International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 2 Issue: 3 08-Apr-2014,ISSN_NO: 2321-3337

# Secure Data Staging Management in Cloud

,J.C.Kavitha [1] , M.Beula kutti [2] , A.Jeba bala churchlin[3]

Asst.Professor, HOD,Department of Computer science and Engineering, Meenakshi College of Engineering, India[1].
Asst.Professor, Department of Computer science and Engineering, Meenakshi College of Engineering, India [2]
PG student , Department of Computer science and Engineering, Meenakshi College of Engineering, India [3]
beu_sundar@yahoo.com[2], afjeba@gmail.com[3]

.

***ABSTRACT—*** *In this paper we study the strategies for efficiently achieving data staging and storing the data on the set of vantage sites in the cloud system. Data staging is achieved and the security to the data items that are cached at vantage sites is provided. User request for accessing a file is processed and frequent access count is maintained for each file. Frequent access count is compared with the threshold. The file that satisfies the threshold is moved to the staging server. For accessing data from the staging server, homomorphic tokens are used. The system implements archive data management in the staging server.*

**Keywords— Cloud computing, Data staging, Homomorphic token.**

## 1, INTRODUCTION

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). A cloud can be private or public. A public cloud sells services to anyone on the Internet. (Currently, Amazon Web Services is the largest public cloud provider.) A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. When a service provider uses public cloud resources to create their private cloud, the result is called a virtual private cloud. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services. Cloud storage is a model of networked enterprise storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Hosting companies operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers and multiple locations. The safety of the files depends upon the hosting companies, and on the applications that leverage the cloud storage.

With increasing data accessibility demands on the cloud, one of the pressing needs of the Cloud Service Provider (CSP) is to efficiently serve the needs of the user who demand data

item in the shortest possible time. For this, data availability maximization needs to be done. A particularly appealing approach to maximize data availability is to stage the requested data to some vantage sites and cache the data for a particular period of time, so that the quality of service for user‟s future access can be greatly improved. This integrated functionality is called data staging.Users usually request data items from cloud and the Cloud service providers (CSPs) provide the requested data items to the users, making the requested data available to the users in a shortest possible time leading to a functionality called data staging. The aim is to achieve data staging of requested data and to provide security to the data that are cached at vantage sites.Section 2 discusses about the literature survey emphasizing the research activities in cloud computing and also overviews about the proposed system. Section 3 presents the architecture for leveraging data staging and security. Section 4 describes about the algorithm used in the system. Section 5 deals with the experimental results. Section 6 mentions the concluding remarks and future enhancements about the project.

## 2, RELATED WORK

B.veeravalli etal., discusses the challenging problem of data distribution on a network based environment. The challenge lies in designing efficient strategies to transfer the data among a set of nodes that demand the data in such a way that the cost of transfer be a minimum. This process utilizes several intermediate resources. For instance, service providers may plan to schedule for transferring the data after carefully considering when and where to transport the data. This scheme is often referred to as reservation based document caching scheme. The entire process of transferring the data incurs monetary cost due to the utilization of resources like communication links and storage cost at any intermediate nodes, etc. The requested data items are called shared data and it has to be cached till the request time instant at an intermediate node before it can be delivered. This process of caching data at intermediate node until a transfer to a desired node involves cost in dollars per minute. Thus, when the set of users demand a shared data in a particular order, total cost of caching and transmission has to be minimized by caching data to some vantage sites and transfer it via vantage links that are cost effective. The optimal caching schemes are used for demand transferring of data across the users, so that minimized cost can be calculated. Since, we minimize the total cost of the set of users demanding the shared data the per user cost is also minimized.

D.Aksoy et al., states about the data broadcasting system. Data broadcasting systems can be distinguished according to whether they are push model or pull model. Using push model, the data items are sent out to the clients with out explicit requests for such items. In pull model, data items are broadcasted by the server in response to request received from clients. This arrangement is referred to as On-demand data broad cast. A key design consideration in the development of a large scale on demand broadcast server is the scheduling algorithm and it is used to select items to broadcast. In large scale systems, majority of the data resides in locations higher latency than server‟s memory and it leads to performance degradation. Scheduling algorithm is not sufficient for this case, to address

this problem. A set of mechanisms that coordinate the broadcast scheduling with the location and retrieval of data item have to be broadcasted. This is called data staging. Three approaches are used based on increasing bandwidth utilization, decreasing the need of fetching an item, decreasing the fetch latency.

P.Srinivaset al., discusses cloud data security. The data can be stored and retrieved using cloud computing. To maintain the data securely in distributed environment i.e., on clouds, an effective and flexible distributed scheme with Token Generation algorithm have been proposed. It has been ensured that data files checking as a secure and dependable cloud storage service. A new scheme was introduced to encrypt with the user specified key parameters to make the resource more robust. A new algorithm was derived which is very light weight and easy to compute. The encrypted blocks are stored into cloud and token checking on this encrypted blocks have been performed which gives more security to data. The data is effectively verified in case of any block modifications of files before storing to clouds by token acknowledgment. The scheme is highly efficient and resilient against attacks like Byzantine server failures, malicious data modification attacks. The two way verification of file blocks is more robust and ensures that data will not be modified before reaching to clouds.

Y.Bartal et al., the discussed problem bears some similarity to the classic file allocation(FA)problem where multiple copies ofa file are maintained via caching, migrating, and replicating at the nodes of a network to minimize communication costs for read/write requests. File copies could be created/deleted at will with zero cost, and file caching cost is also free. As a consequence, there is only a transmission cost defined for file replication, and write cost for file creation/update, which is typically a nonlinear function of the number of copies present. In contrast, existing model co nsiders the caching cost for read-only data which is the major difference that makes the findings to enrich the FA problem.

## 3, SYSTEM ANALYSIS

### 3.1 Existing System

Data staging is achieved on the set of vantage sites with minimum cost for caching and transmitting the requested data. Two phase data staging algorithm is implemented. For phase I, Variety of data as Input; Inferred access patterns as Output. For Phase II, Inferred access pattern as Input; Total cost as Output. Two cost models adopted in the research are homogeneous model and heterogeneous model. In homogeneous cost model, the transmission cost between any pair of nodes are identical while the caching cost at all sites is also identical. The multi-copy algorithm is defined to handle multiple distinct data items. The single-copy algorithm is considered as the special case which provides the optimal solution. In heterogeneous cost model, caching cost and transmission cost were not identical for all nodes.

### 3.2 Proposed System

This system mainly focuses about the data staging process and storing the data to the vantage site or the staging server. And also the system provides security to the data which is at staging server. The user can login to the system by providing username and password if he is an existing user. If a new user comes, an account is created and then login. After login, the user will be able to upload, view and search the data. The upload option will enable the user to upload the files based on category. The view option enable the user to view all the files in the database .The search option provides the particular file which is searched by the user if it is available in the database. User request for accessing a file is processed and frequent access count is maintained for each file. Frequent access count is compared with the threshold. The file that satisfies the threshold is moved to the staging server using Threshold Control Staging (TCS) algorithm. Admin can also view the files which are cached at staging server. A homomorphic token is generated for the file which is at the staging server when ever the request is processed. Using Token Generation (TG) Algorithm the token is generated and it is validated by the Staging server using validate Token(VT) algorithm. If the token is valid then the user can access data from staging server. If it is not valid then the accesses will be denied.

The staging server consist of archive data management using which the performance of the staging server is maintained. The file which are not frequently accessed in the staging server is moved back to the cloud..

## 4, SYSTEM ARCHITECTURE

The user will be given a username and password for future access into the system. If a new user enters into the system, an account with a username and password is created. The user if exists, will be authenticated. The user can upload the audio files into the database. Category based accumulation will be done at this module. There are different categories such as classical, pop, jazz, devotional, etc. The user can also view the uploaded audio file. The user can also search for audio files based on the category. The view option will enable the user to view all the files uploaded. The user request for accessing an audio file is processed and frequent access count is maintained for each file. The frequent access count is compared with the threshold. The audio file that satisfies the threshold is moved to the staging server using Threshold Control Staging (TCS) algorithm. The admin can also view the audio files which are cached at the staging server. For accessing data from the staging server, an optimistic way of providing a homomorphic token is to be proposed. This token needs to be authenticated by the cloud server first before sharing the data to the consumers from the data staging environment. In addition, the token needs to be validated by the staging server before sharing the information to the end users. This provides additional and stable security to the data staging system.

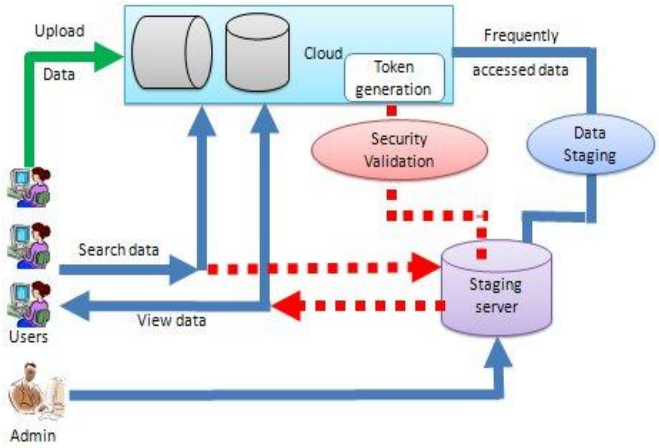


Figure .1 Data Staging and Secured Data Access in Cloud

The high frequently accessed files are kept in staging server to serve the request of the user. These files are maintained in a staging server only for a specific period of time. The files based on least frequent accessed technique are removed from the staging server, so that the performance degradation of the staging server will be reduced.

## 5, ALGORITHM DESCRIPTION

### 5.1 Threshold Control Staging Algorithm:

Threshold control staging algorithm deals with the data/file which is stored in the system. Even as the users are increasingly accessing the data from the cloud, a threshold count is set for the files which are stored. Whenever the file is accessed a count is maintained. For each access the count is increased by one. But when it reaches the threshold value it will be moved to the staging server. The algorithm is given below,

1.**declare** count, thresh, dscount

(Initialize variables- count: Access Count, dscount- access count in staging server))

2.**select** count for a particular file from database

3.**if** count is zero

4.**Begin**

Insert the file id and count to the database

5.**End**

6.**Else**

7.**Begin**

Update the table and increase the count by one for the particular file with its id.

8.**End**

9.**Select**

Assign the count to the „thresh‟ variable

10.**Select** dscount from the data staging server for the particular file

11.**If** thresh>10 and dscount=0

12.**Begin**

13.Move the file to staging server

14.**End**

15.**End**

### 5.2 Token Computation(TC) Algorithm

Token Computation Algorithm computes the key/token for accessing a particular data/file from the staging server. The High frequently accessed files are stored in the staging server. If an user want to access the file they can generate a token by clicking a button. Using this token they can access the file.

**Step 1:** Choose parameters Username F, password L, random digits V, id E .

**Step2:** x=F+L+V+E

Compute x.

**Step3:** calculate Token

Token= Encrypt(xL +xR)

**Step 4:**Update the encrypted token.

**Step 5:**Sent the token to mail id.

**Procedure**

i. create objects for class smtpclient and mailmessage
ii. assign the token value and set the subject Mail.subject=”token”
iii. smtpserver.send(mail).

### 5.3 Validate Token (VT) Algorithm

The token which is sent to the user will be stored .when the user enters the token for verification the following algorithm is implemented.

(1) declare count=0
**(2) Begin**
(3) Select count from the user details table
(4) Select token, userid

(5) Verify token
**(6) End**

## 6,IMPLEMENTAION

### 6.1 C#.NET

**Features of C#.Net**

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There‟s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate. ".NET" is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).

2. A hierarchical set of class libraries.

The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run..NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

### 6.2 SQL SERVER

**Features of SQL-Server**

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL

Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects, They are,

1. TABLE

2. QUERY

3. FORM

4. REPORT

5. MACRO

TABLE

A database is a collection of data about a specific topic.

VIEWS OF TABLE:

We can work with a table in two types,

1. Design View

2. Datasheet View

Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

QUERY:

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

## 7, EXPERIMENTAL RESULTS

Wherever User can login into the system and access the file which they needed. The frequently accessed files are moved to the staging server. Secured access is provided to the file which is in the staging server. Because the file can be accessed directly from the staging server ,cost for accessing the file is reduced compared to the access from cloud.so total cost for accessing a file is reduced..

## VIII. CONCULSION AND FUTUREWORK

Leveraging data staging and security for data access in cloud enables the user to store data into cloud. User can also search and view the data in the system. When ever a user request data item from cloud, the requested data item must be provided to the user with in the shortest possible time. This leads to the functionality called data staging which is caching the requested data to the staging server. Thus the data staging process will greatly improve the user‟s future access to the data. This data staging process is achieved and the data item is moved to the staging server. The system is focused on security. The reason for providing security is unknown users should not access the authorized data. Security will be provided to the data which is cached at staging server by means of homomorphic token.The staging server will be given archive data management in order to resolve storage concern.

## REFERENCES

[1]. Yang Wang, BharadwajVeeravalli, Senior Member, IEEE, and Chen-KhongTham”On Data Staging Algorithms for Shared Data Accesses in Clouds”IEEE Transactions on Parallel And Distributed Systems, Vol. 24, No. 4, April 2013.

[2]. Bharadwaj Veeravalli “Network Caching Strategies for a Shared Data Distribution for a Predefined Service Demand Sequence” IEEE Trans. Knowledge and Data Eng., vol. 15, no. 6, pp. 1487-1497, Nov. 2003.

[3]. D. Aksoy, M.J. Franklin, and S.B. Zdonik, “Data Staging for On- Demand Broadcast” - Proc. 27th Int‟l Conf. Very Large Data Bases(VLDB ‟01), pp. 571-580, 2001.

[4]. L. Breslau, P. Cue, P. Cao, L. Fan, G. Phillips, and S. Shenker,“Web Caching and Zipf-Like Distributions: Evidence and Implications”Proc. IEEE INFOCOM, pp. 126-134, 1999

[5]. P.Srinivas * , G. Rajesh Kumar “Efficient data security using Dynamic Token” check for Cloud Storage Systems International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 6, August – 2012.

[6]. X. Chen and X. Zhang,“A Popularity-Based Prediction Model for Web Prefetching,” Computer, vol. 36, no. 3, pp. 63-70, Mar. 2003.

[7]. Y.Bartal, M.Charikar, and P.Indyk,"On Page Migration and Other Relaxed Task Systems" Theoretical Computer Science,vol. 268, no. 1, pp. 43-66, 2001.

[8]. Y. Bartal, A. Fiat, and Y. Rabani, "Competitive Algorithms for Distributed Data Management (Extended Abstract)," Proc. 24thAnn. ACM Symp. Theory of Computing (STOC '92), pp. 39-50.