ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

LOAD BALANCING WITH MULTIPLE CLOUD SERVICES USING PSO TECHNIQUES

G.Gayathri[1], E.Nandakumar M.E.,(Ph.D)[2] M.Arulprakash M.Tech.,

[1]*Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani*

[23]*Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani*

[1]*gayathrice12@gmail.com*

Abstract : *The cloud computing refers to storing of data from the server without using any hardware and software device.in past decades, significant attention has been devoted to the task allocation and load balancing in distributing systems. Although there have been some related surveys about this subject, each of which only made a very preliminary review on the state of arts of single type of distributed systems. To correlate the studies in varying types of distributed systems and make a comprehensive studies on task allocation and load balancing according to the general characteristics of distributed systems. Based on these general characteristics, this survey reviews the studies on task allocation and load balancing with respect to the following aspects: 1)typical control models; 2)typical resource optimization methods; 3)typical methods of achieving reliability; 4)typical coordination mechanisms among heterogeneous nods; 5)typical models considering network structures. For each aspect, we summarize the existing studies and discuss the future research directions. Through the surveys, the related studies in this area can be well understand based on how they can satisfy the general characteristics of distributed systems.*

Keywords : *datacenter creation, broker creation cloudlet creation, , physical machine allocation, fitness value calculation, synthetic traffic.*

1. Introduction

Distributed systems, in which h the distributed computation-al units are connected and organized by networks to meet the demand of large-scale and high performance computing, have received considerable attention over the past decades [1][2][3]. There are many types of distributed systems, such as grids [4], peer-t o-peer (P2P) systems [5], Adhoc networks [6], cloud computing systems [7], pervasive computing systems [8], and online social network systems[9].Currently, the applications in distributed systems are various, such as web service, scientific computation, and file storage. In general, an application in a distributed sys-tem can be divided into a number of tasks and be executed on different nodes; the performance of an application in distributed system is dependent on the allocation of the tasks comprising the application onto the available nodes, referred to as the task allocation problem [10][11]; if too many tasks are crowded on certain nodes, tasks could be switched from heavy-

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

burden ed nodes to light-burdened ones to reduce the waiting time of tasks at nodes, which is called load balancing [4]. It is commonly stated that task al-location and load balancing are crucial to the distributed systems [12]. In past decades there are significant amount of studies for this area, which could cause people to be puzzled by the enormous amount of wide variety of re-search results. Although there have been some related surveys previously, but most of which only made a very preliminary review on the state of art of one single type of distributed system, such as the survey of load balancing in grids [91] [92], the survey of load balancing in cloud computing [93] [94], and the survey of load balancing in P2P systems [95]. Then, how to correlate the related studies in varying types of distributed systems and make a general taxonomy on them? To solve the above problem, first this survey paper summarizes the typical characteristics of general distributed systems, such as the distributed control, resource distribution, open environments, heterogeneous nodes, and network structures. Then, the survey paper categorizes the existing studies on task allocation and load balancing based on how they can satisfy the characteristics of distributed systems, which includes the following aspects: control models, resource optimization, ensuring reliability, cord-nation mechanisms, and measures to consider network structures. With this survey, a general and macroscopic review of task allocation and load balancing for varying types of distributed systems can be achieved, which can have higher generality much than the previous preliminary surveys for only one single type of systems.

1. System Analysis

1.1 Existing System

Currently, a large number and variety of applications have been extensively developed for distributed systems linking computers and other computing devices together in a seamless and transparent way. In distributed systems, various nodes act autonomously and cooperate with each other, which can achieve the purposes of resource sharing, open-ness, concurrency, scalability, fault-tolerance, and transparency [16][17]. In summary, real distributed systems have the following general characteristics. Lack of a global control unit. Due to the large-scale, dynamic, and heterogeneous nature of distributed system [17], it is difficult to implement a global control unit for the en-tire system. For example, in ad hoc networks, wireless nodes can communicate with each other in the absence of affixed infrastructure and without central control [18]; in sensor networks, many sensor nodes are densely deployed and may not have global identifications because of the large amount of overhead and large number of sensors [19]; inP2P systems, a decentralized architecture is applied in which individual nodes in the network act as both suppliers and users of resources, in contrast to the centralized client-server model where client nodes request access to resources provided by central servers [5]. Distribution and sharing of resources. There are various resources in the systems, such as data, replicas of popular web objects, processors, computational capacity, and disk storage. Resources are district butted among nodes, and node soften agree to share their local resources with each other to

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

implement tasks [20]. To impel Ent the effective sharing of resources, resource management is very crucial, which of-ten includes resource placemen t, resource provision, re-source discovery, and resource negotiation [5] [21] [22].Moreover, resource caching is often used to improve the performance of resource access, where some nodes cache the resources to serve future requests instead of fetching resources from the original nodes [23]. Openness and unreliability of systems. Distributed sys-tems are often open, and some nodes and network structures may be unreliable. For ex ample, there are no efficient ways to prevent malicious peer s from joining open systems, thus the distributed systems are very vulnerable to abuses by selfish and malicious users [24]. Moreover, due to the autonomy of nodes, some nodes may fail to operate independently. Therefore, reliability and fault-tolerance are cru-coal for real open distributed systems. Heterogeneous nodes. In some distributed systems, the nodes are heterogeneous. For example, in grids, the nodes may be attributed to different organizations [20]; in sensor networks, the nodes may have different capacities and sensing ranges [19] [25]; in social network systems, the nodes may have different response times and thresholds for diffusion [26]. When tasks are executed, heterogeneous nodes should coordinate with each other to maximize the total utilities. Moreover, hate erogenous nodes may compete for critical resources when they execute tasks. Constraints of network structures. In a distributed sys-tem, all nodes are constrained by the network structure. In the network structure, each node can only communicate with its neighboring nodes; the communication between two non-neighboring nodes should be relayed by other intermediate nodes [1] [11]. Therefore, the system performance is significantly influenced by the network structures. Accordingly, the tasks in distributed systems always en-counter the following situations: the tasks may be implemented by negotiation among nodes without a central control. In a distributed system, the tasks that require more than one node may be implemented in manner such that many nodes coordinate with each other and negotiate resources. Because distributed systems often lack a global control unit, as st ased above, the negotiation in the task allocation may be implemented by nodes autonomously. the tasks are always implemented by accessing required re-sources distributed in the networks. To execute tasks efficiently, one of the key elements is to improve the performance of accessing the resources necessary for the tasks. Therefore, resource optimization in networks can significantly influence the performance of tasks. the tasks may be implemented unreliably. Distributed sys-tems are open and may be invaded by malicious users [9]; moreover, there may be unreliable resources and communications in distributed systems [16]. Therefore, the tasks may be implemented unreliably, and sometimes the de-sired results may not be achieved [27]. the tasks performed by heterogeneous nodes may vary sig-nificantly and compete for critical resources. Because the nodes are heterogeneous and the number of users may vary dynamically in real time, the tasks may vary significantly at different times and at different nodes. Moreover, the users at different nodes are often self-motivated and will attempt to maximize their own benefits. Therefore, the execution of tasks of different users May co meet for critical resources,

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

which may bring about conflicts in the systems. the performance of tasks may be influenced significantly by network structures. First, the network structures may influence the resource access performance for tasks; thus, the resource optimization in task allocation should consider the network structures. Second, the interaction and coordination among nodes often obey certain network structures, thus a suitable network strut cure can improve the cord-nation performance of nodes in task allocation.

1.2 Proposed System

Control models represent how to control the entire process-is of task allocation and load balancing. In general, there are two prevalent control models in existing studies; one is the centralized control model, and the other is the decentralized control model .The centralized model uses a central controller that should know the status information of the entire system in real time [36]. The central co stroller will make all decisions based on the information that is sent from other nodes [37].The centralized model can be implemented easily, however, it may be sometimes infeasible in real distributed systems that are large and dynamic, where the global information cannot be achieved in real time; moreover, the central con-trolled may become a performance bottleneck of the system. In contrast, the distributed model can be used in those large and dynamic distributed systems, where the tasks can be allocated by nodes themselves and the global control unit is not needed; the nodes make their decisions based on their own perceived information about the system. Fault-tolerance can be achieved because each node can act as the allocator. However, a drawback of the distributed model is that the nodes face relatively high computational costs which may place heavy loads on large systems and make the task allocation processes difficult to control effectively.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

3. System Design

3.1 Flow diagram

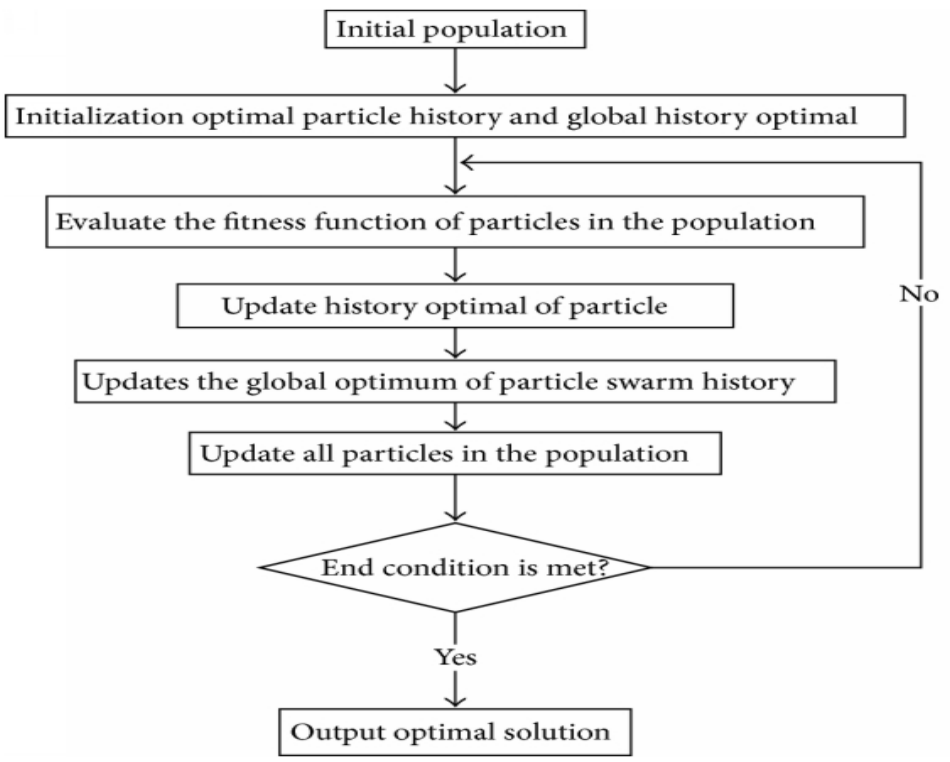
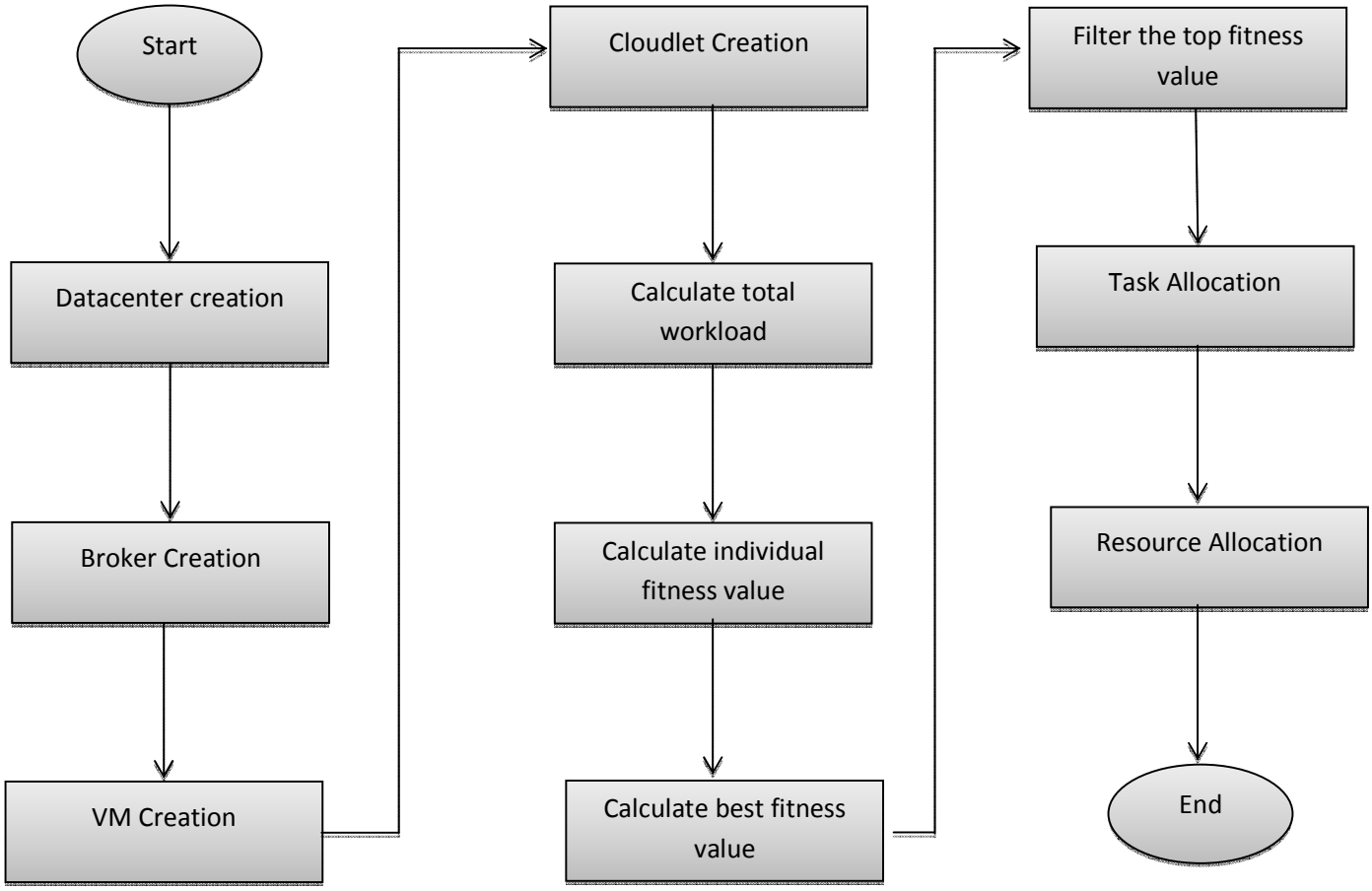


Fig 3.1 Flow Diagram

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

3.2. System Architecture



3.2 System Architecture

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

3.3 Use case diagram

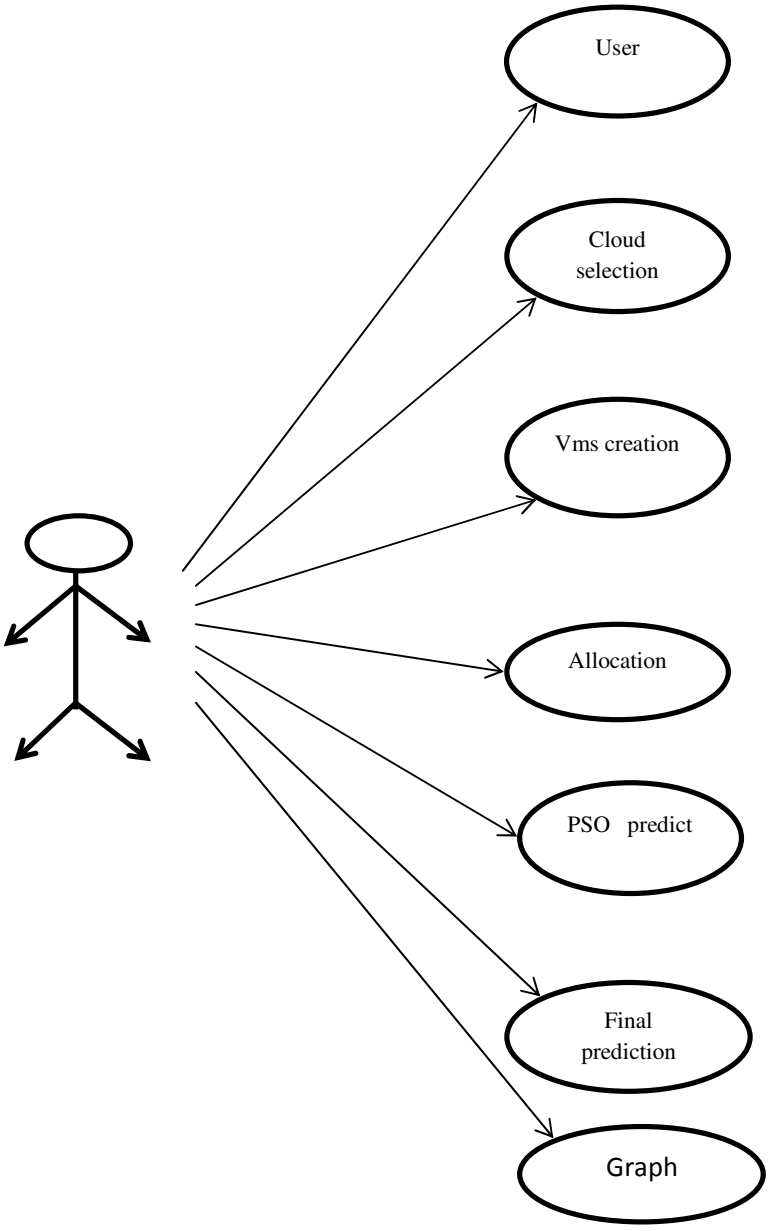


Fig 3.3 Use Case Diagram

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

3.4 Class Diagram

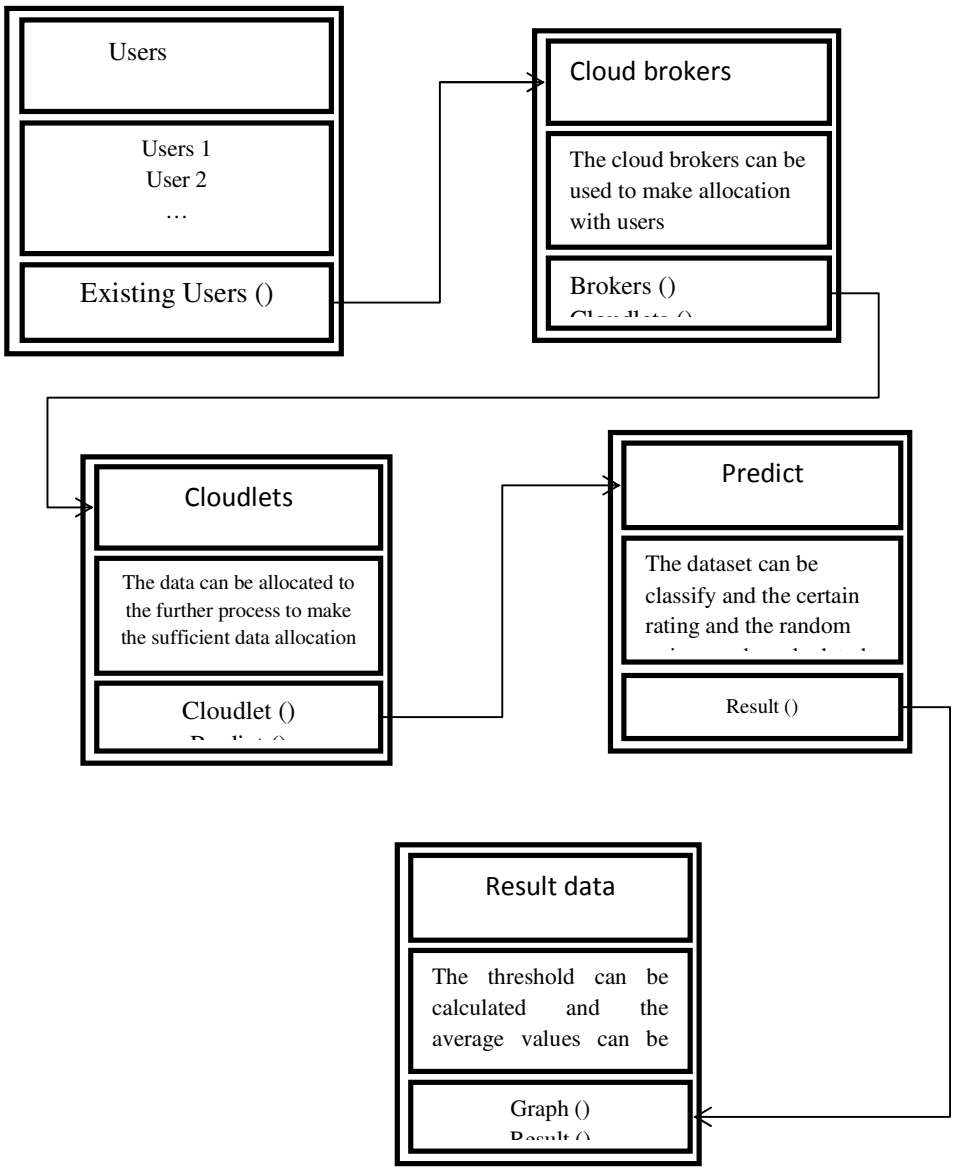


Fig 3.4 Class Diagram

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

3.5 Sequence diagram

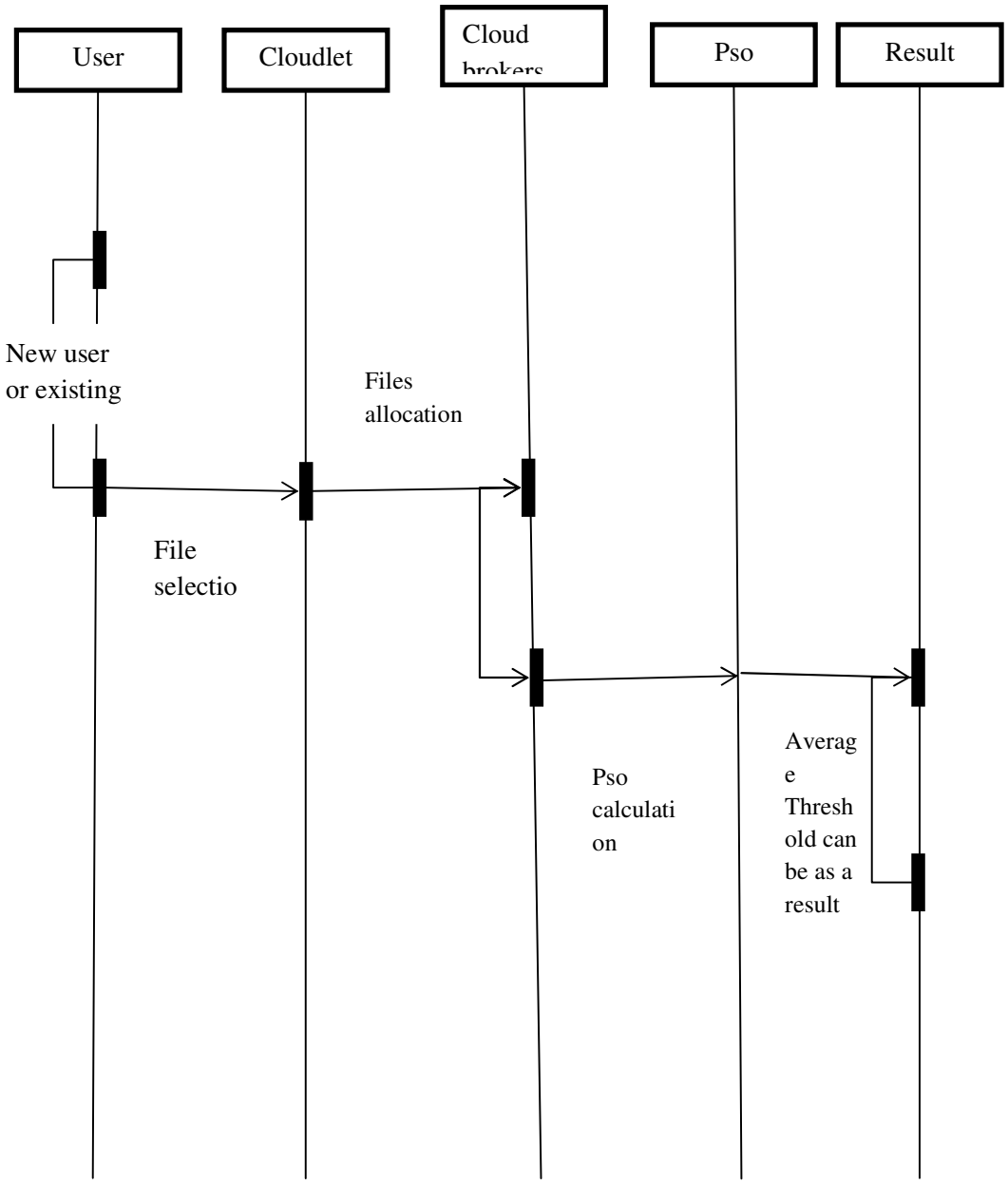


Fig 3.5 Sequence Diagram

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

3.6.ERDiagram

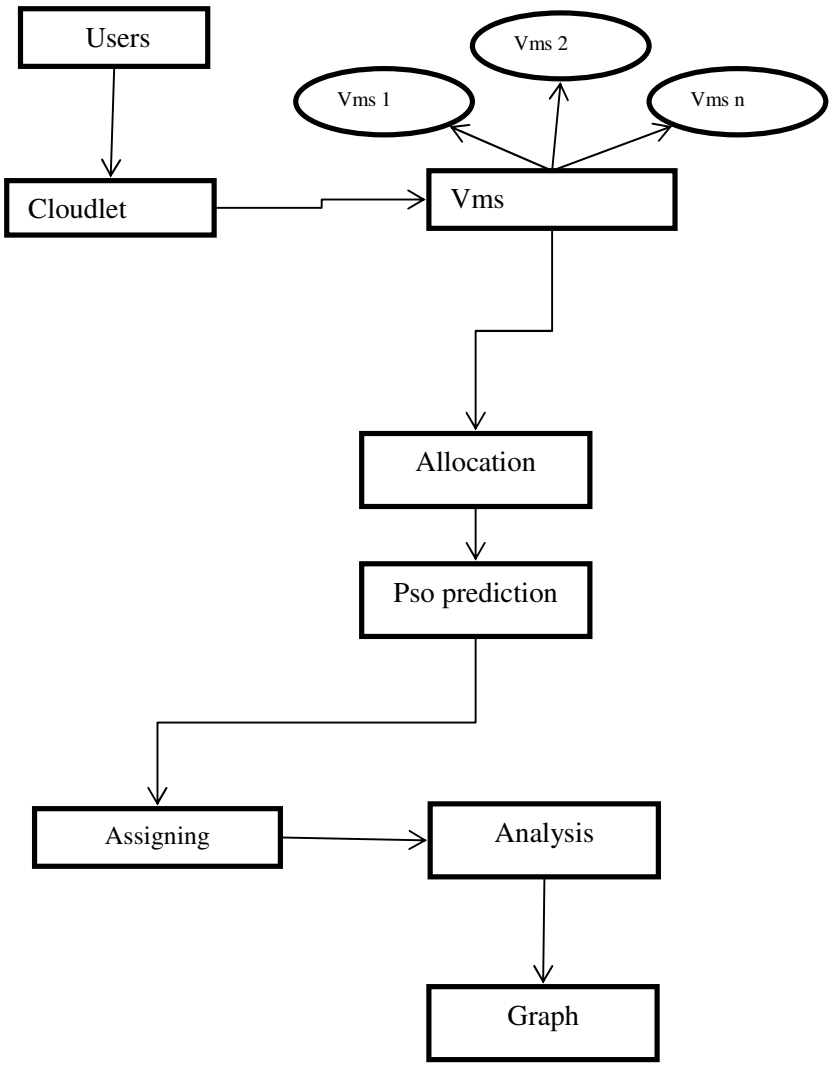


Fig 3.6 ER Diagram

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

4. System Testing

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that "al gears mesh", that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

4.1 Types of Testing

4.1.1 Unit Testing

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

4.1.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

i) Top-down integration testing.
ii) Bottom-up integration testing.

4.1.3 White Box Testing

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

4.1.4 Black Box Testing

- Black box testing is done to find incorrect or missing function
- Interface error
- Errors in external database access
- Performance errors
- Initialization and termination errors

In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behavior of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

4.1.5 Validation Testing

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

4.1.6 User Acceptance Testing

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

4.1.7.Output Testing

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

5. System Implementation

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the system
- Their confidence in the software built up
- Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

5.1 User Training

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important.

Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

5.2 Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

5.3 Operational Documentation

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

5.4 System Maintenance

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than "finding mistakes".

5.4.1 Corrective Maintenance

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

5.4.2 Adaptive Maintenance

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

5.4.3 Perceptive Maintenance

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

5.4.4 Preventive Maintenance

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

6 Modules

- Cloud Parameter Creation
- Job Scheduler
- Applying QPSO
- VM Scheduler

6.1Cloud parameter creation

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

6.1.1 User Registration and policy

User, first register their details and login into the process. Depending upon the policy given, user send request to the SLA Manager. User requirement is matching with the resources in the cloud service providers.

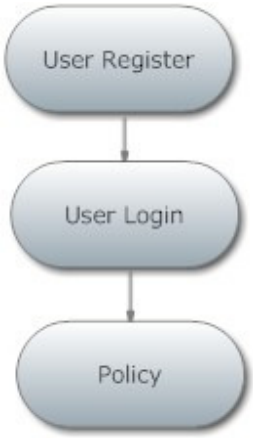


Fig 6.1 Parameter Creation

6.1.2 Vms scheduler

SLA Manager

SLA stands for Service Level Agreement. SLA Manager receive the request from the user. They handover the request to the broker for further process. The service user negotiates with the service broker on the service-level agreement (SLA) details. The service user negotiates with the service broker on the service-level agreement (SLA) details. They eventually prepare an SLA contract. According to this contract, the broker selects, and then presents highly trusted resources to users from the trusted resource pool.

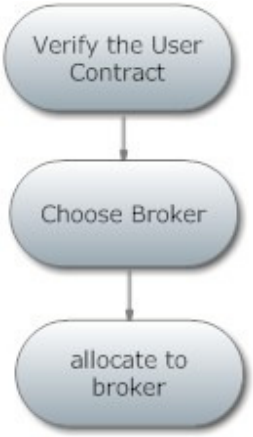


Fig 6.2 Virtual Machine Allocation

6.1.3 Job scheduler

Resource Registration

Each resources in the cloud are register their details and resources details in the cloud for resource allocation. It manages and indexes all the resources available from multiple cloud providers, and obtains information from each particular cloud resource, acting as pricing interface for users, and updating the data-base when new information is available.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

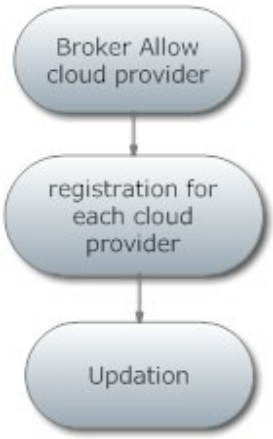


Fig 6.3 Job Scheduling

6.1.4 Trust Evaluation

Each cloud manager registers its service resources through the cloud broker. The service user negotiates with the service broker on the service-level agreement (SLA) details. They eventually prepare an SLA contract. According to this contract, the broker selects, and then presents highly trusted resources to users from the trusted resource pool.

Fig 6.4 Trust Evaluation

6.1.5 Evaluate Trusted Resource

RTD is used to evaluate recent cloud resource service operators, and RTD is evaluated by knowledge of a resource's quality of service. When matchmaking a resource for users, the cloud broker must first consider whether the resource has the required capabilities. The first of these considerations can be evaluated by the resource's availability, which can determine whether a resource has the required capability or not. The second consideration mainly focuses on the reliability and security of the resource, which can be evaluated by the resource's service.

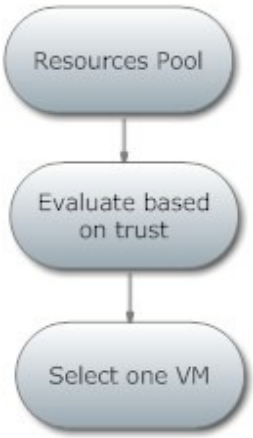


Fig 6.5 Evaluation

6.1.6Applying qpso

Allocation of User Policy

Allocate the resources to the user depends upon the trust evaluation and trust resources in the cloud .Finally, Trusted resources is chosen and send to the broker. Then broker allocate the resources to the user through SLA Manager.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

Fig 6.6 Allocation

7 SYSTEM REQUIREMENTS

7.1 Software Requirements

- O/S : Windows XP.
- Languag : Java.
- IDE : Net Beans 6.9.1
- Data Base : MySQL

7.2 Hardware Requirements

- System : Pentium IV 2.4 GHz
- Hard Disk : 160 GB
- Monitor : 15 VGA color
- Mouse : Logitech.
- Keyboard : 110 keys enhanced
- Ram : 2GB

7.4 Java

Java is a programming language originally developed by James Gosling at Sun Microsystems (now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere." Java is currently one of the most popular programming languages in use, particularly for client-server web applications.

7.5 Java Platform

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware.

End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets. Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would. Just-in-Time compilers were introduced from an early stage that compiles byte codes to machine code during runtime.

Just as application servers such as Glass Fish provide lifecycle services to web applications, the Net Beans runtime container provides them to Swing applications. All new shortcuts should be registered in "Key maps/Net Beans" folder. Shortcuts installed INS Shortcuts folder will be added to all key maps, if there is no conflict. It means that if the same shortcut is mapped to different actions in Shortcut folder and current key map folder (like Key map/Net Beans), the Shortcuts folder mapping will be ignored.

* Database Explorer Layer API in Database Explorer

* Loaders-text-dB schema-Actions in Database Explorer

* Loaders-text-sql-Actions in Database Explorer

* Plug-in Registration in Java EE Server Registry

The keyword public denotes that a method can be called from code in other classes, or that a class may be used by classes outside the class hierarchy. The class hierarchy is related to the name of the directory in which the .java file is located.

The keyword static in front of a method indicates a static method, which is associated only with the class and not with any specific instance of that class. Only static methods can be invoked without a reference to an object. Static methods cannot access any class members that are not also static. The keyword void indicates that the main method does not return any value to the caller. If a Java program is to exit with an error code, it must call System. Exit () explicitly.

The method name "main" is not a keyword in the Java language. It is simply the name of the method the Java launcher calls to pass control to the program. Java classes that run in managed environments such as applets and Enterprise JavaBeans do not use or need a main () method. A Java program may contain multiple classes that have

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

main methods, which means that the VM needs to be explicitly told which class to launch from.

The Java launcher launches Java by loading a given class (specified on the command line or as an attribute in a JAR) and starting its public static void main(String[]) method. Stand-alone programs must declare this method explicitly. The String [] args parameter is an array of String objects containing any arguments passed to the class. The parameters to main are often passed by means of a command line.

7.6 Java a High-level Language

A high-level programming language developed by Sun Microsystems. Java was originally called OAK, and was designed for handheld devices and set-top boxes. Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web.

Java source code files (files with a .java extension) are compiled into a format called byte code (files with a .class extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VMs). Byte code can also be converted directly into machine language instructions by a just-in-time compiler (JIT).

Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer.

Object-Oriented Software Development using Java: Principles, Patterns, and Frameworks contain a much applied focus that develops skills in designing software-particularly in writing well-designed, medium-sized object-oriented programs. It provides a broad and coherent coverage of object-oriented technology, including object-oriented modeling using the Unified Modeling Language (UML) object-oriented design using Design Patterns, and object-oriented programming using Java.

7.7 Net Beans

The **Net Beans Platform** is a reusable framework for simplifying the development of Java Swing desktop applications. The Net Beans IDE bundle for Java SE contains what is needed to start developing Net Beans plug-in and Net Beans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally-signed upgrades and new features directly into the running application.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)
- Window management
- Wizard framework (supports step-by-step dialogs)
- Net Beans Visual Librar

7.8 J2EE

A **Java EE application** or a **Java Platform, Enterprise Edition application** is any deployable unit of Java EE functionality. This can be a single Java EE module or a group of modules packaged into an EAR file along with a Java EE application deployment descriptor.

Enterprise applications can consist of the following:

- EJB modules (packaged in JAR files);
- Web modules (packaged in WAR files);
- connector modules or resource adapters (packaged in RAR files);
- Session Initiation Protocol (SIP) modules (packaged in SAR files);
- application client modules
- Additional JAR files containing dependent classes or other components required by the application;

7.9 Wamp Server

WAMPs are packages of independently-created programs installed on computers that use a Microsoft Windows operating system.

Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

7.10 MySQL

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

Free-software-open source projects that require a full-featured database management system often use MySQL. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, Drupal and other software built on the LAMP software stack.

7.11 Platforms and interfaces

Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interface called MyODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The MySQL server and official libraries are mostly implemented in ANSI C/ANSI C++.

7.12 FEASIBILITY STUDY

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

• Economic Feasibility

• Technical Feasibility

• Behavioral Feasibility

Economic Feasibility

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

Technical Feasibility

This study center around the system's department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

Behavioral Feasibility

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user's request, instead of getting a report, which doesn't contain much detail.

8 SYSTEM IMPLEMENTATION

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the system
- Their confidence in the software built up
- Proper guidance is impaired to the user so that he is

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

8.1 User Training

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

8.2 Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

8.3 Operational Documentation

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

9 Conclusion

We propose SOTS for trustworthy resource matchmaking across multiple clouds. We have shown that SOTS yields very good results in many typical cases. How-ever, there are still some open issues we can apply to the cur-rent scheme. First, we are interested in combining our trust scheme with reputation management to address concerns in users' feedback. A universal measurement and quantitative method to assess the security levels of a resource is another interesting direction. Evaluation of the proposed scheme in a larger-scale multiple cloud environments is also an important task to be addressed in future research.

10 References

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

[1] Azzedin .F and Ridha .A. "Feedback behavior and its role in trust Assessment for peer-to-peer systems,"Telecommunication. System., vol. 44, No. 3-4, pp. 253–266, 2010.

[2] Cayman .S, Gales .A, Chapman .C, Tofutti .G, Rodeo-Merino .L,Vaquero .L.M, K. Nagging and B. Rochwerger, "Monitoring service Clouds in the future internet," in Future Internet Assembly. Amstardam, the Netherlands: IOS Press, 2010, pp. 115–126.

[3] Chaves .S.A, Urinate R.B ,septal .C.W, "Toward an architecture for monitoring private clouds," IEEE Communication Mag.,vol.49, no. 2, pp. 130–137, 2011.

[4] Dragon .N, "A survey on trust-based web service provision Approaches," in Proc. 3rd Int. Conf. Dependability, 2010, pp. 83–99.

[5] Ghosh .D, Canon .R.S, and Ramakrishnan .L. "I/O performance Of virtualized cloud environments," in Proc. 2nd Int. Workshop Data Intensive Comput. Clouds, 2011, pp. 71–80.

[6] Habib .S.M, Rise .S, and Muhlhauser .M, "Towards a trust man-agreement system for cloud computing," in Proc. IEEE 10th Int.Conf. Trust, Security Privacy Comput. Commun. 2011, pp. 933–939.

[7] Liang.Z and Shi .W, "A reputation-driven scheduler for auto-gnomic and sustainable resource sharing in grid computing," Parallel Diatribe. Comput. vol. 70, no. 2, pp. 111–125, 2010.

[8] Hoffman .L.J, Lawson-Jenkins .K, and Blum .J, "Trust beyond Security: An expanded trust model," Communication. ACM, vol. 49, No. 7, pp. 95–101, 2006.

[9] Hwang .K and Li .D, "Trusted cloud computing with secure Resources and data coloring,"IEEE Internet Comput. , vol. 14, no. 5, pp. 14–22, Sep. /Oct. 2010.

[10] Jackson .K.R, Morikis .K, Ramakrishnan .L, K. J. Range, and R. C.Thomas, "Performance and cost analysis of the Supernova factory On the Amazon AWS cloud," Sci. Program. , vol. 19, no. 2-3, pp. 107–119, 2011.

[11] Khan .K.M and Mallei .Q, "Establishing trust in cloud Computing," IT Prof., vol. 12, no. 5, pp. 20–27, Sep. /Oct. 2010.

[12] Kim .H, Lee .H, Kim .W, and Kim .Y, "A trust evaluation model For QoS guarantee in cloud systems," Int. J. Grid Diatribe. Comput. , vol. 3, no. 1, pp. 1–10, 2010.

[13] Legrand .I, Voicu .R, Cirstoiu .C, Grigoras .C, Betev .L, Costan .A, "Monitoring and control of large systems with MonALISA," Com-mun. ACM, vol. 52, no. 9, pp. 49–55, 2009.

[14] Li .X and Yang .Y, "Trusted data acquisition mechanism for cloud Resource scheduling

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 11, March 2017

based on distributed agents," China Commun. , vol. 8, no. 6, pp. 108–116, 2011.

[15] Li .X, Zhou .F, and Yang .X, "Scalable feedback aggregating (SFA) Overlay for large-scale P2P trust management," IEEE Trans. Parallel Diatribe. Syst., vol. 23, no. 10, pp. 1944–1957, Oct. 2012.

[16] Manuel .P.D, S. Tamara Salve, and Barr .M .I .A .E, "Trust managreement system for grid and cloud resources," inProc. 1st Int.Conf. Adv. Comput., Dec. 2009, pp.176–181.

[17] Shafer .J, "I/O virtualization bottlenecks in cloud computing Today," in Proc. 2nd Conf. I/O Virtualization, 2010, pp. 1–7.

[18] Woolskin .R, Obertelli G, Nurmi .D, Osman .S, Yousef .L, and Zagorodnov .D, "The eucalyptus open-source cloud-computing system," in Proc. 9th IEEE/ACM Int. Symp. Cluster Com-put. Grid, 2009, pp.124–131.