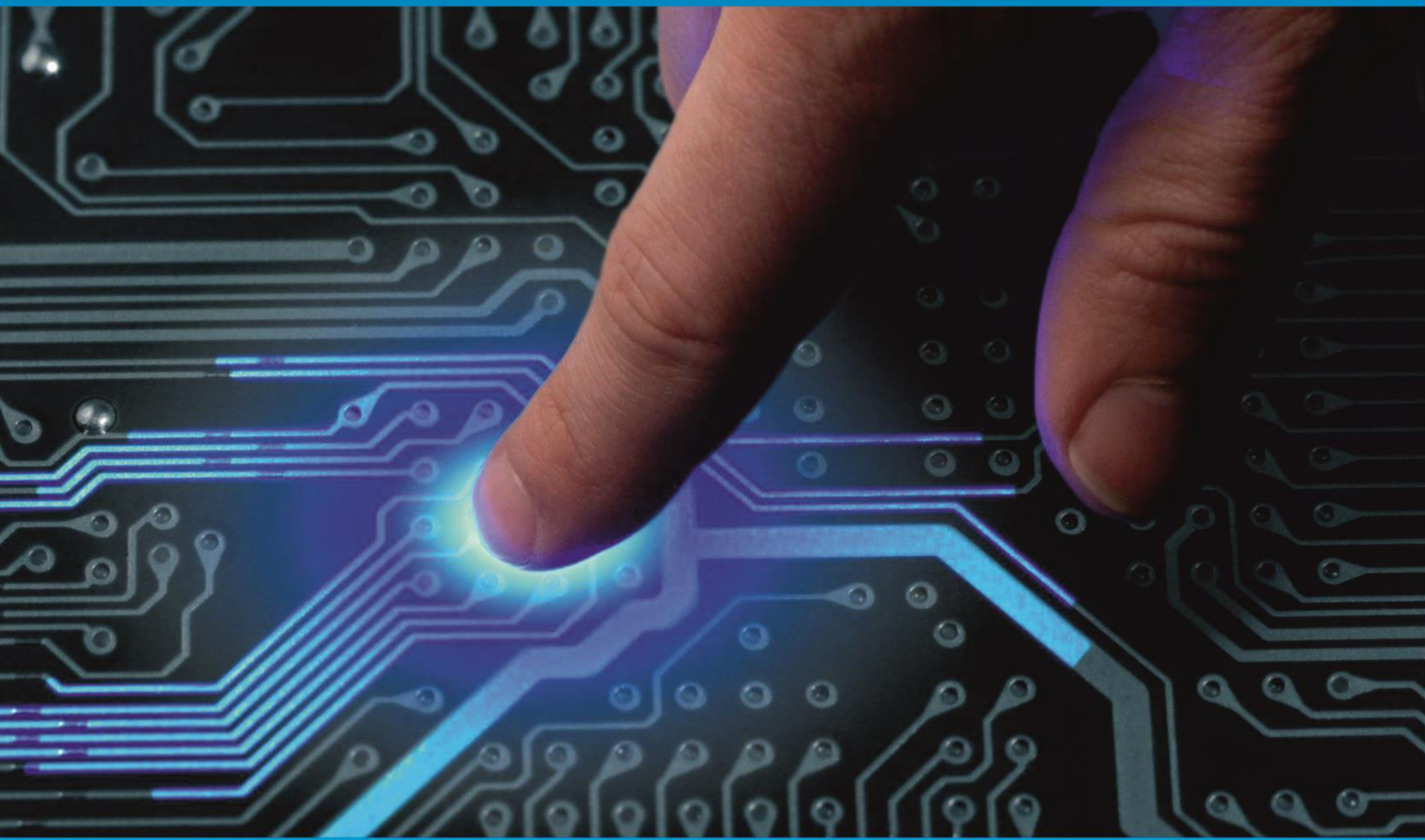
e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 3, March 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE2021. 0903108|

Fault-Tolerant Dynamic Task Scheduling Scheme for Virtualized Cloud Data Centers

T.Thilagavathi, Mr. M. Balamurugan, M.E.,(Ph.D).,

ME Final Year, Department of Computer Science, The Kavery Engineering College, Tamil Nadu, India

Assistant Professor & HOD, Department of Computer Science, The Kavery Engineering College, Tamil Nadu, India

ABSTRACT: Reminder and replication are two basic and widely accepted techniques in distributed fault tolerance computer systems. Traditional System Concept A cloud-based security administrator that makes granular decisions to choose mitigation strategies to protect each customer's outsourced computer assets based on their specific security needs against specific threats. Prioritize the existing system by introducing and deploying ineffective security solutions based on the client's security requirements. Cloud Computing aims to provide cloud consumers with on-demand flexible online resource allocation that pays off. Achieving this goal automatically requires a resource scaling mechanism that maintains application performance in accordance with Service Level Agreements (SLA) while reducing resource costs. In this article, we introduce a cross-correlation prediction approach based on distributed machine learning and formulated the problem of resource planning as a stochastic problem. Then, the Resource Online Scheduling Algorithm (RICH) is used using an optimization technique to achieve a temporal average gain that is close to the value Optimal with a reduction in the gap (1 / V) for CSP while maintaining high system stability and lower Overload to ensure QoS (Quality of Services) for cloud users. Machine learning for the Naïve Bayes classifier is based on the Bayes theorem, which requires strong independence (naive) between the attributes or features (predictors). This predicts the resource requirements of multiple virtual machines running in a cloud infrastructure. Also map network resources dynamically to virtual nodes and links. The agents make sure that the virtual networks have the required resources at all times, but only the necessary resources are reserved for this purpose.

KEYWORDS: Distributed Computing systems, Fault Tolerant workflow scheduling algorithm, Service Level Agreements (SLA), Distributed Machine Learning, Resource online scheduling (RCH) algorithm, Naïve Bayes classifier, Virtual Networks

I. INTRODUCTION

In cloud environment, the conventional roles of providers is classified into: infrastructure providers who maintain the platform and rent resources in accordance with usage-based pricing model, and service providers who host one or many infrastructure providers to rent resources to serve the end users. In cloud, the user pumps-in the data, applications, or any other services over the internet through browser irrespective of the device or location of the user.This is possible because of the infrastructure that is usually provided by the third-party organizations. These vendor parties give access and lease their services following the usual delivery models of cloud-private, public, hybrid or community. The Cloud providers deploy their services in accordance with the need of the end user. The vendor provides the computing services in three ways, namely, Infrastructure-as-a-Service (IaaS), Platform-as-a Service (PaaS) and Software-as-a-Service (SaaS).

Faults do occur within the clouds and when they happen will hamper adversely the services to be provided to the end user in reasonable time. A mistake within the software package or defect within the hardware could lead to disturbance in the way the cloud is expected to operate. It is necessary to recognize different faults occurring on the infrastructure and find ways and means of countering the same with the main aim of providing continued, uninterrupted service to the end customer.

It is necessary to analyses cause and effects of infrastructure faults and finds the ways to mitigate them. The programs which are part of the operating system are generally designed for recovering back to the normal operation in the case of occurrence of the faults. In no recovery is possible, the program dies and nothing of the ongoing session is stored leading to heavy disruption in providing the services to the end-user. An infrastructure fault may result into the termination or failure of the servers releasing all the current work status, session details and uncommitted work done, drawing the client to a great loss. Whenever a server fails, the middleware diverts the requests to the remaining servers.

At the point where the demand reaches to a critical stage, the alternate servers may also breakdown un able to handle the requests. To avoid a situation where in the client suffers from the infrastructure faults, a fault tolerant system should

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |
|| Volume 9, Issue 3, March 2021 ||
| DOI: 10.15680/IJIRCCE2021. 0903108|

be designed to keep the infrastructure function normally even if one of its servers fails. In this paper several architectures have been presented that cater for fault tolerance at infrastructure Level.
In an integrated task scheduling algorithm that takes into account the issues such as VM management and Datacenter management. Our framework relies on predictions made by machine learning algorithms and scheduling and classification policies generated by a Markovian Decision Process (MDP), to adjust its scheduling decisions on the fly. We propose a cross-correlation prediction approach based on The Naïve Bayes classifier is one of the handiest machine learning algorithms. A Naïve Bayesian classification model is specifically handy for very large datasets as it comparatively requires little effort for its build and it also has no complicative repetitive parameter figuring or calculation.

II. RELATED WORKS

1. Cloud Computing – Issues, Research and Implementations

Mladen A. Vouk was told that Cloud computing – a relatively recent term, builds on decades of research in virtualization, distributed computing, utility computing, and more recently networking, web and software services. It implies a service oriented architecture, reduced information technology over head for the end-user, great flexibility, reduced total cost of ownership, on-demand services and many other things. This paper discusses the concept of "cloud" computing, some of the issues it tries to address, related research topics, and a "cloud" implementation available today.

A powerful underlying and enabling concept is computing through service-oriented architectures (SOA) delivery of an integrated and orchestrated suite of functions to an end-user through composition of both loosely and tightly coupled functions, or services – often network-based. Related concepts are component-based system engineering, orchestration of different services through workflows, and virtualization.

2. Adaptive Fault Tolerance in Real Time Cloud Computing

Sheheryar MalikandFabriceHuet was said Cloud support for real time system is really important. Because, today we found a lot of real time systems around us. Their applications range from small mobile phones to larger industrial controls and from mini pacemaker to larger nuclear plants. Most of them are also safety critical systems, which should be reliable. In general, real-time system is any information processing system which has to respond to externally generated input stimuli, within a finite and specified period of time. So the correctness depends not only on the logical result, but also the time it was delivered. Failure to respond is as bad as the wrong response. These systems have two main characteristics by which they are separated by other general-purpose systems. These characteristics are timeliness and fault tolerance. By timeliness, we mean that each task in real time system has a time limit in which it has to finish its execution.

And by fault tolerance means that it should continue to operate under fault presence. Use of cloud infrastructure for real time applications increases the chances of errors. As the cloud nodes (virtual machines) are far from the transceiver (job submitting node, actuator or sensor). Many real time systems are also safety critical systems, so they require a higher level of fault tolerance. Safety critical real time systems require working properly to avoid failure, which can cause financial loss as well as casualties. So there is an increased need to tolerate the fault for such type of systems to be used with cloud infrastructure. For this purpose we had presented a model for the fault tolerance of real time applications running at cloud infrastructure. The basic mechanism to achieve the fault tolerance is replication or redundancy. We had performed this replication in form of software variants running on multiple virtual machines. Due to the replication, cost for renting the cloud resources will increase. But it is really required to avoid the catastrophic loss.

3. FaultTolerance Middleware for Cloud Computing

Wenbing Zhao, P.M. Melliar and L.E. was said the LLFT middleware maintains strong replica consistency of the states of the replicas, both in fault-free operation and in the event of a fault. If a fault occurs, the LLFT reconfiguration/recovery mechanisms ensure that a backup has, or can obtain, the messages and the ordering information it needs to reproduce the actions of the primary. They transfer the state from an existing replica to a new replica and synchronize the operation of the new replica with the existing replicas, to maintain virtual synchrony. The novel contributions of the LLFT middleware include the Low Latency Messaging Protocol, the Leader Determined Membership Protocol, and the Virtual Determinizer Framework. The Low Latency Messaging Protocol provides a reliable, totally ordered multicast service by communicating message ordering information from the primary in a group to the backups in the group. It achieves low latency (minimal overhead in the response time seen at a client), by having the primary make the decisions on the order in which operations are performed and reflecting the ordering information to the backups. It involves fewer messages than prior fault tolerant systems by multicasting messages, piggybacking information, suppressing messages at the backups, etc.

The Leader-Determined Membership Protocol ensures that the members of a group have a consistent view of the membership set and of the primary member of the group.

It effects a membership change and a consistent view more quickly than other membership protocols, by selecting a new primary deterministically, based on the ranks and precedence's of the backups and by avoiding the need for a multi-round consensus algorithm.

The Virtual Determinate Framework renders the applications virtually deterministic by recording the order and results of each non-deterministic operation carried out by the primary, and by guaranteeing that the backups obtain the same results in the same order as the primary.

4. Fault Tolerance Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing:

Jayadivya S K, JayaNirmala S, Mary SairaBhanus said that Workflow is a set of tasks processed in a predefined order based on its data and control dependency. Scheduling these workflows in a computing environment, like cloud environment, is an NP-Complete problem and it becomes more challenging when failures of tasks are considered. To overcome these failures, the workflow scheduling system should be fault tolerant.

In this paper, the proposed Fault Tolerant Workflow Scheduling algorithm (FTWS) provides fau lt tolerance by using replication and resubmission of tasks based on priority of the tasks.

The replication of tasks depends on a heuristic metric which is calculated by finding the tradeoff between the replication factor and resubmission factor. The heuristic metric is considered because replication alone may lead to resource wastage and resubmission alone may increase makespan. Tasks are prioritized based on the criticality of the task which is calculated by using parameters like out degree, earliest deadline and high resubmission impact.

Priority helps in meeting the deadline of a task and thereby reducing wastage of resources. FTWS schedules workflows within a deadline even in the presence of failures without using any history of information. The experiments were conducted in a simulated cloud environment by scheduling workflows in the presence of failures which are generated randomly. The experimental results of the proposed work demonstrate the effective success rate in-spite of various failures. In this paper, fault tolerance is achieved by compromising task replication and task resubmission methods. Tasks are replicated based on heuristic metric and priority of the task.

Heuristic metric is calculated by finding the tradeoff between replication and resubmission factor which gives the replication number based on impact of the resubmission. The heuristic metric is considered because replication alone may lead to resource wastage and resubmission alone may increase make span. Tasks are prioritized based on out degree, earliest deadline and high resubmission impact. Tasks are scheduled to meet deadline by considering priority and thereby to reduce wastage of resources by restricting unnecessary replication of tasks.

III. PROPOSED SYSTEM

In the proposed system, a dynamic and failure-aware framework that can be integrated within Cloud online scheduler and adjust the scheduling decisions based on collected information about the cloud environment. Virtual machine is playing a significant role for improving resources utilization; processing nodes load balancing, application isolation, fault tolerance in virtual machines, to enhance the nodes portability and to maximize the physical server efficiency. To balance the cloud with its resources for better performance with the services to the end users of the cloud and at the identical time, numbers of users are served by application deployments in the environment of cloud is the main task. In an integrated task scheduling algorithm that takes into account the issues such as VM management and Datacenter management. Our framework relies on predictions made by machine learning algorithms and scheduling and classification policies generated by a Markovian Decision Process (MDP), to adjust its scheduling decisions on the fly. We propose a cross-correlation prediction approach based on The Naïve Bayes classifier is one of the handiest machine learning algorithms. A Naïve Bayesian classification model is specifically handy for very large datasets as it comparatively requires little effort for its build and it also has no complicative repetitive parameter figuring or calculation. The proposed algorithm can be easily integrated with Virtual Machine strategies and is fault tolerant and can efficiently deal with energy consumption problem.

IV. SYSTEM DESIGN

The system design shows we dismantle the user's willingness to pay just as a function of bandwidth; we model it as a common function of bandwidth, power and time dedicated to each application. Instead of assuming a lower cost or a fixed cost per user, we consider a general cost as a function of network capacity. A natural goal for machine learning is to maximize aggregated performance goals across multiple traffic classes. Decomposing the fault tolerant problem

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE2021. 0903108|

results in a stable and an optimal solution, where each traffic class is optimized according to its own performance target, using a dynamic scheduling allocation algorithm of bandwidth.

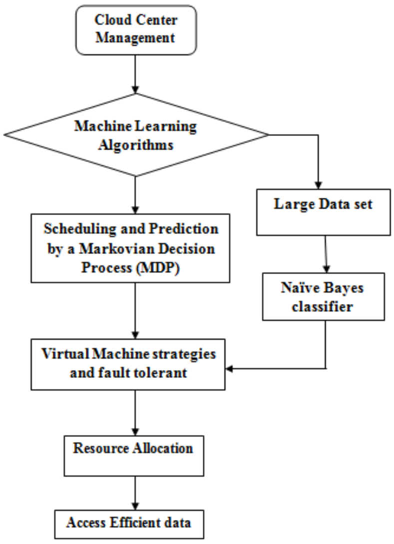


Figure 4.1: System Architecture

V. METHODOLOGY

5.1 Initial Workload Placement

When you power on a virtual machine in a cluster, DRS places it on a suitable host or generates a recommendation, depending on the automation level you choose. Automation levels, also known as migration thresholds, range from conservative to aggressive. VMware vCenter only applies recommendations that match cluster restrictions such as host affinity rules or maintenance. It applies DRS recommendations that can even slightly improve the overall load balancing of the cluster. DRS offers five levels of automation for each cluster that meet your needs.

5.2 Automated Load Balancing

DRS spread the virtual machine workloads across vSphere hosts inside a cluster and monitors available resources for you. Based on your automation level, DRS will migrate (VMware vSphere vMotion) virtual machines to other hosts within the cluster to maximize performance.

5.3 Optimized Power Consumption

Like DRS, vSphere's Distributed Power Management feature optimizes power consumption at the cluster and host levels. When you enable DPM, it compares cluster and host-level capacity with the demand for virtual machines, including the most recent historical demand, and puts hosts into standby mode. As capacity requirements increase, DPM turns on the hosts in standby mode to absorb the extra workload. You can also set up DPM to make recommendations but not take action.

5.4 Traffic Management Optimizing

Traffic Management Optimizing for performance is not the sole design goal. For architecture or a protocol to succeed, an equally important, though sometimes opposing goal is simplicity. Simplicity can be defined as the ease of implementing a protocol or architecture with existing technology, while keeping the overhead low. In this dissertation, current technology trends are used to guide modeling assumptions, as well as understand the implementation possibilities for an algorithm derived from optimization. To support multiple traffic classes in parallel, optimization theory indicates the need for separate resources for each traffic class. One potential implementation is to run each traffic class on a virtual network.

VI. EXPERIMENTAL RESULTS

This section describes the experimental results for the given input data. Here I have implemented my project with frontend as Java. And here I discuss how my project output was shown by screenshots below.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE2021. 0903108|

The VMs Reserve Time Rate (VRTR) is defined as the ratio of the total reserved provisioning time of all VMs in the event of failure over the total reserved provisioning time of all ideal-condition VMs sourced from HEFT and reflects the system resource consumption of the algorithms being compared are faulty. Both metrics above show the cost of scheduling algorithms. The TCR reflects the time cost, while the VRTR displays the resource cost associated with a failure. First, although the resubmission strategy is also used in the proposed system, the resubmission for each task is only passively accepted after all backup copies have failed. Regarding the allocation of resources, the resubmission is actively accepted and limited by the subordinate deadlines of these tasks. In addition, the online reservation customization also changes the fault tolerant type of some tasks from the resubmission to the proposed replication. At the same time, only the resubmission is transferred to IRW. The slightly different time required for a replication and resubmission error for a task can result in an apparent delay in the entire workflow due to other non-running tasks that depend on this task. The delay caused by an error between replication and resubmission becomes more apparent as the task counter increases. In addition to the disruption, the performance variation of VMs also negatively impacts the execution of the workflow. Therefore, we plan to develop a robust scheduling algorithm with a strategy to provide elastic resources for the workflow in the cloud systems

Table 1: Parameters for Generated Workflows

Parameter	Value (min)-(max)
task count	(1100)-(1500)
Width	[0.2, 0.4, 0.8]
Regularity	[0.2, 0.4, 0.8]
Density	[0.2, 0.4, 0.8]
jumps	[1, 2, 3]

VII. CONCLUSION

The cloud computing market is only going to grow as per current trends and in such a case when these dynamic cloud computing environments are more or less prone to failure, effective fault tolerance techniques are a must to ensure reliability over these cloud systems as their failure could be catastrophic. With this proposed model we worked in Proactive fault management domain whereby we first predict the nodes which are most prone to experiencing fault with the Naïve Bayes classifier and then applying fault tolerance techniques to ensure enhanced reliability of the system. This review has studied the concept of Virtual machine migration in cloud computing with the resource utilization and its need for achieving user satisfaction and for enhancing the profit for cloud service providers. With our proposed model, we can enhance the reliability of the system, reliability values are used to estimate the failure probability of any component such as node in our case. Reliability of the system is measured using MTBF factor. With the application of Naïve Bayes, we can reduce the numbers of node failures with the accuracy of nearly 87%.

REFERENCES

[1] P. Mell and T. Grance, "SP 800-145. The NIST Definition of Cloud Computing," NIST, Gaithersburg, MD, United States, Tech. Rep., 2011.
[2] K. Dahbur, B. Mohammad, and A. B. Tarakji, "A Survey of Risks, Threats and Vulnerabilities in Cloud Computing," in Proc. of the International Conference on Intelligent Semantic Web-Services and Applications (ISWSA 2011), 2011, pp. 1–6.
[3] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, "A Survey on Security Issues and Solutions at Different Layers of Cloud Computing," The Journal of Supercomputing, vol. 63, no. 2, pp. 561– 592, 2013.
[4] A. Andrieux and al. Web services agreement specification (wsagreement) OGF, 2007.
[5] D.D. Lamanna, J. Skene, and W. Emmerich. SLAng A language for defining service level agreements. The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, pages 100–106, 20O3.
[6] TAPAS. http ://uclslang.sourceforge.net/. 2013.
[7] mOSic. http ://www.mosaic-cloud.eu/. 2013.
[8] WS-Policy. http ://www.w3.org/TR/ws-policy/. 2013.
[9] Wolfgang Ziegler and Ming Jiang. OPTIMIS SLA Framework and Term Languages for SLAs in Cloud Environment. 2011.
[10] Yousri Kouki, Frederico Alvares de Oliveira Jr., Simon Dupont, Thomas Ledoux: A Language Support for Cloud Elasticity Management. CCGRID 2014: 206-215.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE2021. 0903108|

[11] Sheheryar MalikandFabriceHuet "Adaptive Fault Tolerance in Real Time Cloud Computing" 2011 IEEE World Congress on Service.
[12] Wenbing Zhao, P.M. Melliar and L.E. Mose" FaultTolerance Middleware for Cloud Computing" 2010 IEEE 3rd International Conference on Cloud Computing.
[13] Jayadivya S K, JayaNirmala S, Mary SairaBhanus"Fault Tolerance Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing" International Journal on Computer Science and Engineering (IJCSE)
[14] Ravi Jhawar, Vincenzo Piuri and Marco Santambrogio"A Comprehensive Conceptual System level Approach to Fault Tolerance in Cloud Computing" IEEE.
[15] Fumio Machida, Ermeson Andrade, Dong SeongKim and Kishor S. Trivedi"Candy: Component-based Availability Modeling Framework for Cloud Service Management Using Sys-ML" 2011 30th IEEE International Symposium on Reliable Distributed Systems.
[16] Sun Microsystems, Inc. "Introduction to Cloud Computing Architecture" White Paper 1st Edition, June 2009.
[17] Mladen A. Vouk, " Cloud Computing – Issues, Research and Implementations", Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA,Journal of
Computing and Information Technology - CIT 16, 2008, 4, 235 –246doi:10.2498 /cit.1001391
[18] Alexander Keller and Heiko Ludwig. The wsla framework : Specifying and monitoring service level agreements for web services. J. Netw. Syst. Manage., 11(1) :57–81, March 2003.
[19] N. Ghosh, S. K. Ghosh, and S. K. Das, "SelCSP: A Framework to Facilitate Selection of Cloud Service Providers," IEEE Transactions on Cloud Computing, vol. 3, no. 1, pp. 66–79, 2015.
[20] S. T. Halkidis, N. Tsantalis, A. Chatzigeorgiou, and G. Stephanides, "Architectural Risk Analysis of Software Systems Based on Security Patterns," IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 3, pp. 129–142, 2008.
[21] S. Madria and A. Sen, "Offline Risk Assessment of Cloud Service Providers," IEEE Transactions on Cloud Computing, vol. 2, no. 3, pp. 50–57, 2015.
[22] J. Luna, N. Suri, M. Iorga, and A. Karmel, "Leveraging the potential of cloud security service-level agreements through standards," IEEE Transactions on Cloud Computing, vol. 2, no. 3, pp. 32–40, May 2015.
[23] D. R. dos Santos, R. Marinho, G. R. Schmitt, C. M. Westphall, and C. B. Westphall, "A Framework and Risk Assessment Approaches for Riskbased Access Control in the Cloud," Journal of Network and Computer Applications, vol. 74, pp. 86 – 97, 2016.
[24] V. Bellandi, S. Cimato, E. Damiani, G. Gianini, and A. Zilli, "Toward Economic-Aware Risk Assessment on the Cloud," IEEE Security & Privacy, vol. 13, no. 6, pp. 30–37, 2015.
[25] G. Chen, H. Jin, D. Zou, B. B. Zhou, Z. Liang, W. Zheng, and X. Shi, "SafeStack: Automatically Patching Stack-Based Buffer Overflow Vulnerabilities," IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 6, pp. 368–379, 2013.