Asian Journal of Information Technology 15 (9): 1468-1471, 2016
ISSN: 1682-3915

Optimal Energy Efficient Scheduling in Public Cloud Networks

S. Muthurajkumar, M. Vijayalakshmi and A. Kannan
Department of Information Science and Technology, College of Engineering
Guindy, Anna University, 600025 Chennai, India

Abstract: In a cloud computing, heterogeneous multi-core server processors are present across clouds and data storage centers. Therefore, the total performance of the cloud system can be optimized by providing effective and secured storage and retrieval methods. This paper proposes a new algorithm to optimize power and performance based on load distribution and balancing methods for cloud computing. The proposed algorithm performs performance optimization using speed, time, energy and security. This method has been implemented in a cloud environment and the efficiency of the proposed scheme is compared with the existing schemes and it is found the storage and energy are optimized.

Key words: Energy efficient, data dynamic operation, cloud computing, schemes, India

INTRODUCTION

Cloud computing provides infrastructure, platform and software as a service based on the use of the internet. Moreover, cloud is a distributed network in which a centralized data center provides services to various clients. These services are grouped in the cloud and can be accessed by subscribing through the internet. However, security mechanism for cloud computing are not mature enough to protect the data stored in the cloud. Hence, it is necessary to propose efficient methods for providing security to the data stored in the cloud server.

Storage of large data in cloud using effective storage techniques reduces the maintenance cost and provides location transparency. Hence, a security mechanism which addresses the security challenges is proposed in this paper. Moreover, this work focuses on the proposal of security and power optimization techniques for cloud networks. It also proposes new scheduling algorithms and hence this study focuses on two aspects namely power optimization and optimal scheduling with security.

MATERIALS AND METHODS

In the past, many authors have discussed about data security storage and scheduling. Ren *et al.* (2012), proposed a design which allows users to audit the cloud storage. After auditing, the result ensures strong cloud storage correctness. Their design also supports safe and effective dynamic data manipulation operations. Wang *et al.* (2009), proposed data dynamics for storage security in cloud computing which is single client auditability. The disadvantage is that although all the schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed.

Cevik *et al.* (2012), proposed a power aware routing protocol which considers nodes with higher energy and also the shortest path for performing effective routing. Yang and Jia (2013), proposed a new privacy preserving audit protocol for cloud data storage. Even though this model reduces a considerable amount of cost and provides data security, it consumes a lot of time for dynamically auditing the data stored in the cloud database. Some of the works proposed for handling data that are stored in cloud securely with provable data possession in the server are discussed in the literature (Ateniese *et al.*, 2007; Wang *et al.*, 2011, 2012; Muthurajkumar *et al.*, 2015; Zhu *et al.*, 2012; Tang *et al.*, 2012).

In this study, we have considered a private cloud platform to carry out the implementation. This algorithm were implemented and tested with the same networks parameters which are used in existing works. From the experiments, it is proved that this proposed algorithm are more energy efficient and secured than the related algorithms.

System architecture methods: The architecture of the system proposed in this paper is shown in Fig. 1. It consists of seven components. In this model, user interaction is performed through the user interface module. It accepts user queries and validates them using an intelligent agent. In addition, they are checked for access control if the key checking is successful.

Corresponding Author: S. Muthurajkumar, Department of Information Science and Technology, College of Engineering Guindy, Anna University, 600025 Chennai, India

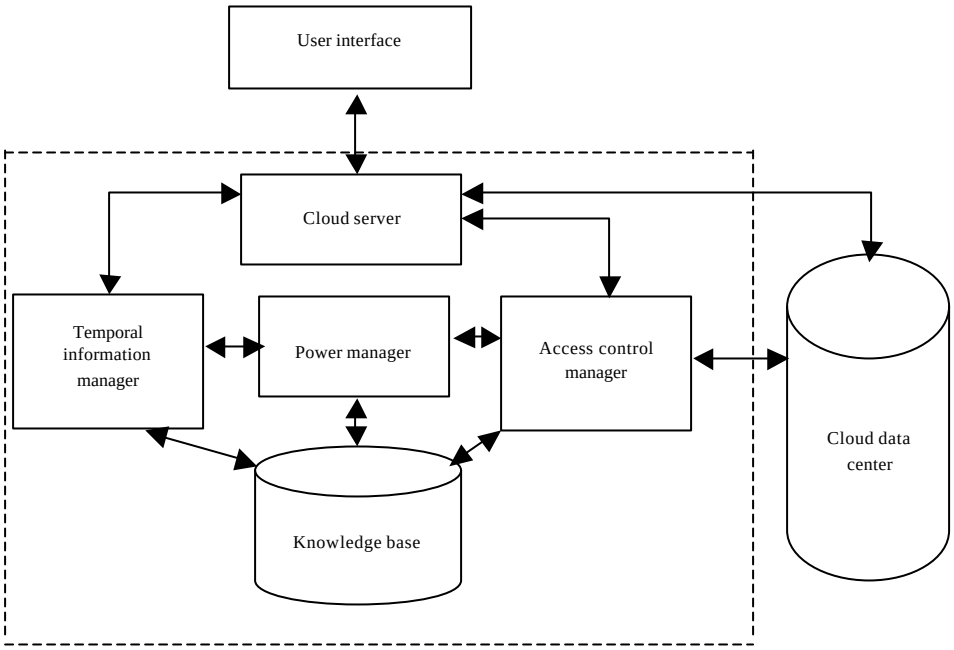

Fig. 1: System architecture

The cloud server manager receives queries from the user interface and passes them to the other module such as access control manager or temporal information manager for key verification if the user's history is doubted. The rule base is used by the power constraint manager which has a collection of active and passive rules and event details. Active rules are automatically fired in response to abnormal events under the conditions defined. The power manager is responsible not only for power management but also for role management activities including inserting, deleting and updating rules based on agent's feedback. The knowledge acquired from the experts on system conditions is stored into the rule base after proper validation. (Fig. 1)

The role data consists of a collection of roles designed by the experts and system designers based on the requirements. It helps the users to store and retrieve large amount of data in various locations. Finally, it supports the cloud database manager present in the cloud data center for performing data manipulation by provides facilities for effective storage and retrieved.

The proposed energy efficient scheduling algorithm methods: In this research, a new algorithm called Energy Efficient Scheduling Algorithm (EESA) has been proposed for improving the performance. This algorithm has been implemented to assign and manipulate roles periodically by considering temporal constraints in order to enhance the security. The proposed EESA Constraints consists of two mechanisms with Temporal and without Temporal.

In this phase, n number of jobs which are submitted through the user interface and s number of servers available in cloud data storage center are considered. In this system, the clients having jobs to be executed communicate a request to the server for executing their job. The major assumption made in this phase is that the numbers of jobs are less than the number of server to due to aggregation. In this algorithm, n clients are denoted by x_1, x_2... and m servers are denoted by y_1, y_2... . The behavior of the system is defined function using the function $f(x,y_i) = a_0+a_1x+a_2x^2+...+a_n$. The energy optimization is performed by obtaining the maxima value for this function in the interval [t1, t2]. The steps of this algorithm are as follows:

- Step 1: Check for client requests from the user interface
- Step 2: Aggregate the request and form n group of jobs
- Step 3: Check the number of servers m available in cloud data center in cloud to form M/M/S queue
- Step 4: Check the weather n < s, If so, serves them without sending to queue
- Step 5: If n = m then begin
  - Calculate the mean service rate $\delta f/\delta x$, $\delta f/\delta y$, $\delta^2 f/\delta x^2$, $\delta^2 f/\delta x\delta y$, $\delta^2 f/\delta y^2$
  - Assume $\delta f/\delta x = 0$ and $\delta f/\delta x = 0$ and calculate the x and y values
  - Solve the (x, y) = (a, b)
  - Calculate $\delta^2 f/\delta x^2_{(a, b)} = A$; $\delta^2 f/\delta xy_{(a, b)} = B$; $\delta^2 f/\delta y^2_{(a, b)} = C$

- $AC - B^2 > 0$ (then move to the conclusion)
- If $A > 0$ or $C > 0$ the minimum group values
- Else maximum group values

- Step 6: For each client assign a server using one to one mapping by applying the follows
  - If n = s, all the server are busy, then the maximum number of customers waiting in the queue will be (n – s) and the mean service rate $\mu_n = s\mu$.
  - Calculate average arrival rate $\lambda_n = \lambda$ for all n
  - Calculate average service rate

$$\mu_n = \begin{cases} n\mu \text{ if } 0 \le n < s \\ s\mu \text{ if } \quad n \ge s \end{cases}$$

  - Find the value of steady state probability of having n customers p_n and p_0

$$p_n = \frac{\lambda_1\lambda_2\lambda_3\ldots\lambda_{n-1}}{\mu_1\mu_2\mu_3\ldots\mu_n} Xp0, \text{for} n \ge 1.p0 = \left[1 + \sum_{n=1}^{\infty} \frac{\lambda_1\lambda_2\lambda_3\ldots\lambda_{n-1}}{\mu_1\mu_2\mu_3\ldots\mu_n}\right]^{-1}$$

  - Find the average number of customers in the Queue (L_q)
  - Probability that an arrival has to wait

$$p(n \ge s) = \frac{\left(\frac{\lambda}{\mu}\right)^2 po}{si\left(1 - \frac{\lambda}{\mu s}\right)}$$

- Step 7: Schedule jobs based on arrival rate and services rate
- Step 8: Apply temporal constraints
- Step 9: Find speed and apply access control policy
- Step 10: If successful output and result

In this algorithm, user inter face is used to choose cloud data center with temporal and hence it will reduce the time complexity of the user. Moreover, it reduces the power within the data center. Since the user interface selects the correct cloud data center using one to one mapping between clients and servers, the servers communicate with any one client on user request. The temporal manager is responsible for monitoring the data centers. The nearest cloud data mapping helps to reduce the power effectively.

RESULTS AND DISCUSSION

In this study, we discuss about the performance analysis of our proposed and existing algorithms have been implemented in JAVA for measuring the actual computation time. From Table 1, it is clear to understand that our proposed EESA with temporal algorithm takes less waiting time in comparison with EESA without temporal algorithms.

Table 1: The waiting time of EESA for various components

No-of CPUS	RAM size (MB)	Hard disk size (GB)	EESA with Temporal Waiting time (ms)	EESA without Temporal Waiting Time (ms)
2	512	10	5677	5687
5	1024	100	4694	4712
5	2048	1000	4109	4120
10	3072	2000	4080	4092
20	4096	4000	1630	1640

Table 2: The number of user request access by temporal power constraint algorithm

Exp. No	No. of User Request Tried	No. of requests denied by EESA with Temporal Algorithm	No. of requests denied by EESA without Temporal Algorithm
1	100	7	8
2	200	9	13
3	300	12	20
4	400	17	27
5	500	21	34

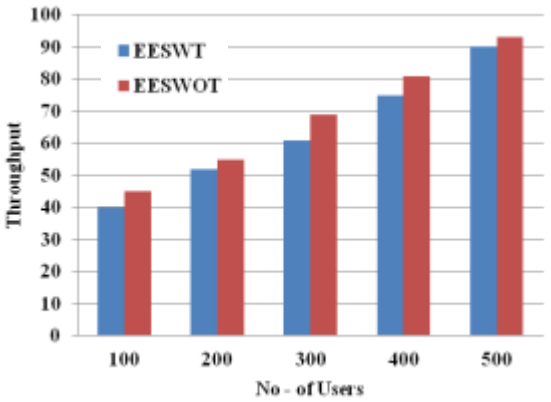

Fig.2: The number of users security level

Figure 2. Shows the number of authorized users throughput level that were permitted by the EESA model, it is observed that the access permission of EESA with Temporal is lower than EESA without Temporal model. Moreover, 5% of less users where denied access in comparison with the existing system and hence the security is enhanced. This is due to the fact that temporal constraints are used effectively to check the abnormal users.

Table 2 shows the number of user requests and requests for access denied by the Round Robin model after checking the identity and temporal power

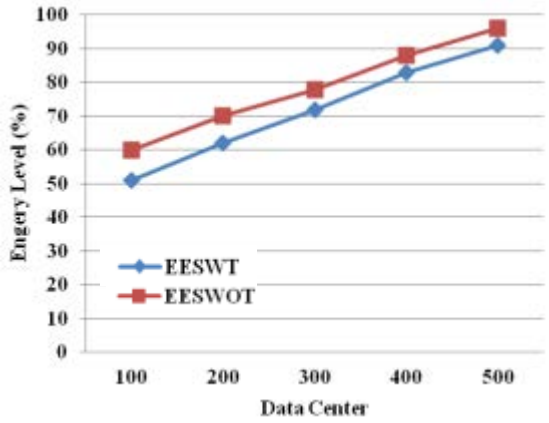

Fig. 3: The number of data centers energy level

constraints. Figure 3. shows the energy take to energy level when they are requested by the user, it can be seen that the energy required for processing revoke is proportional to the number of user sessions and number of users who requested for rollback at a time. Even though the revoking energy increases with the number of users, it decreases after certain limit due to the application logic and presence of intelligent agent who learn the behavior. Therefore EESA with Temporal algorithm consumes less energy when compared to EESA without Temporal Algorithm model.

CONCLUSION

In this research, a new scheduling algorithm for optimizing energy in cloud data centers has been proposed. This algorithm to periodically verify the correctness of the data stored in the cloud server with the help of temporal constraints. In addition to this, the proposed algorithm also supports the existing data storage operations where the data owner is only allowed to modify the data stored in the cloud server. Further works in this scheme is the provision of a rule system for enhancing the decision process.

ACKNOWLEDGEMENTS

The manuscript entitled "Optimal Energy Efficient Scheduling in Public Cloud Networks", S. Muthurajkumar, M. Vijayalakshmi and A. Kannan for the possible publication in Asian Journal of Information Technology. The manuscript has not been previously published, is not currently submitted for review to any other journal and will not be submitted elsewhere before decision is made.

REFERENCES

Ateniese, G., R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, 2007. Provable data possession at untrusted stores. Proceedings of the 14th ACM Conference on Computer and Communications Security, October 29, 2007, Alexandria, Virginia, USA., pp: 598-609.

Cevik, T., A.H. Zaim and D. Yiltas, 2012. Localized power-aware routing with an energy-efficient pipelined wakeup schedule for wireless sensor networks. Turk. J. Electr. Eng. Comp. Sci., 20: 964-978.

Muthurajkumar, S., S. Ganapathy, M. Vijayalakshmi and A. Kannan, 2015. Secured temporal log management techniques for cloud. Procedia Comput. Sci., 46: 589-595.

Ren, K., C. Wang and Q. Wang, 2012. Toward secure and effective data utilization in public cloud. Network IEEE., 26: 69-74.

Tang, Y., P.P. Lee, J. Lui and R. Perlman, 2012. Secure overlay cloud storage with access control and assured deletion. Dependable Secure Comput. IEEE. Trans., 9: 903-916.

Wang, C., Q. Wang, K. Ren, N. Cao and W. Lou, 2012. Toward secure and dependable storage services in cloud computing. IEEE Trans. Serv. Comput., 5: 220-232.

Wang, Q., C. Wang, J. Li, K. Ren and W. Lou, 2009. Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing. In: Computer Security. Backes, M. and P. Ning (Eds.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-642-04443-4, pp: 355-370.

Wang, Q., C. Wang, K. Ren, W. Lou and J. Li, 2011. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst., 22: 847-859.

Yang, K. and X. Jia, 2013. An efficient and secure dynamic auditing protocol for data storage in cloud computing. Parallel Distrib. Syst. IEEE. Trans., 24: 1717-1726.

Zhu, Y., H. Hu, G.J. Ahn and M. Yu, 2012. Cooperative provable data possession for integrity verification in multicloud storage. IEEE Trans. Parallel Distrib. Syst., 23: 2231-2244.