Self-optimization in Autonomic Computing Systems based on the Methodology of Bees Swarm Intelligence

Anselmo Leonardo O. Nhane¹, Mark A. J. Song²

Computer Science Department, Pontificial Catholic University of Minas Gerais, Belo Horizonte, MG, Brasil
¹ Computer Science Department and Information System, National Statistical Institute, Mozambique
² Computer Science Department, Centro Universitario UNA, Belo Horizonte, Brasil

Abstract—This paper proposes a mechanism for self-optimization in autonomic computing systems, inspired in the functioning of the bees swarm in the process of searching for food sources, exchanging information about the located sources and in the allocation of bees to such sources. This analogy is applicable to autonomic computing systems, on which we also seek to continually optimize the system operation by monitoring and adjusting its parameters and evaluating the fitness of proposed solutions. The goal is try to meet 100 Percent of the Demand while minimizing Costs by the system. The methodology follows the component of autonomic control loop. The article includes a case study focused on the statistical information dissemination system of Mozambique, which uses resources from a private datacenter.

Keywords: Autonomic computing, Autonomic computing systems, Self-Management, Self-optimization, The Bees autonomic management

1. Introduction

The increasing complexity in computer systems, Figure 1, applications and services, allied with the growing in the demand for integrated services, are factors that limit traditional mechanisms for software system management with human intervention.

Fig. 1: Example of the IBM Complex System

Mechanisms such as definition of protection policies, elicitation of the system, resource management, troubleshooting, configuration and implementation definition that meets optimization aspects, require a thorough knowledge of the systems. However, the human capacity to intervene and meet all these systems’ needs is a limiting factor (organization challenges), which motivates the scientific community to propose systems with minimal human intervention. That is, the paradigm that aims at reducing administrative overhead by providing self-managing applications.

In poor and developing countries like Mozambique, which has one of the fastest-growing rate of Gross Domestic Product in Sub-Saharan Africa (about 7.3 percent per year), infrastructure is also in constant expansion, accompanied by the increase in the number of users and services provided using software systems, which implies an increased complexity. But these countries also have low availability of trained human resources to deal with the growth of the software systems and infrastructures and also they do not have the budget to ensure the continuous expansion in number of systems. In this context of computing, introducing autonomic computing can make a difference to this group of countries. First, the rate of computer literacy in the whole countries can be minimized, once autonomic computing can take care of its functioning with minimal human intervention.

In 2007, Mozambique had less than one percent of the population with internet access [5], but in 2010, the number of users rose to approximately 2.7 percent and to 4.3 percent in 2012 [7]. This increase is resulting in complex systems in a short period of time to meet the growing demand for current services and products of the Mozambique’s statistical system.

In this paper we propose a self-optimizing mechanism, in autonomic computing system, drawing inspiration from the operation of the bees swarm on its process of looking for food and sharing information in the dance space. In this meter we have confidence that the working process in the hive can be applied to the operation of an autonomic system, which constantly seeks to optimize the system functioning through the monitoring, control and analysis of parameters to identify changes that require configuration changes or allocation of more resources in a situation of efficiency loss.

The article includes a case study, aimed to show how
developing countries can optimize their resource usage, by monitoring demand and making decisions to adjust the level of needed resources from the datacenter, at the dissemination unit of National Statistical System in Mozambique, with budget restrictions.

2. Autonomic Computing

The concept of Autonomic Computing (AC) was introduced by Paul Horn [9]. This concept is inspired by the Human Nervous System in order to find new ways of implementing software systems that are able to answer some challenges posed by the increasing complexity of current IT systems. The human nervous system is the most complex system and it is an example of autonomic behavior in nature. Such system is the master controller of the functioning of the human body, which monitors the changes that occur inside and outside the body. Then, responds appropriately in order to maintain the balance of the functioning of other organs [8]. Complex software systems can be comparable with the human nervous system which controls the most part of body works, eliminating from the human conscious the activities of managing all the actions of the body. So, complex systems must possess autonomic properties that allow them to control part of their operation without human intervention. This paradigm means, the change of the patterns of the systems managed by humans (traditional), into the new era of technology standards managed by the technology itself [19]. Such systems are called autonomic. Several proposals have emerged in the last decade, challenged by the complexity of IT systems. In 2003, for example, Defense Advanced Research Projects Agency (DARPA) proposed self-regenerative systems for military purposes, that can react to unintentional errors or attacks. An example is the Situation Awareness System, from the DARPA, where they intended to create a communication device for location of the soldiers on the battlefield, that can detects and collect data on the presence of the enemy tank, and independently reporting on the location of these to all soldiers. Even in situations of extreme hardship, it can aid in minimizing interference from the enemy [1].

2.1 Autonomic computing system properties

An autonomic system has the following properties: ability to self-configure, self-heal, self-protect and self-optimize. Self-configuration is a property on which the system must ensure, automatically, dynamic adaptation to the changes that occur in their environment. Self-healing is the property on which the system must discover/identify, diagnose and react to the disturbance that can cause the system to malfunction. Self-protection is another property inherent in autonomic systems. This property is in the system to anticipate, detect, identify and protect itself from attacks from any source. Finally, we have the self-optimizing property, through which the system must monitor and adjust its own resources. For this, it is necessary that hardware and software systems maximize the efficient use of resources to achieve the end-user requirements with minimal human intervention. However, several aspects of software engineering, such as life cycle elements, software processes, elicitation of system requirements, self-healing implementation, self-protection implementation, self-configuration and the implementation of self-optimization are still major challenges to be overcome. Thus, the research focus on self-optimization, based on the use of the bees swarm method [10] to optimize the system parameters that govern the system operations in the allocation of resources.

3. The Bees Algorithm

In the natural process of exploring environment, the bees are divided into groups (onlookers, foragers, employed). The Foragers seek those regions with good food (nectar and pollen) sources. After returning to the hive they perform movements known as Waggle Dance, to communicate to the others bees the profitability of the sources found, the distance and amount of these food sources.

Then, the employed bees and onlookers travel to the selected food sources and for the fields with higher abundance of food, more resources are allocated (greater presence of employed bees) [16], [3], [4]. The process repeats and the bees are recruited more and more, in accordance with the food demand.

Karaboga, in his study of the Swarm Intelligence, proposed the bees swarm algorithm for parameter optimization [10]. Successively, the algorithm versions were being updated with substantial improvements [11]. The latest version offers a parallel approach [15], [13], where the artificial agent used is divided into several independent sub-populations.

The distribution of food sources positions are randomly produced to the allowable limit of the parameters. From the Karaboga algorithms illustrated in 1, we have:

\[ x_{ij} = x_{ij}^{min} + \text{rand}(0,1)(x_{ij}^{man} - x_{ij}^{min}) \]  
(1)

Where \( i = 1, 2, ...SN, j = 1..D \). \( SN \) is the number of food source and \( D \) the number of optimization parameters. The employed bees operating the sources calculate food sources and share information with onlookers bees, according to the formula in 2:

\[ v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{ik}) \]  
(2)

Where \( j \) is a random integer in the range \([1,D]\) and \( k \in 1, 2, ...SN \) is a randomly chosen index different from \( i \). \( \phi_{ij} \in [-1,1] \) uniformly distributed. After the \( v_{ij} \) is specified with values that are within the limits of the parameters, a fitness value for minimization problems can be associated with the value \( v_{i} \) by the formula: \( \text{fitness}_{i} = 1/(1 + f_{i}) \) if
\( f_i \geq 0 \) and \( \text{fitness}_i = 1 + \text{abs}(f_i) \) if \( f_i < 0 \), where \( f_i \) is value of the objective function that represents the cost value of the solution \( v_i \).

The greedy selection is used between \( v_i \) and \( x_i \) for each mutant solution, based on minimal cost of each \( v_i \) generated. After all employed bees complete their search for food, they share their information about the amount of nectar and the position of the source with bees onlookers in the dance area. The onlookers bees select food sources according to their profitability, and then they look for new sources of food for a probabilistic choice given by 3:

\[
p[i] = \frac{\text{fitness}[i]}{\sum_{i=0}^{SN}(\text{fitness}[i])}
\]

Finally, if a food source is not improved, then the bees leave the food source and call upon an onlooker to discover a new source in the search area.


The self-optimization process is based on the MAPE-K (Monitor, Analysis, Plan, Execute and Knowledge) control loop proposed in [12], Figure 2.

The sensors,[12] collect data from the managed resource, via a log file. These data are grouped according to the objective criteria that guide the process. Example: collected data from the dissemination service in Statistic office, of the user demand include: time, IP, information concerning data sets, database, number of data cells extracted, information size of extracted data. The data is set to SQL database.

The monitor processes the data collected by the sensor and communicates the level of demand indicators. The processed data are used for the recognition of demanded levels, according to the two following steps: (1) extraction of processing characteristics, from the information passed in the bees dance area, and (2) the classification of the processing level. The extraction of (1) is obtained by applying the Discrete Fourier Transform (DFT), as illustrated at formula 4.

\[
X[k] = \sum_{n=0}^{N-1} x[n]e^{-2j\pi kn/N}
\]

The sensors, [12] collect data from the managed resource, via a log file. These data are grouped according to the objective criteria that guide the process. Example: collected data from the dissemination service in Statistic office, of the user demand include: time, IP, information concerning data sets, database, number of data cells extracted, information size of extracted data. The data is set to SQL database.

The monitor processes the data collected by the sensor and communicates the level of demand indicators. The processed data are used for the recognition of demanded levels, according to the two following steps: (1) extraction of processing characteristics, from the information passed in the bees dance area, and (2) the classification of the processing level. The extraction of (1) is obtained by applying the Discrete Fourier Transform (DFT), as illustrated at formula 4.

The Discrete Fourier Transform (DFT) is a function of the user demand frequency grouping of extracted sample data, at periodic intervals (which can be defined by the users).
The recognition of demand patterns (high, medium, low) is done by classifying the extracted demand intervals. This information is used to map the needs for resources. The clustering is done from the analysis functions \( x(n) \) of the observed demand data, and \( X(k) \) is the transformed form of \( x(n) \) according to the formula in 4.

The example in Figure 4 includes data collected by the sensor according to the methodology of bees, which corresponds to the demand registered in the system environment. The DFT line of graphic shows clustered data, reported in the area of dance (ruled by the methodology of bees) with less amplitude, which helps in finding accurate values to the decision step.

![Fig. 4: DFT and Demand frequency: Service usage for a day constant number of resources](image)

The DFT properties are inherent to the needs of the study, allowing vertical translation of the observed frequencies in order to perform analysis on the situations at low (A), middle (B) and high (C) demands while achieving better precision in demand levels, as Figure 5.

The Monitor uses DFT with the aim of grouping similarity levels by proximity. That is, determining values with the same distribution characteristics to converge within a given range for a certain demand class. This enables the elimination of the influence of extreme values if using the average metric. The adapt case of autonomic monitor, illustrate the 5 the actions of monitor activity.

The step of decision analysis correlates reported event data to identify the current status of the managed element and then suggest actions to be taken in case of performance loss or cost degradation. Then, the BAM use the Bees algorithms and determines which resources are going to be used, correlating the quantities of resources used, and the level of demand identified that optimizes the cost of operation.

This phase uses classified demands from the autonomic monitor, to determine which level of resource is necessary to minimize the cost. In Figure 6, we represent the general overview of the use case for the bees autonomic manager. The bees autonomic manager loop considers that the system interacts with users. And the external component - autonomic manager - is added. The diagrams representation are sourced from Markus in [14], where the study proposes the Adapt Case Modeling Language that can be applied for specification of the autonomic computing systems.

The analysis and decisions also use knowledge to enable decisions in the situations which are planned beforehand, and not in case of dictation of high or low demand. The system in the diagram uses the datacenter (the source for resources) to obtain resources to supply the demands specified.

The resource fitness function is obtained from the cost of each food source: each type of resource has its cost. Each food source includes all types of resources. \( uC^i[j] \) - cost of \( j \) resource in the the \( i \)th food source.

Each mutant solution \( s^i \) is evaluated using the following formula:

\[
fsuC[i] = \sum_{j=0}^{NP-1} s[i] * uC^i[j] \tag{5}
\]

The formula to determine the fitness related to the cost of each class of demand is the following:
\[ f_s(i) = 1 - \frac{f_{suC}[i]}{Bc} \] \hspace{1cm} (6)

Where \( Bc \) is the Budget of demand class, that represents the Total value that enables administrators to specify resource limitations for the system. That is, there is the cost value of each set of resource from the food source related to the demand. In each situation of demand we consider a particular budget of the demand class. This information will help the autonomic bees manage the different cost degradation for each class (\( A \) - lower, \( B \) - medium-size, \( C \) - high) of demand. Based on the fitness value of each solution:

\[
fitness[i] = \begin{cases} 
\frac{1}{1+f_s(i)} & \text{if } f_s(i) \geq 0, \\
1 + \text{abs}(f_s(i)) & \text{if } f_s(i) < 0.
\end{cases}
\hspace{1cm} (7)
\]

Management Levels in the BAM: The bees make decisions in order to achieve global parameter optimization. But the proposal results do not completely guarantee the optimal allocation of resources. Two steps are considered for the decisions. In the first level, we consider the cost of current resource of proposed solution and the total cost of the demand class. The second level indicates at Lower level, resource based on efficiency indicator (like speedup, response time) optimization in order to balance the needed resource, from the bees output proposed parameters.

The knowledge base provides data in addition to information about past decisions. An example of data provided is the time of day, with historical information demand levels. This information adds significant value for predicting the situation of high levels of access to the service.

The example, in Figure 7, illustrates the behavior change of service requests to the systems in the first seven hours, where there was a drop from 5.8 (average value) to nearly 2.0 (average value) of demand. This indicates that demand has fallen about three times the value of the last seven hours. Implying the change of track \( B \) to the range of low demand \( A \). That is, the resource in use are in high cost, so it is necessary to reduce the resource to better attend the current situation.

5. The Case Study: I

In the case study we apply the optimization to the dissemination system of National Statistical Institute of Mozambique, based on a private datacenter.

Background: We consider that there is a computer network that can support the configuration in order to attend the needed of services levels. There are technologies available for self-configuring to attend if the optimization decisions if achieved. These assumptions help to visualize the proposed methodology for the few steps of self-optimization based on the cost in the datacenter for the statistic office.

![Fig. 8: Source of resources: Datacenter](image)

The mean and standard deviation, maximum and minimum values that were observed are key indicators to consider in this range. So the new parameters are defined based on this information. In the example considered, we use the average to select the demand class.

In the decision planning step we evaluate the current demand and observed demand, as the criteria for optimizing based on the specifications of the administrator in the form of policy or from analysis and decision, providing new parameters for the system.

The process of executing decisions occurs when the subset of parameters can improve the current standard of processing resource consumption. The process receives data on the approved update to the next time interval. Then it updates the information on the use of resources in the knowledge base. This step allows the system reconfiguration to attend the suggested levels of demand at low adequate cost of resource usage.

The Private Datacenter of Mozambican statistical service, Figure 8, represents the source of resources for the managed element: the output database of the dissemination unit.

The infrastructure includes a Web-server with non relational database, Pxweb, with .\( Px \) files (PC-AXIS software family format), and it also includes statistical software applications for dissemination. These data set have homogeneous sizes (less than 1Mb). In accordance with the data collected from the Logs of the web-server, 99 percent of queries are characterized by extracting data of smaller size. The main feature is to build the px-matrix and send it to the user. The
user can then either manipulate or view the table. To manipulate tables, there are application tools based on javascript which run in the user side. All the processes are affected by processor, memory, disk, and bandwidth. That is, the optimization process have to consider the four parameters for each food source (datacenter node with processing capacity) in the datacenter.

Fig. 9: Initialization: Up bound and low bound

Control parameters: Number of bees (employed bees + onlookers), NP=6; Max cycle 10; execution time 10000; Number of optimization parameters, D = 4. Number of food sources, which represents data center sites, SN=3; Mutation rate $\phi = 0.05$; the resource bounded by Upper and Lower resources, Fig. 9. Representing the availability of resource that can bee allocated to the system from the datacenter. The cost is considered 1 USD per resource unit, per hour. We consider the budget for A is the minimal.

The output of executions is a set of minimal parameters that represent the A demand situation, that where selected from fitness in formula (5) with an estimated cost 10 USD.

Fig. 10: Output with the A ranked demand

Fig. 11: Output with the B ranked demand

In each interval of time the reward will be difference from the constant utilization of the system resource. In non autonomic systems, mines are used to the demand the cost in use for each interval.

6. Review of Related Works

The question of how to achieve self-optimization in autonomic computing has dominated research in the autonomic computing field, during the first decade. The self-optimization property has received the most contributions since IBM formally introduced the autonomic computing paradigms. The proposals on self-optimization are based on different technique like the Control theory, Utility functions, and Queue theory.

Walsh proposes the use of model-free reinforcement learning based on the use of SARSA [17], [18]. This approach estimates future learning rewards, Q, for each state-action pair. Its learning rule is given by:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

(8)

Where $Q$ is the estimated reward (expected); $s$ is a state, $a$ is an action, $r$ is the reward. $E t$ and $t + 1$, are used to indicate states, present and future and action respectively. $Alpha$ is the learning rate and $gamma$ is the discount factor. The SARSA name is given due to the quintuple $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$ that the algorithm uses.

Van (2010 ) quoted by [6], in his study on "Managing performance and energy consumption in cloud infrastructure ", uses an approach based on utility function to optimize performance and energy consumption of the cloud system, by determining how many resources could be allocated, and how the applications should be put in the infrastructure to optimize energy consumption.
The study proposes one framework on two levels. The first is the application level, where for each application, there is a performance model, measured by the utility function and used expressed in response time. At the second level of infrastructure, there is a global asset manager. The first is responsible for determining the amount of resources in terms of VMs for each application, while the second, by provisioning VMs resulting from the first manager, is responsible for optimizing the allocation of all VMs, as well as to minimize the number of physical machines needed and power consumption.

And the control is done sequentially and periodically, if and only if, there is a difference between current number of VMs and the resulting number of provisioning manager. As a result of the implementation of the provisioning of VMs manager is a set of VMs to be requested or released. Manager allocation of VMs produces a set of VMs with purpose or to create or to destroy or to migrate, as well as to turn off or turn on the power consumption [20]

7. Conclusion

The proposal and ongoing project contribute in the study of autonomic computing, exploring alternatives for achieving self-optimization for the systems. This is done by applying the BAM. This analyse uses a wide cycle for convergence to obtain good results, but they do not interfere in the system once the decision are made in each interval of at least 5 minutes or more depending on system type or number of parameters, and the algorithm convergence is achieved in less than a minute. The example considers the management of resources to the dissemination service, allocating and removing resources provided from the datacenter. This approach helps to maintain the good level of resource usage at low cost based predicted demand. During the observation periods, we show situations where the system was using resources to fulfill the demand for those periods. From Figure 7, the system uses the total resources available for the system during all 13 time units. With the BAM we change from 7 to 13 the amount of resource, form B to A. This result in improvements on the use of resources (e.g. low power consumption, less computer or VM allocated). In general, this causes great money savings. This really fulfills the poor countries, since in these countries there are less literacy rates in computer fields, as well as smaller budgets to maintain the systems. So we agree that using Autonomic computing systems, it will be possible to improve management of total cost for system owners. And so on, minimizing governments challenges in the decision level of the organization in how to introduce the IT systems in remote locations.

Future works: The BAM coordination is the next step to deal with. We agreed that two major situation are important for it: Global parameter optimization coordination - That indicates the best source of resource in term of adequate cost for the demand level, to allow multi-application management.

We intend to develop a framework that will implement the bees autonomic manager and test it in real-time systems for Population Census. We consider that this can help with data collection in distributed environments, by adding and removing resources that can be sourced from external enterprises based on cloud computing in order to minimize cost of the activity.

Acknowledgment

The authors would like to thank the National Statistical Institute(Mozambique) and FAPEMIG for the financial support.

References