|  |  |  |
| --- | --- | --- |
| **Name of the Project**  **Cost Effective Resource Provisioning Approach for Cloud Environments** | Cost Effective Resource Provisioning Approach for Cloud Environments | |
| **Objective/Vision** | Cost optimization is a major concern in cloud computing as owners of large IT infrastructures have to pay a large cost for resource utilization. The infrastructure resources of cloud computing can be widely distributed in different phases according to the needs of different users. In the course of a cloud, implementation users have the flexibility to choose the EC2 instance type that provides the appropriate mix of resources for the target application and workload. The charges are applied on the basis of resource utilization, but it is very high as most of them not used in an effective way.  The main purpose of the system is to create private cloud (test bed) by using (Amazon Account) along with monitoring critical resources like RAM, CPU, memory, bandwidth, partition information, running process information and utilization and swap usages etc. Also, recommend the price reduction strategy. The system also enables optimum utilization of cloud resources. | |
|  | | |
| **User of the System** | | 1. Cloud Controller 2. Individual Node Owners 3. Administrators |
| **Functional Requirements** | | The proposed system which cans monitor VMs (EC2 Instances) on private clouds such as Amazon or Google and offers solutions to decrease infrastructure cost. Resource Monitoring of Cloud Nodes:  **1. Cloud Setup -** Creating private cloud (test bed) by using (Amazon Account).    **2. Resource Monitoring -** monitoring critical resources like RAM, CPU, memory,  bandwidth, partition information, running process information and utilization and swap usages etc.  **3. Authentication and authorization –** we need to connect to existing user’s amazon account using user id and password and fetch all the performance matrix like CPU, RAM, storage etc.  **4. Testing -** In order to evaluate the performance of complete setup, need to deploy resource monitoring and load balancing tools on test bed and evaluate need of available resources.  **Modules:**   1. **Resource Monitoring of Cloud Nodes:**    1. User should be able to view CPU and RAM usage utilization of amazon ec2 nodes    2. CPU and RAM utilization statistics should be dynamic and should refresh every second. 2. Select Cloud Plans for popular clouds like amazon. Cost of service depends on region of server, memory usage, CPU etc. Cloud service providers charge for following services which need to be added in system    1. Storage – Pricing    2. Request Pricing    3. Storage Management Price    4. CPU pricing 3. **Monitor account wise VM Usage of following parameters** 4. CPUUtilization 5. DiskReadBytes 6. DiskWriteBytes 7. NetworkIn 8. NetworkOut 9. StatusCheck 10. **Propose efficient resource utilization** 11. By suggesting memory cutdown 12. By suggesting cpu cutdown 13. By suggesting storage cutdown |
| **Non-Functional Requirements**  **(Performance Requirement, Quality Of Service Requirement)** | | 1. Secure access of confidential data (user’s details). 2. 24 X 7 availability. 99.9% Uptime during business hours. It should mitigate system failure risks. 3. Better component design to get better performance at peak time 4. Flexible service based architecture will be highly desirable for future extension 5. High Scalability. The solution should be able to accommodate high number of customers and brokers. Both may be geographically distributed. |
| **Algorithm** | | 1. **AES Algorithm** 2. **Support Vector Machine (SVM)** |
| **Algorithm Details** | | * 1. **AES Algorithm**   AES is based on a design principle known as a [substitution-permutation](http://en.wikipedia.org/wiki/Substitution-permutation_network) network, and is fast in both software and hardware.[[8]](http://en.wikipedia.org/wiki/Advanced_Encryption_Standard#cite_note-8) Unlike its predecessor DES, AES does not use a [Feistel network](http://en.wikipedia.org/wiki/Feistel_network). AES is a variant of Rijndael which has a fixed [block size](http://en.wikipedia.org/wiki/Block_size_%28cryptography%29) of 128 [bits](http://en.wikipedia.org/wiki/Bit), and a [key size](http://en.wikipedia.org/wiki/Key_size) of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.  AES operates on a 4×4 [column-major order](http://en.wikipedia.org/wiki/Column-major_order) matrix of bytes, termed the *state*, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special [finite field](http://en.wikipedia.org/wiki/Finite_field_arithmetic).  The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text. The number of cycles of repetition are as follows:   * 10 cycles of repetition for 128-bit keys. * 12 cycles of repetition for 192-bit keys. * 14 cycles of repetition for 256-bit keys.   Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.    **Figure 3: -** AES Algorithm Stepwise   * 1. **Support Vector Machine (SVM)** * SVM is a powerful classifier that is able to distinguish two classes. SVM classifies the test image in to the class with highest distance up to the neighboring point in the training. * SVM training algorithm built a model that predict whether the test image fall into this class or another. * SVM necessitate a vast training data to decide a decision boundary and computing cost is very high although we are using single pose (frontal) detection. * The SVM is a learning algorithm for classification which attempt to discover the finest distinguishing hyper plane which minimize the error for unseen patterns.   Image result for svm classifier  Figure 1 : Distinguishing Hyper Plane To Minimize The Error   * The data which cannot be distinguished the input is mapped to high-dimensional attribute space where they can be separated by a hyper plane. This projection is well performed by means of kernels.   Related image  Figure 2: Separating Hyper Plane By Equation   * If training set of samples and the equivalent resultant values {-1, 1}. So SVM intend to get the best separating hyper plane specified by the equation WTx+b that make use of the distance between the two classes as shown in above figure. |
| **General/Design Constraints** | | 1. Application should not use java.io package 2. Cloud node should be ON for monitoring |
| **Optional Features** | | 1. Backup and Restore of system data 2. Scalability of application in case of increase in business need. |
| **User Interface Priorities** | | A. Professional look and feel B. Use of AJAX atleast with all registration forms D. Use of reports |
| **Technologies to be Used** | | JDK 1.8, J2EE,JSP,Servlets,Cloud Database, |
| **Tools to be Used** | | 1. Eclipse Luna 2. Amazon AWS API 3. REST Web Services |
| **Final Deliverable must Include** | | A. Application archive ( .war/.ear ) with source code B. Database backup and DDL Script C. Complete Source code |
| **References** | | 1. Subhas Chandra Misra, Arka Mondal,"Identiﬁcation of a company’s suit- abilityfortheadoptionofcloudcomputingandmodellingitscorresponding Return on Investment",2010 Elsevier. 2. Ryan Chard,Kyle Chard,Rich Wolski,Ravi Madduri,Bryan Ng and Kris Bubendorfer,IanFoster,"Cost-AwareCloudProﬁling,Prediction,andProvi- sioning as a Service",PUBLISHED BY THE IEEE COMPUTER SOCIET, 2017. 3. Xinhui Li, Ying Li, Tiancheng Liu, Jie Qiu, Fengchun Wang,"The Method and Tool of Cost Analysis for Cloud Computing",2009 IEEE International Conference on Cloud Computing. 4. Keith R. Jackson,Krishna Muriki,Shane Canon, Shreyas Cholia, John Shalf Harvey J. Wasserman, and Nicholas J. Wrig,"Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud",2nd IEEE International Conference on Cloud Computing Technol- ogy and Science. 5. AlexandruIosup,Member,IEEE,SimonOstermann,NezihYigitbasi,Mem- ber, IEEE, Radu Prodan, Member, IEEE, Thomas Fahringer, Member, IEEE, and Dick Epema, Member, IEEE"Performance Analysis of Cloud Computing Services for Many-Tasks Scientiﬁc Computing",IEEE TPDS, MANY-TASK COMPUTING, NOVEMBER 2010. 6. AmelieChiZhou,BingshengHeandChengLiuNanyangTechnologicalUni- versity"Monetary Cost Optimizations for Hosting Workﬂow-as-a-Service in IaaS Clouds",IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. X, NO. X, AUGUST 2014. |