Autonomic computing

Hrishikesh Tambe

Department of Computer Science and Engineering, SVERI's College of Engineering, Pandharpur

Third Year Engineering Student

Abstract:

Autonomic computing refers to the self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity to operators and users. Started by IBM in 2001, this initiative ultimately aims to develop computer systems capable of self-management, to overcome the rapidly growing complexity of computing systems management, and to reduce the barrier that complexity poses to further growth.

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 (AC) interacting with each other. An AC 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.

Problem of growing 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 labourcosts exceeding equipment costs by a ratio of up to18:1. Computing systems have brought great benefits of speed and automation but there is now an overwhelming economic need to automate their maintenance.

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.

Autonomic systems:-

A possible solution could be to enable modern, networked computing systems to manage themselves without direct human intervention. The Autonomic Computing Initiative (ACI) aims at providing the foundation for autonomic systems. It is inspired by the autonomic nervous system of the human body. This nervous system controls important bodily functions (e.g. respiration, heart rate, and blood pressure) without any conscious intervention. In a self-managing autonomic system, the human operator takes on a new role: instead of controlling the system directly,

he/she defines general policies and rules that guide the self-management process. For this process, IBM defined the following four

Functional areas:

- 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.

The design complexity of Autonomic Systems can be simplified by utilizing design patterns such as the model-view-controller (MVC) pattern to improve concern separation by encapsulating functional concerns.

Control loops

Basic concepts that will be applied in Autonomic Systems are closed control loops. This well-known concept stems from Process Control Theory.

Conceptual model:-

A fundamental building block of an autonomic system is the sensing capability (Sensors S i), which enables the system 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.

Characteristics:-

Even though the purpose and thus the behavior of autonomic systems vary from system to system, every autonomic system should be able to exhibit a minimum set of properties to achieve its purpose:

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 behavior in response to context or state changes. IBM has set forth eight conditions that define an autonomic system:

1. 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.
2. The system must be able to automatically configure and reconfigure itself depending on thechanging computing environment.
3. The system must be able to optimize its performance to ensure the most efficient computing process.
4. The system must be able to work around encountered problems by either repairing itself or routing functions away from the trouble.
5. The system must detect, identify and protect itself against various types of attacks to maintain overall system security and integrity.
6. The system must be able to adapt to its environment as it changes, interacting with neighboring systems and establishing communication protocols.
7. The system must rely on open standards and cannot exist in a proprietary environment.
8. The system must anticipate the demand on its resources while keeping transpaHrent to users.