



A Novel technology for Improving the Reliability of Cloud Storage System

Geetha Ramadas*¹, Kavitha P¹, Prathikshna M A¹, Thanikonda Bhanuja Chowdary G¹,
Nithya M¹, Parkavi K¹

R.M.K. Engineering College, Tiruvallur, India
hod.eee@rmkec.ac.in

Abstract. Cloud services are gaining popularity and the demand for cloud space is increasing. A lot of research is happening in the field of cloud management and marketing. Effective handling of Cloud service is an important component of cloud architecture and can even be used to act as an intermediate layer between consumers and providers or as a platform for launching marketplace services. Cloud market is very versatile with the concept of pay-as-you-go. The major disadvantage, here, is the hourly billing cycle as user has to pay based on the hourly consumption and most of the consumers may not use full duration for which they have made payment. Majority of cloud service providers give discounts to long-term customers; however, these benefits are not available to transient customers with light computing requirements. A brand-new position, called cloud broker can be proposed, to lower prices for transient cloud consumers. A cloud broker is an intermediate agency between cloud users and providers. A system is proposed where facility to rent a number of reserved Virtual Machines (VMs) from a cloud provider for a reasonable rate, only for the duration for which they are using the cloud. Like cloud providers, cloud brokers use shorter billing cycles. Cloud brokers can dramatically lower prices for users in this way. Cloud brokers can profit from the pricing difference between reserved and on-demand VMs in addition to lowering user costs. This work focuses on the tailor made configuration of cloud brokers and VMs.

Keywords: Cloud service, broker service layer, tariff.

1 Introduction

Cloud computing is becoming the most sought-after technology architecture for enterprises because of its flexibility, security and easy availability. Furthermore, it has attributes like user controlled service facility when in need, universal network access, collecting resources independent of the location, quick fast and flexible resource handling, charges depending on the usage and the risk transfer. The disruptive nature of cloud computing is transforming the way businesses as a whole use information technology. One of the major parameters for this transformation is the outsourcing of the data to the cloud. Maximum benefit can be derived by the individuals or organizations using cloud services as the data storage in cloud

© The Author(s) 2025

A. S. Edward et al. (eds.), *Proceedings of the International Conference on Advanced Research in Electronics and Communication Systems (ICARECS 2025)*, Atlantis Highlights in Engineering 38,
https://doi.org/10.2991/978-94-6463-754-0_57

is very flexible and is based on the demand by a customer. Some of the noteworthy benefits are reduction of the workload of storage handling, location independent access to global data and minimization of investment on the infrastructure. The term "cloud computing" represents a broad variety of innovative concepts involving a sizable number of physically available machines, like the Internet. It can also be interpreted as the ability to manage all the programs that are run on all the connected computers. It is also used to describe distributed computing based on the network. It is most often used to refer to services which are dependent on the network. It is most often delivered using software controlled virtual hardware running on one or more physical machines but appears to be provided by genuine server hardware. Since these virtual servers are physically not available, they may be quickly moved and scaled up or down without affecting the consumer and this makes them resemble cloud. The promotion of hosted services and application service provisioning that runs client-server software on a remote location have helped in spreading awareness about it. Cloud computing utilizes resource sharing, like the electrical network, to achieve coherence and profit. The implementation of cloud computing is the larger idea of pooled services and networked infrastructure. The effectiveness of pooled resources is another crucial component of the cloud. Cloud resources are often shared among several users and dynamically redistributed as necessary. This allows for the distribution of resources to users. For instance, a cloud computing facility could reallocate the same resources while providing a different application (like web servers) to North American clients while still servicing European users during European business hours. We must boost computer power while reducing environmental damage because this way utilizes less electricity, cooling, rack space, and other resources. An organization is considered to be "moving to the cloud" when it transitions from the existing CAPEX strategy which depends on exclusive hardware which depreciated over time to the OPEX model which depends on buying the shared cloud infrastructure and paying based on the usage. "Moving to the cloud" is how it is described. It also alludes to dynamic shifts. Proponents claim that by using the cloud, firms may avoid making large upfront infrastructure investments and instead concentrate on projects that improve their operations. Advocates assert that by enabling IT to more quickly control resources to dynamically varying load using the cloud helps in reducing the requirement for traditional CAPEX (purchase specialized gear and depreciate it over time). Managing cloud computing involves a number of management difficulties. Public cloud clients don't have complete insight or control over the environment because it isn't confined within their own networks, and they don't own the hardware that hosts the cloud environment. To interact with cloud components, users of public cloud services must establish a connection with a cloud provider's built-in architecture. Configuring IP addresses, subnets, firewalls, and data service activities for storage requires the use of cloud APIs. Users of public clouds must participate in cloud infrastructure management since the ability to control these processes depends on the infrastructure and services of the cloud provider. Because end users can distribute programs through self-service portals, capacity management

is difficult in public and private cloud systems. Applications of all sizes appear out of nowhere, consume unforeseen quantities of resources and then disappear. All the cloud environments struggle with chargeback, or granularly charging resource utilization. Because they must charge competitive prices for their services while still making a profit, public cloud service companies face the difficulty of chargeback. Due to overlapping internal resources that may be funded by a single business unit, such as electricity, it may be tiresome task for the establishments to analyze actual resource costs on an instalment basis for users of public cloud services. Chargeback is very simple for private cloud operators, but the trick is figuring out how to assign resources as nearly as possible to real resource utilization in order to maximize operational efficiency. Budget overruns can be dangerous. Hybrid cloud architectures, which integrate traditional infrastructure components with both public and private cloud services, provide unique management issues. Economical challenges with excessive storage or bandwidth use, the dissemination of outdated photos, and security issues if sensitive data winds up on public cloud servers are just a few of them. Controlling information flow in a hybrid cloud architecture is another major issue. Applications hosted by public cloud providers off-premises must exchange information with on-premises clouds, and this information may change frequently. Both public and private clouds must handle the complicated mixture of regulations, rights, and restrictions that are commonly present in hybrid cloud environments. The functioning of servers, storage, networking, and data centre of the public cloud environment are controlled by the respective service providers. Three standard alternatives are commonly available to customers of pooled cloud platform: User self-provisioning: customers can pay the service provider through a web form or console interface. He can make payment for each transaction separately. Advance provisioning: Client signs a contract with a set of service providers based on the information which are already available. He has to pay the fee either as 100% advance or recurring fee at regular intervals. Dynamic provisioning: When a customer requests resources, the provider makes them available and deactivates them when the request is no longer required. Pay-per-use fees are assessed to the client. Software is needed to develop and manage private clouds. Software tools are necessary for managing a private cloud because they enable the setting up of a virtualized pool of compute resources, offer exclusive interface to the end-user, and manage security, resource allocation, tracking, and billing. Due to the customary high degree of virtualization and organizational focus on portable workloads found in cloud settings, management tools for private clouds often focus on services rather than resources. The main focus of the service provider should be in identifying the areas which needs to be controlled mainly what to control, how to control and when to control because compute, network, and storage resources in hybrid cloud environments must be managed across various domains. Some of the policies that should be implemented to manage these domains efficiently include access control, budgeting, and reporting. The functional diagram is shown in fig.1. The end-users are directly or indirectly connected to main service provider. The customers will be able to purchase from a cloud ser-

vice provider, with file upload and file download. The main strategy which are considered are preference matching, cost modelling, optimal calculation. The final step is resource allocation. Applications for cloud brokerage are constructed using current virtualization technologies, cloud platforms, and IaaS/PaaS/SaaS services. They can aid in providing context for broker apps by highlighting the main application focus. Brokerage is a method of resolving disputes between parties. It is crucial to take into account several user views, including those of the broker itself, the service provider and the end user. Each of these groups, including prosumers, have their own lifecycle. They differ in terms of which factors are under whose control and how crucial those aspects are to the action at hand. For instance, Gartner has segmented several supplier and prosumer types inside its brokerage model. The task of the app developer is made simpler by hiding the business-rule coding interface and simplifying the work through programming and architectural characteristics. Frameworks and API libraries have evolved into devops and, more recently, PaaS platforms for multi-language, multi-framework construction and commissioning, development of collaborative apps and its lifecycle management, as examples of how this has happened. For

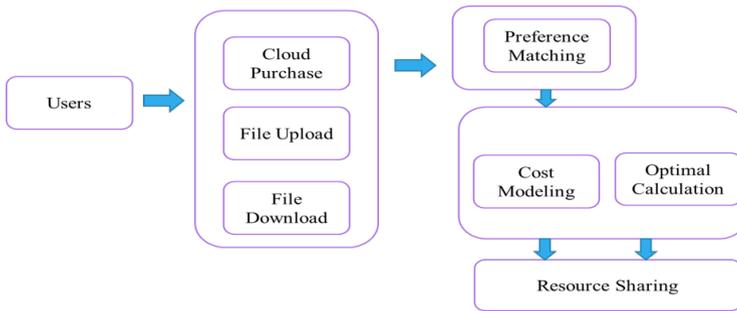


Fig. 1. Functional Block Diagram.

example, the developer does not need to be aware of the flexibility of the IaaS. They will construct the broker application as prosumers. The interface and broker application service are all that are required for app/cloud service users. They stand to gain the most from cloud broking, in my opinion. The Broker Platform Provider, a new actor, is the other major entity whose actions are affected by these technologies. Their main responsibility is creating a platform that meets the needs of both developers and end users. Yet, it frequently happens that the providers not only create the broker platform but also host it and make it available. While analysing and interpreting comparison results, it is important to take into account the various user perspectives. For instance, developer features are crucial for application developers but less important for end users. A new job is established, to the cloud broker, to lower the cost of cloud consumers. It obtains a number of reserved VMs at a competitive price from cloud providers

and rents them out to consumers on demand for reduced cost than what the cloud providers charge. In addition, broker has a shorter billing period than cloud providers.

2 Existing System

The present service taxonomies do not include a separate taxonomy, methodology, or comparison tool for cloud brokers. A taxonomy for inter-cloud architectures is, however, currently being created. They use a framework for comparison that takes into account five key factors: type/organization, architecture, brokering strategy, application type, and awareness. Insufficient for a thorough classification and comparison are the generic definitions offered by Gartner, Forrester, and NIST. An elementary, comprehensive framework and platform dimensions are proposed in recognition of the requirement for a multi-dimensional classification. Additionally, the idea instance lacks a defined language. An easy, multifaceted framework is suggested after acknowledging the requirement for a multidimensional classification. However, it does not offer a systematic strategy for locating, extracting, modelling, and evaluating the framework, nor does it distinguish clearly between application and platform dimensions. Tailormade solutions used for management, brokerage and market will be contrasted using this paradigm. To demonstrate the framework's usefulness while concentrating on the explanation of concerns with cloud broker architecture and technical specifications for service brokerage solutions, as well as to highlight challenges and more general research goals: On-Demand and Reserved Instances are the main payment methods accepted by all cloud providers for their instances. On-demand instances are appropriate for applications with short duration since clients pay for compute resources according to the instances they operate. According to the current plans provided by reputable cloud providers, customers should rent instances for lengthy periods of time, such as from six months to several years. In contrast to on-demand instance pricing, reserved instances provide customers a sizable savings. Short- term customers are not entitled to enjoy this scheme.

2.1 System Architecture Description

A broker application is a kind of cloud-based software system that provides application services to providers as well as the end users. We have examined the requirements and identified two application-related factors such as the scope of the service is described by the management for operation, broker for mediation and market for open collaboration and competition patterns. Referring to cloud delivery models, the target application services. Based on elements of the platform taxonomy, we have categorized broker platforms. Particularly, it is increasingly beneficial to consider the user types' point of view. For instance, three sorts of support are available for developers: A) Jcloud's API library; B) Cloudify's Devops; and C) Cloud Foundry and OpenShift's Complete PaaS. There is a shift towards end-user- focused cloud management from provider-focused solutions

to developer-focused ones. Caast is an illustration of the latter. The following part will go into greater detail on the trends and issues that came up throughout this debate. The assessment of the framework's suitability and completeness comes next. Adequacy. The framework is sufficient since it offers a sufficiently comprehensive structure. Features and capabilities - From the literature, we extracted capabilities and characteristics with fine graininess. Also, to add more dimensions, we used widespread classifications from the software engineering (specification and design) and cloud computing communities. A state-of-the-art application is used to empirically validate the completeness. For this, cases that represent a wide spectrum of cutting-edge systems have been methodically chosen. A review of the concrete descriptions revealed no essential Ideas missing from the framework. Even though some table parts are weakly populated, it is still possible to identify many tools for each feature, demonstrating that the feature in question is present in cutting-edge systems and demonstrating the validity of the corresponding feature category. The solution is therefore suitably appropriate for its intended use. It enables unbiased grouping along dimensions coming from several fields. As a result, it is compatible with how important ideas are generally understood. It also functions as a tool for analysis.

3 Modules Description

Cloud Brokerage Framework: Private cloud platform is established by letting out the virtual machines the cloud brokers. Depending on the condition of its private cloud, the cloud broker decides how to handle service requests from customers, whether they are carried out internally or sent to public clouds. The private cloud's internal resources will be used to fulfil inbound service requests if any VMs are available. However, the cloud broker will resubmit the incoming service requests to public clouds if there are no VMs available and they cannot be handled quickly in the private cloud. The cloud broker loses these customers' business.

3.1 Preference Matching

Users can specify their choice for a buddy as a consumer and a supplier using a straightforward numerical preference matching interface. The preference of users for their friend increases with value. Negative values indicate a lack of interest in interacting with that friend, whereas zero indicates no preference. Several friends receiving the same value suggests that they are uninterested in one another. When preferences are allocated, they are recorded in the database and utilized to create a general model of preferences for allocation involving the user.

3.2 Optimal Price

Our goal is to determine the best pricing that will maximize earnings. It must be determined how to maximize profit in such a way that we cannot acquire

a closed-form solution but can achieve the numerical solution. We demonstrate the impact of pricing and server size n on the net profit over a discrete period of time. The profit is calculated using the precise PL. As the number of connected virtual machines reaches a particular scale, the additional costs caused by adding more virtual machines begin to outweigh the increased revenue from using the SPQT technique. As a result, the overall profit first rises and then falls. Also, the data demonstrates that the best profit and best pricing are all tied to the maximum. A higher max will result in a higher optimal price and more profit for a given system size.

3.3 Optimal Size

In order to maximize profit, n must be chosen in a way that prevents us from obtaining its closed-form expression. We may then use the bisection method to get n 's numerical answer. In we show how the price and system size n affect the net profit in a given time unit. The VM sales price is initially low, and when the price rises, revenue rises as well. Although the cost is raised in line with the increase in sales, the cost increase is less than the increase in revenue, therefore the overall profit is on the rise. Beyond a particular price, raising the VM sales price is no longer able to boost the overall profit since the rising cost begins to outpace the rising revenue. Hence, the profit shows a decreasing trend.

3.4 Resource Allocation

At the social clearing house, the standard procedure for scheduling donations is to first identify an appropriate recipient who has contact with the user. The list of friends for a specific user is used as a guideline for identifying the list of all donors. The preferences of the consumer for each potential friend are then calculated by obtaining preferences from the database. The preferences of the recipient, as an end user, are also computed for each of these friends. To find the best match, this data is then combined and uploaded to the matching service. Using Seattle's resource acquisition techniques, the social clearing section tries to obtain any available information in order to complete the task. The entire process must be repeated if, at the time of allocation, the selected supplier is not accessible. The user is shown the VMs they have been assigned following the allocation procedure. Users can view who is presently utilizing their resources, renew reservations for them, or cancel them using this interface.

4 Technology

The source code for the Java Programming Language is written using plain text files with the .java extension. This source code is, then, compiled as class files using java compiler. Instead of native processor code, a class file includes bytecodes which are the machine language¹ of the Java Virtual Machine (Java VM). The Java Launcher Tool subsequently starts an instance of the Java Virtual Machine

to run the program. Since Java VM is available on a variety of platforms like Microsoft Windows, the Solaris™ Operating System (Solaris OS), Linux, or Mac OS, the class files can be run on these platforms also. Java HotSpot is a very good platform for improving the performance a program as it can do extra tasks at run time. Some of the jobs, like locating speed bottlenecks and redefining commonly used code parts, fall under this category. In response to a Web client request, software developers are enabled to generate HTML, XML and similar documents based on dynamically using JavaServer Pages (JSP which is a Java technology. Embedded Java Script and specific pre-defined actions are enabled in to static content using this technique. Further XML-like tags, termed as JSP actions, are added to the JSP syntax and are used to call built-in functions. This also enables the development of JSP tag libraries, which function as extensions to the basic HTML or XML tags. Using tag Libraries, expansion of the functionality of a web server independent of any platform can be developed. A JSP compiler converts JSPs into Java Servlets. A JSP compiler may provide byte code for the servlet directly or it generates a servlet in Java script. JSPs can also be interpreted instantly, which speeds up reloading of updates. With JavaServer Pages (JSP) technology, dynamic web content may be produced easily and quickly. Rapid creation of server- and platform-independent web applications is made possible by JSP technology. Servlets are CGI programming's replacement in Java technology. These are applications that create Web pages and run on a web server. For a variety of reasons, it is beneficial to develop web pages on the fly. The user-submitted data form the basis of the Web page. The data changes frequently. For instance, systems that handle orders for e-commerce sites and search engine results pages are generated in this manner. The website makes use of information from corporate databases or other sources; for example, a weather report or news headlines page may be dynamically created, returning an older version of the page if it is still up-to-date. Use this, for instance, to create a Web page for an online store that displays the current prices and quantity of inventory. Because a servlet is a Java class, a servlet engine must execute servlets in a Java virtual machine. When a servlet is first requested, or alternatively as soon as the servlet engine is started, the servlet engine loads the servlet class. The servlet stays loaded to handle new requests until it is explicitly cleared or the servlet engine is shut off. One of the striking feature of some Web servers, including Sun's Java Web Server (JWS), W3C's Jigsaw, and Gefion Software's LiteWebServer (LWS) is the availability of a built-in servlet engine. A servlet engine add-on module is necessary for other Web servers, including Netscape's Enterprise Server, Microsoft's Internet Information Server (IIS), and Apache from the Apache Group. All servlet requests are intercepted by the add-on, which then executes them and sends the results back to the client via the Web server. The WAI CoolRunner by Gefion Software, WebSphere by IBM, JRun by Live Software, and ServletExec by New Atlanta are a few examples of servlet engine add-ons. Sun's official Servlet allows one to download the Java Servlet Development Kit (JSDK) and it has the features of all Servlet API classes and a basic servlet-enabled Web server. Most MySQL use cases are covered in the MySQL Reference Manual. Both MySQL

Community Server and MySQL Enterprise Server are covered by this manual. If the manual is unable to provide the solution(s), MySQL Enterprise, which offers extensive support and services, can be purchased to receive assistance. Moreover, MySQL Enterprise offers a large knowledge base library with hundreds of technical articles that address complex issues related to well-known database topics including performance, replication, and migration. An economical family of high-performance, database products, is created and supported by MySQL AB. The premier product of the company is "MySQL Enterprise," a comprehensive collection of software that has been proven in production and comes with tools for proactive monitoring and first-rate support. One of the widely used open-source database program worldwide is MySQL. Some of the largest and fastest-developing businesses in the world, including market leaders like Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, and Booking.com, use MySQL to support their high-volume Web sites, mission-critical systems, and packaged software. With operations across the globe including offices in the US and Sweden, MySQL AB addresses both the needs of business clients and open-source principles. Java programmers can use the Java Database Connectivity (JDBC) framework. to create applications that can access data held in databases, spreadsheets, and flat files. No matter what database management system is employed to maintain the database, JDBC is frequently used to connect a user program to a "behind the scenes" database. JDBC is cross-platform in this sense. This article gives an overview and sample code that shows database access from Java program and utilize the JDBC API classes, which can be downloaded for free from Sun's website. A data source is a database that another software relates to. Several data sources, including Microsoft and Oracle products, already adhere to the Open Database Connectivity standard (ODBC). Several old C and Perl program link to data sources using ODBC. Many of the similarities between database management systems were unified through ODBC. JDBC advances the level of abstraction by building on this capability. Java program can now connect to ODBC-capable database program thanks to JDBC-ODBC bridges.

5 Testing

Errors are located by conducting appropriate tests. The performance of separate components like subassemblies, assemblies, and/or a finished product. Software testing is essential to satisfy user expectations and to ensure that it complies with the specifications. Some of the most important types of testing are given below

5.1 Unit Testing

Based on the design of the test cases for unit testing, testing of the internal program logic is done to ensure that it is accurate and that program inputs result in valid outputs. Verifying the internal code flow is a highly challenging process. For validation, each component of software must be evaluated separately.

5.2 Integration Testing

Integration tests are run to make sure that all of the program are functioning as one cohesive system. Even if each segment has successfully completed unit testing, the test will determine whether the combination of the components is accurate and reliable. This test is crucial to determining the program's dependability and any integration-related problems.

5.3 Functional Testing

Functional tests show that the components being tested are functional and satisfy all business and technical requirements, as well as those listed in the system documentation and user manuals. Functional tests are planned and created with a focus on requirements, key functions, or unique test cases. Additionally, thorough data field coverage, accepted procedures, adhering to protocols, and business process flows must all be taken into account during testing.

5.4 System Testing

This method of testing is used to verify whether the software system matches with standards requirements of system testing. It assesses a configuration to provide predictable and known results. The configuration-oriented system integration test serves as an illustration of system testing. System testing is predicated on process flows and descriptions, with an emphasis on pre-driven integration points and linkages.

5.5 White Box Testing

This type of testing is carried out by testers who are familiar with the inner workings, structure, and programming language. It is employed to examine the areas that are restricted at the black box level.

5.6 Black Box Testing

This test is done with no prior knowledge of the architecture, language, or internal workings of the module being tested. It is a style of testing where the program under test is treated as a mystery.

5.7 Integration Testing

The incremental testing of two or more integrated software components on a single platform is known as "software integration testing" and is done to simulate failures brought on by interface problems. The purpose of an integration test is to confirm that software applications, such as software system components or, at a higher level, software applications at the corporate level, are properly interoperable. The test results are positive.

5.8 Acceptance Testing

The testing phase of any project is vital and demands active participation from the end user. Additionally, it ensures that the system complies with the functional requirements. Results of the tests are successful.

6 Results

To increase cloud service reliability, "proactive fault- tolerance technique to enhance reliability of cloud service in a cloud federation context" is utilized. When PFT strategies are compared, the Classification Comparison Framework and the We Share Method employing the KNN Algorithm are found to be the most advantageous. This algorithm was chosen primarily because it can effectively share unused cloud space with other users, resulting in lower costs for short-term consumers who utilize the pooled resources. The user can choose their finest resource by analyzing the pricing and QoS details of the cloud. The short-term users can save money by utilizing the shared resources. Fig.2 shows the code implementation for the system.

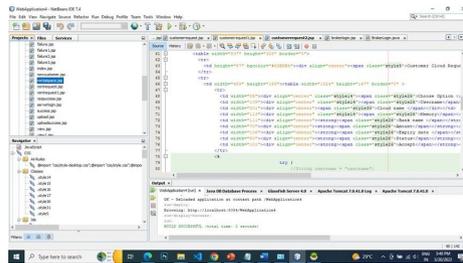


Fig. 2. Code implementation



Fig. 3. Website

7 Results and Discussion

The implementation of this smart street light module demonstrates significant improvements in urban lighting infrastructure management. In the swift fault detection, real-time monitoring and precise fault detection reduce downtime and maintenance costs. The robust database and QR code system ensure accurate location tracking, enabling quick repairs. This eliminates the need for separate location tracking components reduces overall system costs. The QR code system empowers the public to contribute to maintaining urban infrastructure, enhancing community involvement. Optimized street lighting performance conserves energy and reduces light pollution, contributing to sustainable urban development.

TEST CASE ID	TITLE	TEST STEPS	EXPECTED RESULT	FINAL RESULT
1	Android Home Screen	Execute the program	Home screen of the android platform will be visible	Achieved the expected result.
2	Project Home Screen	Click the appropriate icon for running the program	Project splash screen will be visible for milliseconds and will disappear immediately	Achieved the expected result.
3	Sign Up Activity	Fill the required field with Mail ID and mobile number in the correct format	Hassle-free process	Sign Up Activity
4	Login Form Activity	Login using the login credentials	The sign-in button navigates the user to the Main Activity screen.	Easy process
5	Main Activity	Click the view friend's button	The View Friends button navigates the user to the Friend List Screen	Very much user-friendly. Easy navigation to any section
6	Friends List Activity	Click any one of the friends listed	The page navigates to the message activity. The list of friends will be viewed correctly	

7	Message Activity	Enter the message in textbox and click the send button	The message will be sent without any interruption. The textbox must not be empty	
---	------------------	--	--	--

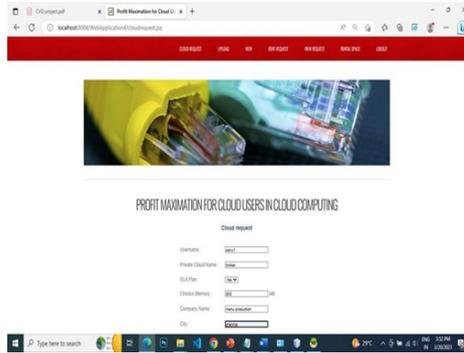


Fig. 4. Cloud request

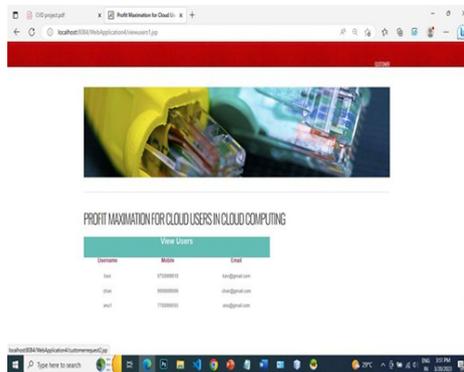


Fig. 5. View users

8 Conclusion

In this suggested approach, we concentrate on configuring a cloud broker and pricing its virtual machines so that its profit can be maximized under the as-

sumption that consumers will pay less. The demand from users, the cost to buy and sell virtual machines, the size of the cloud broker, etc. all have an impact on the profit of a cloud broker. Also, these variables interact, which complicates the analysis of profit. A system that makes it possible for users to share infrastructure resources. Users can run programs on virtualized resources supplied by a cloud broker using our solution. Users can express their preferences for sharing, and matching algorithms are used to enable preference-based socially conscious resource distribution. We Share Method will eventually be integrated with the current cloud brokerage system. With the use of this strategy, individuals can make money by lending out unused cloud resources to peers, both known and unknown. The K-Nearest Neighbour Algorithm (KNN) is used in the WeShare Method to locate users who require cloud resources, and it may be viewed by users who have excess cloud resources.

References

1. Mashayekhy, M. M. Nejad and D. Grosu: Cloud Federations in the Sky: Formation Game and Mechanism. In: IEEE Transactions on Cloud Computing, vol. 3, no. 1, pp. 14-27, Jan.-March 2015.
2. Ray, B., Saha, A., Khatua, S. and Roy, S.: Quality and Profit Assured Trusted Cloud Federation Formation: Game Theory Based Approach. In: IEEE Transactions on Services Computing. <https://doi.org/10.1109/TSC.2018.2833854>.
3. Gill, P., Jain, N., and Nagappan, N.: Understanding network failures in data centers: measurement, analysis, and implications. In: Proc. the 10th ACM Computer Communication Review (SIGCOMM'11), pp. 350-361, 2011.
4. Singh, D., Singh, J., Chhabra, A.: High Availability of Clouds: Failover Strategies for Cloud Computing using Integrated Checkpointing Algorithms. In: IEEE International Conference on Communication Systems and Network Technologies, 2012.
5. Gibson, G. A., Schroeder, B., and Digney, J.: Failure Tolerance in Petascale Computers. In: CTWatch Quarterly, vol. 3, no. 4, November 2007.
6. Egwutuoha, I. P., Chen, S., Levy, D., Selic, B.: A Fault Tolerance Framework for High Performance Computing in Cloud. In: 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012.
7. Miremadi, G., Karlsson, J., Gunnejlo, U., Torin, J.: Two Software Techniques for On-line Error Detection. In: Twenty-Second International Symposium on Computing & Processing (Hardware/Software), <https://doi.org/10.1109/FTCS.1992.243622>.
8. Gizopoulos, D., Psarakis, M., Adve, S. V., Ramachandran, P., Sastry Hari, S. K., Sorin, D., Meixner, A., Biswas, A., Vera, X.: Architectures for Online Error Detection and Recovery in Multicore Processors, March 2011.
9. Bala, A., Chana, I.: Fault Tolerance - Challenges, Techniques and Implementation in Cloud Computing. In: IJCSI International Journal of Computer Science Issues, vol. 9, Issue 1, no. 1, January 2012.
10. Malik, S., Huet, F.: Adaptive Fault Tolerance in Real Time Cloud Computing. In: IEEE World Congress on Services, 2011.
11. Zhao, W., Melliar-Smith, P. M., Moser, L. E.: Fault Tolerance Middleware for Cloud Computing. In: IEEE 3rd International Conference on Cloud Computing, 2010.

12. Okorafor, E.: A Fault-tolerant High Performance Cloud Strategy for Scientific Computing. In: IEEE International Parallel & Distributed Processing Symposium, 2011.
13. Yi, S., Kondo, D., Andrzejak, A.: Reducing Costs of Spot Instances via Checkpointing in the Amazon Elastic Compute Cloud. In: IEEE 3rd International Conference on Cloud Computing, 2010.
14. Singh, D., Singh, J., Chhabra, A.: Evaluating Overheads of Integrated Multilevel Checkpointing Algorithms in Cloud Computing Environment. In: I. J. Computer Network and Information Security, 2012, 5, 29-38.
15. Yi, S., Andrzejak, A., Kondo, D.: Monetary Cost-Aware Checkpointing and Migration on Amazon Cloud Spot Instances. In: IEEE Transactions on Services Computing, vol. month 201X.
16. Sancho, J. C., Petrini, F., Johnson, G., Fernández, J., Frachtenberg, E.: On the Feasibility of Incremental Checkpointing for Scientific Computing. In: Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04), 2004, IEEE.
17. Song, J., Choo, H. K., Lee, K. M.: Application-level load migration and its implementation on top of PVM. In: Concurrency: Practice and Experience, vol. 9(1), 1-19 (January 1997).
18. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., Buyya, R.: CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. In: Software: Practice, 2011 - Wiley Online Library.
19. Buyya, R., Ranjan, R., Calheiros, R. N.: Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. In: International Conference on High Performance Computing & Simulation, 2009. HPCS '09, pp. 1-11.
20. Menychtas, A., Konstanteli, K. G.: Fault Detection and Recovery Mechanisms and Techniques for Service Oriented Infrastructures. In: IGI Global, 2012.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

