Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 10, Issue. 5, May 2021, pg.81 – 86

A Survey on Autonomic Computing: Inspired by the Human Body's Autonomic Nervous System

Radha R
Assistant Professor, Department of Computer Science and Engineering
Alliance University, Bangalore, India
radhaprasadnr@gmail.com
DOI: 10.47760/ijcsmc.2019.v08i12.001

Abstract— Autonomic Computing is a concept that stands for self-management features of distributed computing agents being adaptive to unexpected changes and hiding the complication to users and operators. Introduced by IBM in 2001, this inventiveness ultimately meant to develop computer systems proficient of self-management, to overcome the fast growing intricacy of computing systems management, and to lessen the barrier that complexity poses to added growth. The progress of networks and the Internet, which have presented high accessible and obtainable services, have made atmospheres more multifaceted. The growing intricacy, rate, and heterogenic in distributed computing systems have interested researchers to explore a novel idea to manage with the supervision of intricacy in IT business. Autonomic Computing Systems (ACSs) have been presented for this reason. This paper consists of features of Autonomic Computing Systems, their effects, architecture and challenges.

KEYWORDS: Autonomic Computing Systems, Self-Management Systems, Multi-agent System, Self-healing Systems.

I. INTRODUCTION

Autonomic computing technology is an emerging research topic to solve the software complexity crisis. It is inspired by the function of the human nervous system and is aimed at designing and building systems that are self-managing. Its thoughts is to make systems manage their own under the guidance of management strategies developed by IT managers, through the technologies managing technologies to hide the system complexity.

As a result, the system can realize **self-configuration, self-optimization, self-healing and self-protection.** According to the maturity autonomy, it can be divided into five levels: basic level, management level, predict level, adaptation level, and full autonomic level. In 2001, IBM first proposed the concept of autonomic computing. Then

IBM pointed that complex computing systems should have four properties to achieve self-management, that is, self-configuration, self-optimization, self-healing, and self-protection

To meet self-management, systems should be designed with components that contain an autonomic manager. IBM presented a reference model of autonomic control loop in 2003, which is known as MAPE-K.

- Fig shows an autonomic element (AE) which is comprised of managed resources and an autonomic manager (AM).The Control loop with sensors and effectors together with Monitor, Analysis, Plan, Execute, Knowledge components makes the autonomic element to be self- manage
- The managed resources can be software or hardware resource that is given autonomic behavior in accord with an autonomic manager.

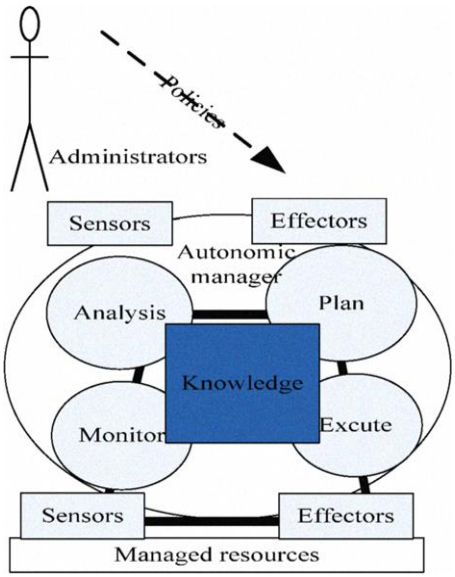

Fig: The reference model of autonomic clement

- Autonomic manager consists of monitoring component, analysis component, planning component, and execution component and knowledge repository.
    - The **monitoring component** provides the ability of self-awareness and detects the external environment.
    - The **analysis component** then carries out autonomic decision-making and decides the adaptive goal of system;
    - **Planning and execution components** achieve the adaptive function when the system state departures from the expected goal.
    - The operation of four components is supported by the knowledge repository

II. AUTONOMIC COMPUTING PROPERTIES AND APPLICATIONS

A. Self-configuration An autonomic computing system configures itself based on the needs of the platform or a strategy provided by IT professionals to adapt to changing environment.

B. Self-healing an autonomic computing system detects, diagnoses abnormalities and makes the appropriate repair measures. Self-healing component can detect system failure and in the case of uninterrupted service, it initiates repair measures automatically based on pre-specified strategy by IT professionals

C. *Self-protection* an autonomic computing system predicts, identifies and prevents the threat which is from anywhere. Self-protection component can detect hostile acts and take appropriate measures to ensure stability of the system.

D. Self-optimization An autonomic computing system automatically optimizes managed resources, and the optimization component adjusts itself to meet the needs of their end-user and business needs.

III. IMPORTANT CONCEPTS AND TOOLS FOR AUTONOMIC COMPUTING SYSTEM MODELS

Autonomic computing is to promote self-manage goal of various components to the whole system. Autonomic computing system involves service-oriented technology, Agent technology, adaptive control theory, machine learning, and optimization theory and so on. It also depends on the maturity of self-manage techniques about the various components resources (such as networks, databases, storage devices and middleware, etc.) The realization of autonomic computing system will result in a noteworthy improvement in system management efficacy.

A. Adaptive Control Theory Adaptive control is one of the most dynamic branches in the modem control theory. Adaptive control system based on dynamic object and the environment, by measuring the input / output information, obtains the dynamic characteristics of the managed object and systematic errors in time. According to the change and a certain design method, it makes decisions and modifies the controller parameters autonomously to adapt control signal to meet the changes of object and disturbance, to maintain optimal control performance of system.

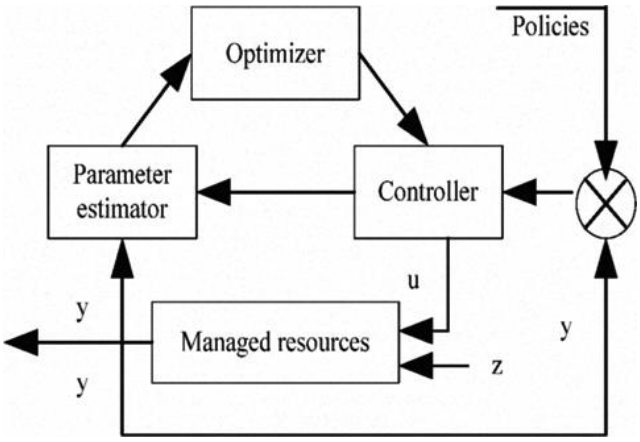

Fig: Autonomic clement based on adaptive control theory

B. Agent Technology Agent called intelligent agent has the properties of reactivity, autonomy and social. It can sense the environment to make a reaction (Reaction Agent) or to achieve the goal-oriented behavior (Deliberative Agent) by plan. Also, agent has been widely known as the key technology which supports the large-scale, open and distributed information systems to achieve dynamic service integration and teamwork. Web Service is a new branch of Web applications, which has the advantages of platform independence, interoperability and so on. Semantic Web technology makes computer understand the meaning of information and complete the intelligent agent function like the human brain. Therefore, based on Agent technology and policy-based management approach, and combined with Web service and Semantic Web technology, we can create a variety of autonomic computing systems. Liao introduced two main methods for building autonomic computing system. One is autonomic element based on Reaction Agent, the other based on Deliberative Agent.

C. *Utility Function* Utility is one of the most commonly used concepts in economics. In autonomic computing, utility function maps each possible state (system performance) of the entity (autonomic element) to a real number, for showing the value which corresponds to system performance (such as reaction time, delay, throughput, etc.). Autonomic computing system uses utility function to show management strategies. Based on the current system performance model, through the maximization of utility, autonomic computing system gets the desired system state and the values of corresponding adjustable system parameters. At last, autonomic computing system adjusts system parameters, making the system achieve the desired state in order to achieve optimization.

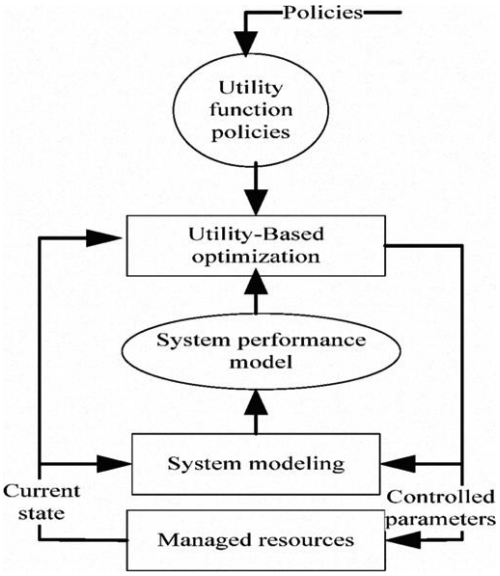

Fig: Autonomic element based on utility function

The autonomic element model of autonomic computing system based on the principle is shown in Fig. Especially in distributed autonomic computing system, utility function provides a favorable framework for self-optimization. In a dynamic complex environment, utility function can make autonomic element optimize computing resources continuously.

IV. CONCLUSION AND FUTURE DIRECTIONS

In a Dispersed Computing System, end users and multiple systems are connected in an exposed, see-through, and geographical large-scalable structure. Therefore, growth and supervision of these systems are dominant complications for IT Industry. IBM projected Autonomic Computing Systems as a key. ACSs cope themselves. Four chief features of ACSs contain self-configuration, self-optimization, self-protection, and self-healing.

A. Autonomic Computing System Architecture: Autonomic computing system itself must be the organizational structure, and the combination and interactive collaboration of autonomic element should be limited. However, current research that autonomic computing system should be what kind of architecture to support its elements effectively for self-management and its elements should be what kind of organizational structure to achieve self-configuration, self-healing, self-optimization , self-protection are not yet ripe. Therefore, the research of generic architecture and prototype for autonomic computing system will remain active.

B. Software Engineering Tools for Autonomic Computing System: The development methodology for autonomic computing system is the key point which achieves a blueprint for autonomic computing. We need more powerful software engineering tools for the analysis, design, development, testing and deployment on autonomic computing system. However, there are a small number of tools which have kinds of different problems. Bigus describes the learning environment ABLE for the development of autonomic computing system. However, ABLE is an agent generation environment based on component reuse. It lacks clear structure, agent deployment and operation

management environment. The features themselves also have no strong self-management support. Thus, with the depth research of autonomic computing, software engineering tools will be a research hot spot.

C. *Strategies/or Autonomic Computing:* The research of autonomic computing strategy in theory and in engineering is still in the initial stage. The main problems are as follows: Understanding for the role of strategy in autonomic computing is not deep enough; there is no powerful specification language, as well as strategic planning methods; there is no systematic strategy implement engineering approach; the strategy position on the life cycle of software systems has not been clear. Therefore, some scholars have presented that strategies for autonomic computing will also be focused on.

REFERENCES

[1]. Tesauro, G., and et al., A Multi-agent systems approach to autonomic computing. IBM Press, (2004) 464-471.
[2]. Ganek,A., IBM Initiatives in autonomic computing and policy. http://www-03.ibm.com/autonomic/letter.shtml.
[3]. DOBSON, S. AND NIXON, P. 2004. More principled design of pervasive computing systems. In Human Computer Interaction and Interactive Systems, R. Bastide and J. Roth, Eds. Lecture Notes in Computer Science, vol. 3425. Springer Verlag.
[4]. Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. 2000. Energy-efficient communication protocol for wireless microsensor networks. In HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8. IEEE Computer Society, Washington, DC, USA, 8020.
[5]. Dashofy, E. M., van der Hoek, A., and Taylor, R. 2002a. An infrastructure for the rapid development of XML-based architecture description languages. In Proceedings Of the 24th International Conference on Software Engineering (ICSE), Orlando, Florida.
[6]. Cobleigh, J. M., Osterweil, L. J., Wise, A., and Lerner, B. S. 2002. Containment units: a hierarchically composable architecture for adaptive systems. SIGSOFT Softw. Eng. Notes 27, 6, 159–165.
[7]. Ganek, A. G. and Corbi, T. A.:The dawning of the autonomic computing era. IBM System Journal, Vol. 42, (2003) 5-18.
[8]. Milojicic, D. S., and et al., Peer-to-Peer Computing. In: Proceedings of the Second International Conference on Peer-to-Peer Computing, (2002) 1-51.

