A Heuristic Approach for Preserving Intermediate Data Set Storage on Cloud using Storage & Regeneration Cost Based on User Preference

[1]Suriyalakshmi Kanagarajan and [2]Saranya Vellaichamy
[1]PG Scholar, Department of Computer Science & Engineering
Sri Vidya College of Engineering & Technology, Virudhunagar, Tamilnadu, India
[2]Assistant Professor, Department of Computer Science and Engineering,
Sri Vidya College of Engineering & Technology, Virudhunagar, Tamilnadu, India

Abstract

Cloud computing offers pay-as-you-go model, where users only pay for their resource consumption. Many large applications utilize cloud computing. These applications generate a lot of essential intermediate data set for future purpose. Storing all intermediate data set is not a cost heuristic approach. At the same time adversary may refer multiple intermediate data set to steal the information. Likewise encrypting every part of intermediate data set will increase computation cost for the user. Our experimental results obtained from a large scale evaluation over real data sets and demonstrate the feasibility of molding the cloud storage system. Our proposed system is used to reduce privacy preserving cost and storage cost on cloud and it is useful for a variety of sensitive applications, such as Medical exploration system, banking, and health care system, etc.

Keywords: *Cloud Computing, Resource Consumption, Intermediate Data Set, Cost Heuristic, Cloud Storage, Privacy Preserving Cost, Storage Cost*

I. Introduction

In recent years, Cloud computing has become one of the most discussed IT paradigms [1]. Cloud builds on many of the advances in the IT industry [12] over the past decade and Presents significant opportunities for organizations to shorten time to market and reduce costs. To improving infrastructure on their own cloud, organizations can consume shared resources and storage resources rather than building and operating [27]. The speed of change in markets creates significant pressure on the enterprise IT infrastructure to adapt and deliver [29]. Cloud computing provides fresh solutions to address these changes. As defined by Gartner, "Cloud computing is a style of computing where scalable and elastic IT- enabled capabilities are delivered as a service to external customers using Internet technologies" [1]. Cloud computing system offers high resources and massive storage to the users [27].

Cloud computing where the user accesses programmers and data which is stored in giant data centers "somewhere in the internet cloud" [28]. Medium-sized companies and start-ups will enjoy low-prices options that enable them to use such a sophisticated infrastructure. The ever –increasing mountains of data first of all has to be saved first and where possible using optimized resources. As data has to be available anytime, anywhere, the storage systems used must provide the data worldwide via the internet on a 24 hour-a-day basis. Data has to be saved several times on a redundant basis in order to ensure that it is not lost due to unforeseen circumstances.

IT manufactures have always been fairly inventive when introducing additional protective functions for data storage around disk arrays [6]. All these procedures are based on the idea of redundancy: keep everything twice or more where possible. This includes at hardware level clusters and girds which means that specific hardware is available several times so that the second device can run with an identical configuration and identical data should there be any problems. The transfer between clusters and girds is based on scaling.

Scientific cloud workflows are deployed in cloud computing environment [18], where all the resources need to be paid for use. The scientific cloud workflow [7] system storing all the intermediate data generated during workflow executions may cause a high storage cost, if we delete all the intermediate data sets and regenerate they every time when needed, the computation cost of the system may well be very high too [11].

In this paper, we propose a heuristic approach for the intermediate data storage of scientific cloud workflows to reduce the overall cost of the system [14]. Intermediate data sets in scientific cloud workflows often have dependency. Along workflow execution they are generated by the task, a task can operate on one or more data sets and generate new one(s) [18].These generation relationship is a kind of data provenance [10]. we created Data Dependency Graph(DDG) [19] is to identify the sensitive data set & usage frequency of the data set based on user preference. The data set will be stored / regenerated based on the user preference & usage frequency. The cloud users identify which data set to be stored and which data sets to be regenerate and also provide privacy by encrypt/decrypt the selected data sets [22]. We design heuristic approach to reduce privacy preserving cost. The storage cost is based on the size of the data set and the regeneration cost is based on the time to regenerate the data set from stored predecessor [2]. The regeneration cost of the data set is less than the storage cost and all the intermediate data sets not stored in cloud as shown in Table 2.

II. BACKGROUND AND RELATED WORK

The related work aims in reviewing the existing literature related to the storage and privacy preserving for intermediate data set with a cost effective approach [10]. The existing work methodology in cost effective approach for large applications in the cloud was examined [13]. Several researches discussed about the issue of trade-off between computational cost [11] and storage cost. The cost effectiveness of privacy preserving for cloud was also analyzed.

A Highly Practical Approach toward Achieving Minimum Data Sets Storage Cost in the Cloud by Dong Yuan et al states a novel highly cost effective and practical storage strategy that can automatically decide whether a generated data set should be stored or not at runtime in the cloud [2]. Users can deploy their applications in unified resources without any infrastructure investment. Also excessive processing power and storage can be obtained from commercial cloud service providers. With the pay-as-you-go model, the total application cost in the cloud highly depends on the strategy of storing the application data sets in the cloud may data set in a high storage cost, because some data sets may be rarely used but large in size; in contrast, deleting all the generated data sets and regenerating them every time when needed may data set in a high computation cost [11]. In a commercial cloud computing environment [20], service providers have their cost models to charge users [1].

In general, there are two basic types of resources in the cloud: Computation and Storage. Popular cloud services providers’ cost models are based on these types of resources. For example, As of 2014, Amazon charged about $0.131/hour ($9.7/month) for smallest micro instance virtual machine learning Linux (or) windows [1]. Storage optimized instance cost $6.820/hour. Data transfer rate $0.12 per GB depending on the direction &month volume inbound data transfer is free. Foster et al [10] proposed a comparison of grid computing [14] and cloud computing. As far as their consent the large applications deployed in the cloud could be more heuristic and cost – effective than the grid. Jawwad Shams et al [13] discussed about the Requirements, Expectations, Challenges, and Solutions in Data intensive Cloud [3]. K.K.Muniswamy-Reddy et al [14] proposed that provenance of the data is vital information when the data stored in the cloud and also discussed the properties of provenance [5]. Adams et al [2] modeled a framework to represent the trade-off between computation cost and storage cost. The author has also proposed an idea to store the input, the process, and the data sets instead of storing the entire data. Deelman et al [7] introduced a model to store only some of the popular intermediate data sets that can save the cost in comparison to always regenerating them from the input data. Security parameters are not in their scope.

At Present, Encryption is used in most of the research to provide data privacy. Ciriani et al [5] introduced the strategy of combining data fragmentation and encryption to encrypt only part of the data. The cloud application handles large data set but data fragmentation technique is not heurisitic for handling large data. Banjamin C.M.Fung et al [3] suggest the anonymization technique reduce the computation required for encryption and decryption instead of standard encryption technique. Anonymization of data alone is not a heuristic approach to provide privacy for the data. Xuyun Zhang et al [15] proposed the strategy that encrypts only part of the data sets. Encryption is based on privacy leakage constraint [4]. Storage and computation parameters are not in their scope.

III. LIFE CYCLE OF COST COMPUTING IN CLOUD

The data sets storage strategy is generic [6]. It can be used in any computation [11] and data intensive applications with different price models of cloud services [17]. To evaluate the cost effectiveness of storage strategy in the Heuristic system, it was compared with different storage strategies such as "store-all, store-none, usage based" strategies. Store-all strategy stores all the intermediate data set [16]. There is no need for regeneration here. Store-none strategy Discard all the data set and there is no storage cost for user [2]. Usage based strategy only consider the usage frequency Parameter for taking decision to either store or discard. The proposed work reduces the cost for cloud user by selecting intermediate data set based on heuristic algorithm. The selected data set is stored in the cloud for future reference.

The storage cost and regeneration cost are calculated dynamically for the intermediate data set. Each iteration defines the computation of IDS at time t1, t2… tn. The Store-all stores all the intermediate data set generated during the computation [11]. The cost of Store-none gradually increases when the data sets are frequently accessed.

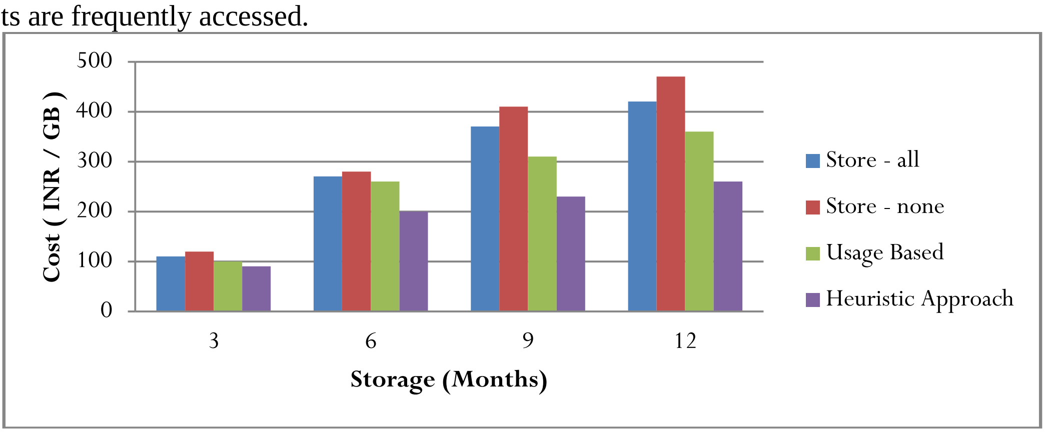


Figure1. Storage Cost Comparison

The usage based storage strategy only stores the popular data set which is frequently accessed by the system [16]. The Heuristic approach stores data set based on cost factors and user constraints. The comparison reports of all the four strategies are shown in Fig1.

IV. PROBLEM DEFINITION

The cost of the cloud user depends upon the strategy of storing and optimizing the computation. Finding a good strategy to selectively store the appropriate data sets and encrypting the necessary part of the data sets is a challenging task. The intermediate data sets are dependent to each other. Regeneration of intermediate data sets not only affects the particular data sets but also the dependent data. Various scenarios are also to be considered before selecting the appropriate data sets. Assume Intermediate data set IDS_1 generated during the computation of IDS. Consider SC_i is the storage cost of IDS_1 and. RC_i is the regeneration cost of IDS_1.

Setting 1:

If SC_i is greater than the RC_i, then the user cannot simply discard the data set and regenerate it. The access frequency of IDS_1 is an important factor here to store or regenerate the data set. If IDS_1 was frequently accessed then the regeneration cost was as many times of accessed. So the access frequency of the data set is to be considered before regenerating the data set.

Setting 2:

If SCi is greater than the RCi and the data set is also rarely used. By this condition we cannot simply discard the data set. Because sometimes users doesn't tolerate delay caused by the regeneration of data set and ready to store the data set at any cost. So the preference of the user also considered for storing the data set.

Setting 3:

Every user has own perspective about their sensitive information. The user preference for sensitive data varies from one user to another. User preference parameter for encryption also to considered.

The tradeoff between storage and computation is the key issue handled in this work. There are various methodologies proposed to store the data efficiently in a static environment [11]. The intermediate data sets stored in the cloud are dynamically changed. For dynamic environment, it is difficult to select the appropriate data set to store. In this work a framework is proposed to select appropriate data to store and to provide privacy in a cost effective approach.

V. A Heuristic Approach For Preserving Intermediate Data Storage

A Heuristic approach for intermediate data Storage and Privacy preserving for Intermediate data sets in Cost effective approach for Cloud Environment provides solutions for handling intermediate data sets in an efficient manner to reduce the cost for cloud user.

Intermediate generated data are the data produced in the cloud computing system while the applications run. In Fig. 2, user runs the application based on the actual data. The generated data produced from the execution of application are mentioned as intermediate data sets IDS_1, IDS_2, …. IDS_n. After the generation of intermediate data sets, relation tree is constructed. The intermediate data sets are sometimes dependent with each other. Relation tree is used to identify the relationship among the generated data set. The successor and predecessor of the data sets are easily found using the relation tree. If P is the parent node of data set R, C and D is the child of R. P is the predecessor of R, C and D are the successor of R. The intermediate data sets are regenerated by the system efficiently using the provenance of the data. Data provenance defines the origin of the data which is more helpful to regenerate the data set from stored predecessor when the regenerated data set was dependent on the stored data set. In a commercial cloud computing environment, the resources are offered by cloud service providers, who have their cost models to charge the users for storage and computation. To calculate the total application cost in the cloud, some important attributes for the data sets are defined.

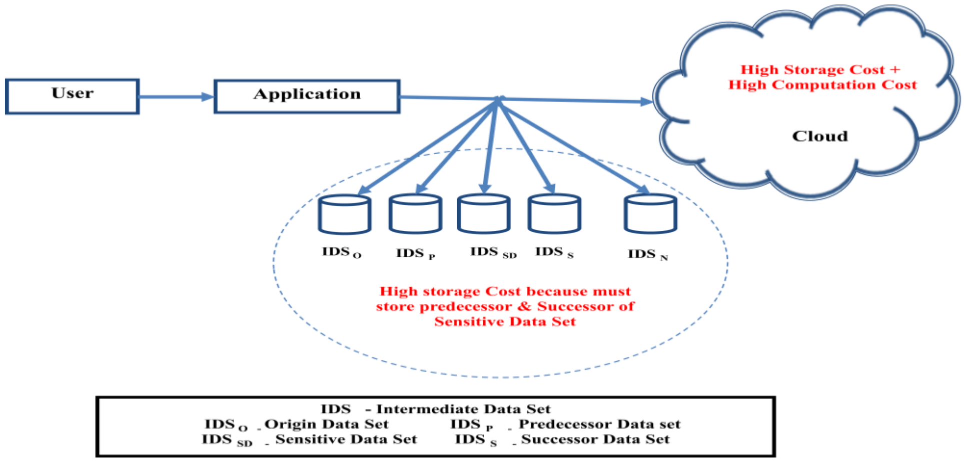

User
Application
High Storage Cost +
High Computation Cost
Cloud
IDS O
IDS P
IDS SD
IDS S
IDS N
High storage Cost because must
store predecessor & Successor of
Sensitive Data Set
IDS - Intermediate Data Set
IDS O - Origin Data Set
IDS P - Predecessor Data set
IDS SD - Sensitive Data Set
IDS S - Successor Data Set

Figure 2. Storage cost in cloud

5.1. Cost Effective model

<RC, SC, UF, PC, UPS2, FL> RC denotes the Regeneration Cost of Data set from its stored predecessors. SC denotes the Cost of Storing Data set in the system per time unit. UF denotes the

Usage Frequency of the Data set. PC denotes the Privacy Cost for the Data set. UPS2 denotes the User Preference for Selecting Sensitive data set. FL denotes the Flag which indicates whether the Data Set is stored or deleted in the system.

IDS_1, IDS_2, IDS_3 … IDS_n denotes the Intermediate Data Set generated from the User Application UA where n is the total number of intermediate Data set.

$$\forall\, IDS_i \in UA, where\ i = 1,2,3 \dots n \qquad (1)$$

In this work, health care application is considered as a working model. All the intermediate data sets generated from the health care application are represented in table structure. The size of the data set is calculated based on the number of rows and columns using (2). The table structure is represented in matrix format. It is represented as

$$Size\ SI = \begin{bmatrix} 0 & \cdots & m \\ \vdots & \ddots & \vdots \\ n & \cdots & m \end{bmatrix} \qquad (2)$$

The storage cost [2] of the intermediate Data set was calculated by multiplying the size of the Data set with the cost per unit size using (3).

$$Storage\ Cost\ SC = \sum_{i=0}^{n} SI_i * CS_i \qquad (3)$$

SI_i denotes the Size of the intermediate Data set and CS_i denotes the Cost for storing the data per unit size. The regeneration cost of the Data set was calculated using (4) by taking the summation of computational cost of retrieving data from stored predecessor up to the required intermediate Data set.

$$Regeneration\ Cost\ RC = \sum_{i=SP}^{IDS} CC_i \qquad (4)$$

SP denotes the Stored Predecessor of the intermediate Data set. CC denotes the Computational Cost of Data set. The SC and CC are varied with every cloud service provider. In our work, we have taken a random cost for SC and CC. TC denotes the Total Cost of computation and storage resources consumption in the cloud per time unit [6], which is also the cost of running the application in the cloud per time unit.

5.2. Privacy preservation

The intermediate data set are stored in the cloud based on the proposed cost model. The data sets produced are vulnerable to the attacks by other users. Cryptographic technique encryption and decryption is used to solve the privacy issue. Existing approach encrypt the entire data set before storing into the cloud. Encrypting all the data set is not a heuristic approach because encryption and decryption require some computation. The user cannot perform any operation easily on the encrypted data set. The frequent access of the data set increases the computational cost.

Encryption and decryption requires computational resources too. The cost of encryption and decryption is combined together. It is denoted as Privacy Preservation Cost (P2C) using (5) [15]. Intermediate Data Set IDS is divided into IDSE which denotes the encrypted data set and IDSU which denotes the unencrypted data set. IDS= IDSE U IDSU and IDSE ∩ IDSU = Ø.

$$Privacy\ Cost = \sum_{di \in D} Si * Fi * Ci \qquad (5)$$

The privacy cost is calculated based on the size of the Data Set Si, Frequency of usage Data Set F_i and Cost per unit size Ci using (5).

5.3. Privacy leakage constraint

The application processed in the cloud may contain sensitive data but all the data in the intermediate data set are not sensitive data. Identifying sensitive attribute alone is not sufficient to preserve the privacy of the data. The user also analyzes the quasi identifier in the data set. Quasi identifier is not a unique identifier. The data collected from many quasi attribute leads to the leakage of privacy [4]. At some time user don’t want to encrypt the data set for reducing the computational cost and delay. The privacy leakage of the intermediate data set was calculated based on sensitive information and quasi identifier [4].

Let Q be the set of quasi attribute and S be the set of sensitive attribute in the intermediate data set IDS. the threshold value for privacy leakage.

$$\forall\, q \in Q \text{ and } \forall\, s \in S \qquad (6)$$

The Privacy Leakage of the intermediate data set was calculated using the entropy method in (7).

$$PL_{IDS} = \mathrm{H(S, Q)}\ \text{-}$$

$$H(S,Q) = \log(|Q|.\,|S|) \qquad (8)$$

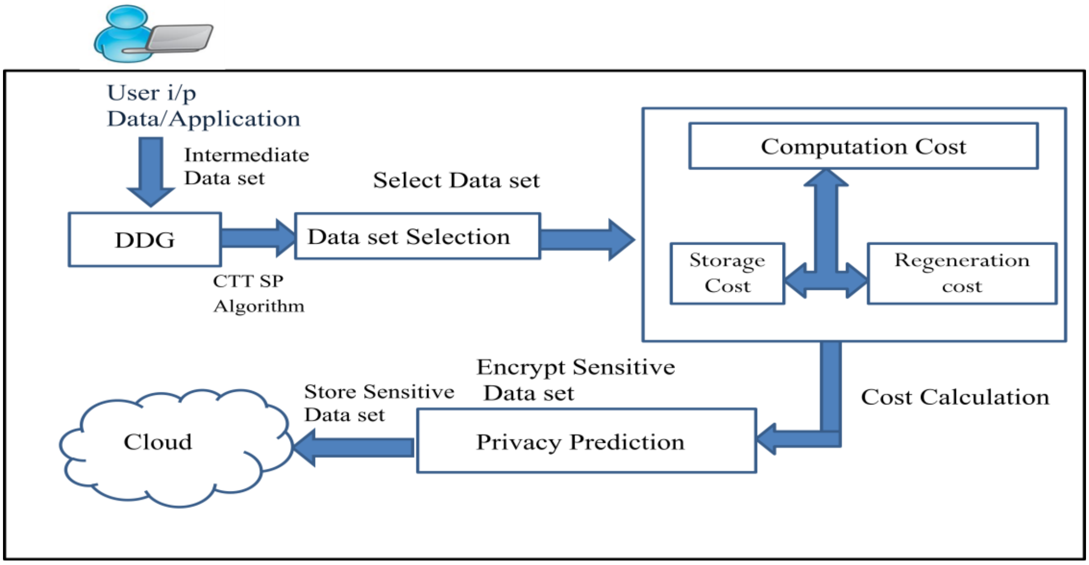
$$H'(S,Q) = -\textstyle\sum_{q \in Q\ s \in S} p(s,q).\log(p(s,q)) \qquad (9)$$

H(S, Q) is the entropy of the sensitive data set and quasi identifier. The value of H(S, Q) is calculated based on the total number of quasi identifier and total number of sensitive data set using (8). H′(S, Q) is calculated based on the probability of sensitive data set and quasi identifier using (9).

VI. Methodologies

The total application cost in the cloud depends on the storage cost and computational cost. Fig. 3 represents the architecture of the proposed work. The user process the application via internet. Lot of intermediate data sets is produced during the computation [11]. The data dependency graph is generated based on the relation between the generated data sets [19].

The storage cost and regeneration cost is calculated based on the user preference, usage frequency and data sets are selected based on the cost efficient constraints. The data sets should be securely stored in the cloud, so it was encrypted and then stored in the cloud.



.

Figure 3. Storage Cost Calculation

Instead of encrypting whole data set, partial encrypting of information is done based on the user preference and privacy leakage constraint [4].

6.1. Heuristic Algorithm

Input: Intermediate data sets generated from application

Output: Selected Intermediate data sets and partial Encryption

```
for ( Every dataset di )
if ( flag is  “not stored” )
if ( UP is set ) or ( UF greater than TV )
TC = SC + PC; // SC based on size
Flag set to “stored”;
Goto L1;
else if ( SC greater than RC )
Find stored predecessor for di;
```

```
TC = UF * ( RC + PC );
flag set to “not stored”;
end if
if ( Flag is “Stored” )
L1: if ( SF set to true ) Find SP from user;
Encrypt (SP);  // Based on TV
end if
end if
end for
```

Where,

PC = Privacy Cost	UP = User Preference	TV = Threshold Value
RC = Regeneration Cost	UF = Usage Frequency	SF = Sense Flag
SC = Storage Cost	SP = Sensitive Parameter	SP = Sensitive Parameter

Cost Transitive Tournament Shortest Path (CTT-SP) Algorithm was used in the existing works. In the proposed Cost Effective model, CTT-SP algorithm was extended with enhanced parameters in addition to the existing ones.

a. Generation cost from direct predecessor
b. Storage cost of data set
c. Usage frequency
d. Provenance of data
e. User preference for storage
f. User preference for sensitive data
g. Privacy Preservation Cost.

The SC is the storage cost of the intermediate data set [2]. The SC is dependent on file size and cost accounted by the service provider for resource usage. The RC is regeneration cost of intermediate data set, UP is the user preference for storing the data set, UF is the access frequency of the data set, SF is the sensitive flag which denotes whether the information is sensitive or not. SP denotes the sensitive parameter that needs to be encrypted. TV denotes the threshold value for usage frequency.

VII. EXPERIMENTAL SETUP

The open source cloud environment tool Eucalyptus is used for cloud management [9] in real time analysis of proposed COST EFFECTIVE model [21]. Eucalyptus is used for managing, scheduling and to communicate with the user, it enables pooling compute, storage and network resources [25]. That can be dynamically scale up or down as application work load change. The storage controller is one of the components of Eucalyptus. The storage controller is similar to Amazon Elastic Block Storage [1]. It is Linux block device that can be attached to a virtual machine. User can create snapshots from Elastic Block Storage Volume snapshots are stored in walrus and made available across availability zones. Walrus allows user to store persistent data, Organize as buckets and objects walrus to create, delete and list buckets; it provides for storing and accessing virtual machine image and user data. The Linux operating system Centos is used. Kernel virtual machine (KVM) is installed on the top of the hardware to virtualized the hardware resources.

The data sets storage strategy is generic [6]. It can be used in any computation and data intensive applications with different price models of cloud services. To evaluate the cost effectiveness of storage strategy in the COST EFFECTIVE system, it was compared with different storage strategies such as “store-all, store-none, usage based” strategies. Store-all strategy stores all the intermediate data set [16]. There is no need for regeneration here. Store-none strategy discard all the data set and there is no storage cost for user. Usage based strategy only consider the usage frequency parameter for taking decision to either store or discard. The proposed work reduces the Cost for cloud user by selecting intermediate data set based on Heuristic Algorithm. The selected data set is stored in the cloud for future reference.

7.1. Dataset

In Scientific application, a large number of intermediate data sets are involved. The data sets are collected from the National Center for Emerging and Zoonotic infection Diseases [24]. The different intermediate data set having on cloud. The intermediate data set associated with diseases. They are “Chikenguniya”, ”Dengue” ,”Ebola”, ”Cancer”,” TB”, ”Anthrax” and ”Headache” etc…we have create ‘N’ number of intermediate data set over 2 categories as the seasonal disease and common disease.

Table 1. Sample Data Set

Disease name	Ebola	Cancer	Tuberculosi
Spread	Monkeys Gorillas Chimpanzee	Lymph Nodes Blood Vessels	Air Tb infection cough sneeze
Illness Germ	Ebola HF Genus: Ebola Virus Family: Filoviridae	Simian t- Lymphotropicvirus Genus: Delta retrovirus Family: Retroviridae	Mycobacterium Tuberculosis
Warning Sign	Fever(greater than38.6ºc) Headache Muscle Pain Diarrhea Vomiting Abdominal Pain	Ulcerates(colorectal cancer) Cough(lung cancer) Fever Tired Enlarged liver Enlarged spleen	Fever Chills Night sweats Loss of appetite Weight loss Nail clubbing
Laboratory Finding	Elisa test IGM Elisa Polymerase chain reaction Virus isolation Immunohistochemistry test	X-rays Ionizing radiation Physical examination Medical imaging Universal screening Selective screening	Skin test Chest x-ray Sputum analysis PCR test
Treatment	Antiviral drugs Intravenous fluids Balancing electrolytes Maintain oxygen status Maintain blood pressure	Surgery Chemotherapy Radiation therapy Hormonal therapy Targeted therapy Palliative care	Antibiotic Ionized Nitroimidazo-oxazine Sutezolid Delamanid Bed aquiline
Prevention	Wearing of protective clothing Avoid blood contact Avoid blood fluids Avoid funeral Avoid contact with bats Avoid nonhuman primates	Environmental factors are Controllable Avoid tobacco Avoid insufficient diet Avoid alcohol Air pollution Background radiation	Do not spend long periods of Time in stuff Use face mask Stop transmission Drug resistant tb
Control	Careful hygiene Avoid funeral Avoid burial Using infection control measures	Don’t use tobacco Eat a healthy diet Maintain healthy weight Protect yourself from sun Get immunized	TB infection control plan Prompt detection of infections patients Airborne precaution

Health care system has moved data storage into cloud for economical benefits. Original data sets are encrypted for confidentiality. Data users like governments or research center access part of original data sets after anonymization.

Intermediate data sets generated during data access are retained for data reuse and cost saving shown in Table 1. Two independent generated intermediate data sets are common disease and seasonal disease .The seasonal diseases are spread some particular season. That time frequency/preference of accessing data set to preserve privacy because this can less cost.

7.2. Methods compared

In the proposed system, the health care application is taken for the evolutionary perspective and its intermediate datasets are computed at various time integral to show the simulation data set In the existing system, the users encrypt all the data set before storing the IDS. The Cost Effective system reduces both the storage cost by selecting appropriate dataset and privacy preserving cost by encrypting only partial data set. The cost of encrypting all the data set is compared with the optimized encryption.

Table 2. Heuristic Algorithm Compared with Existing Algorithm

Modified storage cost calculation algorithm	Existing algorithm
The data sets will be "stored / regenerated" based on user preference and usage frequency.	If particular intermediate data set is no more wanted, The user has to delete all the intermediate data sets.
To "minimize the storage cost" & privacy preservation cost of intermediate data sets in an effective manner.	In cloud Storage & regeneration cost is high because store all intermediate data set and store none intermediate data set.
The DDG is constructed to "identify the sensitive data set & usage frequency of the data set" based on user preference.	The intermediate data dependency graph is constructing to identify sensitive data set; remaining data sets are automatically deleted.
The "Modified Storage Cost Calculation Algorithm" to reduce privacy preserving cost.	Privacy preservation is high because of storing all the intermediate data sets.
The "regeneration cost of the data set is less than the Storage cost", and all the intermediate data sets not stored in cloud.	High Rental for storage even if it is not in use because Upper bound constraints.

The Cost Effective system reduces both the storage cost by selecting appropriate dataset and privacy preserving cost by encrypting only partial data set [4]. The Cost Effective system stores intermediate data set based on cost factors and user constraints [28].

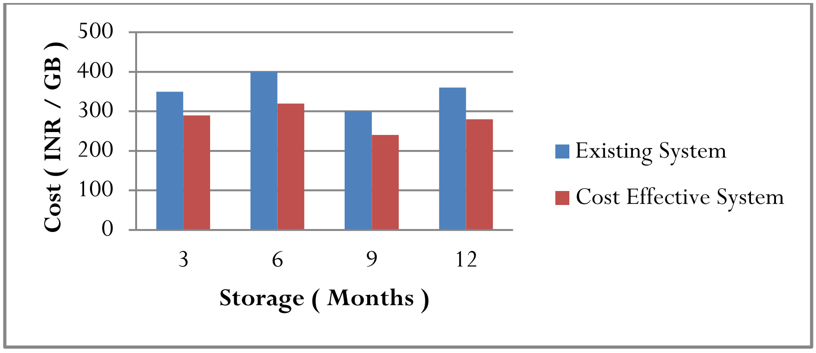


Figure 4.Privacy Cost Comparison

VIII. CONCLUSION

In data and computation intensive applications on cloud, data management is becoming a significant research area. The cloud service provider aims to reduce the cost for cloud user and provide heuristic data management. The user application generates enormous amount of intermediate data set at run time. In existing systems, storing all generated data set in the cloud increases the storage cost shown in Fig 1. Discarding all the generated data set and regenerating them every time whenever it is needed data sets in a higher computational cost. When handling intermediate data set privacy is incorporated by using encryption/decryption technique which also leads to higher computational cost. To resolve this issues Cost Effective is proposed in this work. This encrypts the part of intermediate data sets rather than all for reducing privacy-preserving cost.

IX. FUTURE ENHANCEMENT

In future, Heuristic Approach aims in extending the concept of storage and privacy preserving for intermediate dataset heuristically from single cloud service provider to multiple service provider, when there is a necessity of accessing data from multiple cloud service providers.

REFERENCE

[1] "Amazon Cloud Services" http://aws.amazon.com/, 2014.

[2] Dong Yuan, Yun yang, wenhao Li, Lizhen Cui, Meng Xu and Jinchunchen "A Highly Practical Approach toward Achieving Minimum Data Sets Storage Cost in the Cloud" , IEEE Transaction on parallel and distributed system, june 2013.

[3] Jawwad Shamsi, Muhammad Ali Khojaye, Mohammad Ali Qasmi "Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions", Springer Science and Business Media Dordrecht 2013, April 2013.

[4] Xuyun Zhang, Chang Liu, Surya Nepal, Suraj pandey, and Jinchun Chen "A Privacy leakage upper bound constraint-based approach for cost effective privacy preserving of intermediate data sets in cloud",IEEE Transaction on parallel and distributed system, June 2013.

[5] S.B. Davidson, S. Khanna, V. Tannen, S. Roy, Y. Chen, T. Milo, and J. Stoyanovich, "Enabling Privacy in Provenance-Aware Workflow Systems," Proc. Fifth Biennial Conf. Innovative Data Systems Research (CIDR '11), pp. 215-218, 2011.

[6] V. Ciriani, S.D.C.D. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P.Samarati, "Combining Fragmentation and Encryption to Protect Privacy in Data Storage," ACM Trans. Information and System Security, vol. 13, no. 3, pp. 1-33, 2010.

[7] Dong Yuan, Yun Yang, Xiao Liu, Jinjun Chen "On-demand minimum cost benchmarking for intermediate dataset storage in scientific cloud workflow systems", j. parallel and distributed computing, 2010.

[8] B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," ACM Computing Survey, vol. 42, no. 4, pp. 1-53, 2010.

[9] P. K. Gunda, L. Ravindranath, C. A. Thekkath, Y. Yu, and L. Zhuang, "Nectar: Automatic Management of Data and Computation in Datacenters," in 9th Symposium on Operating Systems Design and Implementation (OSDI2010), Vancouver, BC, Canada, pp. 1-14, 2010.

[10] K.-K. Muniswamy-Reddy, P. Macko, and M. Seltzer, "Provenance for the Cloud," Proc. Eighth USENIX Conf. File and Storage Technology, pp. 197-210, 2010.

[11] I. Adams, D.D.E. Long, E.L. Miller, S. Pasupathy, and M.W. Storer, "Maximizing Efficiency by Trading Storage for Computation," Proc. Workshop Hot Topics in Cloud Computing, 2009. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[12] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the fifth Utility," Future Generation Computer Systems, vol. 25, pp. 599-616, 2009.

[13] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The Cost of Doing Science on the Cloud: the Montage Example," in ACM/IEEE Conference on Supercomputing (SC2008), Austin, Texas, pp. 1-12, 2008.

[14] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in Grid Computing Environments Workshop (GCE2008), Austin, Texas, USA, pp. 1-10, 2008.

[15] Benjamin C. M. Fung, Ke Wang, and Philip S. Yu, Fellow, IEEE "Anonymizing Classification Data for Privacy Preservation", IEEE transactions on knowledge and data engineering, 2007.

[16] Dong yuan, yun yang, Xiao Liu and Jinchun chen "A Cost-Effective Strategy for Intermediate Data Storage in Scientific Cloud Workflow Systems" , Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium.

[17] D. Yuan, Y. Yang, X. Liu, and J. Chen, "A Local-Optimisation Based Strategy for Cost-Effective Data Sets Storage of Scientific Applications in the Cloud," Proc. IEEE Fourth Int'l Conf. Cloud Computing, pp. 179-186, 2011.

[18] D. Yuan, Y. Yang, X. Liu, and J. Chen, "On-Demand Minimum Cost Benchmarking for Intermediate Data Sets Storage in Scientific Cloud Workflow Systems," J. Parallel and Distributed Computing, vol. 71, pp. 316-332, 2011.

[19] D. Yuan, Y. Yang, X. Liu, G. Zhang, and J. Chen, "A Data Dependency Based Strategy for Intermediate Data Storage in Scientific Cloud Workflow Systems," Concurrency and Computation: Practice and Experience, vol. 24, pp. 956-976, 2012.

[20] M. Zaharia, A. Konwinski, A.D. Joseph, R. Katz, and I. Stoica, "Improving MapReduce Performance in Heterogeneous Environments," Proc. Eighth USENIX Symp. Operating Systems Design and Implementation, pp. 29-42, 2008.

[21] H. Khazaei, J. Misic, and V. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queueing Systems," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 5, pp. 936-943, May 2012.

[22] X. Zhang, C. Liu, J. Chen, and W. Dou, "An Upper-Bound Control Approach for Cost-Effective Privacy Protection of Intermediate Data Set Storage in Cloud," Proc. Ninth IEEE Int'l Conf. Dependable, Autonomic and Secure Computing (DASC '11), pp. 518-525, 2011.

[23] B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," ACM Computing Survey,vol. 42, no. 4, pp. 1-53, 2010.

[24] "National Center for Emerging and Zoonotic infection Diseases" http://www.cdc.gov/2014.

[25] "Eucalyptus Open-Source Cloud Computing Infrastructure" http://www.eucalyptus.com/2009.

[26] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, 2010.

[27] L. Wang, J. Zhan, W. Shi, and Y. Liang, "In Cloud, Can Scientific Communities Benefit from the Economies of Scale?," IEEE Trans. Parallel and Distributed Systems,vol. 23, no. 2, pp. 296-303, Feb. 2012.

[28] S.K. Garg, R. Buyya, and H.J. Siegel "Time and Cost Trade-Off Management for Scheduling Parallel Applications on Utility Grids," Future Generation Computer Systems, vol. 26, pp. 1344-1355,2010.

[29] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, ―Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the fifth Utility,‖ Future Generation Computer Systems, vol. 25, pp. 599-616, 2009.

AUTHORS

Suriyalakshmi Kanagarajan is currently pursuing M.E degree in the Department of Computer Science and Engineering at Sri Vidya College Of Engineering and Technology, Virudhungar. She had her Completed Bachelor degree in PSR Rengasamy College of Engineering for Women, Sivakasi. Her more interests include Cloud Computing and Big Data.

Saranya Vellaichamy is working as an Assistant Professor in the Department of Computer Science and Engineering at Sri Vidya College of Engineering and Technology, Virudhunagar. She had completed her Bachelors in Computer Science and Engineering from Kamaraj College of Engg & Tech and Masters in Software Engineering from Anna University, Trichy. And she has a teaching experience of 4 years. Her current research interests include Data Mining and Software Engineering. Also she is a life member of ISTE, ISOC, IAENG and IACSIT.