a NP-hard problem [30]. Unsurprisingly, real-time schedulers are unable to deliver deterministic response times, which are an important metric gauged for system robustness analysis. Security-sensitive real-time tasks running on clouds must be protected against cyber-security threats, which make the design of resource management systems for clouds a grand challenge. To address the aforementioned challenging issues, we will pilot a security- and frequency-aware DVFS model (SF-DVFS) to incorporates security services and energy management in a computing cloud. Please refer to our roadmap elaborated in Section IV-D for a detailed research plan on FS-DVFS.

B. Energy-aware Scheduling

In the past decade, high energy consumption in cloud-based data centers has motivated the research community to develop energy-efficient techniques, among which a growing number of energy-aware scheduling algorithms offer impressive energy savings to computing clusters on clouds [31] [32]. Generally speaking, energy-efficient scheduling approaches can be categorized into two camps, namely, DVFS-based (Dynamic Voltage and Frequency Scaling) and VM-based (Virtual Machine) techniques.

1) DVFS-based Scheduling: Recall that (see Section II-B) DVFS-based schemes strive to make good trade-offs between energy consumption and performance in processors, which are a major player in reducing power consumption of data centers. For example, Garg *et al.* developed the near-optimal energy-efficient scheduling algorithms, where DVFS is employed to minimize carbon emission by scaling down CPU frequency while maximizing profits of cloud providers [33]. Fettes *et al.* designed practical scheduling policies, which seamlessly integrate DVFS and the virtual-machines consolidation scheme to make cloud-based data centers energy efficient [34]. Maroulis *et al.* applied DVFS to curb the energy consumption of MapReduce applications running on computing clusters [35]. Suleiman *et al.* merged the thermal-aware approach and DVFS in a smart way to offer power management in data centers [36]. Duan *et al.* devised an algorithm to judiciously tunes CPU frequency in accordance with QoS requirements [37]. In this algorithm, a prediction method was incorporated to adapt CPU frequency by jointly considering QoS and available slack time. Consequently, the novel scheduler is capable of reducing energy consumption in heterogeneous Hadoop clusters. Similarly, Ibrahim *et al.* mixed the DVFS and machine learning approaches to slash energy consumption in network-on-chips systems (NoCs) [38].

2) Virtual-Machine-based Scheduling: A tremendous effort in building energy-aware schedulers over the past several years has concentrated on dynamical consolidation of virtual machines. A vast majority of such scheduling algorithms aim to manage virtual machines according to dynamic system workload, thereby cutting back the number of physical hosts so that idle hosts are switched off to conserve energy. Recently developed scheduling strategies leverage live migrations of virtual machines to support multiple fields, including scientific workflows and real-time tasks. For example, Xu *et al.* designed an energy-aware resource allocation method to allocate virtual machines in support of scientific workflow executions [39]. After proposing a novel rolling-horizon scheduling architecture for real-time tasks running on clouds, Zhu *et al.* implemented an energy-aware scheduling algorithm called *EARH* for real-time, aperiodic, independent tasks [40].

A wide range of scheduling algorithms were designed to conserve energy consumption in clouds by the virtue of virtual-machine migrations and consolidation. For instance, Khazaei *et al.* proposed a scheduling technique to minimize service delay in clouds by lowering transmission and processing times through virtual-machine migrations [41]. After investigating a way of dynamically consolidating tasks to boost resource utilization and to reduce energy consumption, Hsu *et al.* presented an energy-aware task consolidation (ETC) method to optimize energy efficiency in clouds [42]. To take uncertainties into account, Chen *et al.* employed proactive and reactive algorithms to mitigate adverse impacts of uncertainties on scheduling quality of cloud-based data centers [43].

C. Cloud-aware Scheduling

1) Online Scheduling: Much attention has been paid towards online scheduling of multiple tasks and jobs. For example, Shin et al. extended the conservative back-filling algorithm by utilizing the earliest deadline first and the largest weight first policies to schedule real-time jobs [44]. Ge *et al.* dived into a GA-based task scheduler, which manages waiting tasks through a genetic algorithm with a goal of balancing load [45]. Liu and Han proposed an online scheduler allowing virtual machines to obtain extra CPU shares when blocked by I/O interrupts, thereby curtailing energy-efficiency losses caused by I/O intensive tasks [46].

2) Scheduling for Multi-processors: When cloud computing platforms are fueled by multi-processor systems, scheduling algorithms is focused on enhancing the overall performance of multi-processor systems. For instance, Dorronsoro.et al. presented a two-level strategy for scheduling large workloads on multicore distributed systems, taking into account their total execution time and energy consumption [47]. Kwok and Ahmad devised an array of optimal static algorithms to schedule task graphs with random parameters for multiple homogeneous processors [48]. Similarly, Mohamed and Awadalla proposed multi-processor-based scheduling approaches, namely the modified list scheduling heuristic (MLSH) and the hybrid genetic algorithm (GA) [49].

3) Performance-aware scheduling: Performance-aware scheduling solutions were deployed to optimize system performance measured in terms of response time, makespan, and completion time. Please refer to [50] and [51] for the comprehensive surveys on task and resources scheduling policies that are intended to speed up system performance of clouds. For example, Tang *et al.* designed a self-adaptive scheduling algorithm for jobs running on MapReduce-based computing clusters [52]. This algorithm dynamically decides the start time of each reduce task according to the corresponding job's context such as task completion time and map tasks' output size. Gan *et al.* implemented a genetic simulated annealing algorithm to optimize the makespan of a set of tasks. In this approach, simulated annealing is used to optimize each offspring yielded by the genetic algorithm [53]. Furthermore, an improved genetic algorithm was developed to apply the outputs of Max-Min and Min-Min as initial solutions to schedule independent tasks [54]. Zuo *et al.* proposed a multi-objective ant colony algorithm to address the task scheduling problem. The focal point of this multi-objective algorithm is to minimize makespans by incorporating user-budget costs as constraints during the course of task scheduling [55].

IV. A Research Roadmap

In Section IV-A, we start the roadmap description by presenting the concepts of security services and strengths. Next, Section IV-B discusses the development of security overhead models for various security services. We propose in Section IV-C an idea of incorporate security and frequency awareness into the context of qualify of service (QoS). Finally, Section IV-D presents a security- and frequency-aware DVFS model (SF-DVFS) in clouds.

A. Security Services and Strengths

The security of a cloud computing system entails a capability of keeping various attacks at bay. A security system built for clouds consists of a diversity of security services like data integrity, confidentiality, and authentication. Because security services are implemented by different algorithms, the security services experience various strength associated with computational overhead. For instance, data confidentiality may be furnished by the *RC4* or *AES* cryptographic algorithms. RC4 is a fast algorithm with low memory space overhead [56]. Importantly, Fluhrer *et al.* discovered a few vulnerabilities in the RC4 algorithm, meaning that RC4 is unsafe for any key size [57]. In contrast, AES encryption was rigorously reviewed for potential security loopholes before being standardized by *NIST* in 2001. Compared with RC4, AES is more secure at the cost of high overhead.

The security strength of a cryptographic algorithm largely depends on key size and the number of operation rounds. The key size directly resembles the strength of the algorithm against key search attacks. In the AES case, the key size can be configured at 128, 192, and 256 bits. Theoretically speaking, the number of guesses to crack AES protected data is 3.410^{38} for the 128-bit key, 6.210^{55} for the 192-bit key, and 1.110^{77} for the 256-bit key. On the other hand, expanding the number of operation rounds makes ciphers more secure, because a large number of rounds leaves no trails of original data. Therefore, one may make use of the number of operation rounds to gauge the quality of ciphers against potential cryptanalysis attacks [58].

To optimize the security strength of applications running on computing clouds, we advocate for future efforts to quantitatively measure the strength and computational overhead of different security services implemented by cutting-edge algorithms. It is arguably true that the strength of a security service is proportional to the service's computing and communication overhead, because low-quality security services that bear high overhead should be replaced by either fast-service counterparts or high-quality services with high overhead.

B. Security Overhead Models

Among a variety of security services, confidentiality, integrity, and availability are three common services to safeguard sensitive data. Among these three types of services, we first focus on the security overhead models developed to capture the correlation between strength and overhead in the confidentiality and integrity services. Then, we shed some light on the idea of construction a security overhead model for data availability services.

1) Confidentiality and Integrity: A security service may be implemented by multiple implementation instances, each of which have distinctive security strength and the computing overhead. In this study, we refer to the implementation instances as security service instances or security instances for short. Given a security service, we assign 1 as the strength value of the strongest security instance. The strength values of the other security instances in this service type are normalized based on the strongest instance. The overhead of each security instance should be derived from a program profiling study. Let us take the confidential service security as an example. Table I summarizes the strengths and speed of the encryption algorithms implemented in the five confidentiality instances. Similarly, Table II lists the hash functions supporting the five integrity instances. The details on these security overhead models can be found in the literature [19] [59].

The overhead of the cryptographic instances are measured on virtual machines running on a physical machine powered by a 3.3 GHz duo-core CPU, 2.0 GB main memory, and 400 GB disk [60]. The overhead of each security instance is heavily reliant on the size of data to be protected and the security instance's speed. More specifically, the overhead of securing data equals to data size divided by the speed of the given security instance. Such a security overhead plays a key role in utilizing slack time to adjust security and frequency levels in a resource management system articulated in Section IV-C.

TABLE I. The Encryption Algorithms for Confidential Service.

Encryption algorithms	Strength	Speed (Mb/s)
IDEA	1.00	17.34
DES	0.90	18.21
Rijndael	0.64	39.88
Blowfish	0.36	39.96
RC4	0.30	87.07

TABLE II. The Hash Functions for Integrity Service.

Hash functions	Strength	Speed (Mb/s)
TIGER	1.0	48.03
RIFDMD-160	0.77	71.27
SHA-1	0.63	80.67
RIFDMD-128	0.36	86.97
MD5	0.26	138.12

2) Data Availability: Now we propose an approach to building overhead models for data availability services in cloud storage. High data availability becomes possible with the full support of replication services or erasure code services, which are summarized as follows.

Data replication is a simple yet effective approach to tolerating failures in cloud storage. In case of a lost data block, one replica block is sufficient to fix the problem with the minimum data movements over networks. A high replica factor like triplication boost storage system performance via parallel I/Os [61]. An overhead model dedicated to data replication is comprised of replication service instances representing different replica factors. In this model, a high replica factor offers high data availability at the cost of creating replications. On the flip side, the overhead can be reduced by lowering the replica factor. Intuitively, in this model security levels of data availability are measured by replica factors. The overheads of read operations are in stark difference from those of write operations. A high replica factor leads to fast reads and expensive writes; the opposite is true for a low replica factor. Thus, the overhead model must be separately developed for reads and writes.

Erasure codes are widely adopted in cloud storage housed in data centers [62] [63] [64]. The Reed-Solomon (RS) code is a popular erasure-code solution, thanks to its optimal storage efficiency and high level of data availability tolerance [65]. (k+r,k) RS codes encode source data with a k×(k+r) *Generator Matrix*, which involves a k×k *Identity Matrix* and a k×r *Redundancy Matrix* (see the details in [66]). In RS encoding, parity strips are originated by multiplying k data strips with the k×r redundancy matrix. In the security overhead model for data availability services fueled by RS code, security levels and overhead are obtained from parameters k and r. In general, the large values of r offers a high level of availability (high security level) at an expensive cost of constructing parity strips. Reducing r value curtails the overhead by sacrificing data availability.

C. Security and Frequency Awareness in QoS

The security overhead models articulated in Section IV-B can be incorporated into a QoS model to speculate the time spent in performing assigned security services. Specifically, security overhead prolongs task execution times, which in turn triggers performance degradation. Nevertheless, tasks that are slowed down by such security overhead are acceptable as long as QoS requirements can be fulfilled.

In conventional real-time task models, the worst case excution-time (WCET) and deadlines are two key parameters capturing QoS requirements of real-time applications. Besides WCET and deadlines, CPU frequency is a practical parameter to prescribe QoS requirements. Given memory and I/O resources, a task's execution time largely depends on an assigned processor and its CPU frequency level. It is feasible to convert time-aware requirements into frequency-aware requirements.

Fig. 5 outlines a model of converting frequency requirements from deadlines and WCET specified as timing constraints. In this modeling procedure, task requirements are modeled in the format of minimum frequency requirements. By the same token, security overhead incurred in security-sensitive applications should be integrated into the WCET measures, which are converted into frequency requirements. As a future research direction, tremendous efforts will be dedicated to ways of constructing frequency requirements from WCET values that are reliant on time spent in performing security services. Such security service times will be derived from security overhead model (see also Section IV-B).

We investigate multiple virtual machines running on a group of physical machines modeled as $C = \{c_1, c_2, ..., c_m\}$. Let us define a set of n virtual machines as $V = \{vm_1, vm_2, ..., vm_n\}$ running on machine c, where we have $c \in C$. Each virtual machine is denoted as a pair $vm_i = (a_i, f_i^{req})$, where a_i is the creation time of virtual machine vm_i, f_i^{req} is the minimum frequency requirement of virtual machine vm_i. The correlation between an overall tasks frequency requirement and each virtual machine's frequency requirement is formally expressed as:

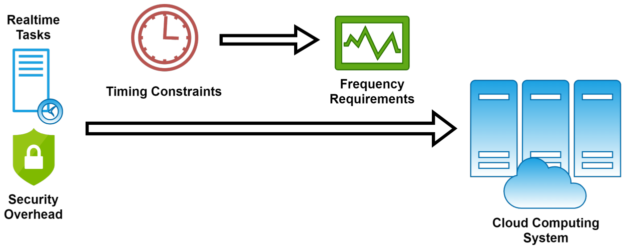

Fig. 5. A procedure of converting frequency requirements from deadlines and WCET specified as timing constraints. Task requirements are modeled in the format of minimum frequency requirements in clouds.

$$f_{V,c}^{req} = \sum_{i \in V} f_{i,c}^{req}. \quad (1)$$

We define a *security-related frequency requirement* co^{fre} as the frequency requirement that is derived from the corresponding security overhead. We layout in Eq. (2) the relation between an overall security-related frequency requirement and each virtual machine's security-related frequency requirement. The security-related frequency requirement of virtual-machine set V running on physical machine c is an accumulated measure of the security-related requirements of all the virtual machines in set V. Thus, we have

$$co_{V,c}^{fre} = \sum_{i \in V} co_{i,c}^{fre}. \quad (2)$$

Considering the minimum security-related frequency requirement $co_{V,c}^{fre}$, we show that physical machine c has a capability to support all the virtual machines in V without violating SLAs as long as the following requirement (3) holds.

$$f_{V,c}^{conf} \geq f_{V,c}^{req} + co_{V,c}^{fre}. \quad (3)$$

where $f_{V,c}^{conf}$ is a frequency configured for virtual-machine set V on physical machine c. To meet specified SLA requirements, one has to regulate the frequency $f_{V,c}^{conf}$ in a way to exceed a threshold of $f_{V,c}^{req} + co_{V,c}^{fre}$.

D. Security- and Frequency-aware DVFS Modeling

Fig. 6 unravels a high-level architecture of the security- and frequency-aware DVFS model or *SF-DVFS*, in which the frequency-aware DVFS, a security overhead model, and security services are seamlessly integrated.

In one of our recent studies [67], we proposed a frequency-aware DVFS model aiming to conserve energy consumption of tasks with QoS requirements. In our DVFS model, the energy consumption of processor c is calculated as:

$$E_c = \frac{\sum_{i \in V} \Gamma_i}{f_c^{max}} \left(\frac{P_c^{sta}}{r_c} + P_c^{dmax} (r_c)^2 \right). \quad (4)$$

where Γ_i is the total number of clock cycles of vm_i, f_c^{max} is the max frequency level of processor c, frequency retio r is the ratio between the current processor frequency f and the

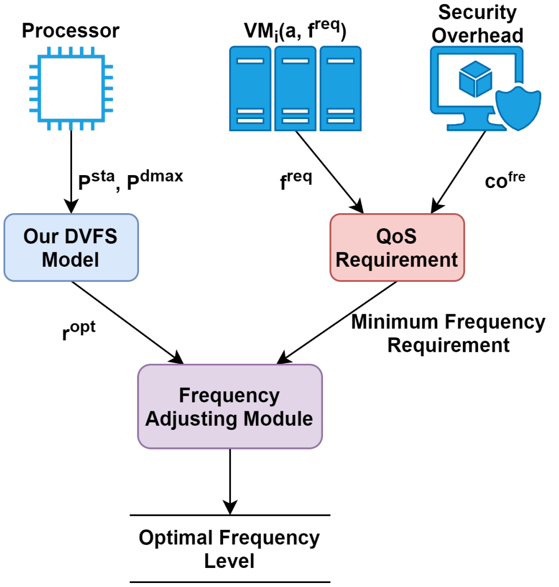

Fig. 6. The security- and frequency-aware DVFS model *SF-DVFS* integrates the frequency-aware DVFS, a security overhead model, and security services in the context of quality of services (QoS).

maximum frequency f^{max} held by a processor. P_c^{sta} is the static power of processor c; P_c^{dmax} is the maximum dynamic power of processor c.

Let r^{opt} be an optimal frequency ratio that curbs the energy consumption in the system. We obtain optimal ratio r^{opt} as:

$$r_c^{opt} = \sqrt[3]{\frac{P_c^{sta}}{2P_c^{dmax}}}. \qquad (5)$$

Given the optimal frequency ratio r^{opt} in Eq. (5), we derive the minimized energy consumption E^{opt} from this ratio r^{opt}, static power P_c^{sta}, and maximum dynamic power P_c^{dmax}. Thus, we have

$$E_c^{opt} = \frac{\sum_{i \in V} \Gamma_i}{f_c^{max}} \left(\frac{P_c^{sta}}{r_c^{opt}} + P_c^{dmax} \cdot (r_c^{opt})^2 \right). \qquad (6)$$

In the system architecture depicted in Fig. 6, the *QoS requirement module* outputs a minimum frequency requirement from two input parameters, namely, (1) the minimum frequency requirement f^{req} and (2) the security-related frequency requirement co^{fre} prescribed in virtual machines. On the left-hand side of the architecture, our *frequency-aware DVFS model* incorporates the static and maximum dynamic power constants to obtain an optimal frequency ratio r^{opt}. Finally, the *frequency adjusting module* compares the optimal frequency ratio and the overall minimum frequency requirement to configure the most appropriate frequency level to reduce the energy consumption of the virtual machines running on a physical machine.

To enhance the system architecture outlined in Fig. 6, we advocate for the following future research directions. First, practical VM consolidation and management policies should be blended with DVFS to build energy-efficient clouds running tasks with QoS requirements. Second, machine-learning-based prediction techniques are expected to boost the performance of the VM consolidation and management policy. Third, the security overhead (see also Section IV-B) largely depends on security levels. Hence, it is desirable to dynamically configure security levels to fulfill QoS requirements in our proposed security-aware energy management system. For example, if QoS requirements are permitted, security service instances with strong strengths should be elected to maximize security in clouds. Otherwise, security levels must be lowered to avert performance degradation.

V. Concluding Remarks

In this paper, we introduced the evolution of cloud computing systems from the three aspects - load balancing, energy conservation techniques, and security issues. We showed that the various types of scheduling policies orchestrate a diversity of resources to optimize resource utilization in clouds. Among all the energy-saving schemes for cloud computing, we focused on DVFS-based and VM-based scheduling solutions to offer energy savings in clouds.

As the research roadmap towards the security-aware energy management in clouds, there are four connected components: (1) security services, (2) security overhead models, (3) security- and frequency-aware QoS, and (4) security-enabled DVFS. Currently, we are in a process of developing a security-aware energy management system for cloud computing environments. Our novel energy management system is expect to achieve high security and energy efficiency in clouds by seamlessly integrating the security services, a security overhead model, and the security- and frequency-aware DVFS model. As a final remark, we emphasize that the development of a security-aware energy management system should incorporate underpinning techniques from multiple areas like machine learning solutions, DVFS techniques, real-time scheduling, security services, security strength evaluation, and security overhead analysis.

Acknowledgment

This work is supported in part by the U.S. National Science Foundation under Grants IIS-1618669, OAC-1642133, and CCF-0845257.

References

[1] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl, “Scientific cloud computing: Early definition and experience,” in *2008 10th ieee international conference on high performance computing and communications*. Ieee, 2008, pp. 825–830.

[2] B. Wilder, *Cloud architecture patterns: using microsoft azure.* ” O’Reilly Media, Inc.”, 2012.

[3] D. Reinsel, J. Gantz, and J. Rydning, “The digitization of the world: from edge to core,” *Framingham: International Data Corporation*, 2018.

[4] “Data Center Market – Global Outlook and Forecast 2018 – 2023,” Arizton 4577457, Tech. Rep., 6 2018.

[5] “Data center power market size, share trends analysis report by product (pdu, ups, busway), by end use (it telecom, bfsi, energy, healthcare, retail), by region, and segment forecasts, 2019 – 2025,” Tech. Rep.

[6] A. Marashi, “Power hungry: The growing energy demands of data centers,” https://www.vxchnge.com/blog/power-hungry-the-growing-energy-demands-of-data-centers, 2019.

[7] "BP Statistical Review of World Energy 68th edition," BP p.l.c., Tech. Rep., 2019.

[8] R. Latif, H. Abbas, S. Assar, and Q. Ali, "Cloud computing risk assessment: a systematic literature review," in *Future information technology*. Springer, 2014, pp. 285–295.

[9] A. N. Khan, M. M. Kiah, M. Ali, S. A. Madani, S. Shamshirband *et al.*, "Bss: block-based sharing scheme for secure data storage services in mobile cloud environment," *The Journal of Supercomputing*, vol. 70, no. 2, pp. 946–976, 2014.

[10] A. Corradi, M. Fanelli, and L. Foschini, "Vm consolidation: A real case based on openstack cloud," *Future Generation Computer Systems*, vol. 32, pp. 118–127, 2014.

[11] W. Kim, M. S. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core dvfs using on-chip switching regulators," in *2008 IEEE 14th International Symposium on High Performance Computer Architecture*. IEEE, 2008, pp. 123–134.

[12] C. Gu, Z. Li, H. Huang, and X. Jia, "Energy efficient scheduling of servers with multi-sleep modes for cloud data center," *IEEE Transactions on Cloud Computing*, 2018.

[13] T.-Y. Chen, T.-T. Yeh, H.-W. Wei, Y.-X. Fang, W.-K. Shih, and T.-s. Hsu, "Cacheraid: An efficient adaptive write cache policy to conserve raid disk array energy," in *2012 IEEE Fifth International Conference on Utility and Cloud Computing*. IEEE, 2012, pp. 117–124.

[14] M. Gieles, S. P. Zwart, H. Baumgardt, E. Athanassoula, H. Lamers, M. Sipior, and J. Leenaarts, "Star cluster disruption by giant molecular clouds," *Monthly Notices of the Royal Astronomical Society*, vol. 371, no. 2, pp. 793–804, 2006.

[15] S. Zaman and D. Grosu, "A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 129–141, 2013.

[16] A. Kaur and S. Kinger, "Temperature aware resource scheduling in green clouds," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2013, pp. 1919–1923.

[17] P. Arroba, J. L. Risco-Martín, J. M. Moya, and J. L. Ayala, "Heuristics and metaheuristics for dynamic management of computing and cooling energy in cloud data centers," *Software: Practice and Experience*, vol. 48, no. 10, pp. 1775–1804, 2018.

[18] Y. Ran, H. Hu, X. Zhou, and Y. Wen, "Deepee: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 645–655.

[19] M. Qiu, L. Zhang, Z. Ming, Z. Chen, X. Qin, and L. T. Yang, "Security-aware optimization for ubiquitous computing systems with seat graph approach," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 518–529, 2013.

[20] J. Li, X. Chen, M. Li, J. Li, P. P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE transactions on parallel and distributed systems*, vol. 25, no. 6, pp. 1615–1625, 2013.

[21] G. Sivathanu, C. P. Wright, and E. Zadok, "Ensuring data integrity in storage: Techniques and applications," in *Proceedings of the 2005 ACM workshop on Storage security and survivability*, 2005, pp. 26–36.

[22] N. K. Gill and S. Singh, "A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers," *Future Generation Computer Systems*, vol. 65, pp. 10–32, 2016.

[23] S. K. Madria and B. Bhargava, "A transaction model to improve data availability in mobile computing," *Distributed and Parallel Databases*, vol. 10, no. 2, pp. 127–160, 2001.

[24] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE transactions on information forensics and security*, vol. 8, no. 12, pp. 1947–1960, 2013.

[25] K. Yang, Z. Liu, X. Jia, and X. S. Shen, "Time-domain attribute-based access control for cloud-based video content sharing: A cryptographic approach," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 940–950, 2016.

[26] S. Abrishami, M. Naghibzadeh, and D. H. Epema, "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 158–169, 2013.

[27] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds," *Future Generation Computer Systems*, vol. 48, pp. 1–18, 2015.

[28] S. Malik and F. Huet, "Adaptive fault tolerance in real time cloud computing," in *2011 IEEE World Congress on services*. IEEE, 2011, pp. 280–287.

[29] T. Xie and X. Qin, "Security-aware resource allocation for real-time parallel jobs on homogeneous and heterogeneous clusters," *IEEE transactions on parallel and distributed systems*, vol. 19, no. 5, pp. 682–697, 2008.

[30] X. Zhu, X. Qin, and M. Qiu, "Qos-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters," *IEEE transactions on Computers*, vol. 60, no. 6, pp. 800–812, 2011.

[31] Z. Zong, A. Manzanares, X. Ruan, and X. Qin, "Ead and pebd: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters," *IEEE Transactions on Computers*, vol. 60, no. 3, pp. 360–374, 2010.

[32] X. Zhang, J.-J. Lu, X. Qin, and X.-N. Zhao, "A high-level energy consumption model for heterogeneous data centers," *Simulation Modelling Practice and Theory*, vol. 39, pp. 41–55, 2013.

[33] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Environment-conscious scheduling of hpc applications on distributed cloud-oriented data centers," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 732–749, 2011.

[34] Q. Fettes, M. Clark, R. Bunescu, A. Karanth, and A. Louri, "Dynamic voltage and frequency scaling in nocs with supervised and reinforcement learning techniques," *IEEE Transactions on Computers*, vol. 68, no. 3, pp. 375–389, 2018.

[35] S. Maroulis, N. Zacheilas, and V. Kalogeraki, "A framework for efficient energy scheduling of spark workloads," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 2614–2615.

[36] D. Suleiman, M. Ibrahim, and I. Hamarash, "Dynamic voltage frequency scaling (dvfs) for microprocessors power and energy reduction," in *4th International Conference on Electrical and Electronics Engineering*, vol. 12, 2005.

[37] K. Duan, S. Fong, W. Song, A. V. Vasilakos, and R. Wong, "Energy-aware cluster reconfiguration algorithm for the big data analytics platform spark," *Sustainability*, vol. 9, no. 12, p. 2357, 2017.

[38] S. Ibrahim, T.-D. Phan, A. Carpen-Amarie, H.-E. Chihoub, D. Moise, and G. Antoniu, "Governing energy consumption in hadoop through cpu frequency scaling: An analysis," *Future Generation Computer Systems*, vol. 54, pp. 219–232, 2016.

[39] X. Xu, W. Dou, X. Zhang, and J. Chen, "Enreal: An energy-aware resource allocation method for scientific workflow executions in cloud environment," *IEEE Transactions on Cloud Computing*, vol. 4, no. 2, pp. 166–179, 2015.

[40] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin, and X. Liu, "Real-time tasks oriented energy-aware scheduling in virtualized clouds," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 168–180, 2014.

[41] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 810–819, 2016.

[42] C.-H. Hsu, K. D. Slagter, S.-C. Chen, and Y.-C. Chung, "Optimizing energy consumption with task consolidation in clouds," *Information Sciences*, vol. 258, pp. 452–462, 2014.

[43] H. Chen, X. Zhu, H. Guo, J. Zhu, X. Qin, and J. Wu, "Towards energy-efficient scheduling for real-time tasks under uncertain cloud computing environment," *Journal of Systems and Software*, vol. 99, pp. 20–35, 2015.

[44] S. Shin, Y. Kim, and S. Lee, "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2015, pp. 814–819.

[45] Y. Ge and G. Wei, "Ga-based task scheduler for the cloud computing systems," in *2010 International Conference on Web Information Systems and Mining*, vol. 2. IEEE, 2010, pp. 181–186.

[46] D. Liu and N. Han, "An energy-efficient task scheduler in virtualized cloud platforms," *International Journal of Grid and Distributed Computing*, vol. 7, no. 3, pp. 123–134, 2014.

[47] B. Dorronsoro, S. Nesmachnow, J. Taheri, A. Y. Zomaya, E.-G. Talbi, and P. Bouvry, "A hierarchical approach for energy-efficient scheduling of large workloads in multicore distributed systems," *Sustainable Computing: Informatics and Systems*, vol. 4, no. 4, pp. 252–261, 2014.

[48] Y.-K. Kwok and I. Ahmad, “On multiprocessor task scheduling using efficient state space search approaches,” *Journal of Parallel and Distributed Computing*, vol. 65, no. 12, pp. 1515–1532, 2005.

[49] M. R. Mohamed and M. H. Awadalla, “Hybrid algorithm for multiprocessor task scheduling,” *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 3, p. 79, 2011.

[50] T. Mathew, K. C. Sekaran, and J. Jose, “Study and analysis of various task scheduling algorithms in the cloud computing environment,” in *2014 International conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2014, pp. 658–664.

[51] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, “Cloud computing resource scheduling and a survey of its evolutionary approaches,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1–33, 2015.

[52] Z. Tang, L. Jiang, J. Zhou, K. Li, and K. Li, “A self-adaptive scheduling algorithm for reduce start time,” *Future Generation Computer Systems*, vol. 43, pp. 51–60, 2015.

[53] G.-n. Gan, T.-l. Huang, and S. Gao, “Genetic simulated annealing algorithm for task scheduling based on cloud computing environment,” in *2010 International Conference on Intelligent Computing and Integrated Systems*. IEEE, 2010, pp. 60–63.

[54] P. Kumar and A. Verma, “Independent task scheduling in cloud computing by improved genetic algorithm,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 5, 2012.

[55] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, “A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing,” *Ieee Access*, vol. 3, pp. 2687–2699, 2015.

[56] B. Schneier and D. Whiting, “Fast software encryption: Designing encryption algorithms for optimal software speed on the intel pentium processor,” in *International Workshop on Fast Software Encryption*. Springer, 1997, pp. 242–259.

[57] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the key scheduling algorithm of rc4,” in *International Workshop on Selected Areas in Cryptography*. Springer, 2001, pp. 1–24.

[58] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*. john wiley & sons, 2007.

[59] W. Liu, S. Peng, W. Du, W. Wang, and G. S. Zeng, “Security-aware intermediate data placement strategy in scientific cloud workflows,” *Knowledge and information systems*, vol. 41, no. 2, pp. 423–447, 2014.

[60] H. Chen, X. Zhu, D. Qiu, L. Liu, and Z. Du, “Scheduling for workflows with security-sensitive intermediate data by selective tasks duplication in clouds,” *IEEE Transactions on Parallel and distributed systems*, vol. 28, no. 9, pp. 2674–2688, 2017.

[61] P. Zhou, J. Huang, X. Qin, and C. Xie, “Pars: A popularity-aware redundancy scheme for in-memory stores,” *IEEE Transactions on Computers*, vol. 68, no. 4, pp. 556–569, 2018.

[62] D. Borthakur, R. Schmidt, R. Vadali, S. Chen, and P. Kling, “HDFS RAID,” 2010, technical Talk. Yahoo! Developer Network.

[63] D. Ford, F. Labelle, F. Popovici, M. Stokely, V. Truong, L. Barroso, C. Grimes, and S. Quinlan, “Availability in Globally Distributed Storage Systems,” in *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI’10)*. USENIX, 2010, pp. 61–74.

[64] J. Huang, P. Zhou, X. Qin, Y. Wang, C. Xie, and J. Jose, “Optimizing erasure-coded data archival for replica-based storage clusters,” *The Computer Journal*, vol. 62, no. 2, pp. 247–262, 2019.

[65] H. Weatherspoon and J. D. Kubiatowicz, “Erasure Coding vs. Replication: A Quantitative Comparison,” in *Peer-to-Peer Systems*. Springer, 2002, pp. 328–337.

[66] M. Manasse, C. Thekkath, and A. Silverberg, “A Reed-solomon Code for Disk Storage, and Efficient Recovery Computations for Erasure-coded Disk Storage,” *Proceeding in Informatics*, pp. 1–11, 2009.

[67] J. Mao, T. Bhattacharya, X. Peng, T. Cao, and X. Qin, “Modeling energy consumption of virtual machines in dvfs-enabled cloud data centers,” in *2020 39th IEEE International Performance Computing and Communications Conference*. IEEE, 2020.