International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-1 Number-1 Issue-1 September 2011

A Novel Cloud Computing Mapping and Management through Class and Object Hierarchy (CCMMCOH)

Naveen Gurjar[1], Ravindra Gupta[2]
M.Tech Scholar, Dept. of Computer Science, SSSIST Sehore, India[1]
Professor in CS Department at SSSIST Sehore[2]

Abstract

The role of data analytics increases in several application domains to cope with the large amount of captured data. Cloud computing has become one of the key considerations both in academia and industry. Cheap, seemingly unlimited computing resources that can be allocated almost instantaneously and pay-as-you-go pricing schemes are some of the reasons for the success of Cloud computing .In this paper we discuss few aspects of cloud computing and also there area. We also propose a novel approach which is cloud computing mapping and management through class and object hierarchy. In this approach we first design a cloud environment where we can analyze several object oriented aspects based on some assumptions. Then we deduce message passing behavior through a backup files based on the properties of object orient like class and object.

Keywords

Cloud Computing, OOP, Cloud Environment, Class and Object

1. Introduction

Cloud computing has become one of the most important evolutions computer science has seen recently. The success for Clouds can be attributed to the ability to provide seemingly unlimited computing resources almost instantaneously and also to the pay-per-use pricing schemes.

The applications designers need to be cognizant of the increasingly becoming virtualized and cloud oriented data center during the application design and architecture activities.

Given a wide and increasing range of virtualization based and cloud oriented platforms and frameworks, it is not always clear that in what different ways can an application be designed to meet the desired application and how to choose the most suitable platform. Cloud computing is concerned with offering and using abstracted IT infrastructures in networked environments, typically the internet. It is commonly agreed that three layers of abstraction exist [1]:

Software as a Service (SaaS), Platform as a Service (PaaS) Infrastructure as a Service (IaaS).While SaaS providers typically run web based applications that clients may access using their web browser and pay for them via subscriptions, PaaS and IaaS are less focused on end users. PaaS is tailored towards companies that want to offer services to customers, but do not want to host the services ontheir own. In cloud computing, each application is often designed a business process which includes a set of abstract services. Each abstract service encapsulates the function of an application component using its interface, and a concrete service or resources is selected (bound) at runtime to fulfill the function.

A Service Level Agreement (SLA) in cloud computing is defined upon a business process as its end-to-end. Quality of Service (QoS) constraints since a business process defines how abstract services interact to accomplish a certain business goal. Since different concrete services may operate at different QoS measures, an appropriate/optimal set of concrete services/resources may be selected so that it guarantees the fulfillment of SLA and cost is minimal. Such problem, the QoS-aware service composition problem, is a combinatorial optimization problem which ensures the optimal mapping between each abstract service and available resources [2], [3]. Since the problem is known as NP-hard [4], it takes a significant amount of time and costs to find optimal solutions (optimal combinations of resources) from a huge number of possible solutions.

The remaining of this paper is organized as follows. We discuss Cloud Computing Environment in Section 2. In Section 3 we discuss about Object Oriented Concepts. In section 4 we discuss about Recent Scenario. In section 5 we discuss about the proposed method. The conclusions are given in Section 6. Finally references are given.

2. Cloud Computing Environment

According to the Microsoft cloud is a metaphor for the internet. It is quite common these days to draw network diagrams that depict the Internet as a cloud hence the use of the word in this instance.

In a typical cloud computing scenario organizations run their applications from a data centre provided by

a third-party – the cloud provider. The provider is responsible for providing the infrastructure, servers, storage and networking necessary to ensure the availability and scalability of the applications. This is what most people mean when they refer to cloud computing i.e. a public cloud.

There is also much talk about private clouds. A private cloud is a proprietary computing architecture, owned or leased by a single organization, which provides hosted services behind a firewall to "customers" within the organization. Some commentators regard the term "private cloud" as an oxymoron. They say that the word "cloud" implies an infrastructure running over the Internet, not one hidden behind a corporate firewall. There is, however, a larger body of opinion suggesting that private clouds will be the route chosen by many large enterprises and that there will be substantial investment in this area. Already vendors are lining up to release products that will enable enterprises to more easily offer internal cloud services.

According to the Microsoft there are several advantages and disadvantages like:

There is a huge amount of hype surrounding cloud computing but despite this more and more C-level executives and IT decision makers agree that it is a real technology option. It has moved from futuristic technology to a commercially viable alternative to running applications in-house. Vendor organisations such as Amazon, Google, Microsoft and Salesforce.com have invested many millions in setting up cloud computing platforms that they can offer out to 3rd parties. They clearly see a big future for cloud computing. Of course, no technology comes without a set of advantages and disadvantages so we've tried to sort to wheat from the chaff when it comes to the reality of cloud computing. In particular, one always has to be cautious in believing the claims of any specific vendor.

There are several reasons to adopt cloud computing like Cost, Scalability, Business Agility, and Disaster Recovery.
Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.
This cloud model promotes availability and is composed of:

[1] five essential characteristics:
1. On-demand self-service;
2. Broad network access;
3. Resource pooling;
4. Rapid elasticity;
5. Measured Service;

[2] three service models:
1. Cloud Software as a Service (SaaS);
2. Cloud Platform as a Service (PaaS);
3. Cloud Infrastructure as a Service (IaaS); and,

[3] four deployment models:
1. Private cloud;
2. Community cloud;
3. Public cloud;
4. Hybrid cloud.

[4] Key enabling technologies include:
1. fast wide-area networks;
2. powerful, inexpensive server computers; and
3. High-performance virtualization for commodity hardware."

In general, a public (external) cloud is an environment that exists outside a company's firewall. It can be a service offered by a third-party vendor. It could also be referred to as a shared or multi-tenanted, virtualized infrastructure managed by means of a self-service portal.

A private (internal) cloud reproduces the delivery models of a public cloud and does so behind a firewall for the exclusive benefit of an organization and its customers. The self-service management interface is still in place while the IT infrastructure resources being collected are internal.

In a hybrid cloud environment, external services are leveraged to extend or supplement an internal cloud.

Diagrammatically, the three (3) types of cloud computing offerings can be depicted as:

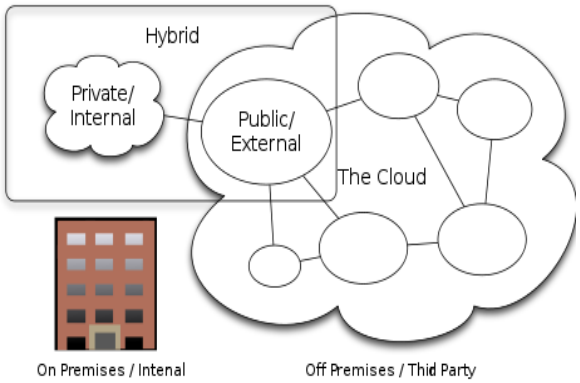

Fig1. Cloud Computing

3. Object Oriented Concepts

Simple, non-OOP programs may be one "long" list of statements (or commands). More complex programs

will often group smaller sections of these statements into functions or subroutines each of which might perform a particular task. With designs of this sort, it is common for some of the program's data to be 'global', i.e. accessible from any part of the program. As programs grow in size, allowing any function to modify any piece of data means that bugs can have wide-reaching effects.

In contrast, the object-oriented approach encourages the programmer to place data where it is not directly accessible by the rest of the program. Instead, the data is accessed by calling specially written functions, commonly called methods, which are either bundled in with the data or inherited from "class objects." These functions act as the intermediaries for retrieving or modifying the data they control. The programming construct that combines data with a set of methods for accessing and managing those data is called an object. The practice of using subroutines to examine or modify certain kinds of data, however, was also quite commonly used in non-OOP modular programming, well before the widespread use of object-oriented programming.

An object-oriented program will usually contain different types of objects, each type corresponding to a particular kind of complex data to be managed or perhaps to a real-world object or concept such as a bank account, a hockey player, or a bulldozer. A program might well contain multiple copies of each type of object, one for each of the real-world objects the program is dealing with. For instance, there could be one bank account object for each real-world account at a particular bank. Each copy of the bank account object would be alike in the methods it offers for manipulating or reading its data, but the data inside each object would differ reflecting the different history of each account.

Objects can be thought of as wrapping their data within a set of functions designed to ensure that the data are used appropriately, and to assist in that use. The object's methods will typically include checks and safeguards that are specific to the types of data the object contains. An object can also offer simple-to-use, standardized methods for performing particular operations on its data, while concealing the specifics of how those tasks are accomplished. In this way alterations can be made to the internal structure or methods of an object without requiring that the rest of the program be modified. This approach can also be used to offer standardized methods across different types of objects. As an example, several different types of objects might offer print methods. Each type of object might implement that print method in a different way, reflecting the different kinds of data each contains, but all the different print methods might be called in the same standardized manner from elsewhere in the program. These features become especially useful when more than one programmer is contributing code to a project or when the goal is to reuse code between projects.

Object-oriented programming has roots that can be traced to the 1960s. As hardware and software became increasingly complex, manageability often became a concern. Researchers studied ways to maintain software quality and developed object-oriented programming in part to address common problems by strongly emphasizing discrete, reusable units of programming logic. The technology focuses on data rather than processes, with programs composed of self-sufficient modules ("classes"), each instance of which ("objects") contains all the information needed to manipulate its own data structure ("members"). This is in contrast to the existing modular programming that had been dominant for many years that focused on the function of a module, rather than specifically the data, but equally provided for code reuse, and self-sufficient reusable units of programming logic, enabling collaboration through the use of linked modules (subroutines). This more conventional approach, which still persists, tends to consider data and behavior separately.There are several properties of object oriented programming by which we can club with the cloud computing to perform better scalability and enhancements. The Properties are shown in Fig 2. Perhaps the most significant characteristic of object-oriented database technology is that it combines object-oriented programming with database technology to provide an integrated application development system. There are many advantages to including the definition of operations with the definition of data. First, the defined operations apply ubiquitously and are not dependent on the particular database application running at the moment. Second, the data types can be extended to support complex data such as multi-media by defining new object classes that have operations to support the new kinds of information.

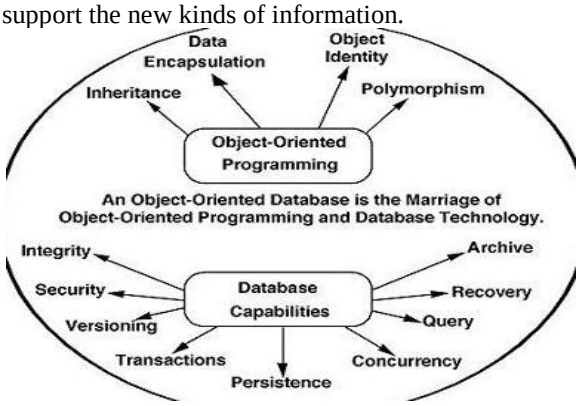

Fig2. Object Oriented Programming

4. Recent Scenario

In 2010, Chia-Feng, Lin et al. [5] proposed about Web Services Distributed Management (WSDM) which is one of the industry standards. However, to implement the WSDM interfaces needs to understand server Web service standards. It increases the complexity and difficulty to build the management system. They simplified the Web service management effort between services using hook technology. Our management systems provide message flow oriented management atomically without modifying service code. Enterprise can control all flows and review them at any time.

In 2010, Hong Zhou and Hongji Yang [6] proposed a novel approach to reengineering enterprise software for cloud computing by building ontology for enterprise software and then partitioning the enterprise software ontology to decompose enterprise software into potential service candidates. Ontology development process includes three steps, namely, building ontologies for source code, data, and application framework respectively, integrating captured ontologies and deploying the final produced ontology.

In 2010, G. Hughes et al. [7] proposed about continues, to describe the structure and operation of an object mapping declarative language and the object oriented system which employs it. Both are currently under development to support the management of these numerous Cloud Computing components. The ultimate aim is to develop a system that combines the rich capability of an imperative assembly with the concise simplicity of a declarative language.

In 2010, Xing Chen et al. [8] describe and construct the Internetware Cloud which focus on middleware management and investigate the reusability of the basic management operations and management processes in the MaaS solution.

In 2010, Chia-Feng,Lin et al. [9] analyze the requirements of access protocols for storage systems based on data partitioning schemes in widely distributed cloud environments. They consider the regular semantics instead of atomic semantics to improve access efficiency. Then, we develop an access protocol following the requirements to achieve correct and efficient data accesses. Various protocols are compared experimentally and the results show that our protocol yields much better performance than the existing ones.

5. Proposed Method

In this paper we proposed a novel Cloud Computing Mapping and Management through Class and Object Hierarchy (CCMMCOH) by using java. In this approach we mainly concentrate on four phases.

1. Interconnectivity
2. Security
3. Resource Sharing
4. OOP Mapping

Phase –I: We develop an environment based on cloud computing. There are two areas in our cloud environment first is cloud area for cloud computation, second is database area for resource sharing. We interconnect the database area and the cloud computing environment for the transactions and also as the backup mechanism. Internally we applied some SQL query for performing some changes and inclusion of data in future. Because if we need some addition future we can update the attribute by alter command and insert the values by insert command.

Phase II: For using the database we use a UA (User Authentication) ID which is generated by the cloud environment and that ID is used for the authentication for intercommunication.

Phase III: Resources are shared among the two environments. Resources are UML data and set of values.

Phase IV: The shared attributes are a series of allowable performance quotients for each noteworthy hardware, software and networking resource. So we can add the shared attributes in the appropriate place. The final step in the periodic analysis process is the Action performed in response to the triggering of an Event.

6. Conclusion

We discuss several aspects of cloud computing including the advantages and disadvantages. We also analyze several asymptotic behavior of cloud computing according with several analyses.

In this paper we discuss few aspects of cloud computing and also there area. We also propose a novel approach which is cloud computing mapping and management through class and object hierarchy. In this approach we first design a cloud environment where we can analyze several object oriented aspects based on some assumptions. Then we deduce message passing behavior through a backup files based on the properties of object orient like class and object.

References

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud Computing and Grid Computing 360-degree compared. In Proc. of Grid Computing Environments Workshop, pages 256–265. IEEE, Jan 2008.

[2] J. Anselmi, D. Ardagna, and P. Cremonesi,A QoS-based Selection Approach of Autonomic Grid Services. In ACM High Performance Distributed Computing, Workshop on Service-Oriented Computing Performance, June 2007.

[3] V. Basili, L. Briand, and W. Melo, "A Validation of Object-Oriented Design Metrics as Quality Indicators," IEEE Trans. Software Eng., vol. 22, no. 10, pp. 751-761, 1996.

[4] Mohsen D. Ghassemi and Ronald R. Mourant,"Evaluation of Coupling in the Context of Java Interfaces", Proceedings OOPSLA 2000.

[5] Chia-Feng,Lin, Ruey-Shyang Wu, Shyan-Ming Yuan , Ching-Tsorng Tsai," A Web Services Status Monitoring Technology for Distributed System Management in the Cloud", 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.

[6] Hong Zhou, Andrew Hugill and Hongji Yang, "An Ontology-Based Approach to Reengineering Enterprise Software for Cloud Computing", 2010 IEEE 34th Annual Computer Software and Applications Conference.

[7] G. Hughes, D. Al-Jumeily & A. Hussain," Supporting Cloud Computing Management through an Object Mapping Declarative Language", 2010 Developments in E-systems Engineering.

[8] Xing Chen, Xuanzhe Liu, Fuzhi Fang, Xiaodong Zhang, Gang Huang," Management as a Service: An Empirical Case Study in the Internetware Cloud", IEEE International Conference on E-Business Engineering.

[9] Ye, Yunqi, Liangliang Xiao, I-Ling Yen, and Farokh Bastani. "Secure, dependable, and high performance cloud storage." In Reliable Distributed Systems, 2010 29th IEEE Symposium on, pp. 194-203. IEEE, 2010.