ISSN 2 2 7 7 - 8 7 2 1

Electronic International Interdisciplinary Research Journal (EIIRJ)
Bi-monthly Reviewed Journal May/June 2013

AUTONOMIC COMPUTING: ARCHITECTURE, APPLICATION AND CHALLENGES

Abhang Swati Ashok

Dept. of IT,

SRES' COE Kopargaon

Abstract:-

Advances in computing and communication technologies and software tools have resulted in an explosive growth in networked-applications and information services that cover all aspects of our life. These services and applications are inherently complex, dynamic and heterogeneous. Similarly, the underlying information infrastructures such as the Internet are complex, heterogeneous and dynamic. This combination exacerbates complexities related to application development, configuration and management, and makes current computing paradigms brittle and inefficient. As a result, applications, programming environments and information infrastructures are rapidly becoming unmanageable, insecure and inefficient when handling runtime changes. This has led researchers to consider alternative programming paradigms and management techniques that are based on strategies used by biological systems to deal with complexity, dynamism, heterogeneity and uncertainty. Autonomic computing is inspired by the human autonomic nervous system that handles complexity and uncertainties, and aims at realizing computing systems and applications capable of managing themselves with minimum human intervention.

Keywords: *Autonomic computing, Human Nervous System, Autonomic component, Autonomic Computing Paradigm*

Introduction

The Autonomic Computing Paradigm has been inspired by the human autonomic nervous system. Its overarching goal is to realize computer and software systems and applications that can manage themselves in accordance with high-level guidance from humans. Meeting the grand challenges of autonomic computing requires scientific and technological advances in a wide variety of fields, as well as new programming paradigm and software and system architectures that support the effective integration of the constituent technologies.

This paper presents an introduction to autonomic computing, its challenges, and applications. In this paper, we first give an overview of the Autonomic Computing, architecture of the nervous system. We then outline the key challenges of autonomic computing and present an overview of existing autonomic computing systems and applications and - Challenges of Autonomic Computing.

Vision of Autonomic Computing

Autonomic computing can be seen as a holistic vision that enables a computing system to “deliver much more automation than the sum of its individually self-managed parts”. A system is considered a collection of computing resources working together to perform a specific set of functions. Eight key features characterize any AC system, cf. [IBM Corporation, 2001]:

- An AC system possesses system identity, i.e., it has knowledge of its components, current status, functions, and interactions with the environment.
- An AC system has the ability of self-configuration and reconfiguration, i.e., it can automatically perform dynamic adjustments to itself in varying and unpredictable environments.
- An AC system performs constant self-optimization, i.e., it monitors its constituent parts and adapts its behavior to achieve predetermined system goals.
- An AC system is self-healing, i.e., it is able to discover the causes of failures and then finds alternative ways of using resources or reconfiguring the system to keep functioning smoothly.

- An AC system is capable of self-protection, i.e., it detects, identifies, and protects itself against various types of attacks to maintain overall system security and integrity.
- An AC system uses self-adaptation to finds ways to best interact with neighboring systems, i.e., it can describe itself to other systems and discover those systems in the environment.
- An AC system is a non-proprietary open solution based on standards that provide the basis for interoperability across system boundaries.
- An AC system has hidden complexity, i.e., it automates IT infrastructure tasks and relieves users of administrative tasks.

Autonomic Components and Systems

This has led researchers to consider alternative design paradigms and management techniques that are based on strategies used by biological systems to deal with complexity, dynamism, heterogeneity and uncertainty – a vision that has been referred to as autonomic computing. Autonomic computing is inspired by the human autonomic nervous system and aims at realizing computing systems and applications that are capable of managing themselves with minimum human intervention. There have been several efforts to characterize the main features that make a computing system or an application autonomic. However, most of these techniques agree that an autonomic system must at least support the following four features:

1. **Self-Protecting**: Be able to detect attacks and protect its resources from both internal and external attacks.

2. **Self-Optimizing**: Be able to detect sub-optimal behaviors and intelligently perform self-optimization functions.

3. **Self-Healing**: Be able to detect hardware and/or software failures and should have the ability to reconfigure it to continue its operations in spite of failures.

4. **Self-Configuring**: Be able to dynamically change the configuration of its resources in order to maintain overall system and application requirements.

Large scale autonomic computing systems can be dynamically composed from smaller Autonomic Components (AComs) where each component supports in a seamless manner any combination of the four properties mentioned above. That means, each ACom can be dynamically and automatically configured, seamlessly tolerate any component failure, automatically detect component attacks and protect against them, and automatically change its configuration parameters to improve performance once it deteriorates beyond certain performance threshold. Once such autonomic components become available, we can dynamically build autonomic computing systems (see Figure 2) to meet any static requirements and runtime changes.

Autonomic Nervous System

The interaction between human body and its external environment is controlled by the nervous system, which consists of ”Central Nervous System” (CNS) and ”Peripheral Nervous System” (PNS). The central nervous system including the brain and spinal cord, is a exchange center of signals from external and internal environment in the human body. The peripheral nervous system connects all the target organs with central nervous system, acts and reacts in response to the external stimuli. The PNS can be further divided into ”Autonomic Nervous System” (ANS) and ”Sensory-Somatic Nervous System” . The Sensory-somatic nervous system controls the

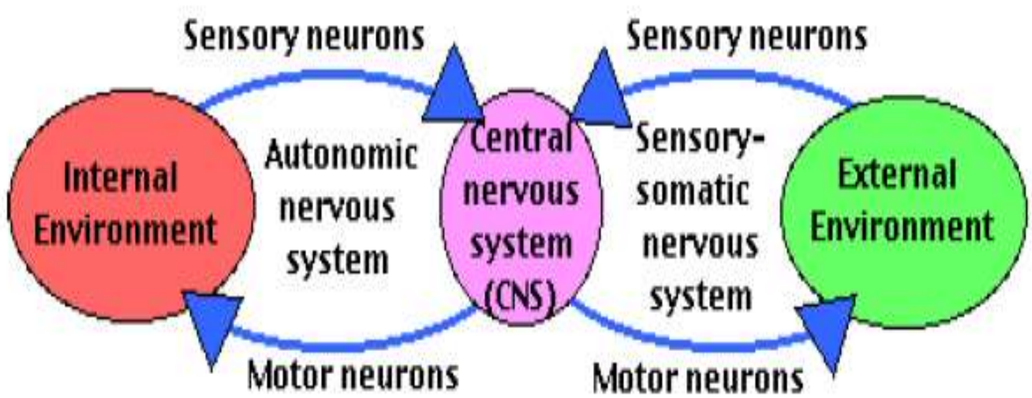

Fig.1. The Nervous System

ISSN 2 2 7 7 - 8 7 2 1

Electronic International Interdisciplinary Research Journal (EIIRJ)
Bi-monthly Reviewed Journal May/June 2013

muscular system and external sensory receptors (i.e, skin), while as the ANS regulates the critical physiological parameters in the internal environment. Two different types of neurons—”sensory neurons” and ”motor neurons”—run in ANS for the purpose of communication between internal environment and CNS. The sensory neurons carry signals from the target organs to CNS, while as the motor neurons carry commands from CNS to target organs and take actions.(see figure 1).

Autonomic nervous system consists of two subparts: “sympathetic nervous system” (sympathetic) and “parasympathetic nervous systems” (parasympathetic). In “The Autonomic Nervous System” it is stated that sympathetic involves “arousal” and “fight or flight reaction”, and parasympathetic controls “relaxation, recuperation and digestion” . Their functions are quite reciprocal. Sympathetic comes into operation under the situation of “any stimulus over an individual’s threshold”. This kind of stimulus could be feelings, noise, light. Sympathetic’ function produces a quickening of the heartbeat, sends more blood into muscles or raises blood pressure. Unlike sympathetic, parasympathetic is activated” after the stimulus has been responded to” and influences the human body in a opposite way. It will make the body re-balance, for example, the raised blood pressure will be lowered, and the increased heartbeat will be slowed.

Through the counteraction of sympathetic and parasympathetic some critical physiological parameters like blood pressure, heartbeat, and body temperature are regulated autonomously. For example, a new student was invited to visit his professor. After meeting with the professor his heartbeat increased because of nervousness. He maybe didn’t notice that, but this process was the result of sympathetic. A few minutes later, after a normal conversation with the professor, his heartbeat returned to the normal level because everything ran ok. This process was the result of parasympathetic.

Each critical physiological parameter is critical to the body life and has a physiological limitation. For example, if the body temperature is too high, that means, autonomic nervous system cannot regulate it alone, the person must have had an illness, and requires an external means to push it back to normal level, otherwise high body temperature would cause unexpected result.

General Architecture of Autonomic Computing

Autonomic computing is a grand challenge. The design of its architecture requires profound knowledge's and technologies in diverse fields. Different architectures may be devised from different viewpoints. From the viewpoint of users autonomic computing should be service-oriented and the providing of services should be distributed. From the viewpoint of developers autonomic computing should be a collection of components, which run alone and offer services to other components. Generally, autonomic computing should be decentralized, and composed of a collection of intelligent elements. These intelligent elements should work autonomously and connect with each other in a way they share services and resources from other elements. This kind of intelligent element is called "Autonomic Element" (AE) in the IBM architectural blueprint for autonomic computing, published in 2003.

In this section a general view of autonomic element and how they form an autonomic computing will be presented.

- **Autonomic Element**

As its name "autonomic" implies, autonomic element is an individual system which runs autonomously and cooperates with other autonomic elements as a whole. Autonomic element consists of one or more resources and a intelligent manager—"autonomic manager", which manages the resources (see figure 2).

The resources managed in autonomic element are called "managed element" . A managed element can be arranged into three categories: "resource layer", "composite resource layer" and "business solution layer" according its existing context. In resource layer a managed element is just like that in a non-autonomic environment, such as a CPU, or a printer, or a storage. In the second layer—composite resource layer—a managed element is a composition of resources for the improvement of performance and availability, such as a web server, or a database. In the third layer—business solution layer—a managed element is a system with a special business objective, for example, it may be an online shopping system or a cashier system. Each managed element has “sensors” and “effectors” which are used to provide internal state and adapt its behaviors in response to the environmental changes.

Autonomic nervous system regulates its critical philological parameters autonomously in our body, so does autonomic element. Autonomic element should also have the ability to maintain its managed elements. This is one task of autonomic manager. It’s another task is to communicate with other autonomic elements to establish a global autonomic computing.

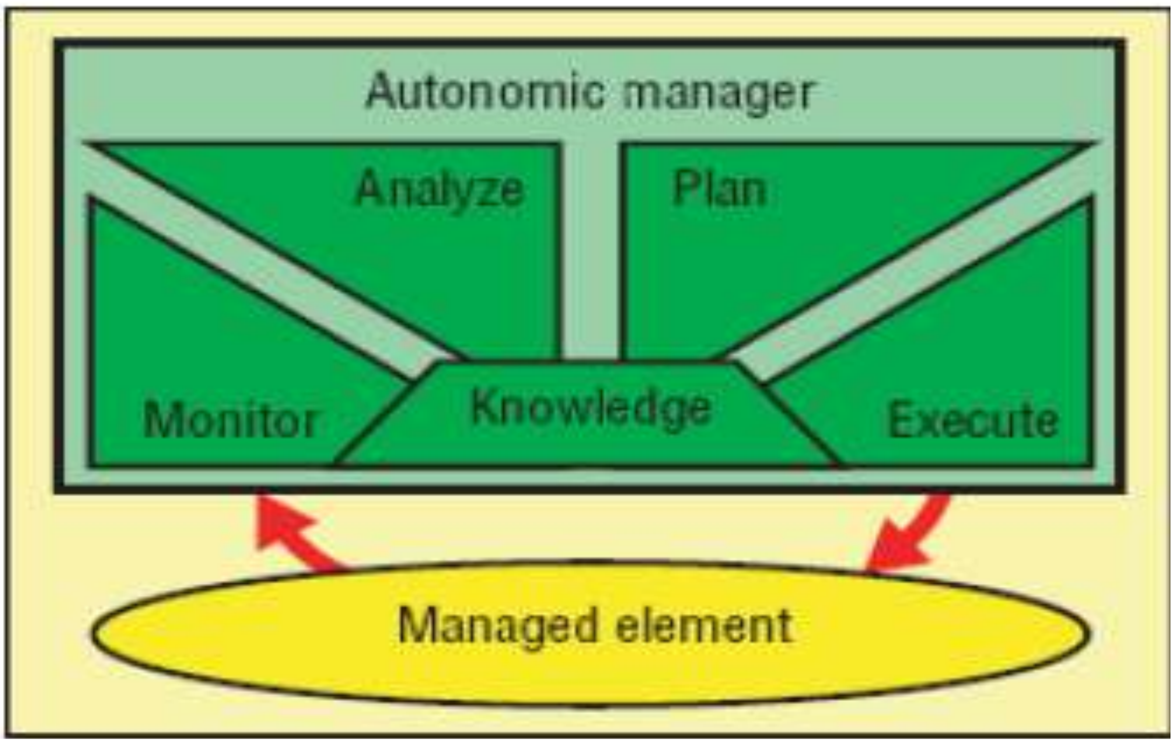

Fig. 2.Autonomic element architecture

In order for managed elements to be maintained autonomously, the knowledge about managed elements plays a significant role. At first, knowledge is gathered about managed elements using their sensors, after that it is correlated and analyzed. Finally, managed elements are affected by the previous results using their effectors. Based on the management of knowledge, a intelligent loop ”Monitor” — ”Analyze” — ”Plan” — ”Execute” (MAPE) are used in autonomic manager to maintain the managed elements.

Details about this intelligent loop are showed as the following:

- In the monitor phase, autonomic manager gathers the metrics and topological information about managed elements using some methods like filtering, collecting and reporting.
- In the analyze phase, autonomic manager consults the objective of this autonomic element, correlates the monitored information and predicts situation and trends.
- In the plan phase, autonomic manager plans the future action based on the analyze results and the objective of this autonomic element.
- In the execution phase, autonomic manager follows the previous plan and changes the behaviors of managed elements using their effectors.
- To provide services to other autonomic elements and adapt its internal state in response to the environmental changes (e.g., changes in other autonomic elements), autonomic manager has also its own sensors and effectors and uses them to communicate with other autonomic elements.

- **Autonomic Computing Architecture**

Autonomic element is a basic component of autonomic computing. It is intelligent because it can not only regulate internal managed elements, but also "facilitate collaborative interaction with other autonomic managers" . The collaboration between autonomic elements represents the structure of a target IT system. In the autonomic computing environment, this structure will be peer-to- peer or hierarchical. The resource sharing and parameters' negotiation within various autonomic elements are normally policy-based. This structured collaboration with various autonomic elements forms the architecture of autonomic computing. For example, we want to build a complex IT system including many previously defined business objectives. To implement this complex system with the concept of autonomic computing, at first we must develop various autonomic element with the previously defined objectives, and then we build the structure of their collaboration based on the policies that express the goal of system. As a result, an entire self-managing system is finally established.

Related Applications to Autonomic Computing:

Autonomic Computing brings new ideas and concepts in reducing complexity. There have been a number of research projects that use autonomic computing technologies in industry and academies. Some of them will be presented in this section.

- **Autonomic Computing Toolkit**

Autonomic computing toolkit presents some technologies and tools which are closely referred to the properties and general architecture of autonomic computing. This toolkit includes

- Autonomic manager engine: It demonstrates the MAPE control process in the architecture of autonomic computing.
- Log and Trace Analyzer: It demonstrates a partial implementation of control loops, including the part of monitoring and analyzing.
- Generic log adapter: It provides a translation from log files into a common event format—Common Base Event—in order for common logs to be acceptable in a autonomic computing environment.
- Resource Model Builder: this Eclipse plug-in demonstrates how to build a special resources into a autonomic computing environment using common resource model.

- **Dynamic Systems Initiative**

Dynamic Systems Initiative (DSI) is a Microsoft approach to reducing system complexity. As we seen in the general architecture of autonomic computing, the role of knowledge in system management is also emphasized in DSI. To benefit from the knowledge concept, DSI defines a common schema —System Definition Model— in order for other software's to be built into its operating environment. Once this model for software is created, it can be captured in system runtime, so that system is manageable autonomously.

- **OceanStore**

OceanStore is a global-scale persistent data storage system from the University of California at Berkeley. It uses a introspection layer to monitor and analyze network information in order to improve performance and fault management. Each data object within OceanStore has its own GUID and is stored in distributed data location.

- **Other Applications**

OptimalGrid provides a solution of the problem of large-scale application by implementing runtime management and dynamic rebalancing. Policy Management for Autonomic Computing implements a autonomic policy management. The Adaptive Enterprise provides a enterprise infrastructure used to manage enterprise knowledge in real time.

Challenges of Autonomic Computing

During the implementation of autonomic computing some related practices show that self-managing, adaptive computing systems can be realized, and have a great perspective. However, developing those autonomic systems are ”beyond the boundaries of traditional computer sciences” and requires a global cooperation of research in diverse fields. The architecture of autonomic computing simplified this work in a large scale, but caused also some new challenges. These challenges can be divided into three categories: standardization challenges, algorithms and methods challenges and management challenges.

- **standardization Challenges**

Autonomic computing is an open computing; it needs a common, standard model in multidimension.

- Representation of autonomic element needs standardization. An autonomic element may represent a special business or scientific objective, and its services should be shared by other autonomic elements. Thus an open, standard model for autonomic element is needed to design autonomic elements.

- ❖ Knowledge management needs a standardization. In the architecture of autonomic computing knowledge is shared in the implementation of managed loop—MAPE. In the analyze phase, autonomic computing needs to understand the meaning of monitored data autonomously and selects the useful information from them. This requires (a) a common log format for the understanding of monitored data; and (b) a common event correlation to determine useful expressions.
- ❖ Services sharing and parameters' negotiation between different autonomic elements need standardization. Different autonomic elements should operate in an unpredictable environment as a whole. They need to utilize their resources efficiently and to be aware of presence of other autonomic elements and external environment. To achieve it, services should be discovered autonomously and be shared within those autonomic elements. This requires a standardization of negotiation protocol, for example, service discovery protocol and service utilization protocol.
- ❖ System wide collaboration needs standardization. Various autonomic elements collaborate with each other and form a great autonomic computing system. The coordination between different autonomic elements is usually policy-based. These policies should (a) exactly express the goal of the complex system; and (b) be understandable by underlying autonomic elements.

Some projects attempt to solve this problem, but a standardization of policy in autonomic computing is still required.

➢ Algorithms and Methods Challenges

Autonomic computing needs a global cooperation in diverse fields. To develop autonomic computing, some algorithms and methods should be newly researched.

- Learning algorithm. Learning algorithm is closely tied to autonomic computing. From problem determination and autonomic remediation to system wide optimization, learning algorithms are used everywhere, but under new conditions, namely, critical services should not be disrupted. The exploration of learning algorithms is different from the traditional one. How exact a error can be allowed, how to improve the performance of learning process, and how to coordinate different learning process, all of that remain a research challenge.

- Process coordination methods. Autonomic computing system consists of a large scale of autonomic elements. Each of them represents a different objective (i.e, database, webserver) and expresses different optimization criteria. Within an autonomic element there run also many processes. How to coordinate such large number of processes to optimize, configure and reconfigure remains a research challenge.

- Attack detection methods. With autonomic computing exchange of information is accomplished in a autonomous way. Autonomic element need not only to understand the incoming information but also to detect active attacks and protect itself against those attacks.

- Management Challenges

The goal of autonomic computing is to reduce the tasks of nowadays administrators. To achieve it, there need new techniques to monitor and visualize what autonomic computing and its autonomic elements do. These techniques must be ”sufficiently expressive of preferences regarding cost vs. performance, security, risk and reliability”.

Conclusion

The increasing complexity makes computing systems brittle, uncertain and unsafe. The solution of that requires a new computing paradigm—autonomic computing— including self-managing characters. In this paper we presented an overview about autonomic computing including its properties, background, architecture and some related applications. At the end of this paper, an outlook about new challenges of autonomic computing is also presented. We think, autonomic computing is an inevitable way towards the solution of increasing complexity, and it needs to be developed in an evolutionary approach that is different from the traditional way.

References

Ganek, A.G., Corbi, T.A.: The dawning of the autonomic computing era, (available via http://www.research.ibm.com/journal/sj/421/ganek.pdf)

Kephart, J.O.: Research challenges of autonomic computing, (available via http://portal.acm.org/citation.cfm?id=1062464&dl=&coll=GUIDE&CFID=15151515&CFTOKEN=6184618)

IBM: Autonomic computing: The solution, (available via http://www.research.ibm.com/autonomic/overview/solution.html)

Kephart, J.O.: Research challenges of autonomic computing, (available via http://portal.acm.org/citation.cfm?id=1062464&dl=&coll=GUIDE&CFID=15151515&CFTOKEN=6184618)

P. Horn. Autonomic Computing: IBM's Perspective on the State of Information Technology. http://www.research.ibm.com/autonomic/, October 2001.

Autonomic Nervous System, http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/P/PNS.html#autonomic