A Study on Architecture of Autonomic Computing-Self Managed Systems

Payal Mittal
Department of CSE
ASET, Amity University
Sector-125. Noida

Abhishek Singhal
Department of CSE
ASET, Amity University
Sector-125. Noida

Abhay Bansal
Department of CSE
ASET, Amity University
Sector-125. Noida

ABSTRACT

As in a complex growing mesh technologies field, autonomic computing is an auspicious new approach for building large scale distributed systems without assistance of any human interaction. This paradigm provides an environment, which has the potential to manage itself and adapt to the changes. The main objective of autonomic environment is to render the system administrator free by achieving self management properties at a higher level. The main characteristics of autonomic systems, which are to be achieved, are Self-healing, Self-optimizing, Self-protecting and Self-configuring. This paper describes the architecture of stable autonomic systems.

Keywords

Autonomic computing, architecture, managed elements, autonomic element.

1. INTRODUCTION

Autonomic Computing conceals the complexity of design and management in system equipments. An autonomic system is to maintain the authenticity of system in various environment by itself autonomic capabilities [1]. There are two perceptions in autonomic computing:"Vision in Administration" in which computing systems that can adapt themselves in given high-level goals from administrator and "Vision in Software Engineering" due to software intrusiveness of I/T-based systems, in which autonomic functionality is consistent and given I/T professionals concentrate on a new and better usage. These self-managed autonomic systems have the capability to guard them against any sort of malicious attack, correspond with use of open fundamentals and foretell user actions [2].

1.1 Background

Over the past era, the expeditious surge in computer technology helped to produce more strained hardware and software applications. As a result, long term feasibility and sustainability are usually ignored which, results in 'ball-of-mud' applications where peripherals have been added in a continued manner without paying any sort of attention to the resulting complex system integrity [3].The increase in the number of IT professionals is directly proportional to optimizing level of system complexity [4].However, the rampant growth of information systems will involves more than expected cost of computer system components so there is a lot of resource wastage. Finally, an Autonomic Computing is a conceivable approach to perform optimization of resources. The goal of autonomic systems is to build self managed systems that prevents and recover from failures.

The motivation of this research paper is that we have been all savouring a stirring journey as the information technology industry has proceeded faster than any other in history because of the lower power expenditure, superscalar processor implementation that led to epidemic growth in industry. So, by embedding autonomic principles into existing system architecture, we can move one step further for achieving success. These principles established a sight of self management for the servers and protecting them against nasty or malignant programs that somehow entered onto the system.

The outline of this research paper comprises of 4 sections. As we have already seen, section 1 gives background knowledge and introduction about autonomic computing. Section 2 proposed the real world examples that show self managed behaviour. The main domain of this research paper i.e., the architecture of autonomic computing is discussed in section 3 in which autonomic management, role of autonomic manager, functionality of autonomic control loop, goals and architecture of autonomic element are taken into consideration. Section 4 describes challenges faced by IT experts to deal with autonomic paradigm. Then, all these sections followed by conclusion and references.

2. PARADIGM OF SELF MANAGED SYSTEMS

Self-Managed Autonomic systems have the intent to shroud the intrinsic complexity to users and IT controllers [14].Here are some real world examples of devices that are showing autonomic behaviours which are as follow:

a. Recently on 6th of February 2014, LG launches its G Flex model which claims to be a smart phone with superhero powers. This device has self healing abilities which got filmed and seemed faster in warm conditions.

b. Either way you need an agent that knows how to work within shared infrastructure and is capable of self registering into the management server. This agent pulls system information and shared repository, then connects with its management server, which self install all its configuration policies back to the agent so it can self configure. This process is entirely automated for the first time when the agent runs it.

3. ARCHITECTURE OF AUTONOMIC COMPUTING

As we know, all software architectures are implemented to perform constraints to be persisted all along the enactment of system [15]. Architectural Models fortifies all the work plans, which consist of fully described execution scenarios [16].In this paper, we recommended a generic architecture to autonomic approach in order to mitigate administrator workload. We substantiate the potency of our software engineering adding self managing capabilities to existing system [16]. The unconventional architecture of autonomic computing contains two entities-Autonomic Element and Autonomic Manager. This architecture consists an adaptable and malleable infrastructure for all basic components of any self managed system that mitigate IT professionals from convolution associated with depiction, expansion and organizing autonomous systems. It contains an autonomic control loop that domineer the flow of work done between sub constituents of autonomic elements.

3.1 Autonomic Management [15]

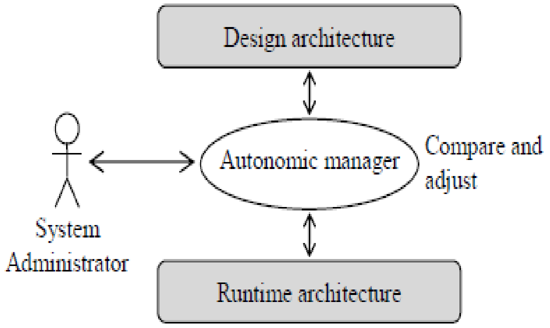

Fig. 1 Autonomic Management [15]

The design architecture of autonomic management system is to give advice to the autonomic manager about sustaining precise software architecture. This architecture enables autonomic manager's work by delaying implementation decisions by the management of variation points. This middle manager entity must have to assure the runtime system integrate to the reference architecture.

Autonomic management provides four characteristics of self managed systems i.e., Self-healing, Self-optimizing, Self-protecting and Self-configuring. These characteristics propose additional following self capabilities [7]:

a. Self-aware- The capabilities of self managed system are apprehensions and knowing of what happens inside and outside both at present through prophecy.

b. Self-adjusting- The abilities are modifications to any situation which controls internal system specification with respect to matching needs.

c. Self-situating- The potentials are independent to self management abilities.

Furthermore, all above self capabilities provide system with amplified characteristics: Anticipation and Openness.

3.2 Role of Autonomic Manager

The basic objective of autonomic managers is to preserve correct software architecture. They have full authority to select and instantiate constituents of architecture and then attach them together. They can delay design conclusions by the use of variation points [15]. As shown in Fig. 2, an autonomic manager has the functionality for controlling the web services adequately. Individual action plan can protect, configure or correct the web service by interacting with its external environment [12].

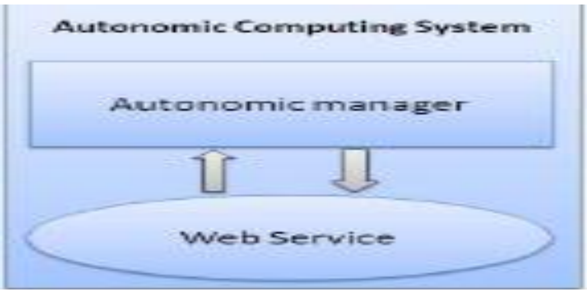

Fig. 2 Autonomic Managers as external layers [12]

3.3 Autonomic Control Loop

The autonomic control loop as shown in Fig. 3, accomplishes four event driven tasks which are as follow:

a. Collect all the application requirements from the environment sensors and in favour of user context knowledge base.

b. After collecting all relevant context, analyze process is performed from inference, economic models, rules and policies and by applying artificial intelligence game theories.

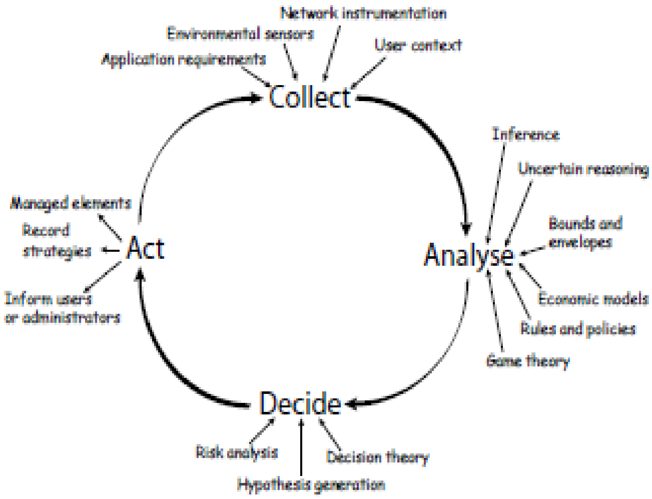

Fig. 3 The autonomic control loop [6]

c. Then analyzed data reached in deciding phase where all risk analysis measures are taken and hypothesis generation are taking into consideration

d. Finally actions are being performed through informing users or administrators by recording strategies or by managed elements.

3.4 Goals of Autonomic Element

As we have known now that autonomic element is considered the heart of autonomic computing architecture. So, there are some objectives of autonomic element which has to be implemented as follows [8]:

a. Programmers should perform computation skills as usual with minimum inhibitions.

b. Autonomic element should be trivial and should not use system resources.

c. It should employ all open benchmarks and basic analogies to develop autonomic element.

3.5 Architecture of Autonomic Element

It comprises managed resources and distributes services to humans or other autonomic elements. It contains sensors, effectors and five component analysis and planning engine which are as follow [10, 11]:

a. Managed Elements- This administrable resource is controlled through a set of incorporated, consists by its effectors and sensors.

b. Effectors- It is an interface which empowers an autonomic manager to manage managed elements through touch points using solicited response to enable call back functions. They also changed the configuration of managed element or autonomic element itself in some unusual way.

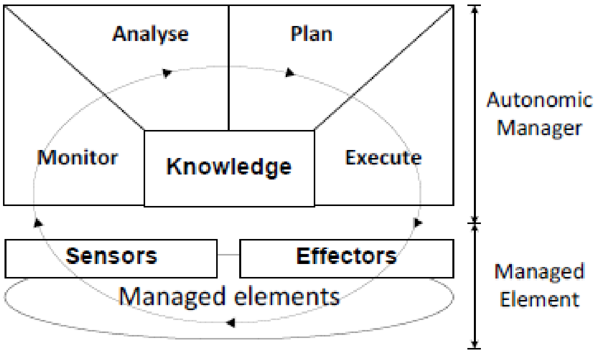

Fig. 4 Autonomic elements [17]

c. Sensors- It enables an autonomic manager to fetch information from managed element through touch points by sending notifications for event driven data retrieval. Sensors collect data about stating transitions of managed elements or autonomic element itself.

d. Monitor- It combines the sensors, trickle the data collected from them and then hides the refined data in knowledge base. It aggregate report details from sensors joined to autonomic manager.

e. Analyzer- It regulates if the autonomic manager can acknowledge by the system policies and also differentiates the collected data against desired sensor values.

f. Execute- It enables the crucifixion of action plans mounted by planner using effectors.

g. Plan- It provides the structure to organize the actions needed to achieve objectives and arrange strategies to correct the progression by planning engine.

h. Touch points- It is one of the main key attribute of self managed systems that constitutes a certain level of indirection. They can also be implementing to monitor some criteria of execution platform as all protocols used to do so. Through it, autonomic managers control other autonomic elements as well.

i. Shared Knowledge- The autonomic manager implements over matrices, performance measured data in its control loop.

All these different modules share a same knowledge through resource details, change exploit policies based on environment plans. There exist self managed architectures for sensors and handheld devices [5]. Different authors discovered different methods to look out multiple characteristics of an autonomic architecture. Amongst of them, perpetuated criteria's are the cost, speed and performance of autonomic response [13].

4. CHALLENGES

Towards the growing complexity in automation technology, an assortment of new hardware and software emerges continuously anticipating in IT sector. There are some open problems pervading for designing autonomic system architecture [19]: We need to determine which one of the techniques is best fitted for which processes that can be used to rampart architecture, testing and flourish autonomic systems. It is virtually in accessible to introduce guiding and activating real components into non-autonomous systems because in some autonomous systems, coupling between components are high. Interpretability and implementing global guidelines for security and configuration are all depend on existence of open standards in which we are lacking. So lack of open standards is a great challenge for building autonomic systems in IT industry. The biggest challenge to design of autonomic systems in information sector are the rules i.e., AC guidelines that govern the transitions to go from a current state to new one [18].

5. CONCLUSION

We have studied the architecture of Autonomous Managed Elements. Through this architecture of autonomous paradigm, there is a scope for making faster implementation of autonomic behavior into non-autonomous systems. There are both short and high-level goals which are provided by autonomic computing such as good consumption of information resources, huge self-managed regulation of IT internal workstation, etc. As a result of systems become autonomous by nature, now machines being capable of reinstatement of human affairs. Most of existing systems provides solutions for specific problems, but not unified refinement for full adaptation to a random environment. Although industry and academic endeavors are starting to grow rapidly in this research field, but there are still many challenges that need to be solved.

6. REFERENCES

[1] Kephart J.Chess D. The vision of autonomic computing, IEEE Computer Society, January 2003.

[2] R. Want, T. Pering, D. Tennenhouse, "Comparing autonomic and proactive computing," IBM SYSTEMS JOURNAL, VOL 42, NO 1, 2003.

[3] B. Foote, J. Yoder, "Big Ball of Mud", in Pattern Languages of Program Design 4, ed. N. Harrison, B. Foote, H. Rohnert, Addison-Wesley, 2000.

[4] P. Horn, "Autonomic Computing: IBM perspective on the state of information technology" , IBM T.J. Watson Labs, NY, 15th October 2001. Presented at Agenda 2001,Scottsdale,AR.(availablehttp://www.research.ibm.com/autonomic/), 2001.

[5] Inmaculada Ayala, self-StarMAS: A Multi-Agent System for the Self Management of AAL Applications/2012, Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.

[6] S. Dobson, S. Denazis, A. Fernandez, D. Gaiti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, "A survey of autonomic communications", ACM Transactions on Autonomous and Adaptive Systems, 1(2):223-259,December 2006.

[7] Carlos C. Insaurralde, Institute of sensors, signals and systems, Heriot-Watt University, Edinburgh: Autonomic Computing Management for Unmanned Aerial Vehicles/IEEE 2013

[8] Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (Ease'07) 0-7695-2809-0/07

[9] IBM Corporation "An architectural blueprint for autonomic computing", white paper, June 2006, 4 Edition.

[10] Miller, Brent, “The Autonomic Edge: Keeping in Touch wid Touch points. http://www-128.ibm.com /developer works/ autonomic/library/acedge5” August 2005.

[11] Steve R. White, James E. Hanson, Ian Whalley, David M. Chess, and Jeffery o. Kephart, “An Architectural Approach to Autonomic Computing”, In Proceedings of the International Conference of Autonomic Computing(ICAC04), pp. 2-9, 2004.

[12] W. Chainbi, H. Mezni, K. Ghedira: “An Autonomic Computing Architecture for self-* Web services. Proceedings of the First International Workshop on Agent- Based Social Simulation and Autonomic Systems (at Autonomics’09), Limassol, Cyprus, September 9-11-2009, Springer.

[13] T. De Wolf and T. Holvoet. “Evaluation and comparison of decentralized autonomic computing systems” IN Department of computer and science.

[14] Mahdi Ben Alaya, Thierry Monteil, “Frameself: A generic context-aware autonomic framework for self-management of distributed systems”, IEEE 21st International WETICE 2012.

[15] Etienne Gandrille, Catherine Hamon, Philippe Lalanda, “ Design and Runtime Architectures to Support Autonomic Management” 7th IEEE International Symposium on MESOCA 2013.

[16] Tomasz Haupt, “Towards Mediation-Based Self-Healing of Data Driven Business Processes” SEAMS 2012.

[17] E. Vassev, M. Hinchey, “Knowledge Representation and Awareness in Autonomic Service-Component Ensembles-State of the Art”, in proc. Int. Symp. On O/C/S-Oriented RT Dist. Comp. Workshops, Newport Beach, CA, USA 2011.

[18] Kephart, Jeffrey O. & Walsh, William E. “An Artificial Intelligence Perspective on Autonomic Computing Policies”, in proc. 5th IEEE Int. Workshop on Policies for Distributed Systems and Networks, Yorktown Heights, NY, USA, 2004.

[19] Mazeiar Salehie, Ladan Tahvildari, “Autonomic Computing : Emerging Trends and open Problems” DEAS’05 St Louis, Missouri USA Copyright 2005