Journal of Data Analysis and Information Processing, 2017, 5, 87-95
http://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Subject Oriented Autonomic Cloud Data Center Networks Model

Hang Qin[1], **Li Zhu**[2*]

[1]Computer School, Yangtze University, Jingzhou, China
[2]Oujiang College, Wenzhou University, Chashan University, Wenzhou, China
Email: 68681700@qq.com, *yeah_1397118@hotmail.com

How to cite this paper: Qin, H. and Zhu, L. (2017) Subject Oriented Autonomic Cloud Data Center Networks Model. *Journal of Data Analysis and Information Processing*, **5**, 87-95.
https://doi.org/10.4236/jdaip.2017.53007

Received: June 2, 2017
Accepted: July 22, 2017
Published: July 25, 2017

Abstract

This paper investigates autonomic cloud data center networks, which is the solution with the increasingly complex computing environment, in terms of the management and cost issues to meet users' growing demand. The virtualized cloud networking is to provide a plethora of rich online applications, including self-configuration, self-healing, self-optimization and self-protection. In addition, we draw on the intelligent subject and multi-agent system, concerning system model, strategy, autonomic cloud computing, involving independent computing system development and implementation. Then, combining the architecture with the autonomous unit, we propose the MCDN (Model of Autonomic Cloud Data Center Networks). This model can define intelligent state, elaborate the composition structure, and complete life cycle. Finally, our proposed public infrastructure can be provided with the autonomous unit in the supported interaction model.

Keywords

Autonomic Cloud Computing, Autonomous Unit, Data Center, Self-Configuration, Service Description

1. Introduction

Autonomic computing concentrates on the self-management characteristics of distributed computing resources, in light of unpredictable changes for hiding intrinsic complexity to operators and users [1] [2]. Meanwhile, data center is a facility used to house computer systems and associated components, such as telecommunications and storage systems. Consequently, cloud data center networks includes redundant or backup power supplies, redundant data communication connections, environmental controls and various security devices. A large data center is the industrial-scale operation used as much electricity as a small town for growing complexity of self-management [2] [3] [4].

This paper proposes a service-oriented autonomic cloud computing self-configuration method, *i.e.*, MCDN (Model of Autonomic Cloud Data Center Networks). It divides the self-configuration into structure self-configuration and interface self-configuration. Consequently, a dynamic contract network protocol is proposed to solve the self-optimization problem of autonomic cloud computing structure, concerning reference group intelligence [5] [6]. We introduce the autonomic in the traditional contract networks, and develop a dynamic contract network protocol under the algorithm description. The protocol can take full advantage of the previous service matching process of autonomous units under the dynamic environment in autonomic cloud data center networks [7] [8].

With the modeling process of autonomic cloud data center networks, a system of autonomic cloud computing is built based on component reuse development process, which is divided into reusable entity layer, autonomous behavior layer, autonomous unit layer and autonomy social layer [9] [10]. Autonomous unit function development and interactive development organic separation are willing to improve autonomous unit function reusability. In the multi-agent environment, we have developed an autonomous calculation of the prototype system with the hierarchical development, which can provide a good support for the rapid development of autonomic cloud data center networks [11] [12].

2. Intelligent Subject in the Cloud

2.1. Intelligent Subject

Intelligent subject and multi-agent system in the subject theory, multi-agent architecture, with the main body of the software engineering and other aspects have made some research results [13]. The main body can be used to refer to a particular environment with autonomy, initiative, social ability, responsiveness and other special hardware or software entity.

The reason why the subject's research is so important is that it can meet the access to the information highway on behalf of people, to solve the information overload and the global scope of the new distributed computing model needs. The main structure of the problem to be solved is what the module by the composition, how to exchange information between them, how the known information affects its behavior and its internal state, and how these modules are grouped in software/hardware together to form an organic whole of the main body.

2.2. Autonomous Calculation

The complexity of the computer software system structure and the computer organization structure is increasing. However, it is that the complexity of the computer system itself has become an urgent need to solve the problem.

The autonomy is inspired by the complex autonomic nervous system, which predicts the system's need in the same way. What is going to be solved is the increasingly complex computing environment in the management and cost problems. The important difference between autonomic cloud data center networks and the autonomic nervous system of the human body is that many of the autonomy decisions made by the body are unconsciously, and the autonomic cloud computing components of the computer system follow the human make. We build the framework for automatic performance optimization for component-based enterprise applications, as shown in Figure 1.

Achieving autonomic cloud computing is an evolutionary process, which can be divided into five levels with its autonomy maturity:

1) *Basic level*: The professionals manage each element of the system separately, install, monitor, maintain, uninstall and so on.

2) *Management level*: System management technology used to collect the scattered system information collected to the console, thus saving the management, in terms of the time required for the information to be collected and synthesized.

3) *Forecast level*: The use of new technology has a number of elements associated with the optimal allocation, and gives the administrator providing decision-making advice.

4) *Autonomous level*: The administrator need release business strategy and objectives. Administrator can monitor business processes and change business strategies.

2.2.1. Self-Optimization

By focusing on the hotspots of the system, these modules can optimize their system overhead, making the framework suitable for long-running systems.

The self-improvement is different from the traditional optimization method, while the traditional optimization method based on system metrics, such as resources availability. According to business objectives, we can get business rules and SLA about how metrics affect business objectives. Once the business rules and service level agreements are defined, the business purpose is defined.

We identify how to design adaptive middleware to support autonomic cloud computing in a pervasive computing environment. The focus of the study is

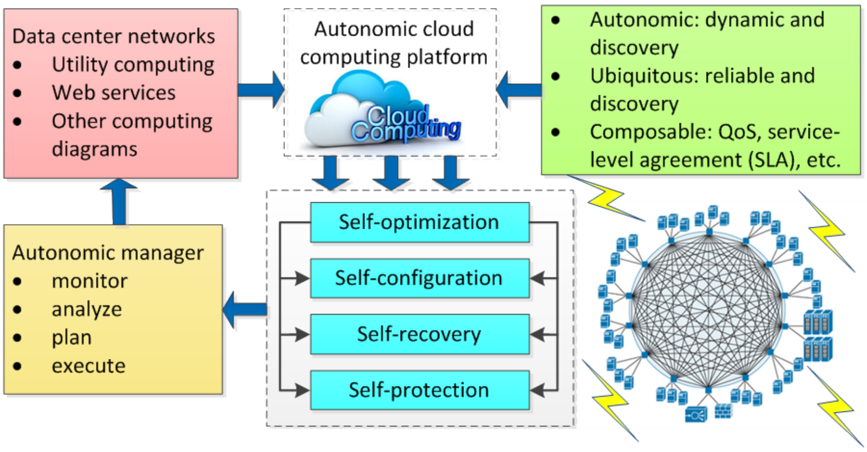
Data center networks
• Utility computing
• Web services
• Other computing diagrams
Autonomic cloud computing platform
Cloud Computing
• Autonomic: dynamic and discovery
• Ubiquitous: reliable and discovery
• Composable: QoS, service-level agreement (SLA), etc.
Autonomic manager
• monitor
• analyze
• plan
• execute
Self-optimization
Self-configuration
Self-recovery
Self-protection

Figure 1. Autonomic computing in MCDN over data center to the Internet.

when the mobile users in the wireless network environment with different devices, how to optimize the network changes connectivity. The goal is to make this change the best, and the application code for the core should be transparent.

2.2.2. Self-Configuration

There is a framework for the deployment of component-based distributed applications and the implementation of self-management. The initial deployment target is represented by a declarative binding language, which can express the component and host the mapping between the components can also be used to express the interconnection between components of the topology. A constraint solver is used to find the configuration that satisfies the target and is automatically deployed.

In the construction of the system or change the system configuration, the operator is often easy to make some mistakes. It describes some algorithms that allow the system to ensure robustness and stability when initializing and changing configurations. They set out a new optimization framework to capture the specific aspects of the system reconfiguration. In a distributed or mobile environment, the connection among sites where the software system is running is often unstable. This method allows the system to monitor connections between nodes when connected self-configuring and self-deploying when lost, increasing the availability of the entire system.

2.2.3. Self-Recovery

We use dynamic meta-programming to insert transactional recovery code at runtime system. Because this recovery method does not require programmers to intervene, but the dynamic conversion code does, it can greatly simplify the recovery design and implementation of complex systems. Consequently, we use decision tree learning methods to diagnose large site failures. The method records the run-time attributes of each request, and uses machine learning and data mining techniques to identify the cause of the failure.

The path followed the fault degree of association is given to different levels, and the nodes are merged according to the observed partial ordering of the system components. The method has a lot of request and run-time information, so as to make statistical analysis meaningful. The systems implemented in this way enable the network to be effective after a failure Restore the network connection.

The recovery simplifies the management in two ways. The first is to reduce the recovery line taken by the recovery the price required for the move, which allows the user to use statistical techniques to fail to capture failures. The second is to simplify the planning and other steps required for troubleshooting.

2.2.4. Self-Protection

A self-management system can continue to monitor the own incorrect configuration, to prevent the system from being hurt. Although there are many constraints that can distinguish the correct configuration from the wrong configuration, these constraints are rarely recorded down, let alone be effectively used to organize.

For a wide range of network attacks, we propose an on-line monitoring and analysis framework for self-protection. They use the software main body to monitor some of the properties of the system and put all the network and computing resources. The software body is to implement the corresponding recovery mechanism. As the peak demand for subscribers is not easy to predict, the current department demand for resources changes. When there is an unpredictable load peak, the system has time to implement the resources. Dynamic peak load protection is used for three technologies: *adaptive short term forecasting*, *online capacity planning capacity planning* and *configuration management*.

3. Autonomic Cloud Data Center Networks Model Framework

3.1. Autonomous Units

We define the autonomous unit as managing the behavior according to the user's policy and interacting with other autonomous units provide a certain service, *i.e.*, computing power, resources, knowledge. Autonomous units can collect information from the environment, make decisions, and root according to the need to adjust their own, making their own self-configuration, self-repair, self-optimization, self-protection capabilities. But the autonomous unit and the main body also have important differences:

1) *Different functions*: The main body is in certain environment autonomy of the computing entity, in addition, the autonomous unit must also be able to self-configuration, self-optimization, self-repair and self-protection.

2) *Different structure*: In order to meet the needs of self-management, autonomous units must have a strategy management, complexity analysis and other general body does not have the internal structure.

3) *Different life cycle*: Autonomous unit has a more complex than the main life cycle.

3.2. Functional Structure

We divide the structure of the autonomous unit into four parts of the shared knowledge function:

1) *Monitoring*: It is about collection, filtering, management, and reporting information. The above resources in the system should have a unified information source model, making information monitoring and collection easier.

2) *Analysis*: The complexity lies in the current environment analysis, construction system model. This makes autonomous units available to environment to learn, and to help the autonomous unit to predict the future.

3) *Planning*: In accordance with the strategy and the agreement, it is to build the target sequence of behavior.

4) *Implementation*: It is about the implementation of the planning in the sequence behavior, and the implementation of process control management.

3.3. Autonomous Unit Mental State

3.3.1. Mental State Model

The belief of the autonomous unit is the most basic element in the mental state of the autonomous unit and the most important element. The state of mind for reasoning and other related issues are based on faith, must rely on faith. The main unit's belief can often be seen as autonomous knowledge bases, which contain rich content, including basic rational, autonomous units familiar with the field of justice, autonomous units in the field of objective facts and data understanding.

The goal reflects the state or behavior to be achieved, usually because the autonomous unit receives a certain strategy, or because of the autonomy of the unit's own beliefs and standards arising from the goal. So the goal is more with the belief of the autonomous unit, which comes from the outside world and the autonomy of the unit faith. The plan actually describes the autonomy unit belief, the autonomy unit behavioral capability, and the autonomous unit organically linked together, indicating that in a certain state of faith, in order to achieve a certain goal, what kind of behavior will be taken.

3.3.2. Behavioral Competencies

Generally, the capacity of an autonomous unit refers to the ability of the autonomous unit to perform certain actions and to perform certain tasks or function, corresponding to a certain behavioral entity, *i.e.*, functional modules. Thus, the description of the capacity of the autonomous unit actually corresponds to the description of the action in the dynamic description logic, indicates what type of action the autonomous unit can perform and, on the other hand, points out what is going on what kind of results. The representation of the capacity of the autonomous unit is the essential part of the belief for the autonomous unit, but it is more involved in the action of the autonomous unit and the state of change.

In the traditional theory of action and logic programming, generally do not specify the action of the operation of the object. The axiom of action in dynamic description logic includes two aspects, one is to specify under what conditions an action is enforceable over the premise of axiom, the other is to specify a move what will happen when executed in a certain state over the successor state of justice.

3.3.3. Behavioral Planning

When there is a certain goal, the autonomous unit must find an effective way to achieve these goals standard under behavioral planning, and sometimes need to modify existing goals with the reasoning process. In order to complete this regulation planning reasoning, autonomous units can be used in two ways: *static planning* and *dynamic planning*.

Dynamic planning is for a certain goal, autonomous units based on their current state of faith, combined with the autonomy of the unit with the domain of justice and their ability to describe, to find an effective way to achieve goals and ways. Combined with the strategy target planning problem, we can give static planning algorithm and dynamic programming algorithm.

3.4. Composition of the Structure

In order to enable the autonomous unit having the function given and implement the mental state model, autonomous unit should have what kind of internal structure. It is considered when defining the structure of autonomous units the following points:

1) *The versatility of the autonomous unit structure*: The definition of the internal structure of the autonomous unit is universal on their own structure needs.

2) *The autonomy of the composition of the structure of the scalability*: The structural scalability makes the autonomous unit can be easily and flexible insert new of the functional modules, without destroying the autonomous unit should have the characteristics.

3) *The composition of the structure of the autonomy of the unit interoperability support*: The interoperability between autonomous units is the autonomous unit, since the main computing system has the necessary prerequisite for self-management characteristics.

Consequently, we define the structure of the autonomous unit as shown in Figure 2.

- *Sensor*: The autonomous unit senses the external environment through the sensor and collects the required information, in light of the autonomous unit through the function module interface on the external environment.
- *Communicator:* In order to improve the interoperability between autonomous units, the single the communication between the elements are based on the mode of message passing.
- *Model construction:* the complexity of the current environment analysis, modeling system model.
- *Planning scheduler*: A system model based on the model construction module, it can be used to achieve the goal according to the policy plan, concerning the behavior of the implementation of scheduling operations.
- *Coordination engine*: The coordination of the interaction between the various modules, communication and collaboration, maintenance of autonomous unit daily operation, to ensure that main unit is always running.

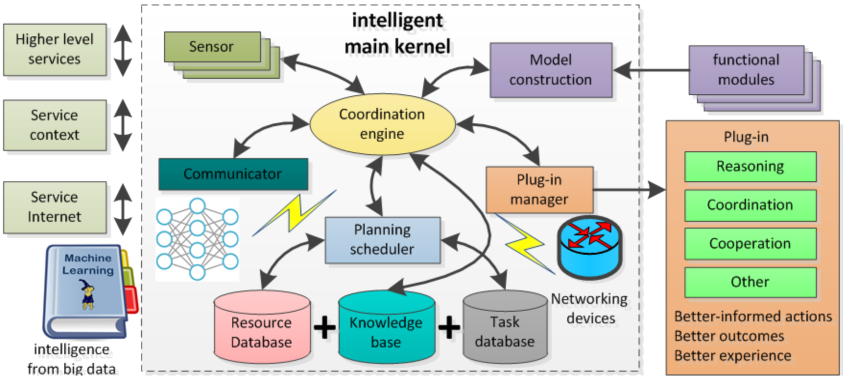

Figure 2. Architecture of autonomous unit over intelligence from big data.

- *Resource database*: It saves the list of autonomous units owned and available resources.

4. Conclusion

This paper presents functional structure, mental state, service description and composition structure, in terms of constraints made for autonomous units. When an autonomous unit satisfies the above issues, it can be presented as a certain self-management characteristic. This paper also develops the MCDN (Model of Autonomic Cloud Data Center Networks) architecture, in terms of the analysis of the autonomous unit under the interactive mode. Finally, the system of independent computing side surface has made a useful attempt, given the framework of the autonomic cloud data center networks model for the standardization of independent work to achieve independent software system.

References

[1] Rittinghouse, J.W. and Ransome, J.F. (2016) Cloud Computing: Implementation, Management, and Security. CRC Press, Boca Raton.

[2] Whaiduzzaman, M., *et al.* (2014) A Survey on Vehicular Cloud Computing. *Journal of Network and Computer Applications*, **40**, 325-344.

[3] Khalil, I.M., Khreishah, A. and Azeem, M. (2014) Cloud Computing Security: A Survey. *Computers*, **3**, 1-35. https://doi.org/10.3390/computers3010001

[4] Gerla, M., *et al.* (2014) Internet of Vehicles: From Intelligent Grid to Autonomous Cars and Vehicular Clouds. 2014 *IEEE World Forum on Internet of Things*, New York, March 2014, 241-246. https://doi.org/10.1109/WF-IoT.2014.6803166

[5] Bitam, S., Mellouk, A. and Zeadally, S. (2015) VANET-Cloud: A Generic Cloud Computing Model for Vehicular Ad Hoc Networks. *IEEE Wireless Communications*, **22**, 96-102. https://doi.org/10.1109/MWC.2015.7054724

[6] Toosi, A.N., Calheiros, R.N. and Buyya, R. (2014) Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey. *ACM Computing Surveys*, **47**, 7.

[7] Avram, M.-G. (2014) Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective. *Procedia Technology*, **12**, 529-534.

[8] Yan, Q., *et al.* (2016) Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges. *IEEE Communications Surveys & Tutorials*, **18**, 602-622. https://doi.org/10.1109/COMST.2015.2487361

[9] Hwang, K., Dongarra, J. and Fox, G.C. (2013) Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Morgan Kaufmann.

[10] Zhan, Z., *et al.* (2015) Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches. *ACM Computing Surveys*, **47**, 63.

[11] Albert, G., *et al.* (2008) The Cost of a Cloud: Research Problems in Data Center Networks. *ACM SIGCOMM Computer Communication Review*, **39**, 68-73. https://doi.org/10.1145/1496091.1496103

[12] Dastjerdi, A.V. and Buyya, R. (2015) An Autonomous Time-Dependent SLA Negotiation Strategy for Cloud Computing. *The Computer Journal*, **58**, 3202-3216. https://doi.org/10.1093/comjnl/bxv053

[13] Gomer, R. and Gerding, E. (2014) Consenting Agents: Semi-Autonomous Interactions for Ubiquitous Consent. *Proceedings of the* 2014 *ACM International Joint Conference on Pervasive and Ubiquitous Computing*: *Adjunct Publication*, 13-17 September 2014, 653-658. https://doi.org/10.1145/2638728.2641682