International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 6, Issue 1, 2019, PP 26-47
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: http://dx.doi.org/10.20431/2349-4859.0601004
www.arcjournals.org

Data Finding, Sharing and Duplication Removal in the Cloud Using File Checksum Algorithm

OSUOLALE A. FESTUS

Department of Computer Science, School of Computing, the Federal University of Technology, Akure, Nigeria.

Corresponding Author:** ***OSUOLALEA.FESTUS, *Department of Computer Science, School of Computing, the Federal University of Technology, Akure, Nigeria.*

Abstract: *Cloud computing is a powerful technology that provides a way of storing voluminous data that can easily be accessed anywhere and at any time by eliminating the need to maintain expensive computing hardware, dedicated space, and software. Addressing increasing storage needs is challenging and a time demanding task that requires large computational infrastructure to ensure successful data processing and analysis. With the continuous and exponential increase of the number of users and the size of their data, data deduplication becomes more and more a necessity for cloud storage providers. By storing a unique copy of duplicate data, cloud providers greatly reduce their storage and data transfer costs. This project provides an overview on cloud computing, cloud file services, its accessibility, and storage. It also considers storage optimization by de-duplication studying the existing data de-duplication strategies, processes, and implementations for the benefit of the cloud service providers and the cloud users. The project also proposes an efficient method for detecting and removing duplicates using file checksum algorithms by calculating the digest of files which takes less time than other pre-implemented methods.*

Keywords:*Cloud computing, data storage, file checksum algorithms, computational infrastructure, duplication.*

1. INTRODUCTION

Cloud computing is an emerging trend in new generation Information and Communication technology [6]. Every user has some amount of data to store in an easily available secure storage space. The advantage of storing data to the cloud for giving out information among friends, to simplify moving data between different mobile devices, and for small commerce to back up and provide disaster recovery capabilities cannot be over-emphasized as the cloud computing concept is basically to make live easy by the solutions emerging with it.

With social networking services such as Facebook, Instagram and so on, the benefits of sharing data have become so numerous such as the ability to share photos, videos, information and events, creates a sense of enhanced enjoyment in one's life and can enrich the lives of some people as they are amazed at how many people are interested in their life and well-being [9].

The benefits cloud users gain from data sharing is summarily higher productivity [6]. With multiple users from different locations contributing to data in the Cloud, the time and cost will be much less compared to having to manually exchange data and hence creating a cluster of redundant documents. As the volume of web documents increases on internet, it is a burden to search engines to provide the relevant information to the user query. In addition, a greater number of duplicates of documents also grows simultaneously on the web which increases the retrieval time and reduces the precision of the retrieved documents.

Data sharing is becoming increasingly important for many users and sometimes a crucial requirement, especially for businesses and organizations aiming to gain profit. Data sharing is becoming prevalent in many industries and organizations. Hospitals are now benefitting from data sharing as this provides better, safer care of patients.

Cloud computing is one of the developing technologies, which helped several organizations to save money and time adding suitability to the end users. Thus, the scope of cloud storage is massive because the organizations can virtually store their data's without bothering the entire gadget. Cloud

Computing provides key benefit to the end users like cost savings, able to access the data regardless of location, performance and security. With more people accessing their files online, an important part of file sharing today is done by taking advantage of cloud storage [3].

Data deduplication has become is a complicated concept in itself in that. Data deduplication removes redundant data and tries to ensure that only unique instance of the data is stored. Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeated data [10]. Data deduplication is also known as single-instance data storage. There are several approaches to deduplication, they include parity bit, repetition code, Cryptographic hash functions, Error-correcting codes, checksum and Cyclic Redundancy Checks (CRC).

Checksum schemes include parity bits, check digits, and longitudinal redundancy checks. Some checksum schemes, such as the Damm algorithm, the Luhn algorithm, and the Verhoeff algorithm, are specifically designed to detect errors commonly introduced by humans in writing down or remembering identification numbers. File checksum concept determines whether there is redundant data in a system, it is implemented by comparing an incoming chunk of file with files that have already been stored using different file attributes like User ID, Filename, Size, Extension, Checksum and date parameters.

2. Related Literature

Cloud storage is a model of computer data storage in which the digital data is stored in logical pools. The physical storage spans multiple servers (sometimes in multiple locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data[11].

Rapid data growth and the need to keep it safer and longer will require organizations to integrate how they manage and use their data, from creation to end of life. Now there is an opportunity to store all our data in the internet. Those off-site storages are provided and maintained by the third parties through the Internet. Cloud storage offers a large pool of storage available for use, with three significant attributes: access via Web services, a non-persistent network connection and immediate availability of very large quantities of storage, and pay for what you use.

2.1. Evolution of Cloud Storage

Cloud storage is an offering of cloud computing based on traditional network storage and hosted storage. Benefit of cloud storage is the access of your data from anywhere. Cloud storage providers provide storage varying from small amount of data to even the entire warehouse of an organization. Subscriber can pay to the cloud storage provider for what they are using and how much they are transferring to the cloud storage.

The cloud storage is an abstraction behind an interface. The storage can access on demand. Cloud provides geographically distributed storage capacity. Cloud storage architecture provides the delivery of service on demand in a scalable and multitenant way.

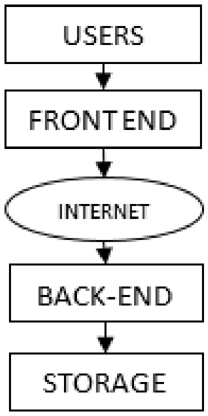


Fig1.*Cloud Storage Architecture [18]*

2.2. Popular Cloud Storage Options

Dropbox

Collaboration: Dropbox gives users the ability of sharing whole folders with other Dropbox users allowing updates to be viewable by all associates. Users can download shared documents directly from Dropbox's web interface without installing the Dropbox desktop client. Storage of files in the Dropbox "Public" folder allows links to files to be sent to both Dropbox and non-Dropbox users; however non-Dropbox link recipients must download the file to access/edit it, and any changes or revisions made to the file by the link recipients will not be seen in the Dropbox version of the file.

Mobile App Support: Documents are easily accessible via mobile devices using the Dropbox mobile app.

Storage: Dropbox offers 2GB of free storage.

Strengths: Primarily in its ease of use. Very simple and user-friendly interface; for example, folders sharing is available by simply right clicking the file or folder on the desktop, and choose Share. You can also determine how quick files are synchronized in Preferences (right –clicking the Dropbox icon). You can also recover deleted files in Dropbox easily than some other options.

Weaknesses: Lowest amount of free storage of the offerings reviewed in this document. Also, when inviting users to share information, the email invitation must be sent to the email address that is associated with a Dropbox account.

Google-Drive

Collaboration: Google Drive Users documents must have a Google Drive account. All updates and editing by associates will be synchronized to Google Drive. For documents that you have right to access, you can receive alerts when changes are made. You can share files with people by sending a link to your file.

Mobile App Support: Google Drive has an Android app which gives you the ability to share information on your Android device using your Google Drive account. You can also share information from Drive with phone contacts.

Storage: Google Drive offers 5GB of free storage.

Strengths: Possesses built in document editor so that programs like Word Editors are not required to be installed on computer in order to edit document. Allows comments to be left on any files stored.

Weaknesses: Sharing is not easy and simple as Dropbox, must use the Google Drive web application to set it up. Also, no ability to set preferences on synchronization speed.

MicrosoftSkyDrive

Collaboration: Associates can access SkyDrive files without registering for a SkyDrive account. You can also update documents simultaneously online with associates.

Mobile App Support: SkyDrive offers both a Window's phone app and an iOS (iPhone/iPad) app. This allows users to view and share as well as edit and update files via mobile devices. SkyDrive files can also be opened via a third-party iOS apps, such as Pages and Keynote.

Storage: SkyDrive offers 7GB of free space.

Strengths: Offers the most storage for free of the options reviewed in this document. Like Google Drive, you can edit documents within the browser, without having to open up a client application like Microsoft Word.

Weaknesses: SkyDrive is less user friendly than Dropbox and Google Drive.

Box

Collaboration: You can share data with both associates that do have Box accounts, and those who don't. Like Dropbox, you can setup a shared folder and invite Box account users for ongoing sharing. You can receive email alerts when files are uploaded, downloaded, or added. You can also set passwords for important files and set time limits for users access to specific files. You have more access to files and documents because security degree can be defined. Box is more for businesses and enterprises, but it is also available for personal use.

Mobile App Support: Users can view, edit, create and share content on-the go. You can find files fast using search bar. You can also upload files from your phone or tablet to Box as well as save files from Box onto your mobile device for offline access.

Storage: Box gives 5GB of free storage.

Strengths: You can store file sizes. Box is organized and user friendly, you can create and organize several layers of folders for all of your data. You can use tagging as a way to keep track of your folders and files. Tags allow you to mark and sort related files that may not be located in the same section of your Box. Box Offers the security option and Content management tools.

Weaknesses: Box doesn't do file synchronization from the computer to box.com as simply as other services do. There is a desktop component called Box Sync, but it's available only to Business and Enterprise account holders for a fee to be paid.

2.3. Common Features of Cloud Storage Options

Many of these services are free to a specific number of gigabytes, with additional storage available for a monthly fee. All cloud storage service provides drag and drop accessing and syncing of folders and files between your desktop and mobile devices, and the cloud drive. They also allow users to collaborate with each other on documents.

Advantages and Disadvantages of Cloud Storage

Advantages of Cloud Storage

- Usability: All cloud storage services reviewed in this topic have desktop folders all systems. This allows users to drag and drop files between the cloud space and their local storage.
- Bandwidth: You can send a web link to recipients through your email and avoid emailing files to individuals.
- Accessibility: Stored files can be accessed anywhere via Internet connection.
- Disaster Recovery: It is highly recommended that businesses have an emergency back-up plan ready every time. Cloud storage can be used as a back- up plan by businesses by a second copy of important files. These files are stored at a remote location and can be accessed via internet connection.
- Cost Savings: Businesses and organizations can sometimes reduce annual operating costs by using cloud storage; cloud storage costs about 3 cents per gigabyte to store data internally. Users can see additional cost savings because it does not require internal power to store information remotely.

Disadvantages of Cloud

- Storage Usability: Be careful when using drag/drop to move a document into the cloud storage folder. This will permanently move your document from its original folder to the cloud storage location. Do a copy and paste instead of drag/drop if you want to retain the document’s original location in addition to moving a copy onto the cloud storage folder.
- Bandwidth: Several cloud storage services have a specific bandwidth allowance. If an organization surpasses the given allowance, the additional charges could be paid. However, some providers offer unlimited bandwidth. This is a factor that companies should consider when looking at a cloud storage provider.
- Accessibility: No internet connection, No access to your data.
- Data Security: There are concerns with the safety and privacy of important data stored remotely. The possibility of private data commingling with other organizations makes some businesses uneasy.
- Software: If you want to be able to manipulate your files locally through multiple devices, you'll need to download the service on all devices.

2.4. Data Finding

Data units are scattered all over the internet as most data items are unorganized, cloud computing tend to bring a definite arrangement pattern to files save on them as all data units on the cloud are assigned a unique index. An advanced search gives users the option of finding files or folders by type, name, title, location, date (taken, modified, or created), size, or property tag. The search locates files and programs stored anywhere in indexed locations, which includes personal folders, e-mail, offline files, and web sites in your History list and ultimately the Cloud. Depending on the file access technology adopted in the cloud storage. Files on the cloud are accessed through a lot of means, for example; files have a Name, Author Name, Size, Keywords all of which are adopted in Search Engine Optimization (SEOs). Searching for a file by the file name is the easiest way of accessing a file over the cloud, then using keywords would help in situations where an external party is searching for a related file and it is done through the use of the Keywords that are defined at the point where the file is stored.

2.5. Data Sharing

Data sharing is the practice of distributing or providing access to digital media, such as data, computer programs, multimedia (audio, images and video), documents or electronic books. File sharing may be achieved in a number of ways. Common methods of storage, transmission and dispersion include manual sharing utilizing removable media, centralized servers on computer networks, Cloud Links, World Wide Web-based hyperlinked documents, and the use of distributed peer-to-peer networking [4]. Confidential data are also stored in the cloud using one or more encryption technique. So only the authenticated members who know the key can access the data but everyone can download.

Types of Data Sharing

Peer-To-Peer File Sharing

Peer-to-peer file sharing is based on the peer-to-peer (P2P) application architecture. Shared files on the computers of other users are indexed on directory servers. P2P technology was used by popular services like Napster, Spotify, and Infinit. The most popular protocol for P2P sharing is BitTorrent.

File Hosting Service

File hosting services are a simple alternative to peer-to-peer software. These are sometimes used together with Internet collaboration tools such as email, forums, blogs, or any other medium. File hosting service, cloud storage service, online file storage provider is an internet hosting service specifically designed to host user files. It allows users to upload files that could then be accessed over the internet after a user name and password or another authentication is provided. Typically, the services allow HTTP access, and sometimes FTP access. Related services are content-displaying hosting services (i.e. video and image), virtual storage, and remote backup.

File Sync and Sharing Services

Cloud-basedfile syncing and sharing services implement automated file transfers by updating files from a dedicated sharing directory on each user's networked devices. Files placed in this folder also are typically accessible through a website and mobile app, and can be easily shared with other users for viewing or collaboration. Such services have become popular via consumer-oriented file hosting services such as Dropbox and Google Drive. Data synchronization in general can use other approaches to share files, such as distributed file systems and version control.

Data Duplication

Cloud Service Providers move large amounts of data over a network and provide access to that data as a service, for example, a basic requirement for any cloud-based data protection solution needs to be the ability to reduce the overall costs of providing the same service that clients could do themselves in their own data centers. One method being used to achieve this goal is data deduplication across multiple end-user clients, where the costs to provide the service is amortized over the number of paying clients. There are multiple methods to remove duplicate data, so service providers and their customers need to be cognizant of the differences between the available solutions and the impact they may have on security and the ability to efficiently move, protect and store data in the cloud in a cost-effective manner. More storage is not the best answer as storage cost money and the increasing number and size of files eventually burdens the company's backup and disaster recovery (DR) plans.

Rather than finding ways to store more data, companies are turning to data reduction technologies that can store less data. Data deduplication emerged as an important part of any data reduction scheme [17].

Data Deduplication Demystified

Data deduplication is basically a means of reducing storage space. It works by eliminating redundant data and ensuring that only one unique instance of the data is actually retained on storage media, such as disk or tape. Redundant data is replaced with a pointer to the unique data copy. Data deduplication, sometimes called intelligent compression or single-instance storage, is often used in conjunction with other forms of data reduction. Traditional compression has been around for about three decades, applying mathematical algorithms to data in order to simplify large or repetitious parts of a file effectively making a file smaller.

The business benefits of data de-duplication include;

- Reduced hardware costs; cloud computing gives businesses the ability to enable employees to gain access to data and applications from anywhere, making them more productive when on the go without the need to adopt high performance computers of their own.
- Reduced backup costs; cloud computing environments enable businesses to scale their compute and storage needs up on as-needed basis, which can keep costs low. Additionally, the cloud architecture moves IT spending from capital to operating expenditures, which makes it easier on the books and simpler to justify. Costs are directly aligned with a business usage so they are easy to predict.
- Automation provides agility; an agile business is a successful business and agility is gained from high levels of automation. Cloud computing services are designed to be heavily automated and self-provisioning, giving end-users the ability to quickie address their needs. Businesses have the ability to more quickly attend to customer demands, which improves service and responsiveness.
- New Business Models: Cloud computing has made it easier to start business innovation initiatives, often enabled by readily available cloud services[1].

Data deduplication primarily operates at the file, block and even the bit level. File deduplication is relatively easy to understand if two files are exactly alike, one copy of the file is stored and subsequent iterations receive pointers to the saved file. However, file deduplication is not very efficient because the change of even a single bit results in a totally different copy of the entire file being stored. By comparison, block and bit deduplication looks within a file and saves unique iterations of each block. If a file is updated, only the changed data is saved. This behavior makes block and bit deduplication far more efficient.

3. Timeline of Deduplication Models

3.1. Location Based

Duplication of data can be removed on various locations. If redundancy of data is to be eliminated on various locations, then that type of data de-duplication technique is called location-based data de-duplication technique.

3.2. Under System De-Duplication

This is the method where the de-duplication hash calculations are created on the target device as the data transfer the system in real time. If the system finds a block that it previously stored on the device it can’t store the new block, just pointer to the already existing block. The advantage of under system de-duplication over after- process de-duplication is that its requirement is small amount of storage space as data is not repeated. On the negative side, it is shortly argued that because hash formations and lookups require so long, it concluded that the data traffic can be slower thereby minimizing the backup maximum output of the system.

3.3. Source Level De-Duplication

When eliminating of redundant data is to be performed on client site or where data is created rather than where data is stored is called source level data de-duplication. Further source level data de-duplication is to be divided into two parts;

Local Chunk De-Duplication; in the local chunk level data de-duplication, redundant data is to be removed before sending it to the destination where the data is to be stored.

Global Chunk De-Duplication; in the global chunk level data de-duplication technique, redundant data is removed at global for each client[7].

3.4. Target Level De-Duplication

During elimination of redundant or similar data, it is to be performed on target site where the data is stored. In this data de-duplication, the client does not know any technique of removal of similar data. This type of technique increases the processing time[7].

3.5. Disk Placement Based

Based on how data is to be placed on disk, data de-duplication technique is to be used either forward reference or backward reference technique.

Forward-Reference Source Level; in the forward-reference source level, recent data chunks are maintained and every other old data chunk are associated with pointers that points forward to the recent chunks.

Backward-Reference Target Level; It introduces the more fragmentation for the past data chunks.

3.6. Single Instance Storage

Removing multiple copies of any file is one form of the de-duplication. Single-instance storage (SIS) environments are able to detect and remove redundant copies of identical files. After a file is stored in a single-instance storage system, all other references to same file will refer to the original single copy. Single-instance storage systems compare the content of files to determine if the incoming file is identical to an existing file in the storage system. Content-addressed storage is typically equipped with single-instance storage functionality. While file-level de-duplication avoids storing files that are a duplicate of another file, many files that are considered unique by single-instance storage measurement may have a tremendous amount of redundancy within the files or between files. For example, it would only take one small element (e.g., a new date inserted into the title slide of a presentation) for single-instance storage to regard two large files as being different and requiring them to be stored without further de-duplication. [16].

3.7. Sub-File De-Duplication

Sub-file de-duplication detects redundant data within and across files as opposed to finding identical files as in Single-Instance storage (SIS) implementations. Using sub-file de-duplication, redundant copies of data are detected and are eliminated even after the duplicated data exist within separate files. This form of de-duplication discovers the unique data elements within an organization and detects when these elements are used within other files. As a result, sub-file de-duplication eliminates the storage of duplicate data across an organization. Sub-file data de-duplication has tremendous benefits even where files are not identical, but have data elements that are already recognized somewhere in the organization. Sub-file de-duplication implementation has two forms. Fixed-length sub-file de-duplication uses an arbitrary fixed length of data to search for the duplicate data within the files. Although simple in design, fixed-length segments miss many opportunities to discover redundant sub-file data. (Consider the case where an addition of a person's name is added to a document's title page, the whole content of the document will shift causing the failure of the de-duplication tool to detect equivalencies). Variable length implementations are usually not locked to any of arbitrary segment length. Variable length implementations match data segment sizes to the naturally occurring duplication within files, vastly increasing the overall de-duplication ratio[20].

3.8. Normalization

It is the process of organizing data to minimize redundancy in the relational database world. The concept of normalization and what we know now as the First Normal Form (1NF) was introduced by [4], the inventor of the relational model. Today there are six normal forms defined but generally, a relational DB table is often described as “normalized” if it is in the Third Normal Form. Normalization involves dividing large, badly-formed tables into smaller, well-formed tables and defining relationship between them. This information about table’s structures and their relations are called metadata (or data about the data). Depending on the degree of normalization, we have more or less information about the DB structure[4].

However, some modeling disciplines such as the dimensional modeling approach to data warehouse design, explicitly recommend non-normalized designs. The purpose of such systems is to be intuitive and have high-performance retrieval of data.

3.9. Map-Reduce

Map-Reduce is a very successful programming model adopted for implementation of data-intensive applications to support distributed computing. Jeffrey et al. introduces Map-Reduce as a master-slave model. The failure of a slave is managed by re-assigning its task to another slave, while master failures are not managed as considered unlikely to happen. Users specify a map and a reduce function. The map function processes key/value pairs and generates a set of intermediate key/value pairs. The reduce function merges all intermediate values associated with the same intermediate key and produces a result as a list of values. The main advantage of Map-Reduce is that it allows for distributed processing of the map and reduces operations. All map processes can potentially perform in parallel and all reduce processes can potentially perform in parallel; provided that their operations are independent of the others [29[.

3.10. File-Based Compare

File system-based deduplication is a simple method to reduce duplicate data at the file level, and usually is just a compare operation within the file system or a file system-based algorithm that eliminates duplicates. An example of this method is comparing the name, size, type and date-modified information of two files with the same name being stored in a system. If these parameters match, you can be pretty sure that the files are copies of each other and that you can delete one of them with no problems. Although this example isn't a foolproof method of proper data deduplication, it can be done with any operating system and can be scripted to automate the process, and best of all, it's free. Based on a typical enterprise environment running the usual applications, you could probably squeeze out between 10 percent to 20 percent better storage utilization by just getting rid of duplicate files[1].

3.11. File Checksum

File checksum shows you how to evaluate checksums or hashes to remove duplicate from your collection. Data duplication removal involves searching storage devices e.g. hard drive, server, etc. for redundant instances of files and selectively deleting them. While the text which follows refers to image files, the same process can be used for any file type on a computer.

As image files take up substantial space, being able to eliminate all but a single instance could significantly decrease the need for storage, especially in cases where there is no strict file handling process when first adding files. File checksum adopts the concepts of file verification (process of using an algorithm for verifying the integrity of a computer file. This can be done by comparing two files bit-by-bit, but requires two copies of the same file, and may miss systematic corruptions which might occur to both files).

3.12. Concepts of File Verification

File verification is the process of using an algorithm for verifying the integrity of a computer file. This can be done by comparing two files bit-by-bit, but requires two copies of the same file, and may miss systematic corruptions which might occur to both files[9].

Integrity verification: File integrity can be compromised, usually referred to as the file becoming corrupted. A file can become corrupted by a variety of ways: faulty storage media, errors in transmission, write errors during copying or moving, software bugs, and so on. Hash-based verification ensures that a file has not been corrupted by comparing the file's hash value to a previously calculated value. If these values match, the file is presumed to be unmodified. Due to the nature of hash functions, hash collisions may result in false positives, but the likelihood of collisions is often negligible with random corruption.

Authenticity verification: It is often desirable to verify that a file hasn't been modified in transmission or storage by untrusted parties, for example, to include malicious code such as viruses or backdoors. To verify the authenticity, a classical hash function is not enough as they are not designed to be collision resistant; it is computationally trivial for an attacker to cause deliberate hash collisions, meaning that a malicious change in the file is not detected by a hash comparison. In cryptography, this attack is called a pre-image attack.

3.13. File Formats in Integrity Check

A checksum file is a small file that contains the checksums of other files. There are a few well-known checksum file formats. Several utilities, such as md5deep, can use such checksum files to automatically verify an entire directory of files in one operation. The particular hash algorithm used is often indicated by the file extension of the checksum file. The ".sha1" file extension indicates a checksum file containing 160-bit SHA-1 hashes in sha1sum format. The ".md5" file extension, or a file named "MD5SUMS", indicates a checksum file containing 128-bit MD5 hashes in md5sum format. The ".sfv" file extension indicates a checksum file containing 32-bit CRC32 checksums in simple file verification format. The "crc.list" file indicates a checksum file containing 32-bit CRC checksums in brik format.

As of 2012, best practice recommendations is to use SHA-2 or SHA-3 to generate new file integrity digests; and to accept MD5 and SHA1 digests for backward compatibility if stronger digests are not available. The theoretically weaker SHA1, the weaker MD5, or much weaker CRC were previously commonly used for file integrity checks[9].

3.14. Checksum Algorithms

Parity byte or parity word: The simplest checksum algorithm is the so-called longitudinal parity check, which breaks the data into "words" with a fixed number n of bits, and then computes the exclusive or (XOR) of all those words. The result is appended to the message as an extra word. To check the integrity of a message, the receiver computes the exclusive or of all its words, including the checksum; if the result is not a word consisting of n zeros, the receiver knows a transmission error occurred. With this checksum, any transmission error which flips a single bit of the message, or an odd number of bits, will be detected as an incorrect checksum. However, an error which affects two bits will not be detected if those bits lie at the same position in two distinct words. Also swapping of two or more words will not be detected. If the affected bits are independently chosen at random, the probability of a two-bit error being undetected is $\frac{1}{n}$.

Modular Sum:A variant of the previous algorithm is to add all the "words" as unsigned binary numbers, discarding any overflow bits, and append the two's complement of the total as the checksum. To validate a message, the receiver adds all the words in the same manner, including the checksum; if the result is not a word full of zeros, an error must have occurred. This variant too detects any single-bit error, but does not do well in more than one bit.

Position-dependent: The simple checksums described above fail to detect some common errors which affect many bits at once, such as changing the order of data words, or inserting or deleting words with all bits set to zero. The checksum algorithms most used in practice, such as Fletcher's checksum, Adler-32, and cyclic redundancy checks (CRCs), address these weaknesses by considering not only the value of each word but also its position in the sequence. This feature generally increases the cost of computing the checksum.

Message Digest: A utility installed which is capable of creating a Checksum file, such as File verification using MD5 Checksums will be used, a spreadsheet application like Microsoft Excel. Its goal is to gather all checksums together for the image collection we want to remove duplicate, sort them by their checksum values, and then use a spreadsheet calculation formula to help us quickly identify the duplicates if any exist. If there are duplicates, then we can use the spreadsheet as our checklist for deleting the files. Summarily, creating a master checksum text file for the image collection, import that text file into a spreadsheet application, locate recurring checksum values within the spreadsheet, find corresponding digital files and verify they are the same and delete all but one instance of the file.

The MD5 algorithm is designed to be fast on 32-bit machines. Additionally, the MD5 algorithm does not need any large substitution tables; the algorithm code can be written quite compactly.

3.15. Md5 Checksum Algorithm

MD5 checksum algorithm which is known as MD5 message-digest is an algorithm that takes as input a message of random length and produces as output a 128-bit fingerprint or message digest of the input. It is estimated that it is computationally impossible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is proposed for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem[14].

We begin by assuming that we have a b-bit message asinput, and that we wish to find its message digest. Here b is an arbitrary nonnegative integer; b may be zero, it need not be a multiple of eight, and it may be arbitrarily large. We imagine the bits of the message written down as follows:

m_0 m_1 ... m_{b-1}

The following five steps are performed to compute the message digest of the message.

Step 1. Append Padding Bits

The message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. That is, the message is extended so that it is just 64 bits shy of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512.

Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

Step 2. Append Length

A 64-bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step. In the unlikely event that b is greater than 2^64, then only the low-order 64 bits of b are used. (These bits are appended as two 32-bit words and appended low-order word first in accordance with the previous conventions.)

At this point the resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16 (32-bit)words. Let M[0 ... N-1] denote the words of the resulting message, where N is a multiple of 16.

Step 3. Initialize MD Buffer

A four-word buffer (A,B,C,D) is used to compute the message digest. Here each of A, B, C, D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first):

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

Step 4. Process Message in 16-Word Blocks

We first state four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word.

F(X,Y,Z) = XY v not(X) Z

G(X,Y,Z) = XZ v Y not(Z)

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X v not(Z))

In each bit position F acts as a conditional: if X then Y else Z. The function F could have been defined using + instead of v since XY and not(X)Z will never have 1's in the same bit position.) It is interesting to note that if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z) will be independent and unbiased.

The functions G, H, and I are similar to the function F, in that they act in "bitwise parallel" to produce their output from the bits of X, Y, and Z, in such a manner that if the corresponding bits of X, Y, and Z are independent and unbiased, then each bit of G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased. Note that the function H is the bit-wise "xor" or "parity" function of its inputs.

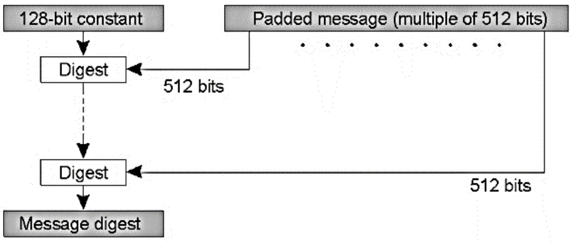


Fig2.*The Structure of MD5 Algorithm (Rivest)*

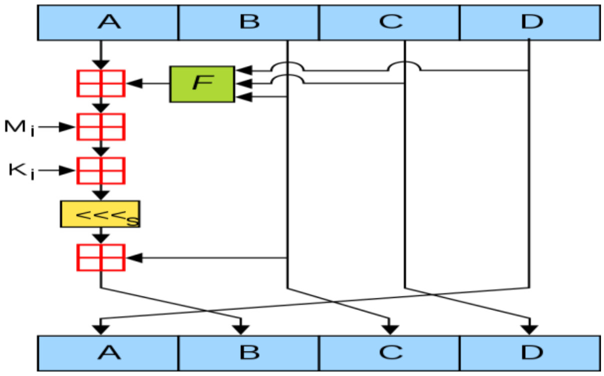


Fig3.*One MD5 operation. (Rivest, 1992)*

4. System Design and Implementation

A comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system. It also focuses on the system functionalities, operations, algorithms and flow chart that were used in the design.

4.1. System Architecture

This system adopts a three-tier architecture, User Interface layer send request to business layer through unified interface, business layer performs database operation after handling the request based on its logic rule, then encapsulate the data which are returned from the database into class format and present for User Interface layer.

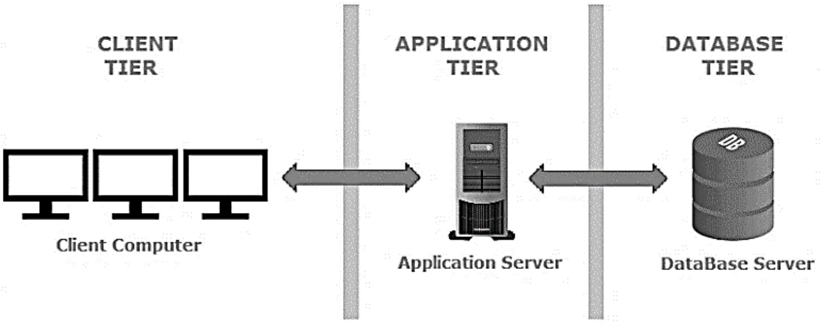


Fig4.*Three tier Architecture.*

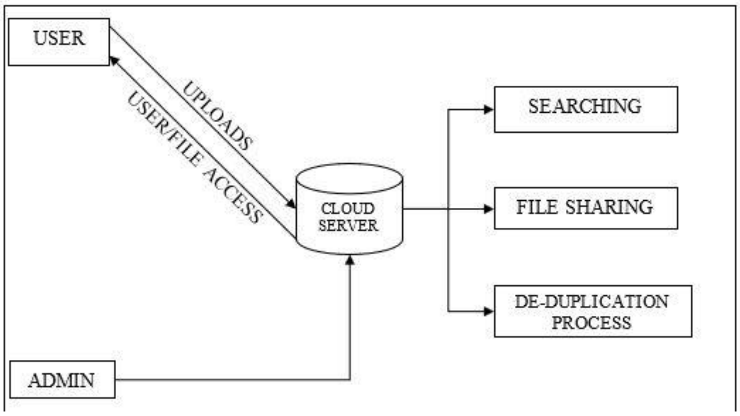
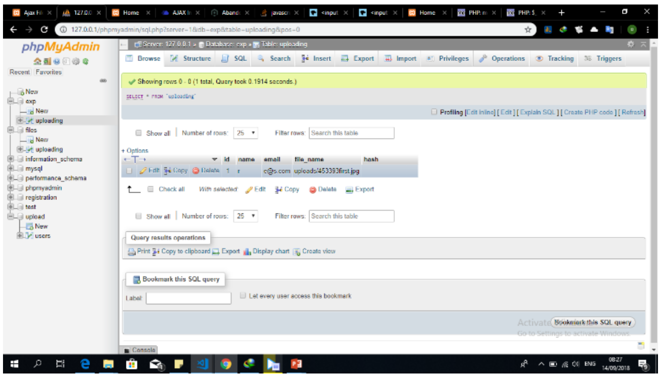


Fig5.*Overall architecture of the system*

4.2. Database Design

The database schema for the website consists of three tables, out of which one table is used to store BLOBs (Binary Large Objects). So, if the files are stored in the file system, the table could be removed. There is one main table to store the primary details of the file uploaded, one table for users and sharing a file within that group, and the last table for organizing the files with use of folders.



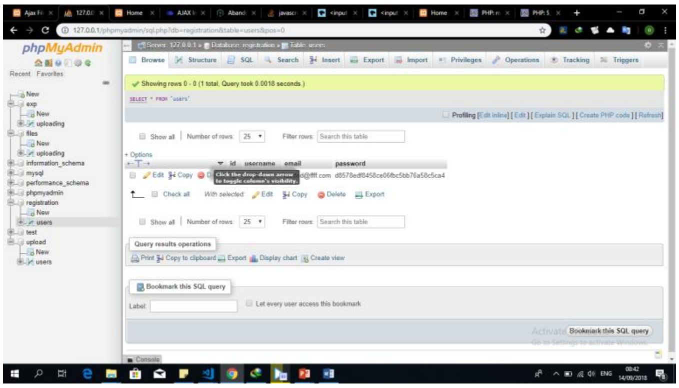
Server: 127.0.0.1 » Database: exp » Table: uploading

Showing rows 0 - 0 (1 total, Query took 0.1914 seconds.)

```
SELECT * FROM `uploading`
```

id	name	email	file_name	hash
1	r	e@s.com	uploads/453393first.jpg	

Fig6.*File database table*



Server: 127.0.0.1 » Database: registration » Table: users

Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)

```
SELECT * FROM `users`
```

id	username	email	password
		d@ffff.com	d8578edf8458ce06fbc5bb76a58c5ca4

Click the drop-down arrow to toggle column's visibility.

Fig7.*User database table*

4.3. System Functionalities

Since the project was done for educational purposes, there was no size limit given to users to upload or download files. The main features of the website are:

User Registration

News users can register to upload files and create their own file system. The user management was easily integrated into the website so Users have their own profile page to update their profile (using profile providers). Other than these users have options to change their password, request for a new password etc.

Uploading Files

This part is the heart of the project where the user can upload a file. File upload takes place in the form of a wizard control, where the user does the upload in a step by step manner. When the user uploads a file, he/she will be given a progress bar indicator to show what the status of the upload is.

Sharing Files

After the upload process, users can choose to share the file(s). This is done by the user generating a link for third parties.

Search

Another important feature available is the extensive search feature. It is a completely AJAX enabled (both simple search and advanced search) with no page refreshes at all. Concepts used to attain this no page refresh feature were PHP, and web services. Search feature searches not just the file name and metadata about it, but through the document files uploaded like .jpg, .jpeg, .pdf, .doc, .xls, .htm, .ppt, .pdf etc. It makes use of the full-text indexing available in SQL Server to search through BLOBs.

4.4. System Algorithm

For this project MD5 (Message Digest 5) hashing algorithm is used, to compare the hash values of files as the major requirement to remove duplicates of a file. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is estimated that it is computationally impossible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

4.5. Mathematical Model

There are four possible functions; a different one is used in each round:

$F(B, C, D) = (B \wedge C) \vee (\sqcap B \wedge D)$

$G(B, C, D) = (B \wedge D) \vee (C \wedge \sqcap D)$

$H(B, C, D) = B \oplus C \oplus D$

$I(B, C, D) = C \oplus (B \vee \sqcap D)$

$\oplus, \wedge, \vee, \sqcap,$ denote XOR, AND, OR, and NOT operations respectively

(Source: MD5; en.wikipedia.org)

4.6. System Pseudo-Code

Step 1. Append Padding Bits

Step 2. Append Length

Step 3. Initialize Message Digest Buffer

Step 4. Process Message in 16-Word Block

Step 5. Output Hash value.

It is to be noted that the MD5 hash is calculated according to this algorithm and values are in little-endian.

```
//Note: All variables are unsigned 32 bit and wrap modulo 2^32 when calculating
varint[64] s, K
varinti
//s specifies the per-round shift amounts
s[ 0..15] := { 7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22 }
s[16..31] := { 5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20 }
s[32..47] := { 4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23 }
s[48..63] := { 6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21 }
//Use binary integer part of the sines of integers (Radians) as constants:
forifrom 0 to 63
    K[i] := floor($2^{32}$ × abs(sin(i + 1)))
end for
//(Or just use the following precomputed table):
K[ 0.. 3] := { 0xd76aa478, 0xe8c7b756, 0x242070db, 0xc1bdceee }
K[ 4.. 7] := { 0xf57c0faf, 0x4787c62a, 0xa8304613, 0xfd469501 }
K[ 8..11] := { 0x698098d8, 0x8b44f7af, 0xffff5bb1, 0x895cd7be }
K[12..15] := { 0x6b901122, 0xfd987193, 0xa679438e, 0x49b40821 }
K[16..19] := { 0xf61e2562, 0xc040b340, 0x265e5a51, 0xe9b6c7aa }
K[20..23] := { 0xd62f105d, 0x02441453, 0xd8a1e681, 0xe7d3fbc8 }
```

```
K[24..27] := { 0x21e1cde6, 0xc33707d6, 0xf4d50d87, 0x455a14ed }
K[28..31] := { 0xa9e3e905, 0xfcefa3f8, 0x676f02d9, 0x8d2a4c8a }
K[32..35] := { 0xfffa3942, 0x8771f681, 0x6d9d6122, 0xfde5380c }
K[36..39] := { 0xa4beea44, 0x4bdecfa9, 0xf6bb4b60, 0xbebfbc70 }
K[40..43] := { 0x289b7ec6, 0xeaa127fa, 0xd4ef3085, 0x04881d05 }
K[44..47] := { 0xd9d4d039, 0xe6db99e5, 0x1fa27cf8, 0xc4ac5665 }
K[48..51] := { 0xf4292244, 0x432aff97, 0xab9423a7, 0xfc93a039 }
K[52..55] := { 0x655b59c3, 0x8f0ccc92, 0xffeff47d, 0x85845dd1 }
K[56..59] := { 0x6fa87e4f, 0xfe2ce6e0, 0xa3014314, 0x4e0811a1 }
K[60..63] := { 0xf7537e82, 0xbd3af235, 0x2ad7d2bb, 0xeb86d391 }
//Initialize variables:
varint a0 := 0x67452301   //A
varint b0 := 0xefcdab89   //B
varint c0 := 0x98badcfe   //C
varint d0 := 0x10325476   //D
//Pre-processing: adding a single 1 bit
append "1" bit to message
// Notice: the input bytes are considered as bits strings,
// where the first bit is the most significant bit of the byte.[48]
//Pre-processing: padding with zeros
append "0" bit until message length in bits ≡ 448 (mod 512)
append original length in bits mod $2^{64}$to message
//Process the message in successive 512-bit chunks:
for each512-bit chunk of padded message
 break chunk into sixteen 32-bit words M[j], $0 \leq j \leq 15$
//Initialize hash value for this chunk:
varintA := a0
varintB := b0
varintC := c0
varintD := d0
//Main loop:
forifrom 0 to 63
varint F, g
if $0 \leq i \leq 15$ then
F := (B and C) or ((not B) and D)
g := i
else if $16 \leq i \leq 31$ then
F := (D and B) or ((not D) and C)
g := (5×i + 1) mod 16
```

```
else if $32 \leq i \leq 47$ then
F := B xor C xor D
g := (3×i + 5) mod 16
else if $48 \leq i \leq 63$ then
F := C xor (B or (not D))
g := (7×i) mod 16
//Be wary of the below definitions of a,b,c,d
F := F + A + K[i] + M[g]
A := D
D := C
C := B
B := B + leftrotate(F, s[i])
end for
//Add this chunk's hash to result so far:
    a0 := a0 + A
    b0 := b0 + B
    c0 := c0 + C
    d0 := d0 + D
end for
varchardigest[16] := a0 append b0 append c0 append d0 //(Output is in little-endian)
//leftrotate function definition
leftrotate (x, c)
return (x << c) binary or (x >> (32-c));
```

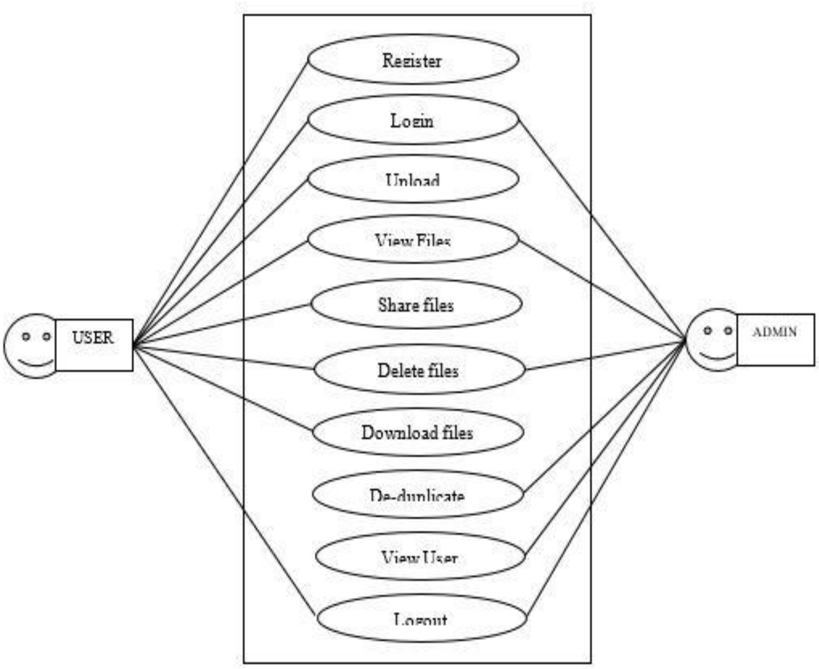


Fig8.*CloudFuta use case diagram*

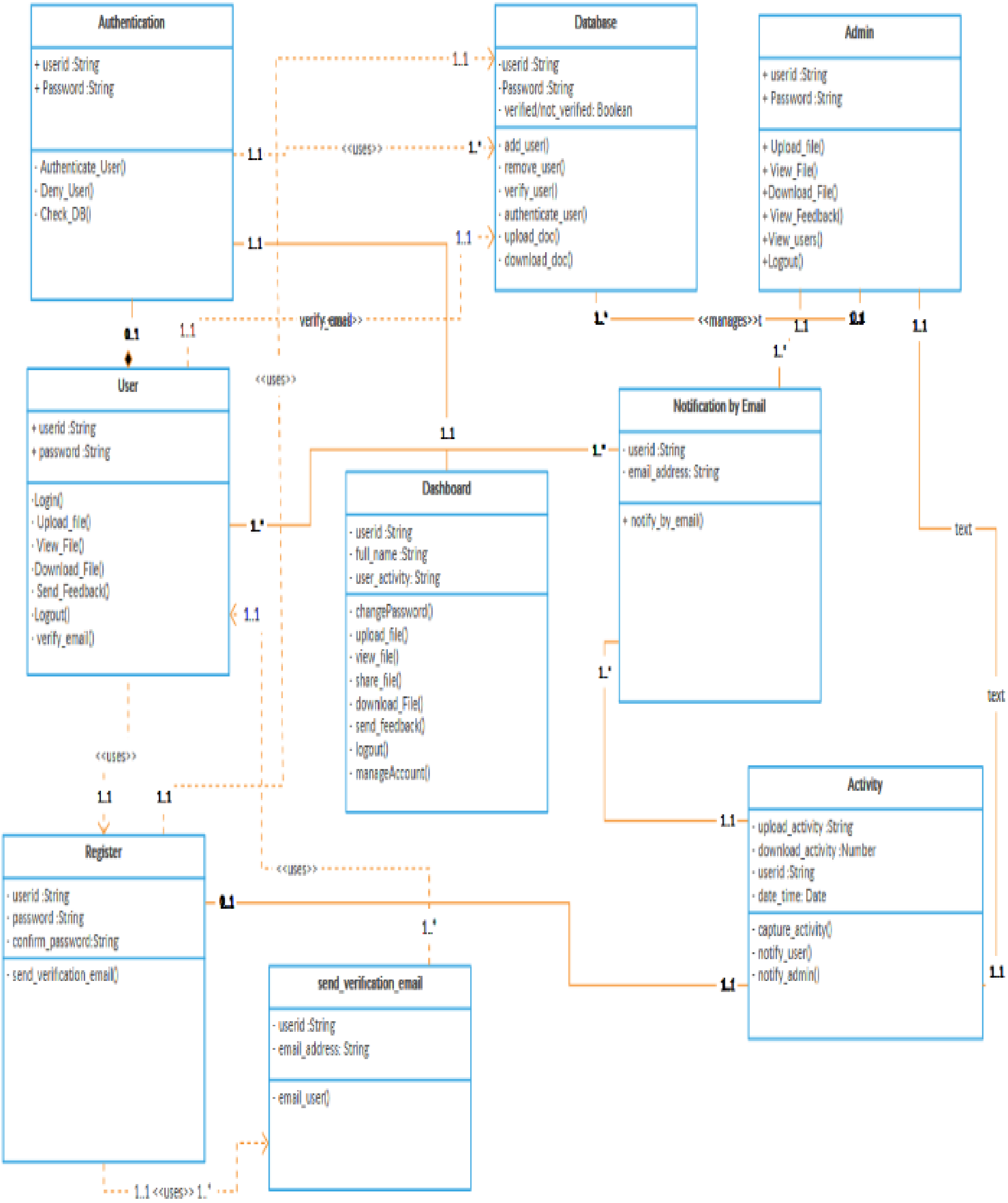


Fig9.*The UML class diagram*

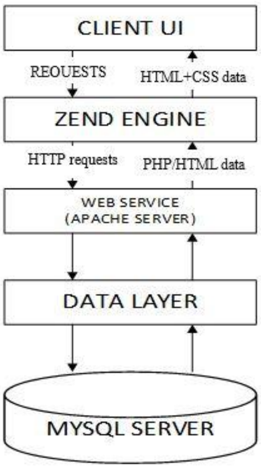


Fig10.*Expanding the three-tier architecture*

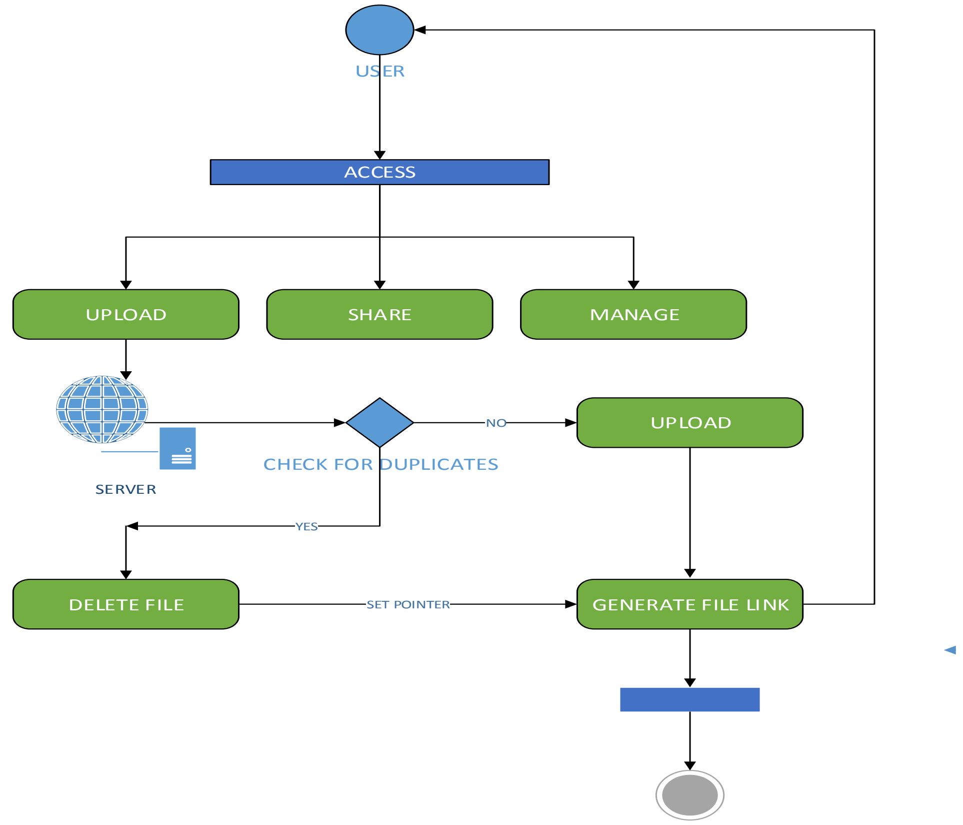


Fig11.*Activity diagram*

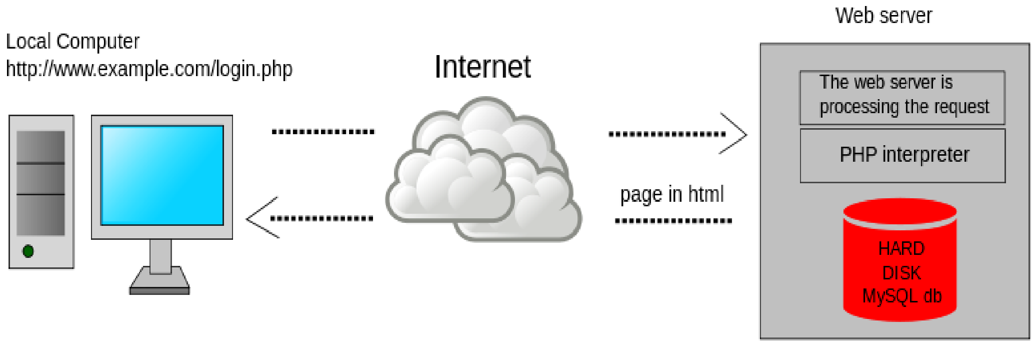


Fig12.*Dynamic web page: example of server-side scripting (PHP and MySQL)*

System Implementation

This chapter focuses on the system unit integration and the overall system implementation and test. It also highlights system Engine and facilities paramount to the proper functioning of the system.

System Requirements

CloudFuta requires that each implementation (central and unit) meet the following hardware and software specifications.

Hardware Requirements

The system would perform optimally on Minimum 350MB Hard Disk space for installation, 4GB HD space required for a typical live system with 1000-2000 events, Recommended minimum CPU - Pentium 4, 3.2GHz (32 bit and 64-bit architectures), Recommended 1GB RAM for a Central Server with Apache, Network Interface card, 4 inches of Graphical display and above. Also, Internet connectivity is highly paramount.

Software Requirements

Any operating system (Linux, MacOSx, Windows, etc.) with a responsive Web browser (Mozilla Firefox, Opera, Google Chrome etc.) installed. To be sure you have the most current details regarding the Platform compatibility with platforms, PHP versions, and Zend's other products refer to the information available online at: http://www.zend.com/store/products/zend-platform/system-requirements.php. The Zend team regularly updates this information.

APACHE

Apache HTTP Server, colloquially called **Apache**, is a free and open-sourcecross-platformweb server, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. The Apache HTTP Server is cross-platform; as of 1 June 2017 92% of Apache HTTPS Server copies run on Linux distributions. Apache supports a variety of features, many implemented as compiledmodules which extend the core functionality. These can range from authentication schemes to supporting server-side programming languages such as Perl, Python, Tcl and PHP. Popular authentication modules include mod access, modauth, mod_digest, and mod_auth_digest, the successor to mod_digest. A sample of other features include Secure Sockets Layer and Transport Layer Security support (mod_ssl), a proxy module (mod_proxy), a URL rewriting module (mod_rewrite), custom log files (mod_log_config), and filtering support (mod_include and mod_ext_filter) [1].

Performance

Instead of implementing a single architecture, Apache provides a variety of Multi-Processing Modules (MPMs), which allow it to run in either a process-based mode, a hybrid (process and thread) mode, or an event-hybrid mode, in order to better match the demands of each particular infrastructure. Choice of MPM and configuration is therefore important. Where compromises in performance must be made, Apache is designed to reduce latency and increase throughput relative to simply handling more requests, thus ensuring consistent and reliable processing of requests within reasonable time-frames.

PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. [19] PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP[13]

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalonegraphical applications. The standard PHP interpreter, powered by the Zend Engine, is a free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

PHP Interpreter

Zend Server is a complete and certified PHP distribution stack fully maintained and supported by Zend Technologies. It ships with an updated set of advanced value-add features designed to optimize productivity, performance, scalability and reliability. Zend Server includes over 80 PHP extensions and supports Apache, NGINX and IIS Web servers. It is delivered as native packages for all leading Linux distributions, Windows, Mac OS X and IBM environments including popular Cloud environments such as Amazon Web Services. Zend Server supports any and all PHP code and provides deep insights into that code including the most popular PHP applications and frameworks like WordPress, Magento, Drupal, Zend Framework, Symfony, and Laravel.

MYSQL

There are two different editions: the open source MySQL community Server and the proprietary Enterprise Server. Out of which, MySQL community Server is most widely used Relational database management system. As discussed above, Apache server uses XAMPP to store all the data like files, username and encrypted password in MySQL database.

XAMPP

XAMPP is an open source platform developed by apache. It is web server solution stack Package. XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also comes with a number of other modules including OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more.Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. The main components in XAMPP windows distribution are; Apache 2.4.34, MariaDB 10.1.34, PHP 7.2.8, phpMyAdmin 4.8.2, OpenSSL, XAMPP Control Panel 3.2.2, Webalizer, Mercury Mail, Transport System 4.63, FileZilla FTP Server 0.9.41, Strawberry Perl 7.0.56 Portable, Perl 5.16.3, mhash 0.9.9.9

Libraries [15]

phpAES

phpAES is a class implementation PHP code that supports encryption cipher of 128, 192, and 256-bit AES. No other extension is required by the library when it comes to compilation into PHP. It is fully functional and compliant with FIPS 197.

PHP Image Upload Class

PHP Image Upload Class is a powerful PHP library that can streamline the process of uploading images into a form. Using this library, a developer can upload image using a file input command. Developers can define user messages outside of the class to help in localization through gettext or similar command.

phpDocumentor

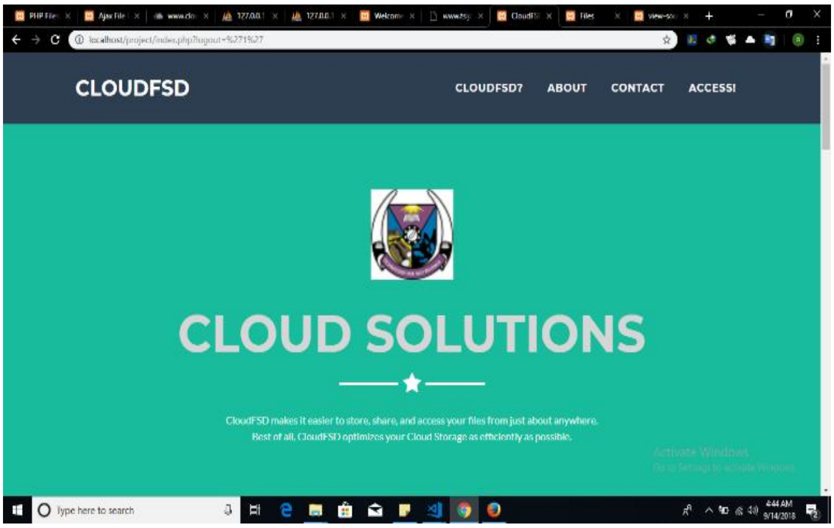
phpDocumentor is a useful auto documentation tool that helps create professional documents using the PHP code. The PHP library supports many different features to add functionality to the site. Some of the value-added features supported by the PHP library include support for incorporating customized documents such as tutorials, linking between documentations, and creating highlighted source code that feature cross-referencing to PHP general documentation. The PHP library can help with automating documentation.

PHP DB Class

PHP DB Class is a great PHP library that helps in PHP and MySQL development. The tool offers easy and convenient access to a database and reduces the amount of coding required to perform the task. In addition, the PHP library offers various debugging features. For instance, developers can use the debugging feature to display requests and the resulting table. They can perform this task just by adding a parameter to the methods of its class.

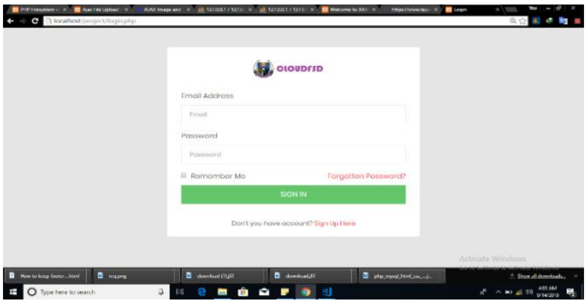
Upload

Upload is a library that simplifies file uploading and validation. When a form is submitted, the library can check the type of file and size



CloudFSD makes it easier to store, share, and access your files from just about anywhere.
Best of all, CloudFSD optimizes your Cloud Storage as efficiently as possible.

Fig13.*system landing page*



localhost/project/login.php

CLOUDFSD

Email Address

Email

Password

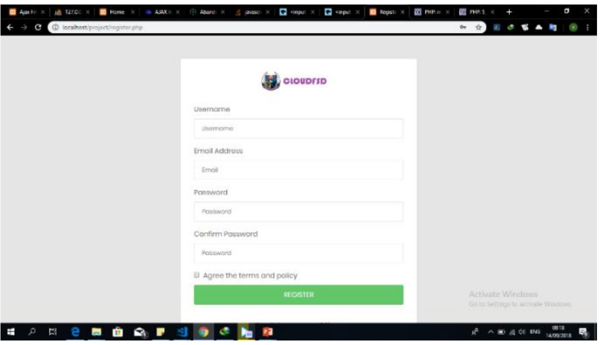
Password

□ Remember Me Forgotten Password?

SIGN IN

Don't you have account? Sign Up Here

Fig14.*System login page*



localhost/project/register.php

CLOUDFSD

Username

Username

Email Address

Email

Password

Password

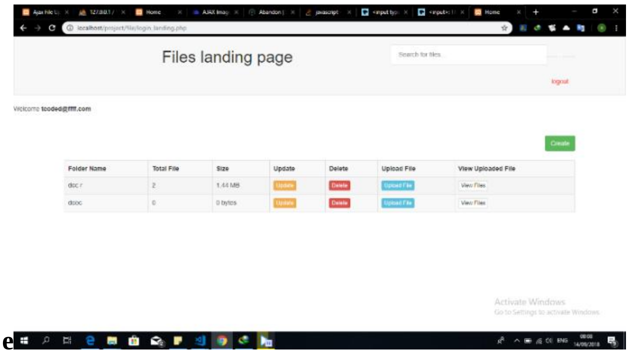
Confirm Password

Password

□ Agree the terms and policy

REGISTER

Fig15.*system registration page*



localhost/project/file/login_landing.php

Files landing page

Search for files

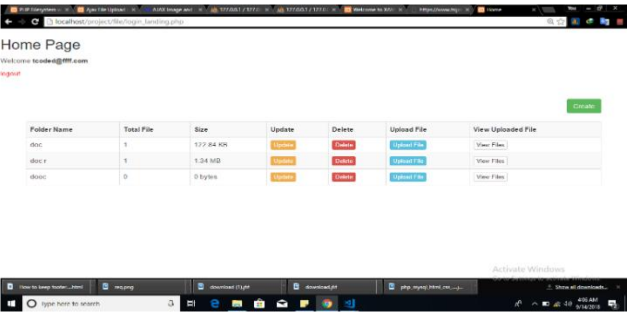
logout

Welcome **tcoded@ffff.com**

Create

Folder Name	Total File	Size	Update	Delete	Upload File	View Uploaded File
doc r	2	1.44 MB	Update	Delete	Upload File	View Files
dooc	0	0 bytes	Update	Delete	Upload File	View Files

Fig16.*system file access page*



localhost/project/file/login_landing.php

Home Page

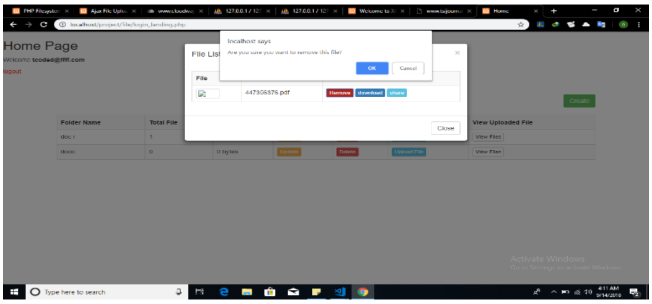
Welcome **tcoded@ffff.com**

logout

Create

Folder Name	Total File	Size	Update	Delete	Upload File	View Uploaded File
doc	1	122.84 KB	Update	Delete	Upload File	View Files
doc r	1	1.34 MB	Update	Delete	Upload File	View Files
dooc	0	0 bytes	Update	Delete	Upload File	View Files

Fig17.*System file upload demo*



localhost/project/file/login_landing.php

Home Page

Welcome **tcoded@ffff.com**

logout

File List

localhost says

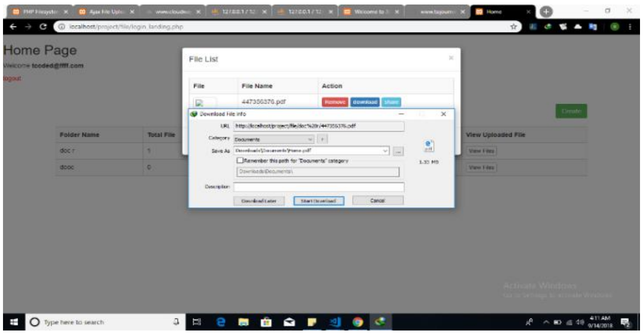
Are you sure you want to remove this file?

OK Cancel

File		
	447356376.pdf	Remove download share

Close

Fig18.*File delete dialogue*



localhost/project/file/login_landing.php

Home Page

Welcome tooded@ffff.com

logout

File List

File	File Name	Action
	447356376.pdf	Remove download share

Download File Info

URL: http://localhost/project/file/doc%20r/447356376.pdf

Category: Documents

Save As: Downloads\Documents\Home.pdf

□ Remember this path for "Documents" category

Downloads\Documents\

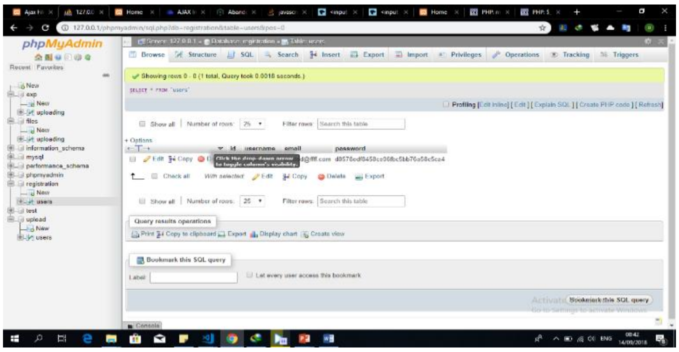
Description:

1.33 MB

Download Later | Start Download | Cancel

Folder Name	Total File	View Uploaded File
doc r	1	View Files
dooc	0	View Files

Fig19.*User File download dialogue*



127.0.0.1/phpmyadmin/sql.php?db=registration&table=users&pos=0

Server: 127.0.0.1 » Database: registration » Table: users

✓ Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)

```
SELECT * FROM `users`
```

id	username	email	password
		d@fff.com	d8578edf8458ce06fbc5bb76a58c5ca4

Fig20.*System database table*

5. Conclusion and Recommendation

Development of a web application which can effectively find data, share data and remove data duplicates in the cloud using file checksum. The web application is useful for organizations dealing with highly redundant operations that requires constant copying and storing of data for future reference or recovery purpose. The technique is a part of backup and disaster recovery solution as it allows enterprises to save data repeatedly and promotes fast, reliable and cost-effective data recovery. For instance, a file that is backed up every week results in a lot of duplicate data and thus, eats up considerable disk space. Data duplicate removal using file checksum run an analysis and eliminates these sets of duplicate data and keeps only what is unique and essential, thus significantly clearing storage space. The key obstacle was all files stored in the database should not contain duplicates of itself. That was encountered by using a software called PHP (Hypertext Preprocessor). The educational benefit of the developed system is a new system was developed to efficiently find data, shares data and remove data duplicates using file checksum in the cloud.

For further improvement, the application should be improved by incorporating features such as social media such as Facebook, twitter to promote the exposure of the content in the webpage massively.

Based on the challenges encountered in the course of the project, it is hereby recommended that more concise work should be done by interested web developers with sophisticated records to simplify the webpage and enlarge database that can take as many data as possible. For a more reliable system, replicated data should be removed to clear more space.

Acknowledgement

With gratitude and humbleness of heart, I am very thankful to the members of the Department of Computer Science, School of Computing, the Federal University of Technology, Akure, Nigeria for availing me an enabling and conducive environment to carry out this research work.

References

[1] Apache HTTP Server, From Wikipedia, the free encyclopedia;https://en.wikipedia.org/wiki/Apache_HTTP_Server; Accessed June 6, 2018.

[2] D. Meister, A. Brinkmann, "Multi-level comparison of data deduplication in a backup scenario", Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference, ACM, pp. 8:1-8:12, 2009.

[3] Daniel J. Abadi, Data Management in the Cloud: Limitations and Opportunities, IEEE Data Engineering Bulletin, Volume 32, March 2009, 3-12.

[4] Edgar J. Codd, 1990; The Relational Model for Database Management Version 2; Addison Wesley Longman Publishing Co., incBooston, MA, USA ISBN:0-201-141-14192-2.

[5] Edwin Schouten; 2012; Cloud Computing Business Benefits; http://wired.com/insights/ 2012/10/5-cloud-business-benefits/

[6] James B., Rajkman B., Zahir T., 2009 MetaCAN: HanessingStorgae Clouds for High Performance Content Delivery; Journal of Network and Computer Application, 1012-1022, 2009.

[7] Jaspreet Singh, 2009; Understanding Data Deduplication, www.druva.com/blog/understanding-data-deduplication/

[8] Kenneth P. Birman, 2012; Reliable Distributed system; Technologies, Web Services and Applications; Google Books.

[9] Morris Dworkin, 2015; NIST Policy on Hash Functions; Cryptographic Technology group, https://csrc.nist.gov/projects/hash-functions/nist-policy-on-hash-functions August 5, 2015; “National Institute of Standard and Technology NIST Special Publication 800-145

[10] NimalaBhadrappa, Mamatha G. S. 2017, Implementation of De-Duplication Algorithm, International Research Journal of Engineering and Technology (IRJET), Volume 04, Issue 09.

[11] O’Brien, J. A. &Marakas, G. M. (2011). Computer Software. Management Information Systems 10th ed. 145. McGraw-Hill/Irwin

[12] Peter Mel; The NIST definition of Cloud Computing, “National Institute of Standard and Technology NIST Special Publication 800-145

[13] PHP 5 tutorials; W3Schools, https://www.w3schools.com/pHP/default.asp

[14] Accessed June, 2018.

[15] Rivest R., 1992 The MD5 Message Digest Algorithm. RFC 1321 http://www.ietf.org/rfc/rfc321.txt

[16] Sandeep Sharma, 2015; 15 Best PHP Libraries Every Developer Should Know; published on; https://www.programmableweb.com/news/15-best-php-libraries-every-developer-should-know/analysis/2015/11/18 ; accessed June 12, 2018.

[17] Single Instance Storage in Microsoft Windows Storage Server 2003 R2Archived 2007-01-04 at the Wayback Machine: https://archive.org/webTechnical White Paper: Published May 2006 access September, 2018.

[18] Stephen J. Bigelow, 2007 Data Deduplication Explained: http://searchgate.org; Accessed February, 2018

[19] Wenying Zeng, Yuelong K. O, Wei S., (2009) Research on Cloud Storage Architecture and Key Technologies, ICIS 2009 Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human Pages 1044-1048.

[20] What is PHP? PHP User contributory notes; http://php.net/manual/en/intro-whatis.php. Accessed June 6, 2018

[21] X. Zhao, Y. Zhang, Y. Wu, K. Chen, J. Jiang, K. Li, "Liquid: A scalable deduplication file system for virtual machine

[22] images", *Parallel and Distributed Systems IEEE Transactions on*, vol. 25, no. 5, pp. 1257-1266, May 2014.

Citation:*OSUOLALE A. FESTUS., (2019). Data Finding, Sharing and Duplication Removal in the Cloud Using File Checksum Algorithm. International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 6(1), pp.26-47. http://dx.doi.org/10.20431/2349-4859.0601004*