International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 4 ǁ April. 2013 ǁ PP.55-61

Autonomic Computing

[1]Sangita, Ritu, [2]Sonal Beniwal

[1](M.tech N.S. (CSE),SES,BPSMV Khanpur Kalan)
(M.tech N.S. (CSE),SES,BPSMV Khanpur Kalan Under the guidance of
(Assistant Prof. in SES, BPSMV Sonipat)

ABSTRACT: *The increasing complexity of computer system results in systems which are prone to errors and disruptions creating major problems for the user. In order to reduce the total cost of ownership and to cope with the rapidly growing complexity of integrating and managing today's computer-based systems, autonomic computing was introduced by IBM in 2001. Autonomic computing is a computer environment that can detect and adjust its system automatically to manage itself without the assistance of any human interaction. This paper introduces autonomic computing, a methodology that focuses on the management of system complexity through the use of methods providing for self learning and self-healing operating systems. The main focus of the CAC in conference ACM 2013 is to be the premier international forum to present the latest research applications, and technologies to make cloud and autonomic computing systems and services easy to design, to deploy and to implement, while achieving the simultaneous goals to be self-manageable, self-regulating and scalable with little involvement of human or system administrators.*

Keyword:-*Autonomic computing, Cloud, Key element, Architecture, Life cycle, Special tracks 2013 conference*

I. INTRODUCTION

"Civilization advances by extending the number of important operations which we can perform without thinking about them." - Alfred North Whitehead

This quote made by the preeminent mathematician Alfred Whitehead holds both the lock and the key to the next era of computing. It implies a threshold moment surpassed only after humans have been able to automate increasingly complex tasks in order to achieve forward momentum. It's the complexity of these systems and the way they work together that is creating shortage of skilled I/T workers to manage all of the systems. Ironically, complexity itself has become part of the problem. But as Mr. Whitehead so eloquently put it nearly a century ago, the solution may lie in automation, or creating a new capacity where important computing operations can run without the need for human intervention. On October 15th, 2001 Paul Horn, senior vice president of IBM Research addressed the Agenda conference, an annual meeting of the preeminent technological minds, held in Arizona. In his speech, and in a document he distributed there, he suggested a solution: build computer systems that regulate themselves much in the same way our nervous systems regulates and protects our bodies. This new model of computing is called autonomic computing.

II. WHAT IS AUTONOMIC SYSTEMS?

"Autonomic Computing" is a new vision of computing initiated by IBM. Broadly speaking, autonomic computing refers to computing infrastructure that adapts [automatically] to meet the demands of the applications that are running in it. Yet, to understand the field of autonomic.from that experience and integrate the results of that learning into their next series of actions or decisions. Autonomic computing systems, in theory at computing, let us consider an interesting parallel to computing systems: our human body. Consider the following everyday events:

- -When you exercise, you start sweating, thus creating evaporation, which then cools your body.
- -When you cut your finger, a clot forms to stop the bleeding.
- -When you walk upstairs you breathe more deeply and your heart beat increases.
- -When you breathe in too much pollen you sneeze

All these body functions have one thing in common: they are performed automatically without you thinking about them. The involuntary actions and reactions of your nervous systems allow your body to cope with changes and threats. Underlying and controlling these actions and reactions are an autonomic nervous

system. The autonomic nervous system maintains the internal environment of our bodies in a steady state by governing involuntary body functions such as respiration and heart rate.
The field of autonomic computing rests upon the parallel of the human body, arguing that computer

III. WHY DO WE NEED AUTONOMIC SYSTEMS?

The first driver of autonomic computing is increasing technological complexity. Today, different vendors adopt different standards for their systems. This presents challenges and makes it difficult for IT personnel to manage an environment containing diverse and heterogeneous infrastructures of typical IT budget are spent on preventing or recovering from crashes.

The second driver of autonomic computing is the increased size of infrastructures. The accelerating pace by which devices are being added to many IT networks (especially the Internet) further complicates the already sophisticated technological environment.
The third driver of autonomic computing is the ballooning costs of systems. Increasing complexity of integrated systems makes the job of maintaining and fixing systems more challenging than ever. In fact, it has been reported that one third to one half knowledge to manage complex IT infrastructures are expensive and, even in today's depressed economy, remain in short supply. Depending on the type of system, labor costs could surpass

Infrastructure costs by factors of 3 to 18. So, the strategy of relying on human intervention to manage IT infrastructure might not be favorable.

The fourth driver of autonomic computing is the shortage of skilled labor. Workers who have the from some unforeseen problems, then they could also learn from that experience and integrate the results of that learning into their next series of actions or decisions. Autonomic computing systems, in theory at least, possess all of these abilities. These are intelligent systems that are able to operate, manage and improve their own operations with minimum or, better yet, with no human intervention. And most importantly, all these functions would be performed without users knowing it and completely transparently to users.

Similarly, for autonomic computing to achieve its full capabilities, different system components have to play coordinated roles. Overtime, autonomic computing shifts the burden of managing systems from people to technologies, freeing IT personnel to focus on other high-value business activities rather than on infrastructures. The term autonomic is derived from human biology. The autonomic nervous system monitors our heartbeat, checks our blood sugar level and keeps our body temperature close to 98.6 °F, without any conscious effort on our part.

The main focus of the CAC in conference ACM 2013 is to be the premier international forum to present the latest research applications, and technologies to make cloud and autonomic computing systems and services easy to design, to deploy and to implement, while achieving the simultaneous goals to be self-manageable, self-regulating and scalable with little involvement of human or system administrators.

IV. KEY ELEMENT OF AUTONOMIC COMPUTING

The elements of autonomic computing can be summarized in 8 key points.

4.1Knows Itself

An autonomic computing system needs to "know itself" - its components must also possess a system identity. Since a "system" can exist at many levels, an autonomic system will need detailed knowledge of its components, current status, ultimate capacity, and all connections to other systems to govern itself.

4.2 Configure Itself

An autonomic computing system must configure and reconfigure itself under varying (and in the future, even unpredictable) conditions. System configuration or "setup" must occur automatically, as well as dynamic adjustments to that configuration to best handle changing environments.

4.3 Optimizes Itself

An autonomic computing system never settles for the status - it always looks for ways to optimize its workings. It will monitor its constituent parts and fine-tune workflow to achieve predetermined system goals.

4.4 Heals itself

An autonomic computing system - it must be able to recover from routine and extraordinary events that might cause some of its parts to malfunction. It must be able to discover problems or potential problems, then find an alternate way
of using resources or reconfiguring the system to keep functioning smoothly.

4.5 Protects Itself

An autonomic computing system must be an expert in self-protection. It must detect, identify and protect itself against various types of attacks to maintain overall system security and integrity.

4.6 Adapts Itself

An autonomic computing system must know its environment and the context surrounding its activity, and act accordingly. It will find and generate rules for how best to interact with neighboring systems.

4.7 Opens itself

An autonomic computing system cannot exist in a hermetic environment. While independent in its ability to manage itself, it must function in a heterogeneous world and implement open standards.

4.8 Hides Itself

An autonomic computing system will anticipate the optimized resources needed while keeping its complexity hidden. It must marshal I/T resources to shrink the gap between the business or personal goals of the user, and the I/T implementation necessary to achieve those goals -- without involving the user in that implementation.

V. AUTONOMIC COMPUTING V/S CURRENT COMPUTING

In an autonomic environment, system components —from hardware such as desktop computers and mainframes to software such as operating systems and business applications —are self-configuring, self-healing, self-optimizing and self- protecting. These self-managing attributes can be compared as given in the table.

Four aspects of self-management as they are now and would be with autonomic computing.

Concept	Current computing	Autonomic computing
Self-configuration	Corporate data centers have multiple vendors and platforms. Installing, configuring, and integrating systems is time consuming and error prone.	Automated configuration of components and systems follows high-level policies. Rest of system adjusts automatically and seamlessly.
Self-optimization	Systems have hundreds of manually set, nonlinear tuning parameters, and their number increases with each release.	Components and systems continually seek opportunities to improve their own performance and efficiency.
Self-healing	Problem determination in large, complex systems can take a team of programmers weeks.	System automatically detects, diagnoses, and repairs localized software and hardware problems.
Self-protection	Detection of and recovery from attacks and cascading failures is manual.	System automatically defends against malicious attacks or cascading failures. It uses early warning to anticipate and prevent systemwide failures.

Fig 1

VI. AUTONOMIC COMPUTING ARCHITECTURE

The autonomic computing architecture concepts provide a mechanism for discussing, comparing and contrasting the approaches different vendors use to deliver self-managing attributes in an autonomic computing system. The autonomic computing architecture starts from the premise that implementing self-managing attributes involves an intelligent control loop. This loop collects information from the system, makes decisions and then adjusts the system as necessary. An intelligent control loop can enable the system to do such things as:

- Self-configure, by installing software when it detects that software is missing
- Self-heal, by restarting a failed element
- Self-optimize, by adjusting the current workload when it observes an increase in capacity
- Self-protect, by taking resources offline if it detects an intrusion attempt.

6.1 Control Loops

A control loop can be provided by a resource provider, which embeds a loop in the runtime environment for a particular resource. In this case, the control loop is configured through the manageability interface provided for that resource (for example, a hard drive).In some cases, the control loop may be hard-wired or hard coded so it is not visible through the manageability interface. Autonomic systems will be interactive collections of *autonomic elements*.

Autonomic computing architecture is shown in the diagram below. This portion of the architecture details the functions that can be provided for the control loops. The architecture organizes the control loops into two major

elements —a managed element and an autonomic manager. A managed element is what the autonomic manager is controlling. An autonomic manager is a component that implements a control loop.

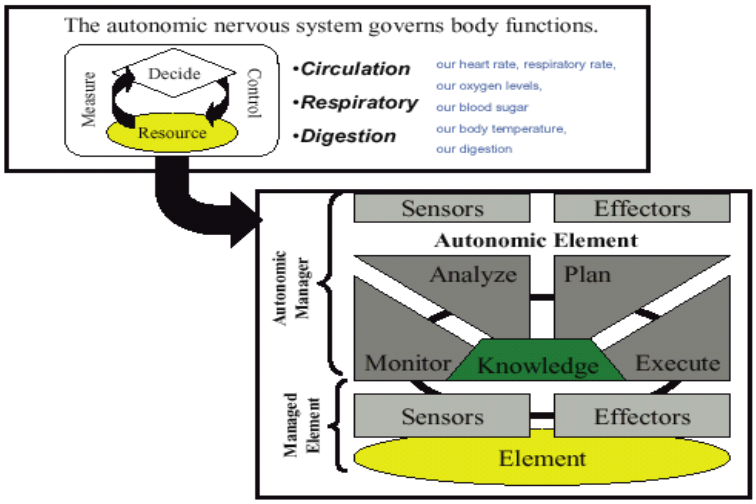

Fig 2

6.2 Managed Elements

The managed element is a controlled system component. The managed element will essentially be equivalent to what is found in ordinary non autonomic systems, although it can be adapted to enable the autonomic manager to monitor and control it. The managed element could be a hardware resource, such as storage, CPU, or a printer, or a software resource, such as a database, a directory service, or a large legacy system.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, you can either use a set of "get" operations to retrieve information about the current state, or a set of management events (unsolicited, asynchronous messages or notifications) that flow when the state of the element changes in a significant way.
- The effectors are mechanisms that change the state (configuration) 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.

6.3 Autonomic Manager

The managed element is a controlled system component. The managed element will essentially be equivalent to what is found in ordinary non-autonomic systems, although it can be adapted to enable the autonomic manager to monitor and control it. The managed element could be a hardware resource, such as storage, CPU, or a printer, or a software resource, such as a database, a directory service, or a large legacy system.The managed element is controlled through its sensors and effectors:

- The monitor part provides the mechanisms that collect, aggregate, filter, manage and report details (metrics and topologies) collected from an element.
- The analyze part provides the mechanisms to correlate and model complex situations (time-series forecasting and queuing models, for example). These mechanisms allow the autonomic manager to learn about the IT environment and help predict future situations.
- The plan part provides the mechanisms to structure the action needed to achieve goals and objectives. The planning mechanism uses policy information to guide its work.
- The execute part provides the mechanisms that control the execution of a plan with considerations for on-the-fly updates.

Knowledge Data used by the autonomic manager's four functions (monitor, analyze, plan and execute) are stored as shared knowledge. The shared knowledge includes data such as topology information, historical logs, metrics, symptoms and policies.

VII. EVOLUTION RATHER THAN REVOLUTION

Most existing systems cannot be redesigned and redeveloped from scratch to engineer autonomic capabilities into them. Rather, self-management capabilities have to be added gradually and incrementally. one component (i.e., architecture, subsystem, or service) at time With the proliferation of autonomic components, users will impose increasingly more demands with respect to functional and nonfunctional requirements for autonomicity. Thus, the process of equipping software systems with autonomic technology will be evolutionary rather than revolutionary.

VIII. GRAND CHALLENGES

A Grand Challenge is a problem that by virtue of its degree of difficulty and the importance of its solution, both from a technical and societal point of view, becomes a focus of interest to a specific scientific community. The difficulty in developing and implementing autonomic computing is daunting enough to constitute a Grand Challenge. At the heart of the matter is the need to bring together minds from multiple technical and scientific disciplines as well as differentiated businesses and institutions to share a sense of urgency and purpose.

8.1 Engineering Challenges

Virtually every following aspect of autonomic computing offers significant engineering .

8.1.1 Life cycle of the autonomic computing

An autonomic element's life cycle begins with its design and implementation; continues with test and verification; proceeds to installation, configuration, optimization, upgrading, monitoring, problem determination, and recovery; and culminates in uninstallation or replacement. Each of these stages has special issues and challenges.

8.1.2 Design, test, and verification

Programming an autonomic element will mean extending Web services or grid services with programming tools and techniques that aid in managing relationships with other autonomic elements. Because autonomic elements both consume and provide services, representing needs and preferences will be just as important as representing capabilities. Programmers will need tools that help them acquire and represent policies.

8.1.2 Installation and configuration

Installing and configuring autonomic elements will most likely a bootstrapping process that begins when the element registers itself in a directory service by publishing its capabilities and contact information. The element might also use the directory service to discover suppliers or brokers that may provide information or services it needs to complete its initial configuration.

8.1.3Monitoring and problem determination

Monitoring will be an essential feature of autonomic elements. Elements will continually monitor themselves to ensure that they are meeting their own objectives, and they will log this information to serve as the basis for adaptation, self-optimization, and reconfiguration.

Monitoring will be important in supporting problem determination and recovery when a fault is found or suspected. Applying monitoring, audit, and verification tests at all the needed points without burdening systems with excessive bandwidth or processing demands will be a challenge.

8.1.4 Relationship among Autonomic Elements

In its most dynamic and elaborate form, the service relationship among autonomic elements will also have a life cycle. Each stage of this life cycle engenders its own set of engineering challenges and standardization requirements.

8.1.5 Specification

An autonomic element must have associated with it a set of output services it can perform and a set of input services that it requires, expressed in a standard format so that other autonomic elements can understand it.

8.1.6 Location

An autonomic element must be able to locate input services that it needs; in turn, other elements that require its output services must be able to locate that element.

8.1.7 Negotiation

Once an element finds potential providers of an input service, it must negotiate with them to obtain that service.

8.1.8 Provision

Once two elements reach an agreement, they must provision their internal resources. Provision may be as simple as noting in an access list that a particular element can request service in the future.

8.1.9 System-wide issues

Other important engineering issues that arise at the system level include security, privacy, and trust, and the emergence of new types of services to serve the needs of other autonomic elements. Autonomic computing systems will be subject to all the security, privacy, and trust issues that traditional computing systems must now address. Autonomic elements and systems will need to both establish . just as human administrators do today, and they will need to do so in an understandable and fail-safe manner.

8.2 Scientific Challenges

The success of autonomic computing will hinge on the extent to which theorists can identify universal principles that span the multiple levels at which autonomic systems can exist—from systems to enterprises to economies. 8.2.1 Behavioural Abstraction & Models

Defining appropriate abstractions and models for understanding, controlling, and designing emergent behavior in autonomic systems is a challenge at the heart of autonomic computing. We need fundamental mathematical work aimed at understanding how the properties of self-configuration, self-optimization, selfmaintenance, and robustness arise from or depend on the behaviors, goals, and adaptation of individual autonomic elements; the pattern and type of interactions among them; and the external influences or demands on the system.

8.2.2 Robustness Theory

A related challenge is to develop a theory of robustness for autonomic systems, including definitions and analyses of robustness, diversity, redundancy, and optimality and their relationship to one another.

8.2.3 Negotiation Theory

A solid theoretical foundation for negotiation must consider two perspectives. From the perspective of individual elements, we must develop and analyze algorithms and negotiation protocols and determine what bidding or negotiation algorithms are most effective. From the perspective of the system as a whole, we must establish how overall system behavior depends on the mixture of negotiation algorithms that various autonomic elements use and establish the conditions under which multilateral —as opposed to bilateral—negotiations among elements are necessary or desirable.

IX. SPECIAL TRACKS IN 2013

The CAC conference main topics are grouped into four special tracks: Autonomic Cloud Computing; Autonomics for Extreme Scales; Autonomic Cybersecurity; and Autonomic Computing Foundation: Tools and Applications. Program vice-chairs in each of these research areas will coordinate the efforts regarding papers focused in these tracks.

a) Autonomic Cloud Computing: Vice Chair Rosa Badia
- Self-management cloud services
- Autonomic cloud applications and services
- Autonomic virtual cloud resources and services
- Cloud workload characterization and prediction
- Monitoring and analysis of behavior of cloud resources and services
- Theoretical frameworks for modeling and analysis autonomic computing systems and services

b) Autonomics for Extreme Scales: Vice Chair Gregor von Laszewski
- Large scale autonomic systems
- Self-optimizing and self-healing at petacomputing scale
- Self-managing middleware and tools for extreme scales
- Experiences in autonomic systems and applications at extreme scales (petacomputing)

c) Autonomic Cybersecurity: Vice Chair Sherif Abdelwahed
- Self-protection techniques of computing systems, networks and applications
- Metrics to evaluate and performance of self-protection algorithms
- Anomaly behavior analysis of autonomic systems and services
- Data mining, stochastic analysis and prediction of autonomic systems and applications
- Metrics to characterize and quantify the cybersecurity algorithms (confidentiality, Integrity, and Availability of autonomic systems
- Datasets and benchmarks to compare and evaluate different self-protection techniques

d) Autonomic Tools and Applications: Vice Chair Jim Dowling
- Benchmarks and tools to evaluate and compare different architectures to implement autonomic cloud systems.

X. FUTURE SCOPE

Realistically, 100 percent autonomic systems will be very difficult to build and will require significant exploration of new technologies and innovations. That's why researches view this as a Grand Challenge for the entire IT industry. People will need to make progress along two tracks- first, making individual system components autonomic and achieving autonomic behavior at the level of global enterprise IT systems. That second track may prove to be extremely challenging. Still another problem to solve: how to design an architecture for autonomic systems that provides consistent interfaces and points of control while allowing for a heterogeneous environment. We could go on, as the list of problems is actually quite long, but it is not so daunting as to render autonomic computing another dream of science fiction. In fact, we're beginning to make progress in key areas. Many established fields of scientific study will contribute to autonomic computing. What we've learned in artificial intelligence, control theory, complex adaptive systems and catastrophe theory, as well as some of the early work done in cybernetics, will give us a variety of approaches to explore. Current research projects at laboratories and universities include self-evolving systems that can monitor themselves and adjust to some changes, "cellular" chips capable of recovering from failures to keep long-term applications running, heterogeneous workload management that can balance and adjust workloads of many applications over various servers, and traditional control theory applied to the realm of computer science, to name just a few.

XI. CONCLUSION

Is it possible to meet the grand challenge of autonomic computing without magic and without fully solving the AI problem? It is possible, but it will take time and patience. Long before we solve many of the more challenging problems, less automated realizations of autonomic systems will be extremely valuable, and their value will increase substantially as autonomic computing technology improves and earns greater trust and acceptance. A vision this large requires that we pool expertise in many areas of computer science as well as in disciplines that lie far beyond computing traditional boundaries.

We must look to scientists studying nonlinear dynamics and complexity for new theories of emergent phenomena and robustness. We must look to economists and e-commerce researchers for ideas and technologies about negotiation and supply webs. We must look to psychologists and human factors researchers for new goal-definition and visualization paradigms and for ways to help humans build trust in autonomic systems. .

No company is going to be able to control all the parts of an autonomic system. It's going to have to be an open source system because there are so many parts. These parts will provide plenty of opportunity for competition.

REFERENCE

[1]. Autonomic Computing:IBM's Perspective on the State of Information Technology, http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf

[2]. 'Trends in technology', survey, Berkeley University of California, USA, March 2002

[3]. IEEE Computer Magazine, Jan 2003

[4]. S-Cube Network, "Self-Healing System".

[5]. E. Curry and P. Grace, "Flexible Self-Management Using the Model-View-Controller Pattern," IEEE Software, vol. 25, no. 3, pp. 84-90, May. 2008.

[6]. ACM conference 2013 www.autonomic-conference.org