International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 5789-5801
© Research India Publications
http://www.ripublication.com

Autonomic Cloud Services to Enhance Secure Data Services

N.Raveendran and DR. Antony SelvadossThanamani

Research Scholar, Bharathiar University
Coimbatore, India
raveeng@gmail.com
Associate Professor& Head,Department of Computer Science
NGM College, Pollachi, India.
selvdoss@gmail.com

Abstract

Cloud computing is an emerging technology model, specifically designed to provide IT services for business and other organizations. It avoids various compatibility issues like hosting ofmultiple servers and other equipment's locally, continuous installations of various software and their up-gradationsand failures in thehardware installed. This technology simplifies the processes and also saves time and money. This proposed work has been concentrated on providing secured data service for the online digital library. Development of a new system comprising the advantages of cloud computing, autonomic computing and service oriented architecture could be beneficial to libraries in the areas such as data security, technology and the user community.

1. Introduction

Cloud computing is an emerging technology at present. Lots of small and medium business industries are increasingly realizing the benefits of cloud computing and are moving towards thiscomputing environment. By using cloud computing, the users will be able to, quickly and dynamically share their resources at minimal costs. Due to this increased efficiency and decreased cost, this service has got a huge popularity in digital libraries of many institutions. The major cloud service providers are Google, Microsoft, Amazon, IBM and Sales force etc. The institutions which use cloud for their digital libraries are little discomfort due to many uncovered potential and critical threats which remain assecurity issues. Security is a major issue in digital library system. The security weakness in digital libraries leads to unauthorized access of confidential information or loss of integrity of the stored data. Also, it may leads to loss of trust for publishers and for other content providers on library.

Complexity is a major problem, which reduces the effectiveness of computing in many environments. Existing, non-autonomic based cloud management system provides more complexity. One of the important factors contributing to the complexity in problem determination is the various ways through which system is reporting about different parts such asevents, conditions, errors, and alerts. Some of the challenges faced by the non-autonomic cloud system [10, 7] are;Co-optimization of Resources, Quality of Service and Reliability. Some of the client site issues faced in cloud computing environments are [13]; Paging, Runtime checking, recovery and verification, Stream filtering, Environment awareness evolution and adaption, efficient client-site execution, catching and pre-fetching.

This paper addresses the open issues of data access control in digital library. It proposes an efficient, scalable and secure data access control mechanism using cloud,service oriented architecture and autonomic computing mechanism.Autonomic computing model is a self-protected, self-healing, self-configured and self-managed one. It is designed to mimic the human body nervous system [5]. In mid-October 2001, IBM released a manifesto observing that the main obstacle to further progress in IT industry is a looming software complexity crisis. IBM cited an application which contains millions of lines of codes. The manifesto pointed out that the difficulty of managing today's computing system goes well beyond the administration of individual software environments. To solve the problem, they tried lot of approaches. But the only option remained is autonomic computing, which can able to manage themselves given high-level objectives from administrators. It is introduced by IBM's Senior Vice President of research Paul Horn to national academy of engineers at Harvard University in 2001.

In [11] Roy Sterritt and Mike Hinchey discussed that the autonomic computing has as its vision, the creation of self-managing systems to address today’s concerns of complexity and total cost of ownership while meeting tomorrow’s needs for pervasive and ubiquitous computation and communication. In [6] Kurian D. and Chelliah P.R., statedthat though the vision of autonomic computing (AC) is highly ambitious, an objective analysis of autonomic computing and its growth in the last decade has thrown more incisive and decisive insights on its birth deformities and growth pains.

2. Autonomic based cloud management system

In this proposed work, existing method of cloud based management system is replaced with implementation of autonomic cloud manager for online digital library system. This research work introduced existing familiar SQL query to control this autonomic based cloud system. The deployment process of autonomic cloud service is a challenging as autonomic and cloud computing are still maturing. Managing such implementation has many open issues, such as, which autonomic policy needs to be applied on full system; and which needs to be applied on partial; so that, the system can result into maximum throughput and becomes highly available. The Autonomic

Manager (AM) will enable online digital library system to be more reliable & highly available. The Autonomic Manager will follow the stages that will include Monitor, Analyse, Plan and Execute in a cyclic manner. The monitor process will be responsible for collecting and organizing data from the required sources. A large portion of knowledge will be from the monitored information over a period of time. This paper introduced autonomic cloud based management system to control the online digital library system which provides runtime decision making.

The cost of implementation for autonomic cloud for digital library systems will be lesser compared to non-autonomic cloud. The maintenance cost of library software without AM will be higher than the development cost with AM. The downtime of system will be reduced so that the cost of maintenance will also be reduced with the presence of autonomic cloud based managers system. The usage of multilevel AM will improve the overall system. Autonomic Cloud infrastructure can be used more efficiently to provide better services compared to the existing system.

2.1 Autonomic Manager

The component of autonomic manager implements the control loop. It is executed through numerous permutations of management tools and products, or distributed by resource provider in runtime environment. Four sections of the autonomic manager are monitor, analyse, plan and execute. The monitor section of the autonomic manager collect, aggregate, filter, manage and report the details which are collected from an element. The analyse part of the autonomic manager provides the mechanisms to correlate and model complex situations. The plan part of the autonomic manager structures the action which is needed to achieve the objectives and goals. It uses policy information to guide its work. The execute part of autonomic manager controls the execution of plan with considerations for on-time updates [12].

Some of the key characteristics of autonomic computing are; the computing system needs to know-itself and comprise components that also possess a system identity. It configures and reconfigures itself under varying and unpredictable conditions. It never settles for the status quo-it always looks for ways to optimize its workings. It performs something akin to healing it must be able to recover from routine and extra-ordinary events that might cause some of its parts to malfunction. A virtual world is no less dangerous than the physical one, so an autonomic computing system must be an expert in self-protection. An autonomic computing system knows its environment and the context surrounding its activity, and acts accordingly. An autonomic computing system cannot exist in a hermetic environment. Perhaps most critical for the user, an autonomic computing system will anticipate the optimized resources needed while keeping its complexity hidden. Self-Healing, Self-Optimizing, Self-Protecting and Self-Configuring are the four fundamental elements of autonomic computing [12].

2.2 Autonomic Manager Architecture

The autonomic architecture splits the loop into four parts which share knowledge. The following sections discuss the parts of autonomic architecture in detail. These four parts work together and provide the control loop functionality. The autonomic manager's knowledge has been built based on the known information about the system. Continuously, learns more about the characteristics of the managed resources. The gathered knowledge has been continuously shared among the four parts, to make decisions by the parts. The control loop carries out tasks as efficiently as possible based on high-level policies. All of the actions in the autonomic control loop either make use of or supplement the knowledge base for the resource. For example, the knowledge base helps the analysis phase of the control loop to understand the information it's getting from the monitor phase. It also provides the plan phase with information that helps in selecting the action to be performed. Also, the entire autonomic control loop needs not to be handled by a single application. Possibly, we may have an application which may use only analyse and monitor phases for e.g. IBM Director, and another application which may use only plan and execute phases for e.g. Toshiba Cluster Perfect.

2.3 Working principle of autonomic based cloud management system

Autonomic computing system solves the complexity problem. The following figure 1 shows the Autonomic Based Cloud Management System. In this diagram, in the resource manager, various components such as application, database, application server, servers, storage device, and networks are included.

These components are a part of the troubleshooting process when a problem occurs. In this system, each component produces multiple log files individually in its own format in various locations. Because there is no cost-effective way to change log files in legacy applications or solutions that have already been deployed, the IBM autonomic computing architecture includes adapters to translate disparate logs into the common format [4]. Adapters help in keeping implementation cost low, and adapters also allow business to use applications from independent software vendors that may not adopt common log formats. Log formats that are familiar enables specialist to easily look for problems in logs and take necessary actions, if needed. This system also illustrates an autonomic manager engine which automates the process that a specialist would use. For instance, IBM has developed the Log and Trace Analyser for Autonomic Computing, which enables the reading of logs in the common format, correlating the logs based on different criteria and viewing the correlated log records [4]. To take the best possible action when it discovers a problem, the manager will rely upon other sources of knowledge. One such source might be a symptom service.

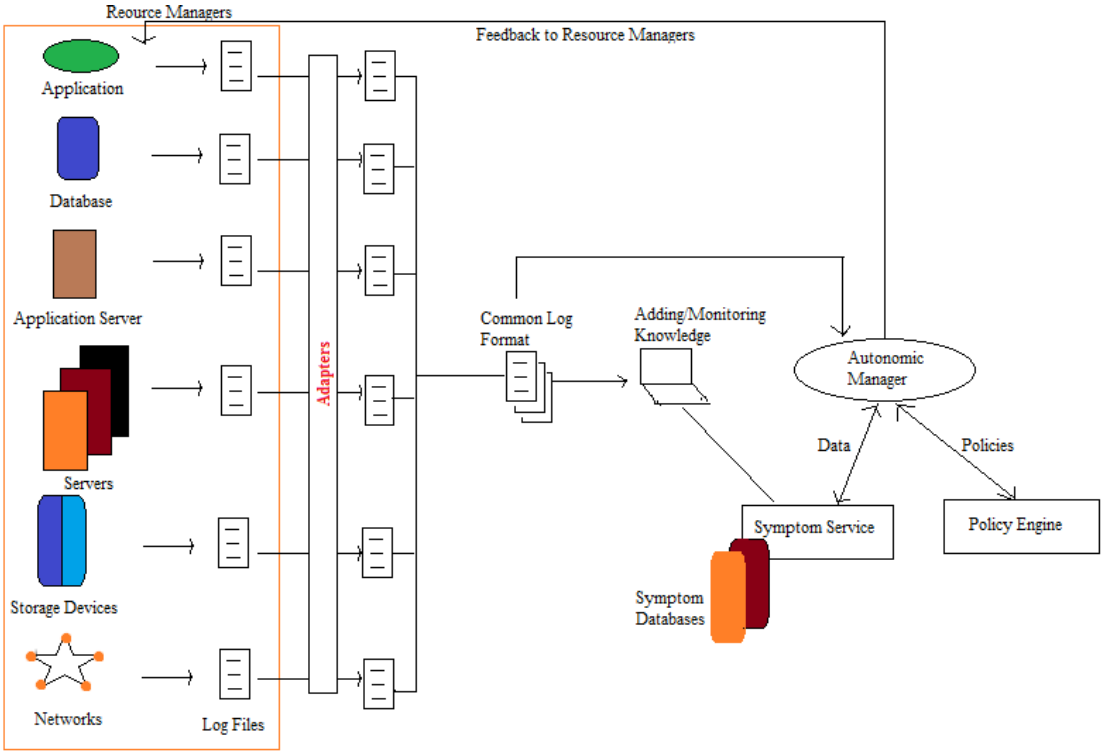

Figure 1: Working principle of autonomic based cloud management system

The symptom service includes a symptom database that contains information about how to detect patterns that indicate problems, how to diagnose that a specific problem has occurred, and how to resolve that specific problem. The symptom database will include a standardized set of interfaces and data formats that facilitate the determination of actionable causes from problem data. In many instances, multiple symptom databases are possible and likely, all presented as part of a symptom service. The process of writing and populating knowledge bases, such as the symptom database, is made simpler by the existence of a common format for log data.

Once the decisions are made about how to best resolve a problem, the autonomic manager may then query other managers, such as a policy engine to determine which corrective actions can be taken. The policy engine matches proposed solutions against rules and policies by ensuring that an action is possible to effect on business-critical processes which is appropriate to the overall situation. For example, the symptom database may report that two separate actions could be used to address a symptom. The first, which would fix the problem, is to reboot a system. The second may be a temporary solution, such as increasing swapper space.

In this example, if there exists a policy which would state that a critical system was not to be rebooted during business hours, the policy service would instruct the autonomic manager to use the temporary solution; the autonomic manager, in turn, would then provide feedback to the resource managers, which would make the necessary changes.

2.4 Autonomic Cloud Storage

In [2] Gabriel Antoniu states, while the cloud computing paradigm is progressively being adopted by companies aiming to deliver large-scale distributed services, such as Amazon, IBM, Google or Yahoo!, the service level provided for data storage remains rather basic. This talk will discuss several issues related to building advanced facilities for data sharing on cloud infrastructures. He discussed how open issues raised by the need for efficient, secure and reliable storage service for data intensive distributed applications running in cloud environments may be addressed by enabling an autonomic behaviour for the cloud storage infrastructure.

2.5 Autonomic manager based data access control

In [8] Omid Mola and Michael A. Bauer, denotes the management of clouds comprised of hundreds of hosts and virtual machines present challenging problems to administrators in ensuring that performance agreements are met and that resources are efficiently utilized. Automated approaches can help in managing such environments. Autonomic managers using policy-based management can provide a useful approach to such automation. They outlined how collections of collaborating autonomic managers in cloud can be a step towards better management of clouds.

In [9] Pannu, H.S., Song Fu et al., denotes the Cloud computing has become increasingly popular by obviating the need for users to own and maintain complex computing infrastructure. However, due to their inherent complexity and large scale, production cloud computing systems are prone to various runtime problems caused by hardware and software failures. Autonomic failure detection is a crucial technique for understanding emergent, cloud-wide phenomena and self-managing cloud resources for system-level dependability assurance. To detect failures, it is essentialto monitor the cloud execution and collect runtime performance data. These data are usually unlabelled, and thus a prior failure history is not always available in production clouds, especially for newly managed or deployed systems.

3. Proposed system

The proposed research uses the autonomic cloud computing technique for secured information sharing. Autonomic computing system enables self-healing, self-protection, self-optimization and self-configuration. The decisions are made on its own by following pre-defined high level policies. This autonomic computing system is made up of autonomic components which interact with each other. It manages them without any human intervention. So instead of controlling the system directly, human defines some rules to direct the self-management process.

In data cloud, users store their data in the cloud and no longer possess the data locally. Thus, the correctness and availability of the data files being stored on the data cloud must be guaranteed. Some of the key issues of existing non-autonomic cloud based system are reliability, co-optimization of resources and quality of service. An unstable cloud storage system is a liability. No one wants to save data to a failure-prone system, nor do they want to trust a company that is not financially stable. While most cloud storage systems try to address this concern through redundancy techniques, there is still the possibility that an entire system could crash and leave clients with no way to access their saved data. Using Autonomic feature we can reduce this risk of unstable systems and can achieve the reliability to a satisfactory level.

The co-optimization of infrastructure, platform and online digital library application is also bigger challenge. Infrastructure, platform, and application can be tuned & optimized separately, but, how they will work in collaboration is still an open challenge and by further research and experiments this issue needs to be addressed. The Quality of Service (QoS) is also the main challenge while implementing cloud services. Cloud Service Providers (CSP) are still not matured enough to provide a highly scalable and reliable services on cloud platform using autonomic features on low cost to the end users. This issue needs to be addressed by cloud service providers to avoid or minimize by dynamically provisioning the right amount of resources in a timely manner.

In this proposed work, existing method of cloud based management system is replaced with an autonomic cloud manager for online digital library system. This proposed work, introduces existing familiar SQL query to control this autonomic based cloud system. The deployment process of autonomic cloud service is a challenge as currently autonomic and cloud computing are still maturing. Managing such implementation has many open issues, such as, which autonomic policy needs to be applied on full system; and which needs to be applied on partial; so that, the system can result into maximum throughput and becomes highly available. The Autonomic Manager (AM) will enable online digital library system to be more reliable & highly available. The Autonomic Manager will follow the stages that will include Monitor, Analyse, Plan and Execute in a cyclic manner. The monitor process will be responsible for collecting and organizing data from the required sources. A large portion of the knowledge will be from the monitored information over a period of time. This proposed work introduces autonomic cloud based management system to control the online digital library system which provides runtime decision making.

The cost of implementation for autonomic cloud for digital library systems will be lesser compared to non-autonomic cloud. The maintenance cost of library software without Autonomic Manager will be higher than the development cost with Autonomic Manager. The downtime of system will be reduced so that the cost of maintenance will also be reduced with the presence of Autonomic cloud based management system. The usage of multilevel Autonomic Manager will improve the overall system. Autonomic Cloud infrastructure can be used more efficiently to provide better services compared to the existing system.

3.1 Enhanced Autonomic Manager in Autonomic Based Cloud Management System

The management of clouds comprised of hundreds of hosts and virtual machines present challenging problems to administrators in ensuring that performance agreements are met and that resources are efficiently utilized. Automated approaches can help in managing such environments. Autonomic managers using policy-based management can provide a useful approach to such automation [8]. Autonomic

Computing (AC) [1] has been widely adopted in Cloud computing as a starting point for responsiveness and autonomy in front of environment changes such as the end users fluctuating number of requests or the infrastructure administrators maintenance. It overcomes the rigid scheme of management, which provides a single Autonomic Manager and no separation of concerns [15].

Autonomic computing is a self-managing computing model named after, and patterned on, the human body's autonomic nervous system. An autonomic computing system would control the functioning of computer applications and systems without input from the user; similar to the autonomic nervous system regulates body system without conscious input from the individual. The goal of autonomic computing is to create systems that run themselves, capable of high-level functioning while keeping the system's complexity invisible to the user. In general, in self-managing autonomic system, the knowledge can be almost any sort of structured data which is used to do a process, especially the processes that can be automated. The data stored on the log files also included with this broad scope of knowledge. The autonomic computing architecture cites the three examples of broad types of knowledge such as; Policy knowledge, Solution topology knowledge and Problem determination knowledge. In a self-managing autonomic system, instead of controlling the system directly, the user defines general policies and rules that guide the self-management process. For this process, IBM defined four functional areas such as:

- Self-configuration - Automatic configuration of components.
- Self-healing - Automatic discovery and correction of faults.
- Self-optimization - Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements.
- Self-protection - Proactive identification and protection from arbitrary attacks.

The following figure 2 shows architecture of autonomic manager;

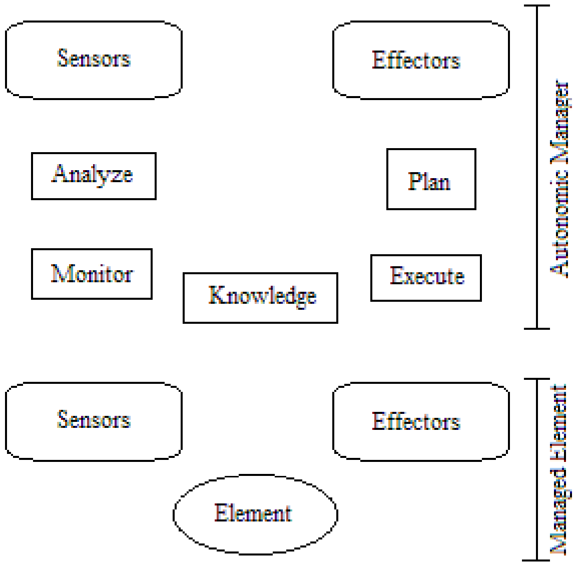

Figure 2: Autonomic Manager Architecture

An autonomic manager is a component, it implements the control loop. The loops can be executed through numerous permutations of management tools and products, or distributed by a resource provider in runtime environment. The structure of the autonomic computing can be defined based on the following four steps; Monitor,Analyse, Plan and Execution.

The monitor function could signal for the plan function to create a new plan. The plan function could signal for the monitor function to accumulate more or less information. The asynchronous collaboration technique, such as a messaging bus, is how the four functions collaborate with one another.

The managed element is controlled through its sensors and effectors. The sensors provide mechanisms to collect information about the state and state transition of an element. To implement the sensors, either use a set of "get" operations to retrieve information about the current state, or a set of management events that flow when the state of the element changes in a significant way. The effectors are mechanisms that change the state of an element. In other words, the effectors are a collection of "set" commands or application programming interfaces (APIs) that change the configuration of the managed resource in some important way.

3.2 Secured Data Services using Autonomic Based Computing

The major obstacle that concerns researchers to concentrate is complexity. However, as the scale and complexity of these systems and applications grow their development, configuration and management challenges are beginning to break current paradigms, overwhelm the capabilities of existing tools and methodologies, and rapidly render the system and applications, unmanageable, brittle and insecure. So, researchers were faced with the task of finding an alternative approach to complexity. After exploring multiple methodologies, researchers were finally introduced the methodology called autonomic computing [12].

It's time to design and build computing systems capable of running themselves, adjusting to varying circumstances, and preparing their resources to handle most effectively the workloads we put upon them. These autonomic systems must anticipate needs and allow users to concentrate on what they want to accomplish rather than figuring how to rig the computing systems to get them there [3]. The name autonomic, the biological concept in which autonomic computing originated is the human body's autonomic nervous system.

3.3 Performance Evaluation

Advanced computing on cloud computing infrastructures can only become viable alternative for the enterprise, if these infrastructures can provide proper levels of non-functional properties. A company that focuses on service-oriented architectures (SOA) needs to know what configuration would provide the proper levels for individual services if they are deployed in the cloud. In this research work, autonomic computing is used for self-healing the system in case of any power failure. In this work, CloudSim open source simulation software is used for performance evaluation. Cloud computing providers assure certain service levels and this is typically done for the platform and not for a particular service instance. This paper focuses on

Autonomic Based Cloud Management System to provide secured data service for digital library.

3.4 Autonomic computing Self-healing Systems vs. General Computing Systems

Complexity in problem determination is reducing the effectiveness of computing in many computing environments. One of the major factors contributing to the complexity in problem determination is the various ways that different parts of the system report events, conditions, errors, and alerts. For example, examine the ways that general computers maintain logs for the system. These logs contain a variety of content in differing formats because solutions are built using disparate pieces and part, often with products from multiple vendors. Generally, computing environments and the obstacle of combining hardware and software components are a typical solution. Most of the logging done today is focusing on reporting data that a product developer considers important for debugging the problem in a single product, as opposed to providing data to debug a solution. This inconsistency in both the format and the content that is made available by products makes it more difficult to write management tools that might ease the complexity issues. To optimize the usefulness and business value of existing and future e-business solutions, major changes and improvements in problem determination must help businesses deal with complexity, ease cross-product problem determination and automate the process of identifying and fixing frequently occurring problems.

Autonomic computing systems are the solution to solving the problem of complexity. Let’s begin with the resource manager. In the resource manager, various components such as applications, database, application server, servers, storage device, and networks are included. Symptoms will be more easily expressed using the common set of terminology and data, alleviating the need for symptoms to be coded using the nuances of how, for example, an individual product says that it has “stopped.” Once the decisions are made about how to best resolve a problem, the autonomic manager may then query other managers, such as a policy engine to determine which corrective actions can be taken. The policy engine matches proposed solutions against rules and policies to help ensure that an action’s possible effect on business-critical processes is appropriate to the overall situation. For example, the symptom database may report that two separate actions could be used to address a symptom. The first, which would fix the problem, is to reboot a system. The second may be a temporary solution, such as increasing swapper space. In this example, if there were a policy that stated a critical system was not to be rebooted during business hours, the policy service would instruct the autonomic manager to use the temporary solution; the autonomic manager, in turn, would then provide feedback to the resource managers, which would make the necessary changes.

3.5 Secured Data Service System Compared with Manual and Autonomic Based Cloud Management System

The table 1 shows the comparison of cloud computing performance before and after implementation of Autonomic Based Cloud ManagementSystem

Table 1: Comparison between Autonomic Based Cloud Management System (ABCMS)and manual system

	Application	Database	Application server	Servers	Storage Devices	Networks
QoS	56	67	58	63	71	58
QoS(ABCMS)	75.7	74.4	80.7	73.9	82.9	71.3
Response Time	75	37	74	79	62	48
Response Time(ABCMS)	81.1	76.2	87	86.3	81.3	77.8
Percentage of Errors	89	51	46	74	58	47
Percentage of Errors(ABCMS)	40	32	30.1	26.6	23	19.8
Down Time	54	55	62	70	64	65
Down Time(ABCMS)	30.1	29.9	27.6	28.1	20.2	22.8
Up Time	46	45	38	30	46	35
Up Time(ABCMS)	69.9	70.1	72.4	70.9	78.8	77.2

On this result, Autonomic Based Cloud Management System proves that it provides best performance compared with manual system. It providesmore error free result. So it proves Autonomic Based Cloud Management System provides secured data service on cloud.

4. Conclusion

Cloud computing becomes increasingly popular within business world during recent years. It provides opportunity to dynamically scale the computing resources for applications. These resources are shared among users using virtualization technology. Several industrial providers run large scale cloud computing platforms which offer different services at various levels of platform, infrastructure and application. It not only operates on local computers, but also, on centralized facilities by various storage utilities and third-parties computing. Cloud computing offers both the applications delivered as services over the internet and system software/hardware in datacentre as service providers. Utilization of cloud resources are an open challenge. Even though cloud computing environment has large setup and cost associated with it, the testing applications and resource allocation policies are highly challenging.

The cloud computing performance has been analysed with the implementation of Autonomic Based Cloud Management's System. Consequently, ten experiments were taken with the help of 'Autonomic Based Cloud Management Systembased secured system. Then, finally,average of these experiments was compared with non-autonomic output dataset. Further, it is proved that 'Autonomic Based Cloud Management Systemprovides secured data service compared with non-autonomic based system.

References

[1] FredericoAlvares de Oliveira Jr, Remi Sharrock et al., "A framework for the coordination of multiple autonomic managers in cloud environments", 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, pp. 179-188.

[2] Gabriel Antoniu, "Autonomic Cloud Storage: Challenges at Stake", 2010 International Conference on Complex, Intelligent and Software Intensive Systems, IEEE, pp. 481.

[3] Horn, Paul, “Autonomic computing: IBM’s perspective on the state of information technology”, In IBM Research, October 2001.

[4] IBM Corporation, “Automating problem determination: a first step towards self-healing computer systems”, www.research.ibm.com/autonomic, 2003.

[5] Jeffrey O.Kephart and David M.Chess, "The Vision of Autonomic Computing", IEEE, January 2003, pp. 41-50.

[6] Kurian, D. and Chelliah, P.R., "An autonomic computing architecture for business applications", IEEE, Information and Communication Technologies (WICT), 2012 World Congress, pp.442 - 447.

[7] N. Raveendran, Dr. Antony SelvadossThanamani et al., "Autonomic Cloud Services using Efficient Data Mining Techniques for Global Online Digital Libraries", International Conference On Academic Libraries 2013, pp. 173-177.

[8] Omid Mola and Michael A. Bauer, "Towards Cloud Management by Autonomic Manager Collaboration", Int. J. Communications, Network and System Sciences, 2011, 4, pp. 790-802.

[9] Pannu, H.S., Song Fu et al., "AFD: Adaptive failure detection system for cloud computing infrastructures", IEEE, Performance Computing and Communications Conference (IPCCC), Dec 2012, pp.71 - 80.

[10] RajkumarBuyya, Xiaorong Li et al, "Autonomic Cloud Computing: Open Challenges and Architectural Elements, Third International Conference of Emerging Applications of Information Technology (EAIT 2012, IEEE Press, USA) 2012, pp. 3-10.

[11] Roy Sterritt and Mike Hinchey, "Tutorial Proposal: Autonomic Computing in Real-Time Systems" [www.artes.uu.se/events/summer05/Hinchey_tutorial.pdf]

[12] Sharee S. Laster, Olatunji, B.S et al., "Autonomic Computing: Towards a Self-Healing System", Proceedings of the Spring 2007 American Society for Engineering Education Illinois-Indiana Section Conference, American Society for Engineering Education, pp. 63-79.

[13] Wei-Tek Tsai, Elston, J. et al., "Real-Time Service-Oriented Cloud Computing", 2010 6th World Congress, 2010, pp. 473-478.

[14] www.wikipedia.org

Authors Biography

Sh. N.Raveendran is working as Scientist in Government of India and is a research scholar of Bharathiyar University, Coimbatore, India. He is having rich experience in ICT industry with primary skills of project management, software development, managing team & clients. He has vast experience of working in the domain of e-Governance. His areas of interests are Cloud computing, Data mining and Knowledge Engineering. Academically, he is having M.Phil., (CSE) and MCA to his credit.

Dr. Antony SelvadossThanamani is presently working as Professor and Head, Dept of Computer Science, NGM Collegeaffiliated to Bharathiar University, Coimbatore, India. He has published more than 100 papers in international/ national journals and conferences. He has authored many books on recent trends in Information Technology. His areas of interest include E-Learning, Knowledge Management, Data Mining, Networking, Parallel and Distributed Computing. He has to his credit 24 years of teaching and research experience. He is a senior member of International Association of Computer Science and Information Technology, Singapore and Active member of Computer Science Society of India, Computer Science Teachers Association, New York.