ISRC Future Technology Topic Brief
Autonomic Computing

Prepared by: Siew Fan Wong
under the supervision of Dr. Blake Ives

What is Autonomic Computing?

Broadly speaking, autonomic computing refers to *computing infrastructure that adapts [automatically] to meet the demands of the applications that are running in it*[7]. Yet, to understand the field of autonomic computing, let us consider an interesting parallel to computing systems: our human body.[1]

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. As various changes occur within the external and the internal environment, the autonomic nervous system triggers functions that guide our bodies to manage the changes by responding to, healing and recovering from these changes, until our bodies eventually return to the homeostatic state[2, 3]. All these prevent our bodies from "crashing" or "shutting down" when we injure ourselves.

The field of autonomic computing rests upon the parallel of the human body, arguing that computer systems should be able to care for themselves, configure themselves, monitor themselves, repair themselves, and plan for future eventualities. By extension, if our computer systems can recover 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[7]. They would prevent small glitches from accumulating and cascading into full-blown outages[3]. And most importantly, all these functions would be performed without users knowing it and completely transparently to users.

To be successful, a fully autonomous computing system must function as a whole in order to achieve the desired benefits[4]. Again, consider the parallel of the human body. It is the self-governing operation of the entire system, and not just individual components that deliver the ultimate benefit. Imagine what would happen if an increase in heart rate is not followed by a corresponding adjustment in breathing and blood pressure; it could be disastrous. 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[4].

Why do we need Autonomic Systems?

Overall, the drivers for autonomic systems are: increasing technological complexity, the increased size of computing infrastructure, the ballooning maintenance costs of infrastructure, and the shortages of skilled labor. Each of these drivers will now be considered in turn.

[1] Note that 'autonomic computing' is an IBM term. There are other terms such as 'self-healing technology', 'holistic computing' and 'introspective computing'[5]. They all ultimately mean the same thing - "computing infrastructure that adapts [automatically] to meet the demands of the applications that are running in it"[7].

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. Often, IT personnel can't integrate different systems. Even when they can, IT personnel still face difficulties adding new systems to an existing environment and then configuring them and managing them. Thus most organizations now spend about three-fourths of their application deployment time and costs on the integration of different systems[6]. Then they must also deliver services across geographical and business boundaries. This means organizations, in addition to managing heterogeneous vendor and technical environments, also have to put extensive efforts into customizing technologies to meet the requirements of different IT policies while delivering unique services to customers. This complexity keeps the cost of managing IT infrastructure high[4]. At times, the complexity causes overruns in IT cost and delays in implementation. This in turns translates into losses in productivity and potentially losses in business opportunities[2].

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[17]. Rapid advances in technology have led to significant improvement in price/performance ratios, thus making technology accessible to many. Today, organizations are no longer dealing with one person accessing one application on a local PC or on a network server. Instead, organizations are seeing thousands, perhaps even millions, of users accessing the same service hosted on one or more servers and potentially at the same time. Relying on human intervention to manage this complexity is unthinkable as the scale and level of complexity extend beyond the comprehension of even highly skilled IT personnel.

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 today's competitive world where customers expect uninterrupted services, even a short break down can cost organizations millions of dollars in lost business. In fact, it has been reported that one third to one half of typical IT budget are spent on preventing or recovering from crashes[3].

The fourth driver of autonomic computing is the shortage of skilled labor. Workers who have the knowledge to manage complex IT infrastructures are expensive and, even in today's depressed economy, remain in short supply. According to a study by researchers at the UC Berkeley, depending on the type of system, labor costs could surpass infrastructure costs by factors of 3 to 18[3]. Therefore, the strategy of relying on human intervention to manage IT infrastructure might not be a favorable strategy in the long run as it might reach a point where existing skilled labors and manpower are not enough to supply the demand required.

What are the benefits of Autonomic Computing?

Autonomic computing, as with the human autonomic nervous system, offer four essential benefits: self-healing, self-optimization, self-protection and self-configuring.

(1) Self-Healing
Self-healing systems detect improper operations and initiate corrective actions before they occur, without disrupting system applications or business processes[4]. Self-healing systems help prevent failure by anticipating potential attacks or damages. In the event of an extraordinary event causing some parts of a system to malfunction, other parts of the system still continue to function as normal. If extra resources are needed to take over current workloads, self-healing systems find alternate way(s) to handle the requests so as to maintain uninterrupted service, and then try to fix the problem automatically and recover from any damages inflicted. Besides recovering from damages, self-healing systems learn from their experiences. Experiences populate their knowledge pools which are then used to prevent recurrences and to make future decisions about events that do repeat.

(2) Self-optimization

Self-optimizing systems efficiently maximize resource allocation and utilization so as to ensure optimal quality of service while continuing to meet users' needs with minimal human intervention[4]. Currently, organizations must adapt to a computing system by manually learning how to use and interact with it, and how to collect and interpret the types of data it produces. IT systems must also be adapted to match business strategies.

With self-optimized systems, organizations could automate much of their manual and complex workloads. Self-optimized systems help automating the management of complex systems, so that human managers could focus on high value activities such as planning and strategizing. Consider, for instance, a marathon runner. She goes to the starting line at the appointed time, sets her mind on the objective and course and then starts to run. The autonomic nervous system in her body optimizes her body response – automatically adjusting her heart rate, her breathing, her blood pressure and, as the elevation of the road changes, the speed of her running with little conscious thought. Self-optimizing features will provide future systems the same kind of anticipation and support capabilities our bodies have. For instance, organizations could set the goal of finishing a complex task within a day. Self-optimized systems will then find the best way to implement the task given available resources and competing task requirements. Thus, self-optimized systems not only optimize system uptime, they also reduce the time and cost for deploying and maintaining IT infrastructures. And, as these systems have the ability to learn, they could proactively adapt themselves to overall business policies and objectives.

(3) Self-protection

A self-protecting system can detect and identify hostile behaviors and take autonomous actions to protect itself against intrusive behaviors[4]. Organizations have long protected their systems by coding rules that identify and block specific events. For example, certain rules are written to look for corrupted data, firewalls are used to restrict and block illegal access to organizational information and virus definitions are employed to guard against potential infections. However, these rules must be based upon pre-written scenarios suspected by IT personnel and thus are only effective against the threats captured in those scenarios. Take the 'Code Red' bug, for instance. In 2001, this virus came unexpectedly to the industry, and in a completely new form. It propagated and multiplied by randomly selecting IP addresses across all of the IP address space. As a result, 'Code Red' infected millions of servers and computer systems and caused an estimated damage of US$2.62 billion worldwide[8]. Even in cases when an event that occurred falls within the precinct of preconceived scenarios, once alerted, IT personnel still have to manually examine the problem, analyze it and then repair the system themselves.

Self-protecting systems, as envisioned, could safeguard themselves against two types of behaviors, (1) accidental human errors, and (2) intentional malicious actions. To protect themselves against the first type of behaviors, for example, self-protecting systems could provide a warning if IT administrators were to initiate a process that might interrupt services. To defend themselves against the second type of behaviors, self-protecting systems would scan for suspicious activities and react accordingly without users being aware that such protection is in process. Besides simply responding to component failure or running periodic checks for symptoms, autonomic system will always remain on alert, anticipating threats and preparing to take necessary actions. Autonomic systems also aim to provide the right information to the right users at the right time through actions that grant access based on the users' role and pre-established policies. Therefore, with self-protecting capability, organizations would be able to consistently enforce security and privacy policies. This could help reduce overall security administration costs, and increase employee productivity and customer satisfaction.

(4) Self-configuring

Self-configuring system adapts dynamically to changes in the IT environment with little human intervention[4]. Existing IT infrastructures require costly human attention. For instance, if organizations want to add a new system to the existing infrastructure pool, they have to physically sit down and configure the system to make sure that the system fits into the existing structure and will function efficiently. Similarly, if workloads increase during certain times of the year, all organizations can do is to either accept a slow down during that time period or to add new systems to the existing infrastructure pool. Adding new systems that will only be used for one or two months a year is not an ideal solution.

Self-configuring systems could help solve these problems. These systems possess a sense of identity; they are equipped with the ability to know how to run themselves and to know how to find and generate rules on how to best interact with neighboring systems. Self-configuring systems adjust and adapt to changing IT environment by tapping into available resources and negotiating the use of their underutilized elements by other systems. For example, when Internet traffic increases, self-configuring systems have the ability to grab servers from the application tier and to reposition these servers so that they would now act as frontline Web servers. The self-configuring systems would then quickly and automatically load the servers from the application tier with all the software necessary for their new role. Once the traffic goes down, the servers would be returned to their original functions[7].

Through the four kinds of features described in this section (self-healing, self-optimizing, self-protecting, self-configuring), autonomic computing can reduce the cost of owning and operating IT infrastructures[4]. Ed Denison, former director of global operations for outsourcing at the Computer Sciences Corporation (CSC) estimated that Terraspring (an example of an autonomic solution) enabled CSC to save as much as $300 per month in labor costs per server[7]. By multiplying that number with the 16,000 servers for just one of its customers, J. P. Morgan Chase, and we see that CSC is looking at a tremendous amount of annual savings. Denison also predicted that the number of administrators per server to improve from the 18 servers per administrator to 40 or 80 or more in the future. In addition, autonomic systems minimize potential human errors as human interaction with infrastructure is minimized. These systems also enhance the organizations' capability to react to change[4]. Collectively these characteristics could enable organizations to operate more efficiently and productively.

Are Autonomic Systems a new concept?

While the term 'Autonomic systems' may be new to many IT managers, the application of the underlying concept in computing environments is not new. In the early days of multiprogramming, for instance, hardware designers developed the concept of bounds registers to ensure that programs stayed within the confines of memory they had been assigned by the operating system, thus limiting the damage one program could impose on another. A more recent example can be found in RAID (Redundant Array of Inexpensive Disks) storage systems[9]. These systems connect multiple small, inexpensive disk drives into one logical storage unit. This logical storage unit uses techniques such as disk stripping, disk mirroring, hot spares and redundant power supplies to automatically switch among redundant disks in order to prevent hard-disk crashes and a total system shutdown. Similarly, SQL Server 7 and SQL Server 2000 have the ability to allocate memory automatically depending on the input/output demands and the size of a buffer pool[2] while 'Spiders' (software agents from search engines) when launched would search the Web for new websites and then return the results to the search engine and update its databases with the new URLs[5]. Other technologies that employ the autonomic concept, include Error-Correcting Code memory, SMART (Self-Monitoring, Analysis and Reporting Technology) for hard disks, Plug-and-Play technology, Window XP, database optimizers, virus management and fault-tolerant servers[2,5]. But, while these technologies move in the right direction, they represent only the early steps towards true autonomic computing.

The concept of autonomic systems extends beyond the IT industry. In the auto industry, for instance, Daimler-Chrysler has long been perfecting Anti-lock Braking Systems (ABS) to ensure driver's and passengers' safety. ABS is comprised of 'electronic sensors and solenoid valves in the wheel hubs'. These sensors and valves used the concept of autonomic systems to prevent the wheels from locking when cars go into a skid[5]. Goodyear and Michelin have created 'Run-Flat' tires that allow drivers to drive safely for a few more miles after a tire punctures. Besides having a reinforced sidewall that act to maintain the chassis level when a tire is deflated, run-flat tires also have sensors that would relay information about the air pressure to the dashboard of the car so that drivers could monitor the pressure levels and act accordingly[5].

What are some of the projects related to Autonomic Computing?

The realization of autonomic computing systems will require the combined efforts and resources of industry and university researchers from throughout the world. The following table provides descriptions of some of the projects already underway in the corporate world.

Organization	**Project(s)**	**Project Description**
HP[2,6]	"Virus-throttling" software	This software has the ability to protect networks from rapidly spreading viruses. It scans network for abnormal activities. When it identifies illegal activities (i.e., viruses attempting to spread rapidly to as many machines as possible), the software instructs computers to go into a 'degraded mode'. This mode allows connection to familiar machines at a very slow rate while delaying or blocking connections to unfamiliar machines, thus, preventing the virus from spreading.
	Utility data center	This is an initiative to develop a set of technologies for planning, designing, managing and supporting data center resources. It has the ability to automatically analyze the typology of an environment and the resources available. Based on the information, it then runs the center using virtualized network resources and virtualized storage resources.
IBM[4]	eLiza	This project is IBM's initiative to add autonomic capabilities into existing products such as their servers. It encompasses such abilities as the detection and isolation of bad memory chips, protection against hacker attacks, automatically configuring itself when new features are added, and optimizing CPU, storage and resources in order to handle different levels of internal traffic.
	Océano	This project uses optimization algorithms to manage servers. It has the capability of figuring out the best way to distribute tasks and the cheapest places to store data. It tries to anticipate demand and prepare computer systems with necessary resources just before they are needed.
Intel[16]	Itanium 2	Intel builds into its Itanium 2 processor features of autonomic computing called the Machine Check Architecture (MCA). MCA is an open architecture that allows systems to continue executing transactions as it recovers from error conditions. It has the ability to detect and correct errors and to report these errors to the operating system software. It also has the capability to analyze data and respond in a way that enables higher overall system reliability and availability.
SUN[10]	N1	N1 has the ability to provision computation, storage and network resources based on the demand for a service. It automates software and hardware installation and configuration for new business service deployment. It also dynamically manages the allocation of IT resources based on pre-defined business policy.

The following table lists all of the projects provides descriptions of some of the projects already underway in the academic world.

University	Project(s)	Project Description
Edinburgh University, UK [11]	Neuromation	This project develops algorithms to structure information based on simple, autonomous units that capture the thoughts and concepts behind the data. With the algorithms, information stored would no longer be dependent on predefined structures. Instead, information is meaningful in its own right regardless of the applications that use it or the structures required to express it. This makes it easier for data to be broken into subsets and transferred seamlessly to other users or devices.
Monash University, Australia[12]	Nimrod-G	This technology provides tools and services that support deadline and budget based scheduling. It has the ability to lease resources at runtime depending on their capability, cost and availability.
University of California, Berkeley[13]	Recovery-Oriented Computing (ROC)	This is a joint project that investigates techniques for building highly dependable Internet services. It emphasizes recovery from failures. Key features include isolation and redundancy, system-wide support for Undo, integrated diagnostic support, and high modularity and restartability.
	OceanStore	This is a global data store designed to provide consistent, highly available and durable storage to users. It allows any server to create a local replica of any data object. These local replicas provide faster access and robustness to network partitions. OceanStore also has internal event monitors that allow it to collect and analyze information such as usage patterns, network activity, and resource availability. Based on the information, it could then adapt to changes in the environment such as outages, service attacks, etc.
University of Maryland, Baltimore County[14]	eBiquity	The goal of this project is to create systems based on the cooperation of autonomous, dynamic and adaptive component of various systems. These systems include mobile/pervasive computing, multi-agent systems, artificial intelligence and e-services. These systems will be composed of a collection of independently designed components that automatically become aware of each other, establish (wireless) communication, exchange information about their basic capabilities and requirements, discover and exchange APIs, and learn to cooperate effectively to accomplish their individual and collective goals.
University of Texas, Austin[15]	Qualitative Reasoning	Several research topics in the group include spatial reasoning and intelligent robotics, access-limited logic for knowledge representation and qualitative reasoning about the physical world.

What challenges remain in implementing Autonomic Computing?

While a variety of efforts are underway to develop particular components of autonomic computing; the challenges in implementing an effective and comprehensive autonomous computing environment requires that development and implementation to also occur at the level of global enterprise IT systems[4]. This requires each individual component in an autonomic system to function properly and for the components to communicate and work efficiently with each other. This will require the development of each individual component in line with agreed upon standards. New algorithms will also be required for new features including the ability to remember previous occurrences, to take previous system experiences and sort the information into good and bad information (i.e., the information to be remembered and the information to be ignored) and then to use that information to improve the rules for making decisions on future actions. Autonomic computing also has to be able to handle and function in an increasingly heterogeneous environment. The technical challenge (i.e., the architecture) on how to actually do all these is still unknown though it will obviously require cooperation from both vendor and user organizations.

Presently, organizations use proprietary IT infrastructures supplied by different vendors. As mentioned before, implementing autonomic computing will require a set of new standards shared among different vendors[4]. This standard must provide a means of identification and communication that will facilitate the sharing of knowledge regarding the management and the development of autonomic IT infrastructure. So, ultimately, an autonomic computing system cannot be a proprietary solution. Instead, vendors must be willing to adopt and implement open standards[17] such as that used by Linux (an open operating system), Apache (an open web server) and the Globus project (a set of protocols to allow computer resources to be shared in a distributed, grid-like manner). Unfortunately, there may be some vendors who are reluctant to do so as adopting open standards may threaten their competitiveness in the market place.

IT management must also be prepared for change as well. First of all organizations (especially IT management) must be convinced of the benefits of adopting autonomic systems. They have to be convinced that such systems will not deprive them of control; but will instead help simplify systems management and thus provide them with more time to deal with other more important issues such as formulating IT policies. IT personnel may also be expected to resist the implementation of autonomic systems as implementation of the systems would threaten their job security. While autonomic computing will not put people out of jobs, it will minimize the number of IS personnel needed to perform certain tasks[4]. Even if organizations accept autonomic systems, they have to adapt and change some of their existing business processes to meet the changes in the IT environment[4]. They need to retrain their employees (especially IT personnel) with new skills and knowledge to implement the systems. The cost (i.e., in terms of time and money) to do so may be so high that some organizations will not be willing to take the risk.

For More Information

Online resources

[4]For IBM's work in autonomic computing, www.ibm.com

[6]To learn more about HP's initiatives in autonomic computing: www.hp.com

[10]To find out more information on the N1 project at Sun, www.sun.com

[11]To know more about the neuromation project at the Edinburgh University, www.neuromation.com

[12]For information on projects at Monash University, Australia, www.gridbus.org

[13]To find out about ROC and OceanStore, www.cs.berkeley.edu

[14]To learn more about ebiquity, www.research.ebiquity.org

[15]To learn about projects on Qualitative Reasoning, www.cs.utexas.edu/users/qr

[16]For information on Itanium 2, www.intel.com

Articles

[3]Gibbs, W. Wayt. "Autonomic Computing." *Scientific American.com*, May 6, 2002.

[17]Horn, Paul. "Why Autonomic Computing will Reshape IT." *ZDNet Australia: News & Technology*, October 16, 2001.

[9]Karp, Mike. "Autonomic Computing: Part1." *Network World Storage Newsletter*, October 2, 2002.

[7]Knorr, Eric. "It's Alive." *CIO*, Oct 1, 2002.

[8]Layman, Jay. "In Search of the World's Costliest Computer Virus." NewsFactor Network, February 21, 2002.

[5]Pereira, Brian. "Autonomic computing brings the healing touch to IT." *Express Computer*, August 19, 2002.

[2]Scott, Karyl. "Computer, Helf Thyself." *InformationWeek*, April 1, 2002.