World Journal of Science and Research. 2(1): 44-53 (2017)

ISSN: 2455 2208

Available online at http://www.harmanpublications.com

World Journal of Science and Research

Harman Publications. All rights reserved

World Journal of Science and Research
Available at online
www.harmanpublications.com

Research Article | Computer Science

A NOVEL WAY OF DEDUPLICATION APPROACH FOR CLOUD BACKUP SERVICES USING BLOCK INDEX CACHING TECHNIQUE

K.Vinitha and R. Selvakumar*

Department of Computer Science, Ponnaiyah Ramajayam Institute of Science and Technology, PRIST University, Thanjavur, Tamil Nadu
*Corresponding author

ABSTRACT

For cloud storage, using deduplication techniques and their performance and suggests a variation in the index of block level deduplication and improving backup performance and Reduce the system overhead, improve the data transfer efficiency on cloud is essential so that, We presented approach on application based deduplication and indexing scheme that preserved caching which maintains the locality of the fingerprint of duplicate content to achieve high hit ratio with the help of the hashing algorithm and improve the cloud backup performance. This paper proposed a novel variation in the deduplication technique and showed that this achieves better performance. Currently, optimized cloud storage has been tested only for text files and pdf files .In future, it can be further extended to use files of other type i.e. video and audio files.

Citation: K.Vinitha and R.Selvakumar (2017) A novel way of Deduplication approach for cloud backup services using block index caching technique *World Journal of Science and Research*. 2 (1): 44-53.

Article Info:

Received on 12th Feb 2017
Accepted on 20th Mar. 2017
Online March 2017

Keywords: Cloud storage, Deduplicatin techniques, Algorithm

***Corresponding author**
R. Selvakumar, Assistant Professor Department of Computer Science, Ponnaiyah Ramajayam Institute of Science and Technology, PRIST University Thanjavur – 613 007, Tamil Nadu

1. INTRODUCTION

The explosive growth of the digital data, data deduplication has gained increasing attention for its storage efficiency in backup storage systems. Today, in the context of user data sharing platforms the challenges for large scale, highly redundant internet data storage is high. Due to this redundancy storage cost is reduces. Storage for this increasingly centralized Web data can be getting by its de duplication. Data deduplication describes a class of approaches that reduce the storage capacity needed to store data or the amount of data that has to be transferred over a network. These approaches detect coarse-grained redundancies within a data set, e.g. a file system; Data deduplication not only reduces the storage space requirements by eliminating redundant data but also minimizes the network transmission of duplicate data in the network storage systems. It splits files into multiple chunks that are each uniquely identified by a hash signature called a fingerprint. It removes duplicate chunks by checking their fingerprints, which avoids byte by byte comparisons. Mainly data deduplication focused on different terms like throughput, advance chunking schemes, other type of storage capacity and clustering method and system workload. As data passes through a cache on its way to or from the storage/processing/networking device, some of the data is selectively stored in the cache. When an application or process later accesses data stored in the cache that request can be served faster from the cache than from the slower device. The more requests that can be served from cache, the faster is the overall system performance. There is a trade-off in cache cost and performance.

PROJECT DESCRIPTION

This process can be entitled as "A Novel Way Of Deduplication Approach For Cloud Backup Services Using Block Index Caching Technique". It uses ASP.NET as Front End and SQL SERVER as Back End. This process can be done through the following modules.

ADMIN

- LOGIN
- VIEW UPLOAD FILES
- VIEW USER DETAILS
- REGISTRATION

DATA OWNER

- LOGIN
- FILE UPLOAD
- REGISTRATION

USER

- REGISTRATION
- LOGIN
- SEARCH DATA

2. SYSTEM DESIGN

2.1 DATA FLOW DIAGRAM

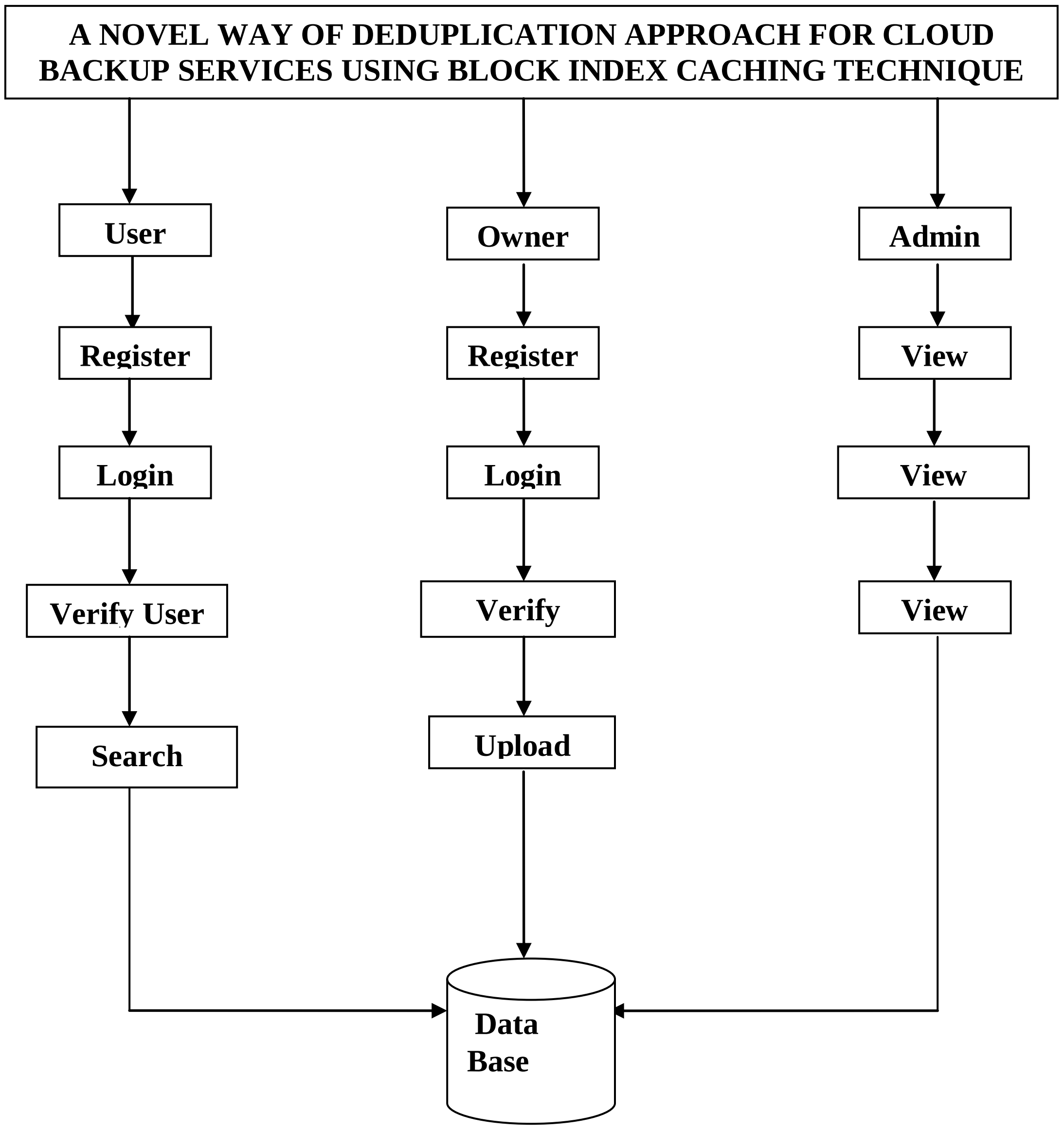

A NOVEL WAY OF DEDUPLICATION APPROACH FOR CLOUD BACKUP SERVICES USING BLOCK INDEX CACHING TECHNIQUE
User
Register
Login
Verify User
Search
Owner
Register
Login
Verify
Upload
Admin
View
View
View
Data Base

TABLE DESIGN

FIELD NAME	DATE TYPE	DESCRIPTION
uname	nvarchar(50)	UserName
sques	nvarchar(50)	Security Question
uid	nvarchar(50)	UserID
ans	nvarchar(50)	Answer
desg	nvarchar(50)	Designation
gpwd	nvarchar(50)	General Password
pwd	nvarchar(50)	Password
imgid	nvarchar(50)	ImageID

DOWNLOAD

FIELD NAME	DATA TYPE	DESCRIPTION
UserId	Numeric	User Id
UserName	Varchar	User Name

USER REGISTER

FIELD NAME	DATA TYPE	DESCRIPTION
UserId	Numeric	User Id
Name	Varchar	Name
Gender	Varchar	Gender
MailId	Varchar	Mail Id
Password	Varchar	Password
PhoneNumber	Numeric	Phone Number
Date	DateTime	Date

3. SOFTWARE DESCRIPTION

FRONT END DESIGN

C#.NET

C# is an object-oriented programming language developed by Microsoft Corporation. C# source code as well as those of other .NET languages is compiled into an intermediate byte code called Microsoft Intermediate Language. C# is primarily derived from the C, C++, and Java programming languages with some features of Microsoft's Visual Basic in the mix. C# is used to develop applications for the Microsoft .NET environment. .NET offers an alternative to Java development. Microsoft's Visual Studio .NET development environment incorporates several different languages including ASP.NET, C#, C++, and J# (Microsoft Java for .NET), all of which compile to the Common Language Runtime.

A new form of iterate employs co-routines via a functional-style yield keyword similar to the one found in the Python language. Anonymous methods provide closure functionality.

Generics or parameterized types support some features not supported by C++ templates such as type constraints on generic parameters. However, expressions cannot be used as generic parameters as in C++ templates. In contrast to the Java implementation, parameterized types are first class objects in the virtual machine, allowing for optimizations and preservation of type information.

Null able value types facilitate interaction with SQL databases. Sample null able type declaration: in Variable Name = null; Partial types allow the separation of a class implementation into more than one source file. This feature was implemented primarily so Visual Studio generated code can be kept separate from developer code. C# version 3.0 introduces several language extensions to support higher order, functional style class libraries. The extensions enable the construction of compositional APIs with the expressive power of query languages in areas such as relational databases and XML. C# 3.0 will include the following new features: Anonymous types: topple types automatically inferred and created from object initializes. Object initializes ease construction and initialization of objects.

Implicitly typed local variables permit the type of local variables to be inferred from the expressions used to initialize them

Implicitly typed arrays: a form of array creation and initialization that infers the element type of the array from an array initialize.

Extension methods make it possible to extend existing types and constructed types with additional methods.

Lambda expressions: an evolution of anonymous methods providing improved type inference and conversions to both delegate types and expression trees.

Expression trees permit lambda expressions to be represented as data (expression trees) instead of as code (delegates).

Query expressions provide a language integrated syntax for queries that is similar to relational and hierarchical query languages such as SQL and XQuery. Microsoft C# developers note that C# 3.0 is byte code compatible with C# version 2.0. For the most part, enhancements comprise purely syntactic or compile-time improvements.

IMPLICITLY TYPED LOCAL VARIABLES

In an*Implicitly Typed Local Variable Declaration*, the type of the local variable being declared is inferred from the expression used to initialize the variable. When a local variable declaration specifies var as the type and no type named var is in scope, the declaration is an implicitly typed local variable declaration. For example:

The implicitly typed local variable declarations above are precisely equivalent to the following explicitly typed declarations: the type and a type named var is in scope, the declaration refers to that type; however, a warning is generated to call attention to the ambiguity. Since a type named var violates the established convention of starting type names with an upper case letter, this situation is unlikely to occur.

The *for-initializer* of a for statement (§8.8.3) and the *resource-acquisition* of a using statement (§8.13) can be an implicitly typed local variable declaration. Likewise, the iteration variable of a foreach statement (§8.8.4) may be declared as an implicitly typed local variable, in which case the type of the iteration variable is inferred to be the element type of the collection being enumerated. In the example

Extension methods have all the capabilities of regular static methods. In addition, once imported, extension methods can be invoked using instance method syntax.

IMPORTING EXTENSION METHODS

Extension methods are imported through *using-namespace-directive*s (§9.3.2). In addition to importing the types contained in a namespace, a *using-namespace-directive* imports all extension methods in all static classes in the namespace. In effect, imported extension methods appear as additional methods on the types that are given by their first parameter and have lower precedence than regular instance methods.

The rewritten form is then processed as a static method invocation, except for the way in which *identifier* is resolved: Starting with the closest enclosing namespace declaration, continuing with each enclosing namespace declaration, and ending with the containing compilation unit, successive attempts are made to process the rewritten method invocation with a method group consisting of all accessible extension methods with the name given by *identifier* imported by the namespace declaration’s *using-namespace-directive*s. The first method group that yields a non-empty set of candidate methods is the one chosen for the rewritten method invocation. If all attempts yield empty sets of candidate methods, a compile-time error occurs.

The preceding rules mean that instance methods take precedence over extension methods, and extension methods imported in inner namespace declarations take precedence over extension methods imported in outer namespace declarations. For example:

In the example, B’s method takes precedence over the first extension method, and C’s method takes precedence over both extension methods.

BACK END DESIGN

SQL SERVER

SQL Server is such a rich and powerful that most people where to begin when they start using it.

SQL Server makes it easy for users even beginners to work with databases, you can Create Table, Edit data and use queries to find the data you want with very little effort, and SQL Server includes wizards that can do the work of designing data entry forms, reports and mailing labels for you.

SQL Server also makes it easy for develops to create applications. It includes as entire programming language, visual basic for applications, and its interface is so powerful that developers can create many custom applications without programming. SQL Server ‘ quires, reports and macros are powerful enough to do most of the work that use to require programming.

When users learn SQL Server that often find they have to wade through long discussions of power features to learn about the simple features they need to work with on their own data. The help system is so extensive and complex discussion of properties, expressions and other advanced features along with instructions on the basis of creating tables, queries, simple forms, reports and mailing labels, which average the user actually needs.

SQL Server has become the best selling database management programs because of its combination of power and ease of use. It is powerful enough that developers can use it to create entire applications. Yet it is easy enough to use that in a short time, beginners can learn to manage their own data with SQL Server.

In others database management programs; the term database is sometimes used to refer to tables that hold data. SQL Server uses the term more broadly. An SQL Server database consists of the tables that hold the data and all he related objects such as quires, forms and reports that are used to manage the data. Microsoft SQL Server will probably be an option on the programming menu, which you can select to start the program.

When you open a database, SQL Server displays the database window sometimes called as the database container, because it contains all the objects that make up the database.

There are two terms you should know, as they are almost always used when computerized databases are discussed. They are simply new names for familiar things.

A record is all the data on a single entity. For example, if have a list of names and address, each record includes the name and address of one person. In the sample table, where you have a list of products, each record includes the name and other details for one product.

A filed is one piece of data that appears up each record. For example, if you have a list of names and addresses the first name might be the first field, the last name might be the second field and the street address might be the third field. In the same table there are fields of product id, product name, supplier and there.

A table is made up of all the similar records you want to work on together. For example you might want to keep the names and address of all your friends in second table.

The simplest and most important type of query is a select query is a select query, which lets you select which data from a table is displayed. You can specify which fields are displayed, enter criteria to specify which records are displayed and specify the sort order of these records.

Forms let you control how data is displayed on the screen. Macros let you automate and speed up your work; they are also used when you develop applications. A macro is a list of actions. Macros can save time for SQL Server users.

MULTIPLE DATABASES

Most Business applications database because there are many types of data that you cannot store effectively in a single table.

When businesses first begin to use computers they discovered they could eliminate repetitious data by breaking up some databases into multiple tables.

When a business kept records on paper, for example, the payroll department had a file with each employee’s name, address, social security number unmanned data on the benefits

Department had a file with each employee’s name, address, Social Security number and data on the benefits to which each was entitled.

The computer can looks up the name and address that go with each payroll record. It can retrieve data so quickly that it can display the name and address ROM one table and the payroll data from another table just as quickly as if they were in a single table.

You only have to enter the basic data once, though. When sometimes you only have to change the address in one table, and it will automatically be changed in all the forms used by all the departments.

Relational Database is not needed for very simple data, such as a mailing list, but they are needed for most business applications.

INTERNET INTEGRATION

The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web application.

SCALABILITY AND AVAILABILITY

The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows 98 through large, multiprocessor servers running Microsoft Windows 2000 Data center edition. SQL Server 2000 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Websites.

EASE OF INSTALLATIONS, DEPLOYMENT AND USE

SQL Server 2000 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites. SQL Server 2000 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems. These features allow you to rapidly deliver
SQL Server applications that customers can implement with a minimum of installation and administrative overhead.

Enterprise Manager is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems. Service Manager is used to control the MS SQL Server (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQL Server Agent processes. An icon for this service normally resides in the system tray of machines running SQL Server. You can use Service Manager to start, stop or pause any one of these services.

Data Transformation Services (DTS) provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

DATABASE

A database is similar to a data file in that it is a storage place for data. Like a data file, a database does not present information directly to a user; the user runs an application that accesses data from the database and presents it to the user in an understandable format.

Database systems are more powerful than data files. The data is more highly organized. In a well-designed database, there are no duplicate pieces of data that the user or application has to update at the same time. Related pieces of data are grouped together in a single structure or record, and relationships can be defined between these structures and records.

When working with data files, an application must be coded to work with the specific structure of each data file. In contrast, a database contains a catalog that applications use to determine how data is organized. Generic database applications can use the catalog to present users with data from different databases dynamically, without being tied to a specific data format.

RELATIONAL DATABASE

There are different ways to organize data in a database but relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data. In a relational database, data is collected into tables (called relations in relational theory).

A table represents some class of objects that are important to an organization. For example, a company may have a database with a table for employees, another table for customers, and another for stores. Each table comprises columns and rows (attributes and tuples in relational theory). Each column represents some attribute of the object represented by the table. For example, an Employee table would typically have columns for first name, last name, employee ID, department, pay grade, and job title. Each row represents an instance of the object represented by the table. For example, one row in the Employee table represents the employee who has employee ID 12345.

When organizing data into tables, you can usually find many different ways to define tables. Relational database theory defines a process, normalization, which ensures that the set of tables you define will organize your data effectively.

While SQL Server is designed to work as a server in a client/server network, it is also capable of working as a stand-alone database directly on the client. The scalability and ease-of-use features of SQL Server allow it to work efficiently on a client without consuming too many resources.

STRUCTURED QUERY LANGUAGE (SQL)

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational databases; the most common is SQL. Standards for SQL have been defined by both the American National Standards Institute (ANSI) and the International Standards Organization (ISO). Most modern DBMS products support the Entry Level of SQL-92, the latest SQL standard (published in 1992).

6.7 ALGORITHM USED

ADMIN:

Step1: Start the process

Step2: The administrator has login into the registered username and password

Step3: To view a block of uploaded files that is accepted by cloud servers and verified by TPA in the multi cloud server.

Step4: The administrator allows the user to download the uploaded file from multi cloud server and that file verified by verifier file using his identity key to download the decrypted data.

Step5: Then the public auditor checks all files integrity and accept the files into cloud.

USER:

Step1: The user only registered user can able to login in cloud server.

Step2: The user to download the uploaded from multi cloud server and that file verified by verifier file using his identity key to download the decrypted data.

TPA:

Step1: The public verifier is able to correctly check the integrity of shared data. The public verifier can audit the integrity of shared data from multicloud with whole data and accept the file.

Step2: The public auditor check all files integrity and accept the files to the cloud server.

Step3: Each server from multi cloud server verify the file block and accept the block of files to verify the verifier.

4. TESTING

DEFINITION OF TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.

- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Valid Input : Identified classes of valid input must be accepted.

Invalid Input : Identified classes of invalid input must be rejected.

Functions : Identified functions must be exercised.

Output : Identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

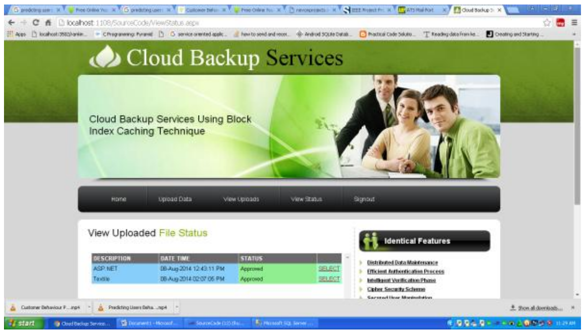
White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

5. SCREEN LAYOUT

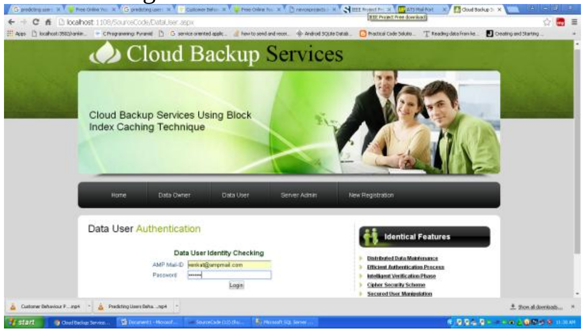
View status


Cloud Backup Services
Cloud Backup Services Using Block Index Caching Technique
Home
Upload Data
View Uploads
View Status
Signout
Identical Features
Distributed Data Maintenance
Efficient Authentication Process
Intelligent Verification Phase
Cipher Security Scheme
Secured User Manipulation

View Uploaded File Status

DESCRIPTION	DATE TIME	STATUS	
ASP.NET	08-Aug-2014 12:43:11 PM	Approved	SELECT
Textile	08-Aug-2014 02:07:06 PM	Approved	SELECT

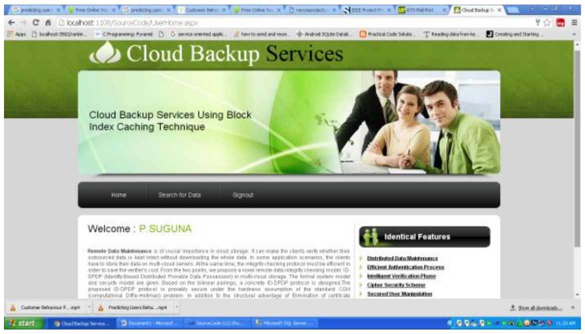
Data user Login


Cloud Backup Services
Cloud Backup Services Using Block Index Caching Technique
Home
Data Owner
Data User
Server Admin
New Registration
Login
Identical Features
Distributed Data Maintenance
Efficient Authentication Process
Intelligent Verification Phase
Cipher Security Scheme
Secured User Manipulation

Data User Authentication

Data User Identity Checking

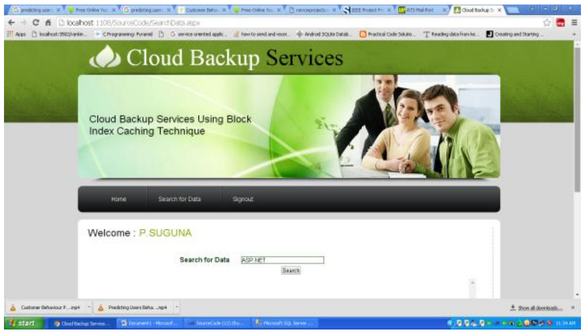
AMP Mail-ID: venkat@ampmail.com
Password: •••••••


Cloud Backup Services
Cloud Backup Services Using Block Index Caching Technique
Home
Search for Data
Signout
Identical Features
Distributed Data Maintenance
Efficient Authentication Process
Intelligent Verification Phase
Cipher Security Scheme
Secured User Manipulation

Welcome : P.SUGUNA

Remote Data Maintenance is of crucial importance in cloud storage. It can make the clients verify whether their outsourced data is kept intact without downloading the whole data. In some application scenarios, the clients have to store their data on multi-cloud servers. At the same time, the integrity checking protocol must be efficient in order to save the verifier's cost. From the two points, we propose a novel remote data integrity checking model: ID-DPDP (Identity-Based Distributed Provable Data Possession) in multi-cloud storage. The formal system model and security model are given. Based on the bilinear pairings, a concrete ID-DPDP protocol is designed. The proposed ID-DPDP protocol is provably secure under the hardness assumption of the standard CDH (computational Diffie-Hellman) problem. In addition to the structural advantage of Elimination of certificate

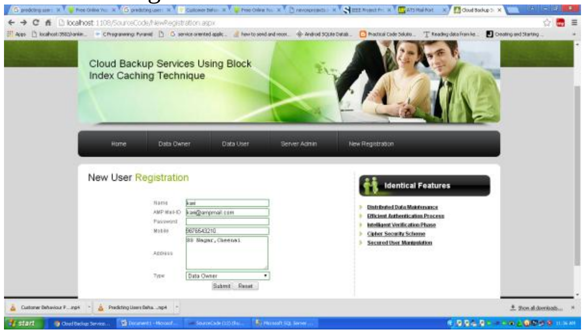
Search for data



Welcome : P.SUGUNA

Search for Data: ASP.NET

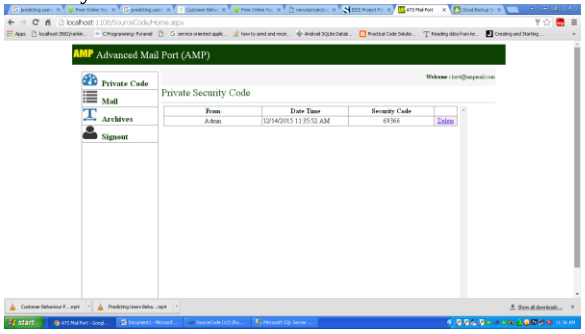
New User Registration


Cloud Backup Services Using Block Index Caching Technique
Home
Data Owner
Data User
Server Admin
New Registration
Identical Features
Distributed Data Maintenance
Efficient Authentication Process
Intelligent Verification Phase
Cipher Security Scheme
Secured User Manipulation
Submit
Reset

New User Registration

Name	kavi
AMP Mail-ID	kavi@ampmail.com
Password	
Mobile	9876543210
Address	SS Nagar,Chennai
Type	Data Owner

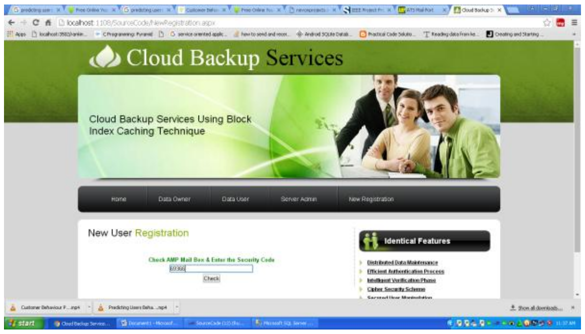
Security code



Welcome : kavi@ampmail.com

Private Security Code

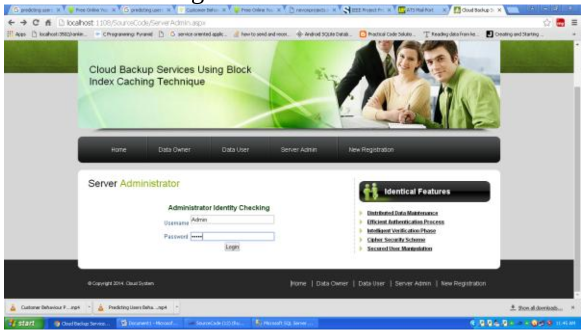
From	Date Time	Security Code	
Admin	12/14/2015 11:35:52 AM	69366	Delete


Cloud Backup Services
Cloud Backup Services Using Block Index Caching Technique
Home
Data Owner
Data User
Server Admin
New Registration
Identical Features
Distributed Data Maintenance
Efficient Authentication Process
Intelligent Verification Phase
Cipher Security Scheme
Secured User Manipulation
Check

New User Registration

Check AMP Mail Box & Enter the Security Code

Administrator Login


Cloud Backup Services Using Block Index Caching Technique
Home
Data Owner
Data User
Server Admin
New Registration
Identical Features
Distributed Data Maintenance
Efficient Authentication Process
Intelligent Verification Phase
Cipher Security Scheme
Secured User Manipulation
Login
©Copyright 2014. Cloud System
Home | Data Owner | Data User | Server Admin | New Registration

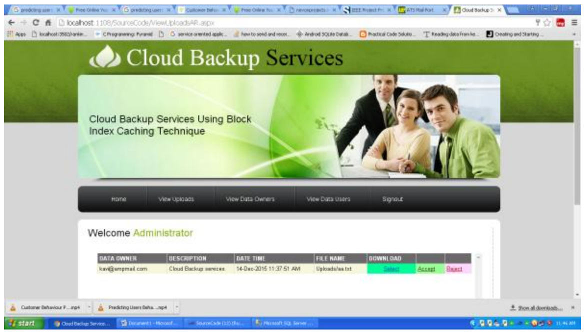
Server Administrator

Administrator Identity Checking

Username: Admin

Password: ••••

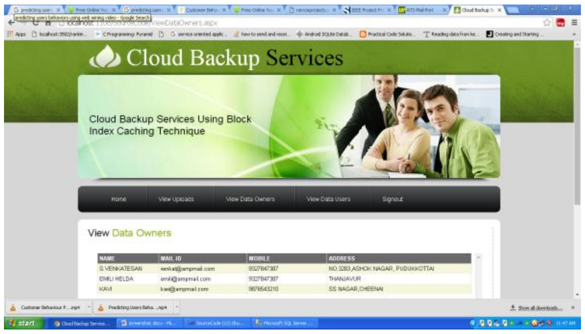
View uploads



Welcome Administrator

DATA OWNER	DESCRIPTION	DATE TIME	FILE NAME	DOWNLOAD		
kavi@ampmail.com	Cloud Backup services	14-Dec-2015 11:37:51 AM	Uploads/aa.txt	Select	Accept	Reject

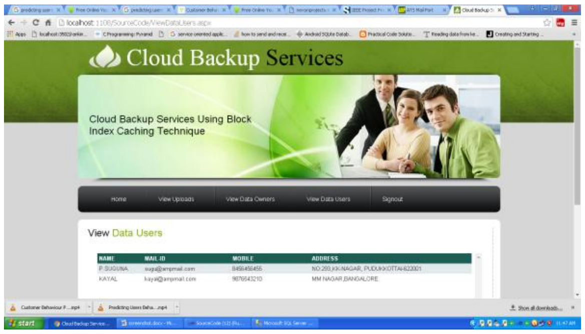
View Data owners



View Data Owners

NAME	MAIL ID	MOBILE	ADDRESS
S.VENKATESAN	venkat@ampmail.com	9327847387	NO.3283,ASHOK NAGAR, PUDUKKOTTAI
EMILI HELDA	emili@ampmail.com	9327847387	THANJAVUR
KAVI	kavi@ampmail.com	9876543210	SS NAGAR,CHEENAI

View data users



View Data Users

NAME	MAIL ID	MOBILE	ADDRESS
P.SUGUNA	suga@ampmail.com	8456456455	NO:293,KK-NAGAR, PUDUKKOTTAI-622001
KAYAL	kayal@ampmail.com	9876543210	MM NAGAR,BANGALORE

CONCLUSION

For cloud storage, using deduplication techniques and their performance and suggests a variation in the index of block level deduplication and improving backup performance and Reduce the system overhead, improve the data transfer efficiency on cloud is essential so that, We presented approach on application based deduplication and indexing scheme that preserved caching which maintains the locality of the fingerprint of duplicate content to achieve high hit ratio with the help of the hashing algorithm and improve the cloud backup performance. This paper proposed a novel variation in the deduplication technique and showed that this achieves better performance. Currently, optimized cloud storage has been tested only for text files and pdf files .In future, it can be further extended to use files of other type i.e. video and audio files.

BIBLIOGRAPHY

1. C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in *Proc. of IWQoS'09*, July 2009, pp. 1–9.
2. Sun Microsystems, Inc., "Building customer trust in cloud computing with transparent seurity," Online at https://www.sun. com/offers/details/sun transparency.xml, November 2009.
3. M. Arrington, "Gmail disaster: Reports of mass email deletions," Online at http://www.techcrunch.com/2006/12/ 28/gmail-disasterreports-of-mass-email-deletions/, December 2006.
4. Amazon.com, "Amazon Web Services (AWS)," Online at http://aws. amazon.com, 2008.
5. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," *Proc. of SecureComm '08*, pp. 1–10, 2008.
6. Google App Engine, Online at http://code.google.com/appengine/.
7. Microsoft Azure, http://www.microsoft.com/azure/.
8. Agrawal et al. Ws-bpel extension for people (bpel4people), version 1.0., 2007.
9. M. Amend et al. Web services human task (ws-humantask), version 1.0., 2007.
10. D. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases.
11. Data Communications and Networking, by *Behrouz A Forouzan*.
12. E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," Trans. Storage, vol. 2, no. 2, pp. 107–138, 2006.
13. D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy. Washington, DC, USA: IEEE Computer Society, 2000
14. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 584–597.
15. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 598–609.

Sites Referred:

16. http://www.asp.net.com
17. http://www.dotnetspider.com/
18. http://www.dotnetspark.com
19. http://www.almaden.ibm.com/software/quest/Resources/
20. http://www.computer.org/publications/dlib
21. http://www.developerfusion.com/

Source of support: Nil;
Conflict of interest: None declared