IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

A Survey on Energy-Aware Fault Tolerant Strategies in Cloud Computing

Kamaljit Kaur[1*], Kuljit Kaur[2]

[1]Department of Computer Engineering & Technology, Guru Nanak Dev University, Amritsar, Punjab, India
[2]Department of Computer Science, Guru Nanak Dev University, Amritsar, Punjab, India

Corresponding Author: kamal.aujla86@gmail.com, Tel.: +91-84270-06522

DOI: https://doi.org/10.26438/ijcse/v7i5.787800 | Available online at: www.ijcseonline.org

Accepted: 23/May/2019, Published: 31/May/2019

***Abstract*—** With the advent of technology, the computational demands of users are increasing day by day. Cloud Computing is among the most trending technologies satisfying the computationally intensive demands of users. Cloud computing has exploited virtualization technology to provide on demand provisioning of resources, results in increased complexity of cloud infrastructure, thus faults are inevitable. These faults may result in failure causing serious loss to the organizations. Techniques used for fault management usually require additional resources increasing the consumption of energy. Moreover, cloud infrastructure also consumes a lot of energy and is the major contributor to carbon content. Growing demands and limited renewable resources had led to serious energy crises. Thus energy efficient fault tolerant solutions are needed to tolerate faults and provide reliable, scalable and flexible availability of cloud services, preventing system failure and minimizing energy consumption at the same time. Fault tolerance and energy efficiency are the crucial issues which must be simultaneously considered in order to ensure availability, performance, and reliability of the cloud computing services. This paper describes the basic concepts of faults, errors, and failures. It also discusses different fault tolerance strategies and the trade-off between energy efficiency and fault tolerance.

***Keywords*—**Checkpointing, Energy efficiency, Fault Tolerance, Migration, Replication

INTRODUCTION

In recent years, the rapid growth of IT infrastructure is triggering the need for high computational resources, leading towards the formation of huge data centres and increasing the energy consumption. A solution to this is the application of cloud computing. Cloud Computing is among the most trending technologies which fulfils the need of computationally intensive demands of users. Cloud Computing allow the clients to access large number of resources which includes storage space, computation power, network , applications and services as requested by users over the global network[12]. Cloud Computing introduces the concept of everything as a Service, mostly referred as XaaS where X can be Software, Infrastructure, Hardware, Platform, Data or Business etc. [13]. The user need not to worry about the initial investments on the resources since cloud computing provides an approach for leveraging computing resources with same ease as utilizing common utilities such as natural gas, water , electricity supply on pay per use basis through the concept of utility oriented computing thus ensuring Quality of Service at the same time [14]. Cloud computing has exploited virtualization technology to satisfy the computational demands of the clients. Due to the rising service demands of the users the cloud infrastructure is increasing day by day which consumes lot of energy during its operation and is the major contributor to carbon content which is harmful for both human health and environment. The huge servers hosted in the datacentres dissipate a large amount of heat and need cooling systems to regulate the optimal temperature which in turn also adds to the carbon footprints and results in energy crises due to high power consumption.

A data centre under cloud infrastructure comprises of thousands of physical nodes and single physical node consists of multiple virtual machine instances, each having its own operating system and work isolated from each other thus the complexity of cloud infrastructure is increasing due to which faults are inevitable [15]. In any enterprise data centre, the continuous availability and reliability of services is of foremost concern and any interruption due to faults may lead to serious consequences besides the loss may not be recovered at times. This necessitates the need for fault tolerant solutions. This paper gives a review on fault tolerance with primary focus on energy aware fault tolerance strategies.

In Section II general Energy efficient fault tolerant cloud computing architecture is discussed. Section III, discusses the basic concepts of fault, error and failures. In section IV, fault tolerance strategies are classified as reactive and proactive fault tolerance. Section V discuss the various energy management strategies in cloud computing that can be merged with the fault handling strategies in order to obtain maximum energy saving. Trade-off between fault tolerance and energy efficiency is discussed in section VI and comparison of fault tolerance strategies is also highlighted. The survey is concluded in the end.

ENERGY EFFICIENT FAULT TOLERANT CLOUD COMPUTING ARCHITECTURE

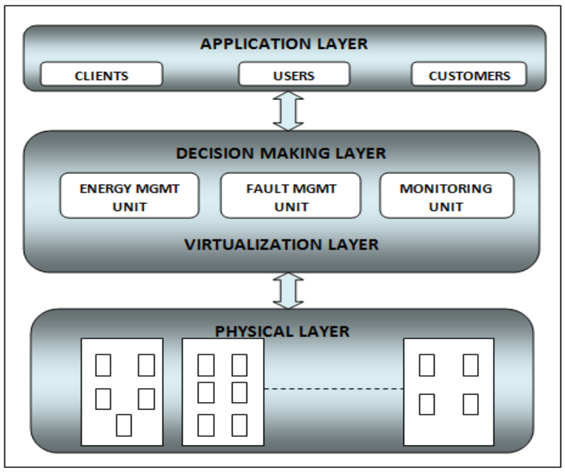

Figure 1. Energy efficient fault tolerant cloud computing architecture

To resolve the above specified issues of rising energy crisis and fault rate, there is a need for optimized energy aware and failure-aware strategies. To implement these strategies, a cloud computing architecture is required. figure 1 depicts layered view of cloud computing architecture to save energy along with managing faults effectively.

Application Layer: This layer is liable for delivering services to the user. User submits request to broker which acts as an interface between the cloud consumer and provider. It also ensures that the desired quality of service must be provided in accordance with SLA constraints.

Decision Making Layer: This is the crucial layer of cloud computing architecture comprising of various modules. The energy management module ensures the selection of optimal energy management strategy as discussed in section IV in order to save maximum amount of energy. The fault tolerance module ensures the selection of efficient fault tolerance strategy as explained in section III in order to increase the reliability and availability of cloud services. Monitoring Module keeps track of different activities, components in order to take appropriate decision for any undesirable activity.

Virtualization Layer: This layer overlaps with the decision making layer and is responsible for managing the virtual resources. Virtual machine migration and consolidation is performed to manage failure and energy consumption as directed by the fault tolerance and energy management modules of decision making layer.

Physical Layer: This layer comprises of physical machines for gratifying the computational or storage demands of cloud users. A data centre under cloud infrastructure comprises of thousands of physical nodes and single physical node consists of multiple virtual machine instances each having its own operating system and work isolated from each other [15].

BASIC CONCEPTS(FAULTS, ERROR AND FAILURES)

A. Faults

Fault represents any imperfection at the ground layer of abstraction that causes unwanted modification in the state of system [6]. A system is a collection of interoperating components which interacts with other systems and subsystems [3]. According to this survey system represents cloud computing infrastructure including hardware (network, servers) and software (cloud management system, appliances) components [18]. Faults can be active or dormant. Fig 1 illustrates that the activation of faults causes error [4]. Faults can be subdivided into numerous types on the basis of its behavior and consequences. Permanent, intermittent and transient are the type of faults that can be distinguished on the basis of fault behavior [8]. Permanent faults are irreversible in nature. Intermittent faults are referred as a repeated malfunction occurring in the same location at random time intervals. Transient faults are temporary in nature occurring at different locations which can be resolved by restarting the system again [9]. Crash faults and byzantine faults can be distinguished on the basis of their consequences. Crash faults halts the system's functionality and can even lead to shutdown such as a power outage. Byzantine faults on the other hand causes the system to behave arbitrarily producing incorrect output such as software bugs, malicious software [10] [11].

Figure 2. Relationship between fault, error and failure

B. Error

An error causes a system to deviate from its normal functionality and often results in failure [7]. Errors can be propagated among different components within the system causing correlated failures [4].Error can be classified as detected and undetected. An error is said to be undetected if the transition in system state remains unidentified [3].

C. Failures

Failure is an event, happens when theobtained results deviates from correct results. Failure refers to the state of misbehavior perceived in case of incorrect output [1] [3]. Activation of fault causes an error which in turn results in failure. At infrastructure level overutilization of hardware resources such as processor, memory modules, hard disks, power supply etc. can cause failure [16]. Server, network and power are primary resource failures considered at this level [2]. At application level logical errors in application development, misconfigurations, malicious software, etc. may lead to failures. The operating system present on virtual machine can crash due to bugs in kernel causing temporary loss of services. This degrades performance of the system. Applications running on the system can also abruptly terminate. Such events cause the host to slow down indefinitely [17]. When the performance of the system degrades providing slow services, this event is termed as a partial failure [4].

I. Fault Tolerance Strategies

Fault Tolerance is the ability to tolerate faults and provides required services preventing system failure. Fault Tolerance ensures service availability, reliability and serviceability [21]. Fault tolerance can be separated in different phases like fault detection, fault avoidance and fault recovery phase. In fault avoidance phase, probability of occurrence of fault is predicted and actions are taken in advance to avoid failures. Fault recovery phase lowers the effect of failure taking appropriate measures to recover the loss. On the basis of these phases fault tolerance can be categorized as

- Proactive Fault Tolerance
- Reactive Fault Tolerance

These categories are further sub classified in fig 3.

Reactive fault tolerance strategies are used in case where failure has already occurred and recovery mechanisms are implemented to recover the loss for providing continuous and reliable services. The downtime prevails for a certain time interval due to the occurrence of failure in reactive approaches. These strategies require additional resources to implement recovery mechanisms as in case of replication same job is executed on multiple hosts to provide backup in case of failure.

Checkpointing is a reactive fault tolerant strategy in which system's state is stored at different locations. If host fails, the safe state can be retained and execution can be continued from the same state instead of rebooting the system from initial state [24] [26]. Coordinated checkpointing is a technique in which synchronization between tasks exists when checkpointing decisions are made whereas in uncoordinated checkpointing there is no synchronization between the tasks when checkpointing decisions are made, rather independent checkpoints are created. Time saving checkpointing mechanism is discussed in [37] entails lesser storage space, provides fault tolerance along with efficient resource utilization.

Replication is a mechanism which create replicas of same process. Replicas must be executed at different locations to provide backup in case of failure [25]. In active replication, same process is executed on all the replicas simultaneously so that the system performs the desired functionality. Within passive replication a single processing component or main replica performs the desired functionality while the back-up replicas conserves the machine's state throughout the usual execution cycles. Identical copies from back-up take control of the execution procedure provided that the main replica fails. The BFTCloud model [36] used replication mechanism to provide reliable services ensuring Quality of Service at the same time. Multi-level Replication mechanism is proposed in [38] to ensure reliability and availability of high priority applications.

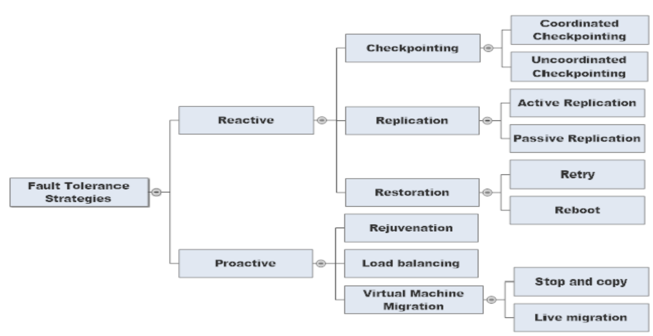

Figure 3. Fault Tolerance Strategies in Cloud Computing

Retry and reboot strategies are generally used in the presence of transient faults that can be resolved by restarting the system [9] .

Proactive fault tolerance strategies are used to prevent the occurrence of failure. Measures are taken to avoid faults by predicting the behavior of system at regular time intervals and decisions are made on the basis of predictions. Proactive approach ensures continuous and reliable service availability with negligible downtime. System's resources are constantly monitored to estimate the possibility of fault occurrence.

In Software Rejuvenation approach application is restarted after regular time interval to prevent software aging and avoid failures [39]. An automation technique that activates rejuvenation in case of any abrupt behavior identified in system is discussed in [40] to avoid failure. Warm-VM reboot, an efficient Rejuvenation approach is proposed in [41] to avoid aging and performance degradation of system.

Virtual Machine migration is optimization strategy in which deteriorating virtual machine is migrated to healthier physical machine without shutting down any machine. Virtual Machine migration ensures transfer of virtual machine to different physical machine without causing service disruption. Virtual Machine migration can further be classified as stop and copy and live migration.

In stop and copy migration the source virtual machine is stopped and its entire content is copied and transferred to destination after which execution is resumed at destination end therefore downtime prevails for a certain time period. In live migration approach the state of source virtual machine is repeatedly copied and migrated to the destination without causing service interruption therefore negligible downtime prevails in this case.

In [23] PCFT approach is proposed to anticipate a deteriorating physical machine on the basis of CPU temperature which is migrated to optimal physical machine to avoid failure. Enlightened post copy mechanism is discussed in [42] allows faster transfer of virtual machines' state providing an improved version of live migration. In [43] Adaptive pre-paging technique is used to remove transmission of identical pages and removal of free memory pages, done by using DSB technique. In [44] Remote Page-fault Filtering mechanism is proposed. This approach reduced faults by allocating new memory pages rather than fetching them from host. A generic algorithm based on process migration to handle faults is proposed in [45] checkpointing and restart policies are deployed to recover loss in case of failure. Virtual Machine migration strategy is used as fault tolerance approach for reconfiguring RVDM infrastructure in [46]. Faulty node is determined on the basis of reliability.

Comparison of different fault tolerance strategies is shown in Table 1 highlighting the various quality of service parameters and their relationship with fault tolerance and energy efficiency is discussed below.

- Energy efficiency: Energy efficiency refers to the consumption of minimum amount of energy to perform the requested task.
- Downtime: The time for which a system is unable to provide the desired services is defined as downtime. Energy efficient fault tolerant systems must have negligible downtime consuming minimum energy at the same time.
- Mean time to repair: The time required for failure recovery is referred as mean time to repair also known as recovery time. Energy efficient fault tolerant systems must utilize minimum resources, consuming minimum energy while providing fast failure recovery.
- Mean Time Between Failures(MTBF): The time between two consecutive failures is referred as MTBF. This parameter must be high for providing reliable availability of services.
- Overhead: The overheads involved for implementing desired algorithm. Energy efficient fault tolerant solutions should involve minimum overheads.

Table 1. Comparison of fault tolerance strategies

Strategies	Reduced Downtime	Mean time to repair	Energy efficient	ReducedMTBF	Reduced Overhead
Checkpointing	✓	✓	✓	X	✓
Replication	✓	✓	✓	X	X
Software Rejuvenation	X	✓	X	✓	X
Load Balancing	✓	X	✓	✓	✓
VM Migration	✓	X	✓	✓	✓

II. Energy Management Starategies

Energy management strategies are deployed to utilize energy efficiently so that energy saving can be considerably increased controlling the rising energy crisis. It also reduces the carbon footprints by improving efficient utilization of hardware as well as software resources. Moreover it also saves organizations cost by promoting the optimal use of cooling equipment and underutilized resources. According to energy oriented scenario by year 2020 [65] up to 223479GWh energy and $95*10^9 can be saved.

This section describes the various energy management strategies that can be used along with fault tolerance strategies discussed in section IV to minimize energy consumption. Energy management techniques could be extensively grouped according to design and performance aspects as

- Dynamic Energy Management
- Thermal Management
- Dynamic Voltage and frequency scaling

These categories are further sub-classified in fig 4.

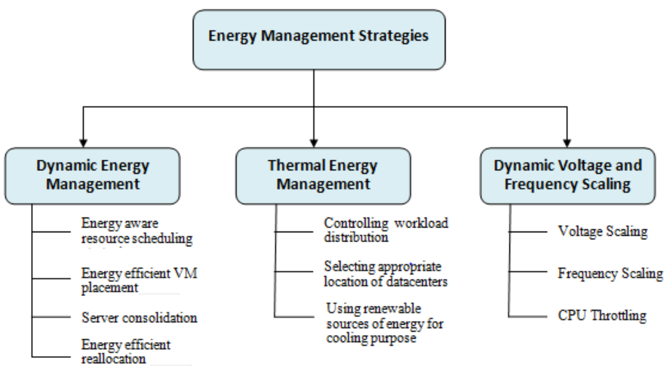

Figure 4. Energy Management Strategies in Cloud Computing

A. Dynamic Energy Management

Cloud computing datacenters includes hardware and software components [18]. In dynamic energy management strategies the idle machines are dynamically powered down or placed in inactive mode to minimize energy usage. Energy efficiency can be achieved by using:

- Energy aware resource scheduling strategies
- Energy efficient VM placement strategies
- Server consolidation

Energy aware resource scheduling strategies comprises of various algorithms that cloud providers deploy to allocate variable resources in a virtual environment in order to provide the desired services to users while consuming minimum amount of energy at the same time.

Various energy efficient resource scheduling algorithms are suggested in [71] based on greedy heuristics to save maximum amount of energy and migration based fault tolerance strategy is implemented to prevent failures. The proposed approach minimize power consumption and number of migrations while providing the desired quality of service. Meta heuristics algorithms can also be combined with migration strategies to minimize power consumption. Ant Colony algorithm is used with live migration approach in [74] to implement dynamic VM consolidation for saving energy. Live migration strategy used in this approach to prevent failures and provide negligible downtime, enhance the continuous availability and reliability of resources proactively. The key goal is to decrease energy consumption and prevent partial failure, which arises due to performance degradation of the system.

Energy efficient VM placement strategies deal with the mapping of virtual machines on the host while considering energy consumption and the user requirements simultaneously.

Optimal mapping of virtual machines to the underlying cloud computing infrastructure is suggested in [75]. Virtual machines are allocated to physical node to minimize communication overheads, and power consumption. Flexible nature of this approach allow user to specifically select the power model. Distinct resources are continuously monitored to identify the abnormalities so that appropriate measures can be taken to avoid failure. VM migration technique in used in [72] along with efficient utilization of energy. Underutilized servers are identified according to dynamic threshold value. Virtual Machines (VM) from underutilized servers are migrated and idle servers are switched off to save energy. Heuristic algorithm based virtual machine placement policies with time and bandwidth as quality of service requirement are compared to save energy and to reduce migration cost. Cloud computing technology provides huge computational services to user that involve huge amount of data processing. In [77], S. Wang *et. al.* has given VM placement policy to provide data intensive services by considering energy as the prime parameter. Particle Swarm Optimization algorithm is used to provide optimal solution for VM allocation and ensures that the desired services are provided in accordance with the QoS constraints.

Power stability algorithm is proposed in [79] for allocating VM to physical nodes to obtain better stability factor. Allocation strategy is implemented to reduce energy consumption and total number of migrations. Performance is analyzed in cloudsim environment and saved 23% energy as compared to MBFD algorithm.

Considerable amount of energy can be saved by deploying server consolidation strategy in which the workload is migrated to the optimal servers so that underutilized severs can be turned off to reduce energy consumption.

First Fit algorithm is used in [19] for initial allocation of resources. The dynamic behavior of workload may lead to failure due to overutilization of host resources hence migration approach is implemented to prevent such failure. Moreover replication strategy is also deployed to recover lost data in case of failure. Workload Consolidation is triggered after regular time intervals to save thoroughgoing energy consumption. VMs are consolidated to reduce cost and power consumption in [73]. The experiment is conducted using heterogeneous nodes and irregular workload is applied which may escalate response time causing failure. Single and multiple VMs are migrated based on deterministic algorithms in accordance with the SLA constraints to prevent failures occurring due to dynamic nature of workload. This approach use maximum correlation policy for VM selection. The algorithm designed in [76] analyze the relationship between routing and energy efficiency. This algorithm suggest the ideal number of switches (network elements used for connecting host) to be used to save maximum amount of energy in cloud datacenters. Fat tree topology is used to handle network failures. This paper concentrated on traffic engineering domain to reduce energy consumption. Energy consumption is reduced by 50% using this approach.

B. Thermal Management

Thermal Management refers to the capability of controlling the temperature by reducing the amount of heat dissipated by the system. Due to the rapid growth of IT infrastructure heat dissipation has increased having worst impact on the reliability of system. The variable workload also adds to the heat dissipation in datacenters. In order to regulate optimal temperature cooling systems are required which consumes high amount of energy. Cooling systems used to regulate the optimal temperature of datacenters contributes to about 25-50% of power consumption [66]. Energy can be saved by:

- Controlling workload distribution
- Selecting appropriate location of datacenters.
- Using renewable sources of energy for cooling purpose.

Automation of thermal load management systems can be achieved by estimating the variable workload distribution and the thermal configuration of datacenter based on the information conveyed by the heat sensitive sensors. Weatherman prototype is projected in [78] based on machine learning methods that predict the thermal behavior of datacenters. Workload is assigned using placement heuristics to minimize cost and power consumption. VMAP+ technique is projected in [70] that use virtual machine consolidation to save maximum amount of energy. Decisions are made on the basis of information obtained from various sensors that sense numerous thermal activities of cloud computing infrastructure. This approach predict the irregular heat production at different areas in datacenter and suggest the view of heat imbalance.

Locations having lower temperature geographically should be selected for creating datacenters as cooler locations provide low-cost cooling. The availability of sustainable energy resources such as solar and wind energy is the key challenge in these locations [80]. Realistic energy consumption model is used to estimate energy utilization on the basis of Minimal Percentage Supply metric.

Non renewable resources are a major source of power production in India. It is estimated that the demand for these resources will rise to 75% till year 2020 and 81% till year 2040 [67] which will ultimately increase the carbon footprints. Thus there exist a requirement to use sustainable resources to reduce power consumption. Strategies exploring the use of renewable resources in cloud infrastructure are discussed in [68]. The energy required by datacenter is estimated. It is evaluated that up to 30% energy can be saved by using systems based on direct current instead of alternate current. The effects of reducing power consumption while considering profit margins are highlighted in [69].The term power usage effectiveness is used for overall energy consumption. Wind turbines are used for producing electricity. The results are compared using different pricing models.

C. Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling is the method of reducing energy utilization by lowering frequency and voltage. Decreasing voltage or frequency decreases the energy consumed. Energy efficiency at this level can be achieved by scaling down the performance of various components such as processors, storage space, memory etc.

Although DVFS provides significant energy cut down and saves considerable amount of energy, but this may also increase fault rate and performance degradation hence this approach can be wisely used for applications where the reliability of services is not the major concern. The effect of DVFS technique on fault rate is observed in [81]. Results demonstrate that frequency and voltage scaling approach for controlling energy utilization can considerably alter the performance of system and decrease its reliability so energy savings through frequency and voltage scaling needs to be carefully weighted while maintaining required level of reliability. Frequency scaling technique is suggested in [82] which determines the direct energy consumption of nodes at variable frequencies in order to determine the optimal frequency to save maximum energy while ensuring minimum performance degradation at the same time.

The work recommended by [87] use EDA (Earliest Deadline first algorithm for task scheduling with major focus on reducing energy utilization and augmenting fault handling. Transient faults are handled by the use of checkpointing strategy. The voltage supply is modified on the basis of available storage energy in the system and jobs are executed at different voltage levels to save maximum amount of energy. Cielo framework based on game theory is presented in [83] that use DVFS technique for energy saving. The main objective of this framework is to allocate resources dynamically and maximize energy saving and resource utilization. NBTI (Negative Bias Temperature Instability) based DVFS technique is proposed in [84]. Major aim is to reduce processor energy consumption and increase throughput. This approach requires additional resources to monitor voltage and temperature. Calibration at regular intervals enhances the monitoring in order to estimate the possibility of failure. Energy consumption is reduced by 61% using this approach.

TRADE-OFF BETWEEN FAULT TOLERANCE AND ENERGY EFFICIENCY

Fault Tolerance is the capability to offer trustworthy services consistently even in the existence of faults. Fault tolerance can be achieved by implementing proactive and reactive fault tolerant strategies as discussed in section III. These mechanisms require additional resources to reduce failures. The increase in these resources triggers the energy consumption having worst impact on the enterprise’s profit margins and on environment. For Example checkpointing technique needs extra storage space to save logs and checkpoints. Snooze [19] used replication mechanism to achieve fault tolerance which uses extra resources to successfully complete the same job.

The cloud infrastructure consumes lot of energy. The huge servers hosted in the data centers dissipate a large amount of heat and need cooling systems to regulate the optimal temperature which often augments the energy utilization. Energy consumption can be reduced by utilizing system resources effectively such as turning off idle resources to reduce energy consumption. For instance virtual machine consolidation mechanism leverages live migration to consolidate virtual machines and idle nodes are powered down to increase energy saving [20]. In act of host failure due to some fault at the infrastructure or application level and if there is no recovery mechanism implemented then all the tasks performed by the machine needs to be restarted which further increases energy consumption. Thus fault tolerance and energy consumption must be simultaneously considered to achieve overall gain and to ensure reliable, continuous, scalable and flexible availability of cloud services.

This paper also reviews the existing research on energy-aware fault tolerance strategies. On one hand fault tolerance is achieved using virtual machine migration by migrating deteriorating virtual machine to healthier machine ensuring reliable, scalable, flexible and uninterrupted availability of services. On the other hand energy efficiency is achieved by utilizing minimum resources and switching off the idle resources.

In Table 2 techniques are described briefly and compared on the basis of parameters such as

- Migration time: Overall time required for transferring a virtual machine from source to destination. Energy efficient fault tolerant systems must utilize minimum resources, consuming minimum energy having lower total migration time to provide fast migration.
- Down time: The time for which a system is unable to provide the desired services is defined as downtime. Energy efficient fault tolerant systems must have negligible downtime consuming minimum energy at the same time.

Table 3 briefly discusses the energy aware fault tolerance techniques in which checkpoints are used for recovery in case of failure. The state of system is continuously stored in coordinated or uncoordinated manner at some reliable location from where it can be restored whenever required. Checkpointing is a time saving approach as execution can be resumed from the save state or checkpoints unlike retry and reboot where complete rollback is required as whole process needs to be executed again. Energy efficiency in this approach can be achieved by using optimal number of checkpoints which require minimum storage space and utilizes systems resources efficiently. Table 3 also compares the various energy efficient checkpointing strategies on the base of parameters:

- Checkpointing frequency: Checkpoint frequency is defined as the interval between the checkpoints. Checkpoints can be stored synchronously having regular time interval or asynchronously with irregular time intervals. It is a key factor which impacts the time required by system to recover from an unexpected failure. In case of longer intervals between checkpoints if the system crashes, more time will be needed to recover the loss.

Table 2. Energy-aware virtual machine migration strategies for fault tolerance

Authors name, year	Technique	Description	Benefits	Drawbacks
Jin et al.[27] 2009	Adaptive compression of migrated data.	The information is compressed and transmitted at every step. An appropriate decompression method is used to obtain the original information at the target node.	Improved resource utilization, reduces migration time and saves energy.	Overhead increased due to compression algorithm
M. R. Hines et al.[43] , 2009	Dynamic self-ballooning (DBS)	Adaptive Pre-Paging technique is used to avoid exchange of identical pages. Free memory pages are removed using DSB technique.	Overall migration time is reduced.	Downtime increases
Liu et al. [28] 2010	Hierarchical copy algorithm	Range of altered pages, peak value and overall write interrupt is monitored If write interrupt < peak value, Altered pages will be exclusively transferred to the target node. Dirty pages are transmitted in final round to reduce overall iterations.	Migration time is reduced as number of iterations are reduced which saves energy.	Extended monitoring of memory image is required.
Ma et al.[29]2010	Improved Pre-Copy strategy	The frequently updated pages are exclusively transmitted in the final step of iteration.	Reduces overall migration time by 32.5% and 34%.	The downtime is more
Zhang et al. [30] 2010	Migration with Data Duplication (MDD)	Similar memory pages are identified by using hash based finger prints. RLE is used to compress data to be transferred.	56.60% decrease in total data transferred.	Hash FingerPrints can result inconsistent output.
Ma et al.[31] 2012	Memory exploration and encoding (ME2)	Valuable pages are recognized and compressed by deploying Run Length Encoding (RLE) method.	Reduces overall migration time as well as downtime.	Higher overhead
Hu et al.[32] 2011, Johnson et al. [18] 2013	Time series prediction technique	Historical statistics are used to identify the frequently modified pages to avoid migration of identical pages.	Energy is saved as less operations are needed for migration.	Only high dirty pages are considered
H. Asai et al.[46] 2013	Hot cloning for migration	Disk images are copied from a host to virtual machine.	Parallel execution reduces energy consumption.	Consistency needs additional overhead.
Jung et al. [33] 2013	VM migration using checkpointing	Virtual machine is migrated for handling out-of-bid failure, starts execution from saved checkpoint.	Rollback time, task waiting time is reduced.	Memory utilization is high, so high cost.
B. Jiang et al. [34] 2013	Priority-Based Live Migration strategy	Dirty pages are generated by high priority virtual machines, are transferred to the destination node after a certain threshold value. Dirty pages of Low priority applications are transferred using stop and copy approach.	Downtime is reduced by 57%.	Total migration time is not considered
K. Su et al. [44] 2015	Remote Page-fault Filtering mechanism	In this approach page faults are reduced as new memory pages are allocated rather than fetching them from host.	Improved performance.	Downtime is not reduced
Kim [35], 2015	Parallel migration approach-breaking chain	Virtual machine list is partitioned and parallel migration is performed. Breakup virtual machines are picked repeatedly of all the separate migration chains	Parallel execution mechanism reduced the amount of energy consumed. Virtual	1.6 – 5.8% additional PMs are needed for parallelizing

	migration	unless the overall migration time is minimized.	machine relocation time is reduced by 21.9 to 62.0%	chained migrations
T. Hea et.al. [88], 2019	Live VM Migration	Performance of Sequential and parallel migration is evaluated in cloud data centers enabled with Software-Defined networking.	Hybrid-post-copy migration performs better than pre-copy migration.	Evaluation of effect of SDN-latency on migration is missing.

Table 3: Energy-aware checkpointing strategies for fault tolerance

Authors name, year	Technique	Description	Benefits	Drawbacks
B. Meroufel et al.[85] 2014	Intra-server checkpointing	All the virtual machines simultaneously created checkpoints and used timer for coordination.	Increased resource utilization and storage access time	Checkpointing rates interval increased cost
R.Rajachandrasekar et al. [49] 2015	Coordinated checkpointing	Additional input output layer is used to reduce energy consumption during I/O checkpoint request phase.	Power consumption is decreased by 48%	Large scale implementationis missing.
P. Chi et al.[50] 2014	Local Checkpointing	MLC STT-RAM is the key component. Soft bit is used to keep record of functioning data and hard-bit is used to save checkpoint.	Recovery time and performance overhead is considerably reduced	Checkpoint overhead is slightly increased.
D. Jung et al. [86] 2011	SLA based checkpointing	Checkpoints are created on the basis of price and time peak values. Possibility of failure time is determined on the basis of price history.	Improves performance and reduces number of checkpoints.	A hybrid approach for spot instances is needed.
D. Ibtesham et al. [63] 2014	Rollback-Recovery Strategy	Segments checkpoint/restart based applications within three stages: Executing the application, Saving the checkpoint and resume execution from the checkpoint.	Energy is saved due to reduction in runtime.	Checkpointing frequency is increased.
S. Di et al.[53] 2013	Checkpoint-Restart Strategy	Checkpoints are created on execution time based on failure and workload prediction.	Speeds up job execution.	Large scale deployment is missing.
M. Salehi et al.[52] 2016	Two-State Checkpointing technique	In absence of faults, irregular checkpoint intervals are opted while regular checkpoint intervals are considered in presence of faults.	Amount of energy consumed is decreased by 26%.	This technique is less reliable.
R. Melhem et al.[51] 2004	Nonuniformcheckpointing	Initially less number of checkpoints are considered, later checkpoint interval is raised as the job deadline is reached.	Energy consumption can be reduced by 68%.	Implementation is missing
B. Nicolae et al.[54] 2011	BlobCR framework	Worked in two phases. Firstly job's state is locally saved using process-level checkpointing. Afterwards the virtual machine is stopped and snapshot of disk is saved into the checkpoint repository.	Saves storage space and also reduces performance overheads	More overheads for long-run applications.
S. Agarwal et al.[48] 2004	Incremental Checkpointing	Altered memory blocks are detected using hash function. Split and merge algorithm is used to automatically divide the job memory into variable sized blocks.	Reduces checkpoint file size by 70%	Experimentation on large scales requited
E. Sayed et. al.	Analysis of	Checkpoints are optimized to save energy.	Slight increase in	Implementation

[64] 2014.	energy-performance trade-off for checkpointing.	Energy aware checkpoint interval is determined for static checkpointing and MTBR is considered for adaptive checkpointing technique.	runtime can reduce energy consumption.	of suggested formulas is required.
J. Ansel et al.[47] 2009	DMTCP transparent checkpointing	Distributed Multithreaded Check-Pointing package used forked checkpointing approach to minimize the time to write checkpoint image on disk.	Minimizes checkpoint time to 0.2 seconds.	Application may slow down.

Table 4: Energy-aware replication strategies for fault tolerance

Authors name, year	Technique	Description	Benefits	Challenges
D Boru et al.[55]2013	Data Replication	Replicas of frequently modified data items are created based on access rate. Limited bandwidth is provided to each replica to ensure its availability.	Performance is enhanced while reducing energy consumption at the same time.	Test bed implementation is required.
X. You et al. [62] 2013	E^2ARS Energy – Efficient Adaptive replication	Data is divided into optimal sub-sections to provide parallelism. Least number of replicas are created ensuring availability. Replicas are stored on the basis of energy aware placement policy.	Adaptive gear-shifting is used to increase energy conservation.	Reduce overhead.
B. Mills et al. [57] 2014	Shadow replication	The replicas or shadows are executed simultaneously along with main process at variable speed. Speed of the primary shadow is increased after failure detection to increase throughput.	Energy consumption is decreased by 40%	Message queue growth rate is increased due to variable speed of shadows.
W. Lang et al.[58] 2009	Mirror replication	Replicas of data is formed and stored at some reliable location. Load balancing is provided using dissolving chain and blinking chain strategies along with replication.	Nodes may be switched off to conserve energy	Predication mechanisms can be opted with this approach.
H. Goudarzi et al.[59] 2012	EVRP algorithm	Dynamic programming is used to generate the number of replicas on the basis of cpu and bandwidth utilization.	Improves resource availability and saves 20% energy.	Cooperation between replicas need to be improved.
X. Cui et al.[60] 2016	Lazy shadowing	Main process is executed at higher speed to obtain maximum throughput and the replicas (shadow) are executed at lower speed to save energy, in case of failure the execution speed of shadow is increased to complete the process successfully within its deadline.	Increases fault tolerance capability	Implement dynamic shadow allocation.
Y. Lin et al. [61] 2017	Energy-Efficient Adaptive File Replication System (EAFR)	Number of replicas varies on the basis of popularity of files. More replicas are created for frequently accessed files. Frequently accessed files are saved in hot servers while others are stored in cold servers to save energy.	Replication time along with energy utilization is decreased.	Prediction of optimal number of cold servers is required.

Shorter checkpoint intervals on other the hand provide speedy recovery, at the expense of increased resource utilization which increases energy consumption. Energy consumption can be considerably reduced by efficiently managing checkpointing during checkpoint operation.

- Number of checkpoints: Ideal amount of checkpoints can be determined to recover from failure while ensuring minimum time and energy is consumed during execution. In case of full checkpointing complete state of system is saved to a secure storage consuming significant amount of storage resources to save the entire system's running states. So to reduce this overhead, incremental checkpointing approach is used which only saves the modified page thus reducing the number of checkpoints stored. Hybrid checkpoint, combine the two approaches, can be used for obtaining effective results.
- Size of checkpoint: Complete state of the system is saved in case of System Level Checkpointing (SLC) where as in Application Level Checkpointing(ALC) the programmer can decide the checkpointing parameters. Application level checkpointing reduces the size of checkpointing file using minimum resources and conserve power simultaneously.

Replication technique is often used to store redundant information (replicas) to enhance the reliability of cloud services as discussed in Section III. The additional resources utilized by the replicas cause escalation in energy consumption. To overcome this issue there is a need of energy efficient replication approach which consumes minimum energy for performing its operation. Table 4 briefly describes the energy aware replication techniques deployed for recovery in case of failure [56]. In table 4 various replication strategies are compared according to the factors such as:

- Number of replicas: Optimal increase in number of replicas results in increased reliability and decreased energy consumption due to parallel execution. If the number of replicas exceed this optimal range it may increase the amount of energy consumed as addition power required for execution of replicas will overcome the amount of energy saved due to parallel execution.
- Communication Delay: Rapid growth in the size of datacentres has increased in the amount of energy consumed as a significant part of energy is lost in the form of communication and transmission overheads. Energy can be saved by decreasing communication delay by making data available and closer to computing resources. This can be achieved through energy efficient data replication.
- Size of replica: Energy can be saved by reducing the size of replica as lesser resources would be required for storage.

Conclusion

Rising computation demands have led to complex infrastructure more prone to faults and increased energy consumption. Thus energy aware fault tolerant solution is the need of the hour. Fault tolerance and energy consumption must be simultaneously considered to achieve overall gain and to ensure reliable, continuous, scalable and flexible availability of cloud services. The fault tolerant strategies must be implemented such that minimum energy is consumed. The main aim of this survey is to provide an unbiased overview of existing fault tolerant energy management strategies. Using optimal features of suggested approaches, hybrid energy efficient approach to tackle faults can be devised in the future.

References

[1] F. Salfner, M. Lenk, M. Malek, "A survey of online failure prediction methods", ACM Computing Surveys (CSUR),vol no 42(3), pp.10,2010

[2] R. Jhawar,V. Piuri, "Fault tolerance management in IaaS clouds" In Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference, pp. 1-6,2012.

[3] A. Avižienis, J.C. Laprie, B. Randell, and C. Landwehr. "Basic concepts and taxonomy of dependable and secure computing," IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11–33, 2004.

[4] M. Grottke, R. Matias, and K. Trivedi, "The fundamentals of software aging," in Proc. IEEE International Conference on Software Reliability Engineering Workshops, Nov. 2008.

[5] S. Marc, Robert W. Wisniewski, J.A. Abraham, S.V.Adve, S. Bagchi, P. Balaji, J. Belak et al. "Addressing failures in exascale computing." International Journal of High Performance Computing Applications, vol. 28, no. 2, pp. 129-173, 2014.

[6] F.Gartner, "Fundamentals of fault-tolerant distributed computing in asynchronous environments," ACM Computing Surveys, vol. 31, no. 1, pp. 1–26, Mar. 1999.

[7] B. Javadi, J. Abawajy, and R. Buyya, "Failure-aware resource provisioning for hybrid cloud infrastructure," Journal of Parallel and Distributed Computing, vol. 72, no. 10, pp. 1318 – 1331, 2012.

[8] M.Lackovic, D.Talia, R.T. Calasanz, J.Banares, and O.Rana, "A taxonomy for the analysis of scientific workflow faults," Proceedings of the 13th IEEE International Conference on Computational Scienceand Engineering,pp.398–403, 2010

[9] J. Wei, L. Rashid, K. Pattabiraman and S. Gopalakrishnan, "Comparing the effects of intermittent and transient hardware faults on programs", Dependable Systems and Networks Workshops, IEEE, pp. 53-58, 2011.

[10] M. Castro , B. Liskov, "Practical byzantine fault tolerance and proactive recovery", ACM Transactions on Computer Systems (TOCS), v.20 n.4, p.398-461, 2002.

[11] R. Jhawar, V. Piuri, "Fault Tolerance and Resilience in Cloud Computing Environments" in Computer and Information Security Handbook, Morgan Kaufmann, 2013.

[12] Q. Zhang, L. Cheng, R. Boutaba, "Cloud computing: state-of-the-art and research challenges", J Internet Serv Appl , springer, 2010, pp. 7–18, 2010.

[13] B.P. Rimal , E. ChoI "A taxonomy and survey of Cloud Computing Systems", IEEE Fifth International Joint Conference on INC, IMS and IDC, pp. 44-51, 2009

[14] C. N. Höfer, G. Karagiannis, "Cloud computing services: taxonomy and comparison", Journal of Internet Services and Applications, Springer, pp.81-94, 2011.

[15] R. Buyya and Chee Shin Yeo, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the

5th Utility", Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, 2009

[16] C.C. Meixner, C.Develder, M.Tornatore, B. Mukherjee, "A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications", IEEE Communiation Sureys, pp. 2244-2281 2016

[17] P.D. Kaur, Kanupriya, "Fault Tolerance Techniques and Architectures in Cloud Computing-A Comparative Analysis", International Conference on Green Computing and Internet of Things, IEEE, pp. 1090-1095, 2015

[18] T. Mastelic , A. Oleksiak , H. Claussen , I. Brandic , J.M. Pierson , A. V. Vasilakos, "Cloud Computing: Survey on Energy Efficiency", ACM Computing Surveys (CSUR), v.47 n.2, p.1-36, 2015

[19] E. Feller, L. Rilling, C. Morin, R. Lottiaux, and D. Leprince "Snooze: A Scalable, Fault-Tolerant and Distributed Consolidation Manager for Large-Scale Clusters," Proc. 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, pp. 125-132, 2010

[20] A. Beloglazov, R. Buyya, "Energy Efficient Allocation of Virtual Machines in Cloud Data Centers", 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010

[21] D. Sun,Guiran, Chang, C. Miao, X. Wang, "Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments", The Journal of Supercomputing 66, Vol no. 1, pp. 193-228, 2013

[22] C. Engelmann, G. R. Vallee, T. Naughton, and S. L. Scott "Proactive Fault Tolerance Using Preemptive Migration", 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing, IEEE, pp 252-257, 2009

[23] Liu, Jialei, S. Wang, Ao Zhou, S.A P Kumar, and R. Buyya "Using Proactive Fault - Tolerance Approach to Enhance Cloud Service Reliability", IEEE Transactions on Cloud Computing, 2017

[24] Goiri, F. Julia, J. Guitart, and J. Torres, "Checkpoint-based fault tolerant infrastructure for virtualized service providers," Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS'10), pp. 455-462, 2010

[25] P. Das, P. M. Khilar, "VFT: A Virtualization and Fault Tolerance Approach for Cloud Computing", Proceedings of 2013 IEEE Conference on Information and Communication Technologies(ICT 2013) , (pp. 473-478) , 2013

[26] A. Zhou, S. Wang, Z. Zheng, C. Hsu, M. Lyu, and F. Yang, "On cloud service reliability enhancement with optimal resource usage," IEEE Transactions on Cloud Computing, pp. 452-466, 2016

[27] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan. "Live virtual machine migration with adaptive, memory compression", In Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE Inter-national Conference on, pages 1 –10, 2009

[28] F. Ma, F. Liu, and Z. Liu, "Virtual machine migration based on improved pre-copy approach," In Proc. IEEE Int'l Conf. Software Engineering and Service Sciences, pp.230-233, 2010.

[29] X. Zhang, Z. Huo, Jie Ma, Dan Meng, "Exploiting Data Deduplication to Accelerate Live Virtual Machine Migration", 2010 IEEE International Conference on Cluster Computing, pp 88-97, 2010

[30] B. Hu, Z. Lei, Y. Lei, D. Xu, and J. Li, "A time-series based precopy approach for live migration of virtual machines", IEEE 17th International Conference on Parallel and Distributed Systems, pp. 947-952, 2011

[31] J. Arputharaj Johnson "Optimization of migration downtime of virtual machine in Cloud"IEEE, pp. 1-5, 2013

[32] Y. Ma, H. Wang, J. Dong, Y. Li,and S. Cheng, "ME2: efficient live migration of virtual machine with memory exploration and encoding", IEEE International Conference on Cluster Computing, pp 610-613, 2012.

[33] D. Jung, S. Chin, K. Chung, H. Yu, "VM migration for fault tolerance in spot instance based cloud computing," in Grid and Pervasive Computing, Springer, vol. 7861, pp. 142-151, 2013.

[34] B. Jiang, J. Wu, X. Zhu, D. Hu, "Priority-Based Live Migration of Virtual Machine", Springer, pp. 376–385, 2013.

[35] C. Kim, C. Jeon, W. Lee, S. Yang, "A Parallel Migration Scheme for Fast Virtual Machine Relocation on a Cloud Cluster." The Journal of Supercomputing, Springer, pp: 4623–4645, 2015.

[36] Y. Zhang, Z. Zheng, and M. R. Lyu. "BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing." In Cloud Computing (CLOUD), 2011 IEEE International Conference, pp. 444-451, 2011.

[37] B. Egger, Y. Cho, C. Joe, E.Park, J. Lee "Effcient Checkpointing of Live Virtual Machine Migration", IEEE Transactions on Computers, pp. 3041 – 3054, 2016.

[38] M. Zhao, F. DUgard, K.A. Kwait, C.A. Kamhoua "Multi-level VM replication based survivability for mission-critical cloud computing", IEEE International Symposium on Inegrated Network Management, 2015.

[39] D. Bruneo, S. Distefano, F. Longo, A. Puliafito, M. Scarpa. "Workload-based software rejuvenation in cloud systems."IEEE Transactions on Computers, vol. 62, no. 6, pp.1072-1085, 2013.

[40] L. Silva, J. Alonso, and J. Torres, "Using Virtualization to Improve Software Rejuvenation," IEEE Trans. Computers, vol. 58, no. 11, pp. 1525-1538, 2009.

[41] K. Kourai, S.Chiba "A Fast Rejuvenation Technique for Server Consolidation with Virtual Machines", IEEE International Conference on Dependable Systems and Networks, pp. 245-255, 2007.

[42] Y. Abe, R. Geambasu, K. Joshi, M. Satyanarayanan "Urgent Virtual Machine Eviction with Enlightened Post-Copy", 12th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp. 51-64, 2016.

[43] M.R. Hines, K. Goplanan "Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning", Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pp.51-60, 2009.

[44] K. Su, W. Chen, G. Li, Z. Wang "RPFF: A Remote Page-fault Filter for Post-copy Live Migration", IEEE International Conference on Smart City/SocialCom/SustainCom together with DataCom 2015 and SC2 2015, pp.936-943, 2015.

[45] I. Egwutuoha, S. Chen, D. Levy, B. Selic, R. Calvo, "Energy efficient fault tolerance for high performance computing (hpc) in the cloud", Sixth International Conference on Cloud Computing (CLOUD), IEEE, , pp. 762–769, 2013.

[46] H. Asai "P2V Migration with Post-copy Hot Cloning for Service Downtime Reduction", IEEE Third International Conference on Cloud and Green Computing, pp. 1-8, 2013.

[47] J. Ansel, K. Arya, and G. Cooperman, "DMTCP: Transparent checkpointing for cluster computations and the desktop," in Proc. of the Int'l Parallel and Distributed Processing Symp. (IPDPS). Rome, Italy: IEEE, 2009, pp. 1–12.

[48] S. Agarwal, R. Garg, M. S. Gupta, and J. E. Moreira. Adaptive incremental checkpointing for massively parallel systems. ICS Proceedings of the 18th Annual International Conference on Supercomputing, pp. 277–286, 2004.

[49] R. Rajachandrasekar, A. Venkatesh, K. Hamidouche, D. K.Panda "Power-Check: An Energy-Efficient Checkpointing Framework for HPC Clusters", 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 261-271, 2015

[50] P. Chi, Cong Xu, T. Zhang, X. Dong, Y. Xie, "Using Multi-Level Cell STT-RAM for Fast and Energy-Efficient Local Checkpointing", IEEE/ACM International Conference, pp. 301-308, 2014

[51] R. Melhem, D. Mosse, E. Elnozahy, "The Interplay of Power Management and Fault Recovery in Real-Time Systems" IEEE Transactions on Computers,, VOL. 53, NO. 2, pp.217-231, 2004

[52] M. Salehi, M. K. Tavana, S. Rehman, M. Shafique, A. Ejlali, and J. Henkel, "Two-State Checkpointing for Energy-Efficient Fault Tolerance in Hard Real-Time Systems", IEEE Transactions on Very

Large Scale Integration (VLSI) Systems, Vol 24, No 7, pp.2426-243, 2016

[53] S. Di, Y. Robert, F. Vivien, D. Kondo, C. Wang, F. Cappello1, “Optimization of Cloud Task Processing with Checkpoint-Restart Mechanism”, High Performance Computing, Networking, Storage and Analysis (SC), 2013 International Conference, pp. 1-12, 2013

[54] B. Nicolae, F. Cappello “BlobCR: Efficient Checkpoint-Restart for HPC Applications on IaaS Cloudsusing Virtual Disk Image Snapshots”, Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, pp. 1-36,2011

[55] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, A. Zomaya, "Energy-Efficient Data Replication in Cloud Computing Datacenters", *CCSNA* Cluster Computing, Vol no. 18(1), pp.385-402, 2015

[56] M. A. Haque, H. Aydin, D. Zhu, "Energy-Aware Standby-Sparing "Technique for Periodic Real-Time Applications", Sparing "Technique for Periodic Real-Time Applications", Computer Design (ICCD), 2011 IEEE 29th International Conference, pp. 190-197, 2011.

[57] B. Mills, T. Znati, R. Melhem, K. B. Ferreira, R. E. Grant, “Energy Consumption of Resilience Mechanisms in Large Scale Systems” 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, IEEE, pp. 528-535, 2014

[58] W. Lang, J.M. Patel, J.F.Naughton “On Energy Management, Load Balancing and Replication”, ACM SIGMOD Record, pp. 35-42, 2009

[59] H. Goudarzi, M. Pedram “Energy-Efficient Virtual Machine Replication and Placement in a Cloud Computing System” 2012 IEEE Fifth International Conference on Cloud Computing, pp. 750-757, 2012

[60] X. Cui, T. Znati, R. Melhem, ”Adaptive and Power-Aware Resilience for Extreme-Scale Computing” 2016 Intl IEEE Conferences, pp 1-9, 2016

[61] Y. Lin, H. Shen, “EAFR: An Energy-Efficient Adaptive File Replication System in Data-Intensive Clusters”, IEEE Transactions on Parallel and Distributed Systems, pp. 1017-1030, 2017

[62] X. You, L. Zhou, J. Huang, J. Zhang, C. Jiang and J. Wan, “E2ARS : An Energy-Effective Adaptive Replication Strategy in Cloud Storage System, Applied Mathematics & Information Sciences An International Journal, pp. 2409-24019, 2013

[63] D. Ibtesham, D. DeBonis, D. Arnold, K.B. Ferreira. "Coarse-Grained Energy Modeling of Rollback/Recovery Mechanisms." InDependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on, pp. 708-713. IEEE, 2014.

[64] N. El-Sayed, B. Schroeder. "To checkpoint or not to checkpoint: Understanding energy-performance-I/O tradeoffs in HPC checkpointing." Cluster Computing (CLUSTER), 2014 IEEE International Conference, pp. 93-102, 2014.

[65] E.A.S Sarma, J.N. Maggo, A.S. Sachdeva, “India’s energy scenario in 2020” 17th World Energy Congress, Houston, TX, pp. 13-18, 1998

[66] F. Ahmad, T. N. Vijaykumar “Joint optimization of idle and cooling power in data centers while maintaining response time”, ACM SIGPLAN, pp. 243–256, 2010

[67] Indian Energy Outlook 2015 , http://www.worldenergyoutlook.org/india

[68] W. Deng, F.Liu, H. Jin, B. Li, D. Li, “Harnessing renewable energy in cloud datacenters: opportunities and challenges” IEEE Network, pp. 48-55, 2014.

[69] M. Ghamkhari, H. Mohsenian-Rad, “Energy and Performance Management of Green Data Centers: A Profit Maximization Approach”, IEEE Transactions on Smart Grid, pp. 1017-1025, 2013

[70] E.K. Lee, H. Viswanathan, D. Pompili, “Proactive Thermal-aware Resource Management in Virtualized HPC Cloud Datacenters”, IEEE Transactions on Cloud Computing, pp. 1-14, 2015

[71] A. Beloglazov, J. Abawajy, R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud Computing”, Future generation Computer System, pp.755-768, 2012

[72] A. Dalvandi, M. Gurusamy, K.C. Chua “Time-aware VMFlow Placement, Routing and Migration for Power Efficiency in Data Centers” IEEE Transactions on Network and Service Management, pp. 349-362, 2015

[73] A. Beloglazov, R. Buyya, “Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers”, Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, pp. 755-768, 2012.

[74] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, H. Tenhunen, “Using Ant Colony System to Consolidate VMs for Green Cloud Computing”, IEEE Transactions on Services Computing, pp. 184-198, 2015

[75] K.K. Nguyen, M. Cheriet “Environment-aware Virtual Slice Provisioning in Green Cloud Environment”, ” IEEE Transactions on Service Computing, pp. 507-519, 2015

[76] L. Wang, F. Zhang, J. A. Aroca, A.V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, “GreenDCN: A General Framework for Achieving Energy Efficiency in Data Center Networks”, IEEE Journal on Selected Areas in Communications, pp. 4-15, 2014

[77] S. Wang, A. Zhou, C.H Hsu, X. Xiao, F. Yang, “Provision of Data-intensive Services through Energy- and QoS-aware Virtual Machine Placement in National Cloud Data Centers”, IEEE Transactions on Emerging Topics in Computing, pp. 290-300, 2016

[78] J. Moore , J.S. Chase, P. Ranganathan, “Weatherman: Automated, Online, and Predictive Thermal Mapping and Management for Data Centers”, IEEE International Conference on Autonomic Computing, pp. 155-164, 2006

[79] K. Singh, S. Kaushal, “Energy Efficient Resource Provisioning Through Power Stability Algorithm in Cloud Computing”, Proceedings of the International Congress on Information and Communication Technology, Springer, pp 255-263, 2016

[80] E. Sheme , N Frashëri , S. Holmbacka, S. Lafond, D. Lucanin , “Datacenters powered by renewable energy: A case study for 60 degrees latitude north”, Software, Telecommunications and Computer Networks (SoftCOM), 2016.

[81] D.Zhu, R. Melhem, D. Mossé. "The effects of energy management on reliability in real-time embedded systems." IEEE/ACM International Conference on Computer Aided Design, ICCAD, pp. 35-40, 2004.

[82] V. Sundriyal, M. Sosonkina, “Initial Investigation of a Scheme to Use Instantaneous CPU Power Consumption for Energy Savings Format”, E2SC '13 Proceedings of the 1st International Workshop on Energy Efficient Supercomputing, 2013

[83] Y. Ren, J. Suzuki, C. Lee, A.V. Vasilakos, S. Omura, “Balancing Performance, Resource Efficiency and Energy Efficiency for Virtual Machine Deployment in DVFS-enabled Clouds: An Evolutionary Game Theoretic Approach”, GECCO Comp '14: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation,pp. 1205-1212, 2014

[84] M. Basoglu, M. Orshansky, M. Erez, “NBTI-Aware DVFS: A New Approach to Saving Energy and Increasing Processor Lifetime”, IEEE International Symposium on Low-Power Electronics and Design, pp. 253-258, 2010

[85] B. Meroufeland, G. Belalem “Adaptive time-based coordinated checkpointing for cloud computing workflows”, Scalable Computing: Practice and Experience, pp. 153–168, 2014

[86] D. Jung, S.H. Chin, K.S. Chung, H. Yu, J. Gil “An Efficient Checkpointing Scheme Using Price History of Spot Instances in Cloud Computing Environment”, IFIP International Federation for Information Processing, pp. 185-200, 2011

[87] A. Kumar, A. Bashir, "Improved EDF Algorithm for Fault Tolerance with Energy Minimization." In Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on, pp. 370-374, 2015.

[88] T.Hea, A.Toosib, R. Buyyaa, "Performance Evaluation of Live Virtual Machine Migration in SDN-enabled Cloud Data Centers", Journal of Parallel and Distributed Computing, Elsevier, pp. 1-40, 2019.

Authors Profile

Kamaljit Kaur was born in Kapurthala, Punjab, India.She completed B.Tech. with Distinction from Punjab Technical University, and M.Tech. with Gold Medal from Guru Nanak Dev University, Amritsar. She is pursuing PhD in the field of Cloud Computing from Guru Nanak Dev University, Amritsar. From 2007 to 2008 she was employed with Dr. B.R.Ambedkar NIT, Jalandhar as Lecturer and from 2010 to 2012, she worked as an Assistant Professor at Lovely Professional University, Jalandhar. She joined Guru Nanak Dev University, Amritsar in July 2012 where she is currently working as an Assistant Professor.Her research interests are in Resource Provisioning in Cloud Computing, Resiliency in Cloud Computing and Distributed Systems, Big Data, IoT. Kamaljit Kaur has published and presented more than 45 papers in scientific journals and international conferences.

Kuljit Kaur Chahal is an assistant professor in the Department of Computer Science, Guru Nanak Dev University, India. She received her PhD degree in computer science from Guru Nanak Dev University, India. Her research interests are in distributed computing, Web services security, and open-source software

