Decision-Theoretic Model to Support Autonomic Cloud Computing

Alexandre Augusto Flores, Rafael de Souza Mendes, Gabriel Beims Bräscher, Carlos Becker Westphall, Maria Elena Villareal
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianopolis, Brazil
e-mail: alexandre.flores@posgrad.ufsc.br; rafaeldesouzamendes@gmail.com; brascher@lrg.ufsc.br; westphal@inf.ufsc.br; maria@lrg.ufsc.br

***Abstract*— Much effort has been made to provide a Cloud Computing (CC) autonomic management. Thus, related works are discussed and the need of a full autonomic model with stakeholders is presented. Moreover, this paper introduces a full model of cloud environment to support decision making in autonomic systems. This model is based on an economic utility view of cloud computing, control theory and autonomic computing. It innovates by introducing the concept of conjuncture and imaginary elements (essential elements to forecast and to a non-stationary model). Mathematical modeling is used to formally define a model and a model implementation overview is given.**

Keywords-cloud computing; autonomic computing; decision-theoretic planning; cloud model.

I. INTRODUCTION

The widespread use of computing devices has introduced a drastic change in the way that computing is produced, distributed and consumed. A strong trend is the concept of cloud computing (CC), which is basically a paradigm that deals with economical activity of production, distribution and consumption of computing. According to Kephart et al. [1], the difficulty of managing computer systems goes beyond managing software isolates. The CC dynamic integrates heterogeneous environments and introduces new levels of complexity, outperforming the levels of human capacity [2]. The result is a demand by autonomics clouds.

Although many works propose the automation of CC management, none of them has a model that represents all the stakeholders involved.

This work presents a new CC view based on economy, and utility leading to a useful approach to cloud management. Using a holistic definition, we propose a model to CC management derived from our model introduced in [3]. This generic model can be used to subsidize many decision-making processes and is presented using a mathematical modeling of principal elements and their relationship with eachother.

This paper is organized as follows. Section II addresses the relevant literature and presents our view of CC. Section III presents CC needs for autonomic management based on related works. Section IV describes our proposed model with mathematical representations and presents a simplified class diagram. Finally, we draw conclusions and suggest possibilities for future research.

II. LITERATURE REVIEW

A. Cloud Computing definition

In this section, we will introduce three CC definitions chronologically. Those references brief our view of the evolution of CC definition over the last years.

Foster et al. [4] have an interesting definition for CC: a widely distributed computing paradigm driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.

Fosters definition is relevant mainly for two reasons: Firstly, he defines CC as a paradigm, and secondly, understands the economic influence at cloud

Furthermore, Buyya et al. [5] have a more complete view which recognizes CC as a paradigm for delivering computing resources as an utility, like gas and water.

Later, the National Institute of Standards and Technology (NIST) [6] defines CC as:

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

The five essential characteristics stated by NIST are: on demand self-service, broad network access, resource pooling, rapid elasticity and measurable service.

As demonstrated, CC definition has changed in the last years from an economic view to a pragmatic and limited understanding. NIST definition is an attempt to allow comparisons between services. However, they recognize the limitation and state that the service and deployment models defined form a simple taxonomy that is not intended to prescribe or constrain any particular method.

Because we see the CC phenomenon more like Foster et al [4] and Buyya et al. [5], our view of CC is:
the economic activity that focuses on mass production, distribution and consumption of computing. This computing has abstracted logical and physical resources and

prominent commercial frontiers between the stakeholders who produce and consume it.

B. Autonomic Computing

The autonomic computing (AC) concept is based in the human autonomic nervous system that governs our heart rate and body, thus freeing our conscious brain from the burden of dealing with these and many other low-level, yet vital, functions [1]. The overall goal of Autonomic Computing is the creation of self-managing systems; these are proactive, robust, adaptable and easy to use.

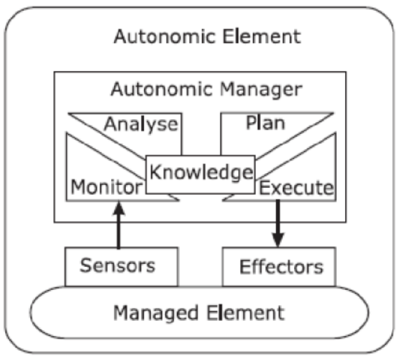

Figure 1. IBM's MAPE-K reference model for autonomic control [6]

A fundamental element that figure in AC bibliography is the MAPE-K control cycle (Figure 1), that consists in Monitor, Analyze, Plan, Execute and Knowledge elements.

For an autonomic system, as shown in [7], to be able to perform self-management, four main abilities must be present: self-configuration, self-optimization, self-protection and self-healing. To achieve these objectives a system must be both self-conscious and environment-conscious, meaning that it must have knowledge of the current state of both itself and its operating environment.

Huebscher et al. [7] define four degrees of autonomicity which can be used to classify autonomic managers and give us the focus, architecturally, that it has been applied. Those elements are:

Support: focuses on one particular aspect or component of architecture to help improve the performance of the complete architecture using autonomicity.

Core: the self-management function involves the core application. It is a full end-to-end solution.

Autonomous: it is also a full end-to-end solution, but the system is more intelligent and it’s able to self-adapts to the environment.

Autonomic: this is the most complete level where the interest is in higher-level human based goals like service-level agreements (SLAs), service-level objectives (SLOs) or business goals are taken into account.

C. Control Theory

Control theory uses engineering and mathematics to deal with the behavior of dynamic systems. The objective of a control system is to make de output y behave in a desired way by manipulating the plant (system) input u [8].

Therefore, we present the first four steps to design a control system, stated by Skogestad [8]:

1. study the system plant to be controlled and obtain initial information about the control objectives;
2. model the system and simplify the model if necessary;
3. analyze the resulting model determine its properties;
4. decide which variables are to be controlled outputs;

Those steps will be mentioned furthermore as the Design Process (DP).

Control Theory often uses transfer functions as a representation, in terms of spatial or temporal frequency, of the relation between the input and output of a linear system. On the other hand, to model complex systems, such as a multi-objetive system, Modern Control Theory often uses a state approach instead of transformation. The system’s state is a set of values representing environment.

CC environment management can be classified as a multi-objective multivariable control problem in a time-discrete system. We can assume the dynamics of the CC system to be controlled by several actors where each of the actors has the aim of optimizing its results along the trajectory determined by vectors of control parameters chosen by all players together [9]. A stochastic approach can be used resulting in a Stochastic Multiplayer Game (SMG).

In this class of problem, Nash, Pareto and Stackelberg optimization principles are often used with cooperative and non-cooperative game-theoretic models. To deal with complex systems control, another known strategy is to use Markov Decision Process (MDP) to select the best sequence of actions to been taking. Now we revise those concepts.

1) Nash Equilibrium

Nash equilibrium, proposed by John Nash [10], describes a situation where no player can increase his payoff by unilaterally switching to a different strategy.

2) Pareto optimal

The Pareto optimal is achieved only when a player can become better off in the game without making any other individual worst off.

3) Stackelberg games

A Stackelberg game solution is formulated to model a leader-follower joint optimization problem as a two-level optimization problem between two decision makers.

The upper-level decision maker (leader) announces his decisions to the lower level (follower). Next, follower makes his own decisions and then feeds decisions back to the leader. This implicates in a mathematical program that contains sub-optimization problems as its constraints [11].

4) Markov Decision Process

MDP is a discrete time stochastic control process. MDP provides a mathematical modeling using decision epochs, actions, system states, transitions functions and functions rewards or cost functions.

Broadly speaking, MDP encodes the interaction between an agent and its environment where every action takes the system to a new state with a certain probability (determined by the transition functions). Choosing an action generates a reward or a cost determinate by reward function.

Policies are prescriptive of which action to take under any circumstance at every future decision epoch. The agent objective is to choose the best sequence of action (policy) under optimum criteria [12].

III. CLOUD COMPUTING CONTROL NEEDS

In this section, we review and show how the scientific community is dealing with autonomic computing to manage Clouds. Firstly, works related to the need of a full autonomic model are presented. Secondly, the need for stakeholders in our model is explained.

A. *Full autonomic model*

When Sharma [2] designs and implements a system to automate the process of deployment and reconfiguration of the cloud management system, he recognizes that capacity estimation of a distributed systems is a hard challenge. He also states that this challenge is intensified by the fact that software components behave differently in each hardware configuration.

Assuming that we cannot predict how software will perform in any particular hardware, cloud manager be dynamic enough to adapt to these differences. Despite Sharma [2] recognizing this, his approach involves only elasticity performed by nodes allocation based on SLOs, monitoring and forecast.

In [13], autonomic energy-aware mechanisms for self managing changes in the state of resources is developed to satisfy SLAs/SLOs and achieve energy efficiency. Unlike [2], this work focuses on power consumption. It also introduces a more complete model, involving not only physical machines and Virtual Machines (VMs), but expanding on it with customers and a service allocator (interface between the Cloud infrastructure and consumer).

Fitó et al. [14] propose an innovative model of self-management of Cloud environments driven by Business-Level Objectives. The aim is to ensure successful alignment between business and IT systems, extending business-driven IT [15]. In this work, typical IT events and risks during the operation of Cloud providers, such as SLAs or SLOs violations, are not dealt with.

However, Beloglazov [13] shows that many optimization techniques are contradictory. To this end, two techniques are considered: one aimed at the consolidation of VMs and increasing the amount of physical resources in cases of workload peaks; and the other at de-consolidating VMs in cases of node overheating incorporating additional constraints.

Therefore, when the presented models are implemented in ad-hoc approaches, they aim to satisfy only a few users or autonomic computing objectives. As demonstrated, in many cases the models have different granularity levels (hardware level, service levels and business goals). These models cannot be integrated naturally and as a result it is difficult to achieve full management of the Cloud environment.

Palmieri et al. [16] have presented a rich application of game theory to schedule tasks on machines in a multi user environment. They use a temporal model based on time slots to promote each agent interaction scene, but do not consider uncertainty. The game-theoretic approach supports the synergy of agents' objectives in a non-stationary way.

To improve overall system performance, Palmieri et al. [16] introduce a peer-to-peer negotiation method, without a central regulator, that influences agent decisions about its strategies. However, this model is limited by granularity of decisions. Their model is limited because it involves only tasks and schedule.

Thus, we believe that cloud computing needs a full model at the autonomic level as presented by Huebscher et al. [7]. The model is a base for decision-making. A broad, generic, and extensible model can be used with many decision-making processes and can help researchers find the best techniques.

The cloud model must be broad enough to involve all cloud components, stakeholders and their goals. Thereby, it will allow a global understanding permitting the system manager to be able to pay attention to all cloud variables and seek synergy between them. By generic we mean that it must work in any CC system. Extensible characteristic can be understood in two ways: firstly in terms of system variables, the system must deal with undefined variables; and secondly recognising that it is not a final model and specifics scenarios may require new components.

B. *Stakeholder*

The first step stated in the DP creates the necessity to obtain information about the control objectives. Autonomic computing goals are some control objectives for a CC autonomic manager. Others control objectives are relative and are different in many works, such as [17] [18] [19].

In [20], the following objectives are used for resource allocation and re-provisioning and are represented as use cases:

Acceleration: This use case explores how clouds can be used as accelerators to reduce the application time to conclude by, for example, using cloud resources to exploit an additional level of parallelism.

Conservation: This use case investigates how clouds can be used to conserve allocations, within the appropriate runtime and budget constraints.

Resilience: This use case investigates how clouds can be used to handle the unexpected..

Another example of objectives can be obtained for [13]. A high-level architecture for supporting energy-efficient service allocation in a Green Cloud is proposed. Energy-efficient service allocation is one objective in this work.

Sharma [2] presents two approaches on decisions for dynamic provisioning: cloud provider centric and customer-centric. Cloud provider centric approaches attempt to maximize revenue, while a customer centric approach attempts to minimize the cost of renting servers.

Taking into account Sharma [2], we believe that the objectives presented by Kim et al. [20] and by Beloglazov et al. [13] are relative in what concerns autonomic computing.

This relativism refers to the scope, time and user perspective, or stakeholder.

Stakeholder is a broader concept than users or actors. The term stakeholder involves not only users and cloud consumers, but it also involves the cloud itself, the cloud provider and related parties.

Thereby, we have established the following definition for management of CC as an activity of configuring manageable computational resources to meet and reconcile the interests of various stakeholders, maintaining and increasing the flow of value through the cloud over time.

Thus, we understand that what many authors call objectives, in order to have a complete management at an autonomic level, should be treated as stakeholders' interests.

IV. Proposal

In this section, we present our proposed model and his building process. Aiming to construct a cloud control model that really automates the whole system, we propose a model using as reference the mathematical modeling of Control Theory.

Resulting model of this process is the basis for decision process in CC and it supports the plan phase of MAPE-K. Essential elements of this model are: Stakeholders; Interests; Cloud state; Actions; Events; Conjuncture and Imaginaries elements. Those elements will be presented in the next sections followed by an implementation overview.

A. Essencials elements

1) The Cloud State

The cloud state is a representation of cloud in a specific moment. It represents a static view, just like photography of the Cloud domain. In Markov decision process and in control theory a state is often represented as a tuple of monitored variables and stationary set of all possible states is S. However, in CC, the set of all possible states at time t can be different at the time $t+1$ because monitored variables in a Cloud change in time, creating different sets of possible states.

The controlled variables stated at step 4 of DP are a sub set of monitored variables. Those are represented as dimensions (D_t) in our model. So D_t is the finite set of all monitored variables at time t. For example, (1) represents the resulting set of: CPU of physical machine one ($p1.c$), its memory ($p1.m$) and its state ($p1.s$); CPU of virtual machine one ($v1.c$) and its memory ($v1.m$); and the router usage ($r1.u$).

$$D_t = \{'p1.c',' p1.m',' p1.s'{,}' v1.c',' v1.m',' r1.u'\} \qquad (1)$$

The dimension index X_{dt} represents all possible values of a dimension at time t, where $X_{dt} \in \mathcal{X}_{Dt}$. The relation of D_t and $\mathcal{X}_{Dt}$ is a bijective function ($dx\colon D_t \rightarrow \mathcal{X}_{Dt}$). So $\mathcal{X}_{Dt}$ can be represented as a set of sets (2), where the first element (X_{1t}) is the index of first dimension at time t, which represents $p1.c$, line 2 is X_{2t} and represents $p1.m$, and so on. This relation is represented by function dx_t.

$$\mathcal{X}_{Dt} = \begin{Bmatrix} '\{'10\%','50\%','80\%','100\%'\}', \\ '\{'20\%',' 40\%',' 60\%',' 80\%',' 99\%'\}', \\ '\{'on','off','reboting','slepping'\}', \\ '\{'10\%','50\%','80\%','100\%'\}', \\ '\{'20\%',' 40\%',' 60\%',' 80\%',' 99\%'\}', \\ '\{'low',' high'\}' \end{Bmatrix} \qquad (2)$$

The set of possible states consists of the cartesian product of each set X_{dt} in $\mathcal{X}_{Dt}$. The consequence is that each element s_t in S_t is a tuple $(x_1, x_2, \ldots, x_n)$, where x_1 is one element of X_{1t}, x_2 is one element of X_{2t} and so on. Thus, we can represent the S_t as (3).

$$S_t = \prod_{d_t \in D_t}^{d_t} X_{d_t} \in \mathcal{X}_{D_t} \qquad (3)$$

2) Stakeholders and Interests

As explained before, ad-hoc objectives are not sufficient to deal with the CC management problem. So, in our model we use a stakeholder interests approach.

The aforementioned acceleration objective, achieved through the allocation of new VMs, is translated in our model as interest of a stakeholder in a state with new VMs. This interest could induce the allocation of more VMs. In this case we can also observe that our model can represent the interests of all involved parties, and the manager could balance the interests using Stackelberg games principles and search for a Pareto optimum or a Nash equilibrium. Allocating more VMs may be interesting for a cloud consumer; however, it can be detrimental to the whole system if, for example, the environment is already overloaded.

Economic problems are normally modeled using a utility function which represents the usefulness of something at a particular time. Extrapolating this concept, we propose an interest function v_t (5) that gives the interest of a stakeholder in a particular state at time t.

$$v_t\colon U_t \times S_t \rightarrow \gamma \qquad (4)$$

As result, (4) returns γ where γ is a real number between -1 and $+1$ $(\gamma \in \mathbb{R} \mid -1 \leq \gamma \leq 1)$. So zero represents a neutral interest, positive numbers represent real interest in a particular state and negative numbers represents a non-interest.

We also define a function du_t (5) that maps all dimensions that a stakeholder (u) can change, where U_t is set of all stakeholders at time t and $u \in U_t$.

$$du_t\colon D_t \rightarrow U_t \qquad (5)$$

3) Actions and Events

Once we have introduced the concept of stakeholders, interests and the cloud state, we present the action that allows the connection between these concepts. The stakeholders can affect and change cloud state directly, through actions, and indirectly, through their interests that are passed to the system manager and that can be translated into actions.

Control theory usually chooses a configuration to get the system to a better state. Therefore, MDPs and SMGs usually understand that an action leads to a new state. In [3], we had good adherence to management needs using MDP and actions, but we refined that model and conclude that cloud state can change in an unexpected way because of unpredicted events.

Stakeholders or the system manager can take an action and lead the system to a new desired state with a certain probability, given by function (6).

$$p_t: S_t \times A_t \times S_t \rightarrow \mathbb{R}_{[0,1]} \tag{6}$$

Events are similar to actions and can also change the cloud state. The main difference between them is that events are not planned or even carried out by a stakeholder. An event can be a hardware problem, a software failure or even a power outage, for example.

In addition, the set of all possible actions and events are not stationary, resulting in A_t and E_t. This is because some of them only make sense in some specific state. For example, the action of turning on a server only exists if the server is off at that time. The same occurs with events, a fault in software, for example, can only happen if the software is installed and running. So events and actions are related to states as:

$$s_t R a_t \subset S_t \times A_t \text{ and} \tag{7}$$
$$s_t R e_t \subset S_t \times E_t. \tag{8}$$

4) Cost fuction

Every action has a related cost. The cost implies in a reduction of a stakeholder interest. Cost function can be defined as (9).

$$c_t: S_t \times A_t \times U_t \rightarrow \mathbb{R} \tag{9}$$

5) Conjuncture and Imaginary Elements(Future)

Here, we define our concept of model conjuncture and its natural derivation, the imaginary elements.

a) Conjuncture

Conjuncture represents the system's structure at a particular time. When new structure elements are added or removed, the conjuncture changes. That is why this element is so important, as what is true in an environment that has, for example, 1 server and 2 VMs, may not be true when the environment grows and has 100 servers and 1000 VMs.

So, for the presented elements we postulate the conjuncture at time t as:

$$cj_t = (D_t, X_t, S_t, U_t, A_t, E_t, dx_t, du_t, v_{t,} c_t, p_t). \tag{10}$$

Other elements can be added to (10) because we are dealing only with essential elements.

b) Imaginary Elements

The following example demonstrates the need of imaginary elements: The environment has one cloud provider and one server. The server at workload peaks uses all available resources and satisfies the SLAs for all consumers. If we give more resources to one of the users, the SLAs will be compromised. The question is: should the system add new resources? Given this, a system manager can infer Nash equilibrium and not allocate more resources. However, a human manager, in that situation, will analyze the whole system, including business goals, and predict new cloud consumers and new demand in the future. So he could identify other needs and have a better plan.

The greatest advantage that a human manager has over autonomic management algorithms is the capacity of human beings to speculate about the future environment. So in order to develop a good plan it is necessary to choose appropriate future actions, based not only on present interests, but possible future interests that may be generated as a consequence of any of these actions.

So, our model can map future imaginary elements, supposing a new conjuncture so that the autonomic manager can take it in to account.

6) Implementation overview

The following implementation overview aims to better explain our model. In Figure 2, a class diagram depicts our proposed model.

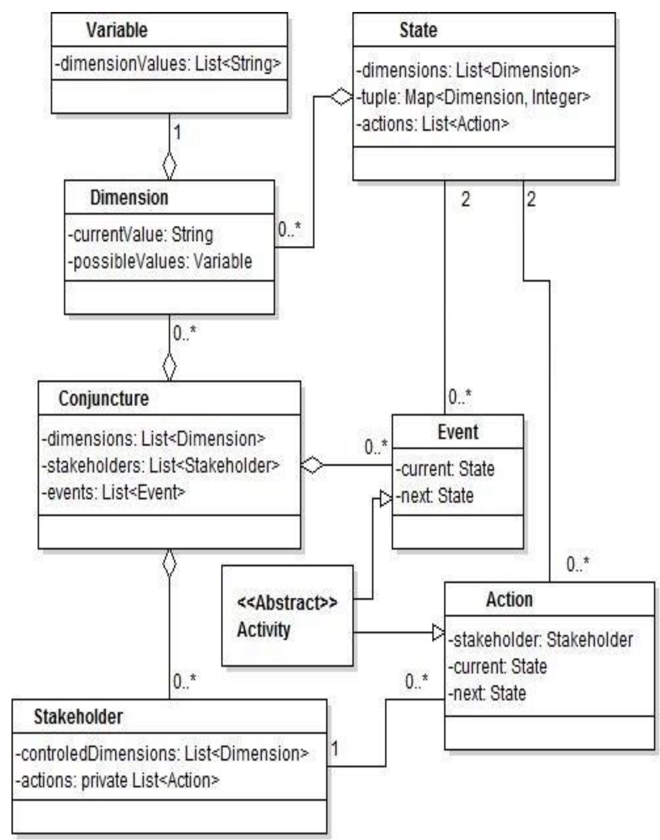

Figure 2. Class Diagram

As shown in (10), conjuncture is the system's core. It has a relationship with dimensions and their possible values, stakeholders, states, actions and events. Although conjuncture class can contain all of them, directly, it is not the only nor the best way to design the system with all elements contained in one class.

ISBN: 978-1-61208-398-8

Following Figure 2, conjuncture associates directly with events, as they come from an unknown source. Also, it must contain stakeholders, which define a set of controlled dimensions and their actions. Finally, it maps states, indirectly, using all the dimensions from the monitored environment, considering possible states as an aggregate of dimensions. Consequentially, all states can be generated from arranged combinations of possible values in every dimension.

With all sets of components defined, half of the system is modeled. However, the functions, as previously described, by (4), (5), (6) and (9) are not yet defined.

V. CONCLUSION

Based on the view of CC presented, it was possible to base the management model for decision-making on a perspective of public utility management and not only on a data center management perspective.

The presented model gives a solid mathematical base to research political behaviors of CC. Also, using the formalisms that were researched, this work introduced CC management as a multi-player game with high level objectives (Pareto optimal and Nash equilibrium) and presented holistic interests independent of CC architecture or implementation.

Finally, this work presented a new concept of “imagination”, essential for a human-like CC management.

For future work CloudSim will be extended to simulate and validate the proposed model and to compare the results with other solutions. CloudSim is a framework to simulation of emerging CC infrastructures and management services.

Following, a multi-strategy approach will be developed. Using Nash equilibrium, Pareto optima, max satisfaction and others in the simulator will be able to choose the best one to implement.

At least, possibilities for future research are:

- Implement a pilot of proposed model using results obtained from simulation;
- Improve the model, if necessary;
- Extend the pilot.

REFERENCES

[1] J. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41-50, 2003.

[2] U. Sharma, "Elastic resource management in cloud computing platforms," University of Massachusetts, Ph.D. thesis 2013.

[3] R. Mendes et al., "Decision-Theoretic Planning for Cloud Computing," in *The Thirteenth International Conference on Networks*, 2014, pp. 191-197.

[4] I. Foster, Y. Zhao, I. Raicu, and L. Shiyong, "Cloud Computing and Grid Computing 360-Degree Compared," in *Grid Computing Environments Workshop*, Austin, TX, 2008, pp. 1-10.

[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandicc, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599-616, 2009.

[6] P. Mell and T. Grance, "SP 800-145. The NIST Definition of Cloud Computing," National Institute of Standards, Gaithersburg, MD, United States, 2011.

[7] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing - degrees, models, and applications," *ACM Computing Surveys (CSUR)*, vol. 40, no. 3, p. 7, 2008.

[8] S. Skogestad and I. Postlethwaite, *Multivariable feedback control Analysis amd Design*, Wiley New York, Ed., 2007, vol. 2.

[9] D. Lozovanu and S. Pickl, *Optimization and Multiobjective Control of Time-Discrete Systems*.: Springer, 2009.

[10] J. Nash, "Non-cooperative games," in *Annals of mathematics*, 1951, pp. 286-295.

[11] Y. Liu, Y. Ji, and R. J. Jiao, "A Stackelberg Solution to Joint Optimization Problems: A Case Study of Green Design," *Procedia Computer Science*, vol. 16, pp. 333-342, 2013.

[12] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*.: John Wiley & Sons, 2009.

[13] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristicsfor efficient management of data centers for Cloudcomputing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768, 2012.

[14] J. O. Fitó, M. Macías, F. Julia, and J. Guitart, "Business-Driven IT Management for Cloud Computing Providers," *CloudCom*, pp. 193-200, 2012.

[15] J. Sauvé, A. Moura, M. Sampaio, J. Jornada, and E Radziuk, "An Introductory Overview and Survey of Business-driven," in *Business-Driven IT Management, 2006. BDIM'06. The First IEEE/IFIP International Workshop on*, 2006, pp. 1-10.

[16] F. Palmieri, L. Buonanno, S. Venticinque, R. Aversa, and B. Di Martino, "A distributed scheduling framework based on selfish autonomous agents for federated cloud environments," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1461-1472, 2013.

[17] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *The Journal of Supercomputing*, vol. 54, no. 2, pp. 252-269, 2010.

[18] A. Corradi, M. Fanelli, and L. Foschini, "VM consolidation: a real case based on openstack cloud," *Future Generation Computer Systems*, pp. 118-127, 2014.

[19] D. Loreti and A. Ciampolini, "Policy for Distributed Self-Organizing Infrastructure Management in Cloud Datacenters," , 2014, pp. 37-43.

[20] H. Kim, Y. El-Khamra, I. Rodero, S. Jha, and M. Parashar, "Autonomic Management of Application Workflow on Hybrid Computing Infrastructure," *Scientific Programming*, vol. 19, no. 2, pp. 75-89, 2011.

[21] Y. O. Yazir et al., "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," in *IEEE 3rd International Conference on*, 2010, pp. 91-98.

 ISBN: 978-1-61208-398-8