RESEARCH PAPER | Computer Science | Volume : 5 | Issue : 4 | April 2015 | ISSN - 2249-555X

An Integrated Self-Managing Framework for Autonomic Computing Systems

KEYWORDS

Smt. Kalpana Dalwai

Assistant Professor, Dept. of Computer Science, Karnatak University's Karanatak Science College, Dharwad.

ABSTRACT *A computer system would satisfy the requirements of "autonomic computing", if the system can configure and reconfigure itself by knowing the operating environments, protect and heal itself from various failures or malfunctions. In order to know the environments and detect failure, an autonomic system needs the capability of acquiring the information through self-monitoring. Once the sequence of events leading to a series of disasters are figured out, it is required to predict and control the system management process through a number of automated learning and proactive actions.*

Introduction

The system makes decisions on its own, using high-level policies; it will constantly check and optimize its status and automatically adapt itself to changing conditions. An autonomic computing framework is composed of autonomic components interacting with each other. Autonomic components can be modeled in terms of two main control loops (local and global) with sensors (for self-monitoring), effectors (for self-adjustment), knowledge and planner/ adapter for exploiting policies based on self- and environment awareness. Driven by such vision, a variety of architectural frameworks based on "self-regulating" autonomic components has been recently proposed. A very similar trend has recently characterized significant research in the area of multi agent systems. However, most of these approaches are typically conceived with centralized or cluster-based server architectures in mind and mostly address the need of reducing management costs rather than the need of enabling complex software systems or providing innovative services. Some autonomic systems involve mobile agents interacting via loosely coupled communication mechanisms.

IBM has set forth eight conditions that define an autonomic system:

- The system must know itself in terms of what resources it has access to, what its capabilities and limitations are and how and why it is connected to other systems.
- The system must be able to automatically configure and reconfigure itself depending on the changing computing environment.
- The system must be able to optimize its performance to ensure the most efficient computing process.
- The system must be able to work around encountered problems by either repairing itself or routing functions away from the trouble.
- The system must detect, identify and protect itself against various types of attacks to maintain overall system security and integrity.
- The system must be able to adapt to its environment as it changes, interacting with neighboring systems and establishing communication protocols.
- The system must rely on open standards and cannot exist in a proprietary environment.
- The system must anticipate the demand on its resources while keeping transparent to users.

Problem in complexity

Forecasts suggest that the number of computing devices in use will grow at 38% per year and the average complexity of each device is increasing. Currently, this volume and complexity is managed by highly skilled humans, but the demand for skilled IT personnel is already outstripping supply, with labour costs exceeding equipment costs by a ratio of up to 18:1. Computing systems have brought great benefits of speed and automation but there is now an overwhelming economic need to automate their maintenance.

Design and maintain the complexity of interactions. They state the essence of autonomic computing is system self-management, freeing administrators from low-level task management while delivering better system behavior.

A general problem of modern distributed computing systems is that their complexity, and in particular the complexity of their management, is becoming a significant limiting factor in their further development. Large companies and institutions are employing large-scale computer networks for communication and computation. The distributed applications running on these computer networks are diverse and deal with many tasks, ranging from internal control processes to presenting web content to customer support.

This creates an enormous complexity in the overall computer network which is hard to control manually by human operators. Manual control is time-consuming, expensive, and error-prone. The manual effort needed to control a growing networked computer-system tends to increase very quickly.

Four functional areas of Autonomic Computing

Self-configuration: Automatic configuration of components.

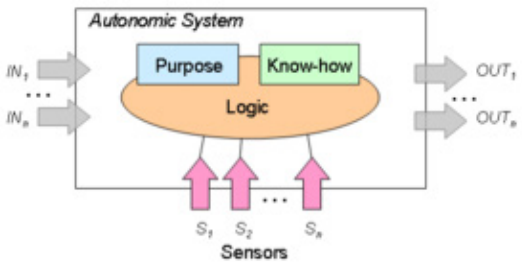
Self-healing: Automatic discovery and correction of faults.

Self-optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements.

Self-protection: Proactive identification and protection from arbitrary attacks.

Volume : 5 | Issue : 4 | April 2015 | ISSN - 2249-555X

Conceptual model



A fundamental building block of an autonomic system is the sensing capability (Sensors Si), which enables the sys tem to observe its external operational context. Inherent to an autonomic system is the knowledge of the Purpose (intention) and the Know-how to operate itself (e.g. bootstrapping, configuration knowledge, interpretation of sensory data, etc.) without external intervention. The actual operation of the autonomic system is dictated by the Logic, which is responsible for making the right decisions to serve its Purpose, and influence by the observation of the operational context (based on the sensor input).

This model highlights the fact that the operation of an autonomic system is purpose-driven. This includes its mission (e.g., the service it is supposed to offer), the policies (e.g., that define the basic behavior), and the "survival instinct". If seen as a control system this would be encoded as a feedback error function or in a heuristically assisted system as an algorithm combined with set of heuristics bounding its operational space.

Characteristics of Autonomic Computing

Automatic

This essentially means being able to self-control its internal functions and operations. As such, an autonomic system must be self-contained and able to start-up and operate without any manual intervention or external help. Again, the knowledge required to bootstrap the system (*Know-how*) must be inherent to the system.

Adaptive

An autonomic system must be able to change its operation (i.e., its configuration, state and functions). This will allow the system to cope with temporal and spatial changes in its operational context either long term (environment customization/optimization) or short term (exceptional conditions such as malicious attacks, faults, etc.).

Aware

An autonomic system must be able to monitor (sense) its operational context as well as its internal state in order to be able to assess if its current operation serves its purpose. Awareness will control adaptation of its operational behaviour in response to context or state changes.

Features

1. Introduces and examines the design of systems and applications that can self-manage with minimal human intervention.
2. Considers the scale, complexity, heterogeneity and dynamism in modern networks and tools to address them.
3. Outlines system, strategies and implementation scenarios to automate complex, tedious and routine activities.
4. Presents the infrastructures, software tools and middleware for enabling autonomic computing systems.
5. Real system implementations from research, industry and academia.

Security in an autonomic computing environment

System and network security are vital parts of any autonomic computing solution. The ability of a system to react consistently and correctly to situations ranging from benign but unusual events to outright attacks is key to the achievement of the goals of self-protection, self-healing, and self-optimization. Because they are often built around the interconnection of elements from different administrative domains, autonomic systems raise additional security challenges, including the establishment of a trustworthy system identity, automatically handling changes in system configuration and interconnections, and greatly increased configuration complexity. On the other hand, the techniques of autonomic computing offer the promise of making systems more secure, by effectively and automatically enforcing high-level security policies.

Future Enhancement

The process of providing the self-management capabilities into an IT infrastructure is an evolutionary process. It must be implemented by every organization through the revision and adaptation of autonomic computing technologies and skills. The IT industry in particular, software systems will further develop self-management capabilities to help improve staff productivity, increase IT business resiliency and reduce operating costs. Following are the future scope for extending the work in creating and enhancing the self-managing autonomic computing capabilities into the software products for improved operational efficiency, supporting business needs and workforce productivity.

Conclusion

I have learned about autonomic computing and the techniques that are used, at the time of writing, to design and implement self-managed software systems. The purpose is clearly to help readers to understand, develop and maintain autonomic systems. I and my colleagues also have a discussion about next-generation software engineering techniques, approaches and tools that would be required to meet future computing system requirements.