Overview of Autonomic Computing Systems

Taranjeet Singh, Sammer Kandpal

Abstract— Today software systems are becoming more dynamic, unmanageable, heterogeneous and complex day by day. IT companies are facing problems in maintaining, storing, and managing software systems. Solution to these problems is to develop software systems based on Autonomic computing. Researchers and scientists have put their ideas to develop software systems based on autonomic computing. These software systems can manage themselves by their own or with minimal human interference and they only require high level guidance from human experts. Autonomic computing based software systems can maintain and adapt themselves according to the changing IT environment so they are solution to increasing dynamism, unmanageability and increasing complexity. This paper tends to provide a survey of autonomous systems, their challenges, characteristics, architecture and applications.

Index Terms— Self management, Autonomic Computing, Autonomic Software Systems, Self Properties, Autonomic Element, Knowledge Source, Managed Element, Manual Managers, Touch Points.

1 Introduction

Improvements in technology, methods and software tools have resulted rapid growth in services, applications and information infrastructure. Systems are becoming more dynamic, heterogeneous, complex and uncertain due to which they became more brittle, insecure and unmanageable. IT companies on other hand have encountered problems in maintaining, development and management of these complex systems. All these current issues in system led the researchers to investigate new paradigm which is based on Biological systems called "Autonomic computing" to handle these complexities. As biological systems such as Human Body have ability to heal themselves, protect from dangers manage its internal state, adapt external environments etc[3], similarly Autonomic computing systems have similar characteristics as biological systems. This conceptwas used to implement DB2 Configuration Advisor and Tivoli Risk Manager by IBM [1]. The term Autonomic is associated to an environment that automatically responds to any faults, problems, failures of systems etc [2]. The aim of autonomic computing is to design and develop a system that can manage itself without external intervention or interference as our human nervous system do. Idea of autonomic computing systems was given by Paul Horn (IBM Senior Researcher) in 2001[3].Autonomic Computing is an environment that has the ability to dynamically adapt to any changes according to high-level business policies [4]. And also it is an evolution to handle the growing complexity of systems [5].These system have characteristics or self-properties such as self-configuring, self-healing, self-optimizing and self-protecting, these four are major characteristics called self-CHOP[6], open, context/environment aware, anticipatory, self-adapting, self-aware, self-managing, self -recovering etc. Any software system can be injected with self -properties and can be converted into autonomic software system. These software systems have ability to manage themselves without any or with minimum human interference or involvement. They have freed human administrators and users from their complexities, once autonomous software is developed according to policies then there is almost no involvement of humans in managing them. Policies are the set of rules that are defined for autonomic systems to achieve their goal; actually these are business policies which are converted into low level or system understandable form by designers and developers of autonomous software systems. There are lots of challenges in designing, developing and maintaining autonomous software systems. This paper provides characteristics, architecture, applications and tends to identify challenges and issues of such systems.

2 Autonomic Computing Systems

The Autonomic Computing is inspired from human body's autonomic nervous system [3]. Although it is new field but it is combination of theories from several existing areas such as control theory, adaptive algorithms, robotics, software agents, distributed systems, real time systems, machine learning, artificial intelligence etc. Autonomic computing software system is a system injected with self-properties and has ability to manage itself and adapt to any changing environment, tune its resource to meet user requirement. These systems work on the basis of situations they observe or sense and take actions accordingly, without interruption of IT professional. According to Paul Horn's definition, an autonomic computing system is a self- management system with eight elements [3], these elements are:-

- Computing system must "know itself" and consists of components that also possess a system identity.
- It must have ability to configure and reconfigure itself under changing and unpredictable environmental conditions.
- An autonomic computing system always looks for ways to optimize its workings.
- An autonomic computing system must be able to recover from events that might cause some damage i.e. it must heal itself.
- An autonomic computing system must be able to protect itself without human involvement.
- An autonomic computing system must know its internaland external environment and should act accordingly.

- An autonomic computing system should be open as it cannot exist in a hermetic environment.
- An autonomic computing system should anticipate the optimized resources while keeping its complexity hidden form the users.

As Autonomic computing is emerging and new field so there are some issues and problems which are to be sorted out while designing, developing and maintaining these systems. Some of issues and challenges in autonomic computing are explained in this section.

- Measuring characteristics of autonomous system is a challenge; this can be achieved by developing a framework by which can bridge gaps between quality factors and autonomic characteristics [1].
- While designing and developing autonomic systems IT professional faces problems in identifying techniques and methods that are best suited for its development. Once appropriate techniques are identified then it is difficult to combine them. It is a challenge of developer to develop autonomic system from existing system; the existing system can be non-autonomic or semi autonomic system [1].
- While performing requirement analysis of autonomic software systems the major challenge is to capture requirements of the stakeholders, then translating, modeling and relating these requirements as per expectations of stakeholders [7].
- In self-healing and self-protecting characteristic challenge is that how the system will be able to detect the faulty components, protect itself and how will it recover from it. After healing the system there should not be any adverse effect on the system and it should not divert from its goals [7].
- Translation of policies in these systems is other challenge as all the high level policies are firstly converted into low level policies which are understood by the system. Various techniques and algorithms should be used for performing translation [7].
- When we are using many autonomic characteristics together in a system then our challenge is how we will coordinate them, how these characteristics will be related to each other and while executing their jobs major concern is priority of these characteristics in which they will execute their tasks [7].

3 CHARACTERISITCS

Characteristics of autonomic computing based systems are sub-divided into major and minor characteristics. There are four major characteristics i.e. self-configuring, self-healing, self-optimizing and self-protecting. Minor characteristics are open, context aware, self-aware, self-adapting, anticipatory, Self-governing, self-monitoring, self-managing etc. Figure 1 shows self characteristics autonomous system. Earlier software system was installed and configured manually and it was very time consuming, but using self configuring property software can be installed, configured and reconfigured dynamically with minimum or no human involvement, using this property autonomic system must adjust and adapt itself in changing environments automatically using the policies provided by IT professional for meeting defined business goals. Self configuring provides maintainability, functionality, portability and usability to the system [1].

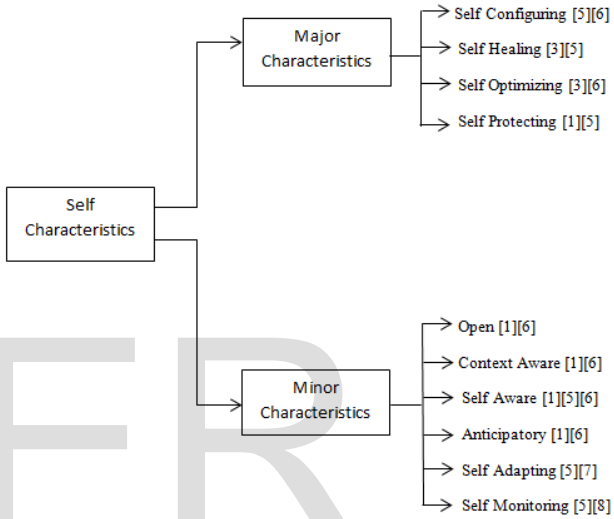

Figure 1: Characteristics of Autonomic systems

It can heal itself and its components, it identifies fault, repair or replace it automatically using self-healing characteristic. Self-healing components identifies the malfunctioning and then take corrective action without disrupting IT environment. Self -healing can be done in two modes i.e. reactive mode and proactive mode. In reactive mode when fault occurs then healing components react accordingly and try to recover from it. In proactive mode system continuously monitors and tries to prevent system from faults or any disruption. The main objective of self-healing is to maximize availability, maintainability, survivability, and reliability of the autonomic application and system [1]. By self optimizing characteristic autonomic software systems will try to be more efficient, tune its components dynamically, improve its performance and execution and system will find optimal ways for improving overall utilization, it provides efficiency, maintainability and functionality to autonomous systems. Resource utilization and workload management are two important aspects of self optimization [6]. Autonomic system must be capable enough for detecting and protecting itself from faults, threats, attacks (both internal and external) and must maintain overall system security and inte-

grity. Self protecting provides reliability and functionality to the systems [1]. These systems must be able to detect hostile behaviors and other problems from the reports generated by sensors and must be able to defend them. Hostile behaviors can be virus attacks, accidental attacks, malicious attacks, unauthorized access, system failure and denial of service attacks etc. [2].An autonomic software system cannot survive in hermetic environment it is always developed for heterogeneous environment so it must be open [1]. As human body is aware of its surrounding environment similarly an autonomic system must be aware of its surrounding environment or its context and must be able to react to changes in environment, this property is called context awareness [1, 6]. An autonomic system should anticipate optimal required resources while hiding its complexity from the end user and it must also try to satisfy user's requirements i.e. it must be anticipatory [1]. An autonomic system should be able to adapt to any changes in the environment such as addition of new component, removal of component, execution platform changes, business policy changes etc, it should be self adapting [7]. An autonomic system must be able to manage itself, its resources, its components, parameters and environment this is called self-managing [8].An autonomic system must be self- aware i.e.it should be aware of its current state, resources and component. It must be also aware about which resources can be borrowed and which resources can be shared. This characteristic is also called self-knowledge [6, 9].

4 ARCHITECTURE OF AUTONOMIC COMPUTING SYSTEMS

Architecture of autonomic system consists of five building blocks these are Autonomic manager, Knowledge source, Touch point, Manual manager and Enterprise service bus ;they work together to provide self-management capabilities to such systems [4].These building blocks are explained in this section.

- **Autonomic Manager**: It uses control loop for performing management functions on the Managed element. Control loop contains four functions i.e. monitor, analyze, plan and execute (MAPE) and it can be used to manage any hardware or software component. Autonomic manager can be software agent consisting of two elements i.e. autonomic element and managed element. Managed element can be any hardware, software, database server or entire system consisting of sensors and effectors. Sensors gather and sense information of current state of managed element and store this information in knowledge base, it is also called probes or gauges [10]. Effectors help autonomic manager to manage managed element through touch-points. Control loop consists of four functions i.e. monitor, analyze, plan and execute. These functions are explained by Ahuja.et.al [2]. Monitor function monitors system, it collects, aggregate and filter the reports collected from managed element through touch points, these reports can be details of topology, metrics, configuration settings etc. Analyze function of control loop analyzes the changes in the IT environment and takes decisions about monitored changes. Plan function generates change plans using the policy information and the execute function executes these plans made by plan function. Once change plan is generated by autonomic manager than execute function executes some action to modify the current state of managed element via actuators [11].Figure 2 shows the architecture of the autonomic element consisting of autonomic manager, sensors, effectors and managed element. Data is collected from managed element through sensors, then it is passed to control loop for appropriate action to be taken through effectors. The four components or functions of MAPE loop work together to achieve the objective by exchanging the knowledge [12].

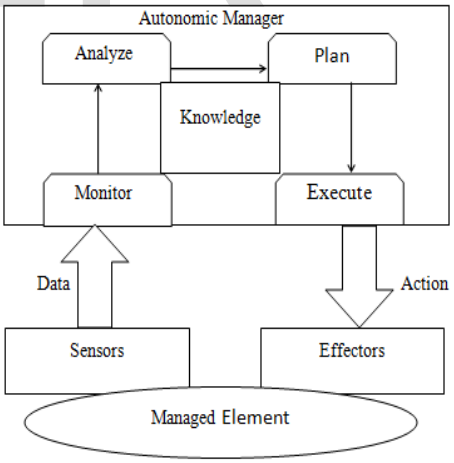

Figure 2: Architecture of Autonomic Element

- **Knowledge Source:** It contains the knowledge of particular type of data that autonomic manager uses for performing management functions. Knowledge source can be implemented through registry, dictionary, database or other repository [2]. Knowledge can be information of policies, change in plans or requests etc. Autonomic manager can obtain knowledge in

- *Taranjeet Singh is currently Assistant Professor in Apex Institute of Technology in AKTU Uttarpradesh, India, PH-9837114133. E-mail:mrtaranjeetsingh1992@gmail.com*
- *Sammer Kandpal is currently Assistant Professor in Apex Institute of Technology in AKTU Uttarpradesh, India, PH-9456505227. E-mail: Sam.coer15@gmail.com*

three ways [4], first one is that knowledge can be passed to it by this way autonomic manager receives policy knowledge. Second method is that knowledge can be retrieved from any external source and third method is autonomic manager can itself generate knowledge through the data collected from sensors.

- **Touch-points:** Touch-points perform some management operations and identify current state for a managed element. Manageability interface provides communication between autonomic manager and touch points [4]. Managed element is controlled by manageability interface through sensors and effectors. Sensors use "get" operations to identify the current state of managed element. Effectors use "set" operations to allow the managed element or resource to change its state in some way.

- **Manual Managers:** With the help of manual managers IT professional is able to perform some management operations manually. It provides a user interface through which IT professional can delegate management functions to autonomic manager [2].
- **Enterprise service bus:** It provides interaction among the building blocks of autonomic system i.e. autonomic manager, touch-points, manual manager etc by connecting and integrating them [4].

5 Applications of Autonomic Computing Systems

There have been many research efforts in industries as well as in academics to develop system based on autonomic computing. Some of existing applications based on autonomic computing are described in this section.

Application	Self-Property	Function
AntHill [1, 9]	Self configuring, Self optimization and openness.	Provide aid in designing and implementing P2P applications. Used for file sharing.
AutoAdmin [1, 5, 9]	Self tuning, self- administrating.	Helps database to track data usage and adapt application requirements.
Oceano [1, 5, 9]	Self configuring, self optimizing, self-aware and anticipatory	It develops and design cost effective prototype for data storage in utility power plant.
Software Rejuvenation [1]	Self-healing, self-protecting and self-awareness.	Used for developing techniques that manages faults and protects system from failures.
SMART [1, 5, 9]	Self configuring, self optimization, anticipatory, context aware and self-aware.	It provides self-management capabilities to database, reduces its complexities and improves its performance.
Autonomia [1]	Self-healing, self-configuring, self-optimizing, self-protecting and openness.	Provides all tools that are required to control and manage for maintaining quality of system to the developers.
ANTS [8, 10, 13]	Self-healing, self-configuring, self-optimizing, self-protecting	It is space mission of NASA (National Aeronautics and Space Administration). An ANT is an abbreviation for Autonomous Nanotechnology Swarm.
Optimal Grid[1, 11]	Self-configuring, self-optimization, anticipatory and open.	It simplifies creating and manages large scale grid application and optimizes system performance.
N1[1]	Self-configuring, self-optimization and open.	It manages data centers by providing resource virtualization, service provisioning and policy automation techniques.
Adaptive Enterprise [1]	Self-configuring, self-optimization, open and anticipatory	It provides techniques by which users can develop system in three levels. These levels are business, services and resource.
eBiquity[1]	Self-configuring, self-optimization, open, context-awareness and self-awareness.	This project was designed to identify the interaction between mobile pervasive computing, multi-agent systems and artificial intelligence techniques.

These applications and project contains some characteristics of autonomic computing not all as it is nearly impossible to develop a system with all the characteristics of autonomic computing but IT loves to make impossible possible, with the efforts of researchers and scientists it would be possible in future but lot of work is still to be done. The table shown below lists some of the applications of autonomic systems.

6 CONCLUSION

Autonomic computing based software systems are very powerful systems that are suited for only heterogeneous and dynamic IT environment and also help developers to develop systems which possess self managing behavior and also act. As properties of these systems are best suited for IT industry, many researchers, scientists have started working on these systems since 2001. A lot of research is being done and better applications are being developed day by day. Clearly these systems are shifting complexity of IT environment to themselves. In this paper we tried to put forward idea of autonomic systems, their challenges and applications of these systems. Many IT companies and researchers from various universities are working on this domain and have developed many applications successfully. Even NASA seems very much interested in autonomic computing. Autonomic systems can be used for developing better future of IT industry.

ACKNOWLEDGMENT

The authors wish to thank Dr. Avadhesh Kumar Galgotias University (India) for his support and help.

REFERENCES

[1] Salehie, M., Tahvildari, L.,"Autonomic computing: emerging trends and open problems", ACM, 2005.
[2] Ahuja, K., Dangey, H., "Autonomic Computing: An emerging perspective and issues", International conference on issues and challenges in Intelligent computing techniques (ICICT), IEEE, pp. 471-475, 2014.
[3] Horn, P., "Autonomic Computing: IBM's Perspective on the state of information technology", IBM, 2001.
[4] IBM White paper, "An architectural blueprint for autonomic computing", 2003.
[5] Sterritt, R., Parashar, M., Tianfield, H., Unland, R., "A concise introduction to autonomic computing", Advanced engineering informatics science, pp. 181-187, 2005, ELSEVIER.
[6] Nami, M.R., Sharifi, M., "Autonomic computing: A new approach", In proceedings of the first Asia international conference on modeling and simulation (AMS'07), IEEE, 2007.
[7] Salehie, M., Tahvildari, L., "Self-Adaptive software: Landscape and Research Challenges", ACM Tranactions on Autonomous and Adaptive systems, Vol. V, No.N, March, 2009.
[8] Dobson, S., Sterritt, K., Nixon, P., Hinchey, M., "Fulfilling the vision of Autonomic Computing", IEEE Computer Society, 2010.
[9] Parashar, M., Hariri, S., "Autonomic computing: An overview", Springer-Verlag Berlin Heidelberg, pp. 247-259, 2005.
[10] Huebscher, M.C., McCann, J. A., "A survey of Autonomic computing-degrees, models and applications," ACM Journal, pp.1-31, 2008.
[11] Li, X., Kang, H., Harrington, P., Thomas, J., "Autonomic and trusted computing paradigms," Springer-Verlag Berlin Heidelberg, pp.143-152, 2006.
[12] Muller, H. A., Kienle, H.M and Stege, U., "Autonomic computing now you see it, now you don't design and evolution of autonomic software systems,"Springer- Verlag Berlin Heidelberg, pp.32-54, 2009.
[13] Kephart,J.O., Chess.D., "The vision of autonomic computing" , IEEE Computer society, pp.41-50,2003
[14] Sterritt, R., "Apoptoticcomputing: Programmed death by default for computer-based systems,"IEEE Computer society, 2011.
[15] Cybenko, G., Berk, V.N., Souza, I.G., Behre, C.,"Practical autonomic computing," In proceedings of the 30th annual international computer software and applications conference (COMPSAC'06), IEEE, 2006.