# Survey on the Standard Operating Procedures for Data Backup and Recovery

Mayukh Mondal
*CSE with Spec. in IoT*
*Vellore Institute of Technology, Vellore*
20BCT0133

Arnish Bhardwaj
*CSE with Spec. in IoT*
*Vellore Institute of Technology, Vellore*
20BCT0157

Pamidighantam Meghana Srilekha
*CSE with Spec. in IoT*
*Vellore Institute of Technology, Vellore*
20BCT0123

***Abstract*—Big data applications have ushered in an era when data storage backups are becoming increasingly crucial. Backup techniques are adaptable, backup contents are secure and trustworthy, and backup and recovery are easy and practical. Cloud computing is rapidly growing and it is all about user privacy, usage simplicity and costs a significant degree recovering user data. Each time a restoration operation is carried out, the client has the option to backup both the data and the entire virtual machine. The modern digital era's data backup techniques are briefly explored and addressed.**

***Index Terms*—Data Backup and Recovery, Cloud Storage, Recovery Techniques, Disaster Recovery, Business Continuity**

## I. Introduction

With the unstoppable growth of digitization, data backup strategies are more important than ever for businesses seeking to improve the dependability and availability of their information systems. However, because backup operations are not free, a data-driven strategy is needed to determine how frequently and what types of backups should be performed. In this study, we propose a comprehensive mathematical methodology for exploring backup policy design space and enhancing backup kind and interval in a system or process. Database systems are frequently used to store data within cloud-based services and applications. The database system's backup and recovery functions are critical. Regular backups of systemic data and stored data help database administrators avoid data leakage and corruption [1]. Administrators restore the database from backup when a disaster strikes the content management system, such as component failure, corruption of data induced by bugs, malfunctions, or other causes. Cloud-based computing services and software products, in specific, require a method for efficiently backing up ever-increasing amounts of data.

For database backup and recovery, two approaches are often used [2]. A physical backup is one method that transmits files including the database's architecture and table files. Physical backup is straightforward since backup and retrieval may be achieved by simply copying database files. A logical backup, on the other hand, analyses the whole database to generate query statement sequences that may be used to reconstruct the database itself. Backup data can be transferred between foundational file systems, operating systems, and MySQL variants since the logical backup is based on results of commands, mostly query-based.

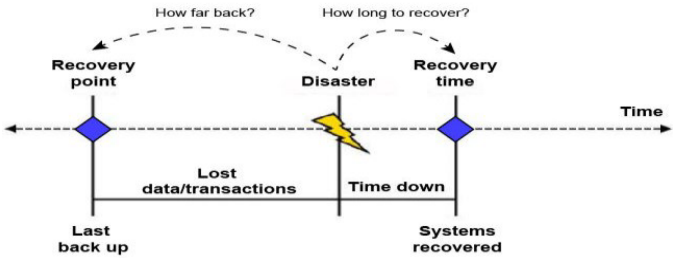


Fig. 1: Restore points throughout the lifecycle of a data entry in the server [4].

### A. Disaster Recovery

An important factor to consider when thinking of handling data in today's world is that the volume and scale of data dealt with have increased considerably. The possibility of loss and corruption of the data has also increased, as has the stakes in keeping it safe. This is where the need for data recovery modules arises. Numerous forms of disasters, whether natural or man-made, can disrupt an organization's operations and result in catastrophic data loss [3]. The purpose of crisis response techniques is to provide functionalities to recover lost digital documents from backup devices if the primary source is compromised. Implementing such solutions presents a number of problems, including the burden on efforts and financial complexity. When catastrophes are exploited as a service, they may be fixed and data recovery rates can be raised at a fair cost. This enables an organisation to sustain or restart important job functions rapidly after a disaster due to its great flexibility. The purpose of DR is to keep the organisation running as normally as feasible.

### B. Geographic Separation:

The geographic separation is a crucial critical element to remember during data recovery. Sites should be spatially isolated from the primary sources to reduce the impact of catastrophes that may cause collateral damage due to geographical proximity. This could, in turn, result in longer delays for query resolution. The longer response time and data recovery will be proportionately caused by distant backup locations. There is a delay in the data transfer back and forth due to the speed of light; the constraints for carrying out deduplication are met

when the transmission distance to the backup location is no longer than a 100 milliseconds. Technologies not relying on concurrency and synced time-clocks can improve speed over greater transmission areas, but they can increase data loss after a disaster.

We discovered various approaches in this study that have their distinct methods of backup and recovery. In general, everything These technologies concentrate on three distinct features, like cost management, data duplication, and security concerns. This method is entirely focused on its backup and recovery function.

## II. Data Backup Strategies

Data and its processing has progressed rapidly since the deployment of secondary storage devices. Currently, cloud computing has emerged as the new industry standard for requisitioning storage and computing resources for most digital-enabled companies [5] [6]. Thanks to its capacity to provide internationally distant resources, it is becoming more ubiquitous in everyday computing. This rapidly-emerging concept is a collection of rules and processes that, in most cases, are backed by relevant infrastructure, allowing the organisation to recover rapidly from disasters and provide recovery and continuity [7]. If the system failed or if a natural or man-made calamity happened, there is a potential of data loss, which may also result in financial loss. Cloud computing processes are networked systems that share resources. Many people are sharing concurrent storage and computational resources. As a result, we require a strong method to restrict unauthorized users from obtaining your vital and helpful data. Cloud-based data repository and backup solutions enable you to back up and restore critical and sensitive information if they are damaged.

Industries are shifting their on-site and off-site resources to the cloud for long-term and active usage, and eliminating the need for maintenance and up-scaling costs for the physical resources. Thus, in the current scenario, information is not only stored physically at local devices, but also distributed throughout the network. The method of access and security for such devices will have to cater not only to the threats of physical intrusion, but also network-based remote intrusion. The current information and data scenario requires handling of a lot more threats, at the cost of providing easy-access and liability-free resources to the client companies [8].

Thus, the data backup strategies can be divided into main types:

1) Recovery and backup of data against local threats
2) Recovery and backup of data against network-related threats

The methodologies that can be implemented to protect against these threats can either target each component separately, or target the data servers as a whole. For cloud-based backup, having a stable connection at all times is necessary, as well having sufficient bandwidth to handle the usual data volume without any server overload. They must also account for any transmission loss of data via parity bits and checksums, and have provisions for retransmision of data. Whether the data

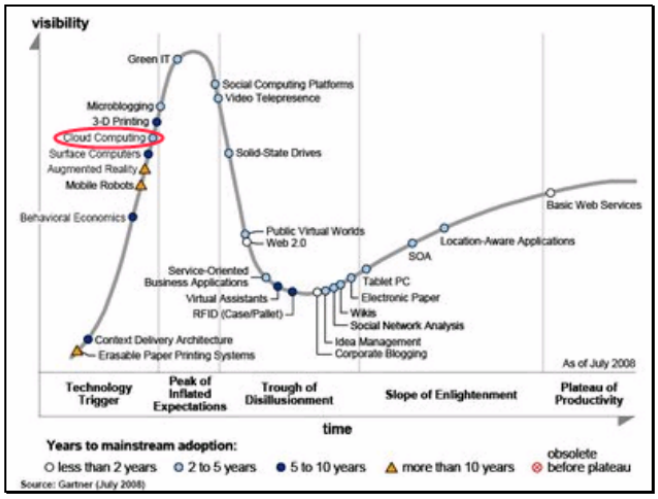


Fig. 2: Cloud computing is a rapidly growing sector [9]

should be concurrently updated on the backup servers, or periodically, must be determined according to the nature of the data being stored on the server.

The data backup strategies for the current digital era, therefore, must cater to these requirements:

1) Physical vulnerabilities such as damage, loss or gradual wear-and-tear of components
2) Human-related errors such as accidental data corruption, loss of access
3) Network vulnerabilities such as network outage, poor bandwidth
4) Directed, local malicious attacks via files, flash storage devices, infected files in the server
5) Malicious attacks through the network via open, unsecured channels and ports, or Directed Denial of Service (DDoS) attacks

### A. Failsafe Mechanisms

When we talk about implementing such replication or backup server-based architectures, there is one fundamental functionality that must be retained. When the primary or a secondary server fails, there must be an algorithm in place that allows the operational server privileges to pass to the next active server automatically. This process must happen with the minimum possible delay, and must ensure that the end-user interface and experience is not affected as a result of this switchover. This mechanism is of paramount importance to those companies that provide resources that need to be hosted/accessible at all times [11]. Depending on the physical locations and the type of networks that the servers form, an order of precedence must be defined clearly that decides the "next" primary server in any case of failure.

For a more modular and modernized approach, we could also design the components of the application resource such that each module of the application is designed in a abstraction-oriented approach. This would imply that when a

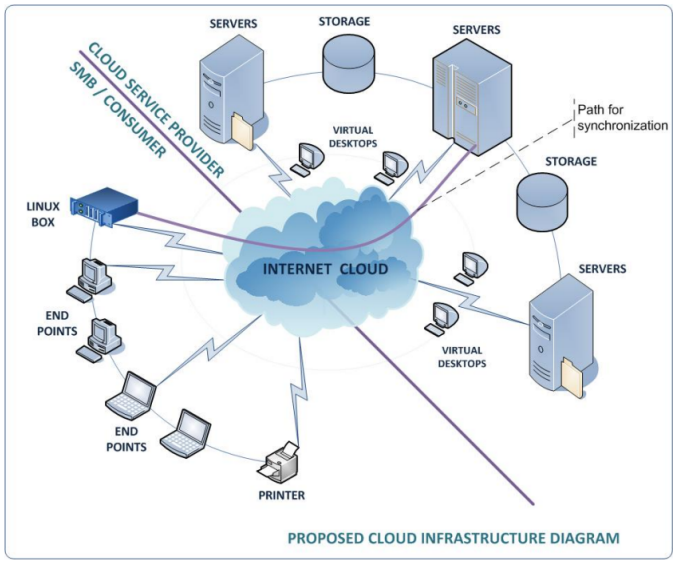


Fig. 3: A typical cloud implementation [10].

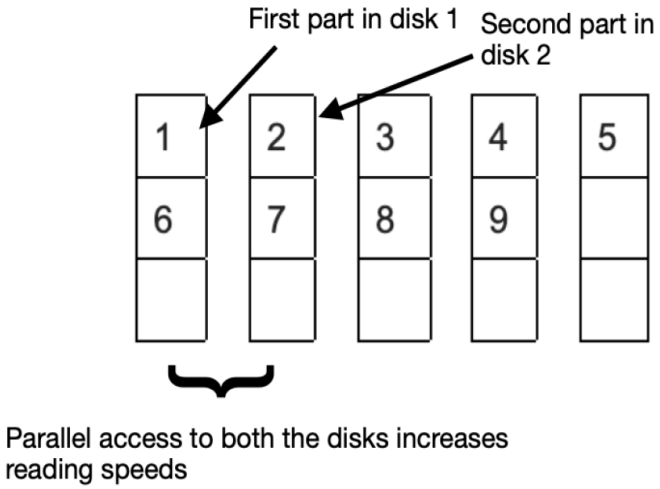


Fig. 4: Functional representation of a RAID Level 0 backup strategy.

single module of the application is corrupted on the primary server, the following operating procedures are followed:

1) First the operation is not interrupted. The role of primary server is switched over to the next server in the order of precedence.
2) Now, the module that has been compromised is identified, and its relevant data is located.
3) The uncorrupted data for this module is retrieved from a secondary server.
4) The data is updated, and the corrupted data is overwritten/deleted.
5) The current switched hierarchy continues, or reverted at the next possible moment. The affected server is able to resume normal operation.

This ensures that physical and logical resources invested for that server are not wasted completely, and the harmful data is salvaged to protect the server. This is a cost-effective approach compared to the alternative of replacing the affected server completely and formatting the data.

### *B. Redundant Array of Inexpensive Databases*

The issues faced by traditional disk-based storage techniques are chances of failure, limitations of read/write speeds, and bottleneck of physically available memory. Redundant Array of Independent Disks (RAID) is a technique that can solve all of these problems. It makes several disk drives function as though they were a single disk drive. There are a variety of configurations for RAID. A dedicated disk controller card is typically used to implement hardware RAID. The card handles all RAID-related tasks and has direct control over the individual drives in the arrays it's connected to. The arrays handled by the card look to the local system behaving as conventional storage drives with the right driver. Although some ATA-based RAID controllers exist, most RAID controller cards operate with SCSI devices.

One of the features of RAID-implemented databases/applications is that the end-user's experience is not affected by the architecture of distributed storage in the back-end. The whole architecture operates overall as if all the information was stored on a single location, on a single storage device. For database-oriented applications using RAID, the results to SQL queries could be stored by RAIDb controllers. The resolution and consistency of the cache are at the controller's control. To further improve performance flexibility, further features can be provided, such as connection pooling. The range of services available through the RAIDb controller is unrestricted. For some users, inspection, consistency testing, data records, or intrusion prevention services may be beneficial.

*1)* ***Level 0 RAID implementation****:* Two or more disk drives make up a RAID 0 array. It employs no redundancy. Each drive's usable storage capacity is partitioned into chunks that are n-fold of the drive's inherent size of blocks. Data is written to the array in chunks, one at a time, to each disk in the array. Some of the benefits are bigger overall available size (simpler to store data files that have higher volume), better read/write speed (load distribution across all of the devices in the array). There is no wasted space since all of the capacity on all of the array's devices is accessible for data storage. The disadvantage of RAID level 0 is that it is less dependable. For a RAID 0 array to be available, every drive must be operational. A single breakdown of a unitary component in an N-drive RAID 0 array wipes out a proportionate chunk of the data, making the overall configuration worthless.

*2)* ***Level 1 RAID implementation****:* It employs two identical disk devices - all data is written to all the drives. When writing data to such an array, there must be two separate write processes: one to each disk. The data-read procedure, however, it is sufficient to be done once from either disk. The benefits of using this are improved redundancy (backup in case of single drive failure) and improved read performance (distribution of

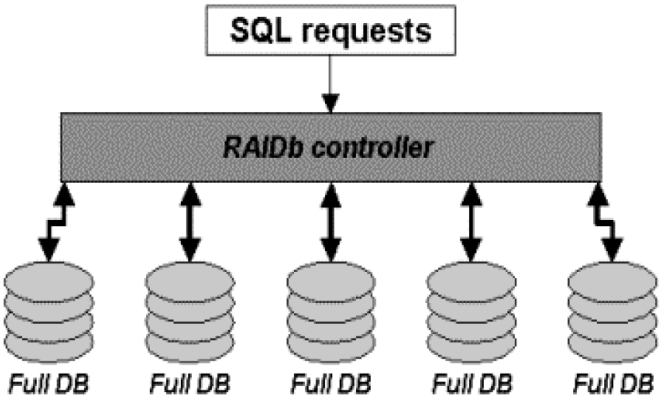


Fig. 5: A database-oriented representation of a RAID Level 1 scheme [12].

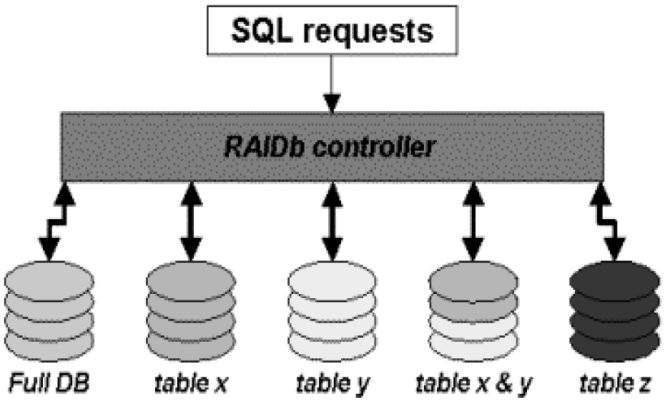


Fig. 6: A database-oriented representation of a RAID Level 2 scheme [12].

read loads). Its drawbacks include size limitations (equal to size of the largest drive) and write I/O delays (must be done by both drives simultaneously).

*3)* ***Level 2 RAID implementation****:* A compromise between RAID-0 and RAID-1 is RAID-2. It offers provisional duplication so that the level of replication can be adjusted for the greatest read/write throughput. Each data chunk must be accessible on at least two nodes in order to comply with this redundant array scheme.

*4)* ***Level 5 RAID implementation****:* It aims to integrate the advantages of the zeroth and first level implementations while reducing their individual problems. A RAID 5 array is made up of numerous drives separated into portions. Some disk space is used in a redundant manner to increase dependability. During operation, this configuration must include at least three storage devices of the same size. Each device is partitioned into elementary sectors, and data is written to each chunk in the sequence in which it was created. If one of the disks in the array fails, data can be recovered using chunks carrying parity. The data of a particular chunk from each disk is mathematically combined to compute the parity in that chunk. Any data updation must be accompanied by its corresponding

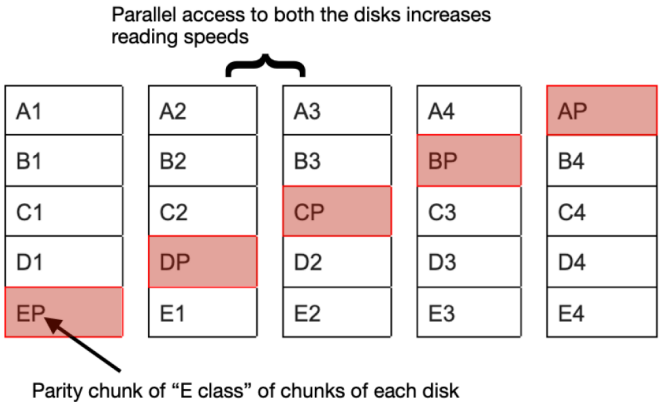


Fig. 7: Functional representation of a RAID Level 5 backup strategy.

parity's updation. This also implies that whenever data is put to the array, it must be written to at least two drives: the data drive and the parity chunk drive. The parity chunks are uniformly distributed among all drives. Although it is feasible to dedicate a particular drive to carry just parity, the frequent updation of parity might cause the parity drive to gradually lose its optimality in performance. This influence is lessened by uniformly dispersing the parity information across the array.

A RAID 5 array provides many advantages such as improved redundancy (metadata about parity bits may be utilized to rebuild the lost or corrupted data pieces) 1, faster reading (read I/O activity is uniformly distributed across all drives and only 1/Nth of the total accessible capacity is allocated to redundancy in an N-disk RAID 5 array). A RAID 5 array also has a disadvantage that is reduced writing speed because write performance is lower than a single drive because each write operation causes least two subsequent operations in the physical devices.

### *C. Replication-based data backup*

Compared to the traditional backup methodologies, this technique overcomes the implementation complexity and bandwidth crowding problems. When considering replication-based backup, we must analyze the data and see if it falls under one of the following categories [12]:

- Data being updated into the server makes key changes to the kernel or middleware of the application
- Data being updated makes changes only to the primary site
- Data being updated makes changes to a specific secondary site
- The application relies on the eager/immediate transaction updation scheme
- The application relies on the lazy transaction updation scheme

Taking all the above critera into consideration, we have the following types of replication:

*1) **Synchronous Replication**:* This refers to the backup strategy where the backup occurs instantaneously, that is, the secondary backup sites are updated with new data concurrently with the addition of new data in the primary data server. The read/write transaction is considered complete only after each of the storage subsystems on the secondary data server report completion of their updation tasks of the current data. This type of updation scheme is useful for continuously varying data, such as sensor data or metrological data, which needs to be logged in real-time. This can also be essential for financial transaction logs for banks and other payment gateways, where a large number of transactions take place at any given time. A natural consequence of this scheme is that the response time of the system becomes very high, and increases proportionately with the number of secondary servers.

The critical conditions arise for this replication scheme when the connection between the two servers is either severed or compromised to limited bandwidth, or data corruption or loss. These cases are not implicitly handled by the replication scheme itself, and separate measures must be placed to ensure that these instances do not cause total breakdown of the system [13].

*2) **Asynchronous Replication**:* Also called the *deferred update strategy*, this is the *lazy transaction updation* equivalent of synchronous replication - the updation to the secondary servers is only carried out periodically. Hence, here the read/write transaction is considered complete for the primary server as soon as its storage subsystem reports completion of the data store operation. The frequency of updation can be decided depending upon the nature of data, and the volume of data usually handled by the company, and the sensitivity of the data. High-priority, sensitive data must be backed up as soon as possible to prevent any foreseeable mishap, whereas as low-priority data can be backed up when the network is relatively idle, and client activity is low.

A natural consequence of implementing this strategy is that the image of the primary server, stored on the secondary servers are usually outdated, that is, there is a time-delay visible in the data records. This could pose problems if there is a total outage just before the periodic backup is completed. This possibility must be considered and it must be adjudged if such a threat could be expected given the nature of the company, and nature of the data.

This strategy is well-suited for networks with low-bandwidth connectivity, however it is less secure. It is not affected by network distance and transmission speed during the update operations, and its ideal use case scenario is disk-to-disk backups.

### *D. Remote Secondary Failsafe Server*

The term ”remote data backup server” refers to a server that is situated at a remote location and saves all of the data from the primary cloud (far away from cloud). Additionally, it uses the data from the remote location if the data in the central repository is lost. The goal is to help query sources in gathering information from other data sources in the event that

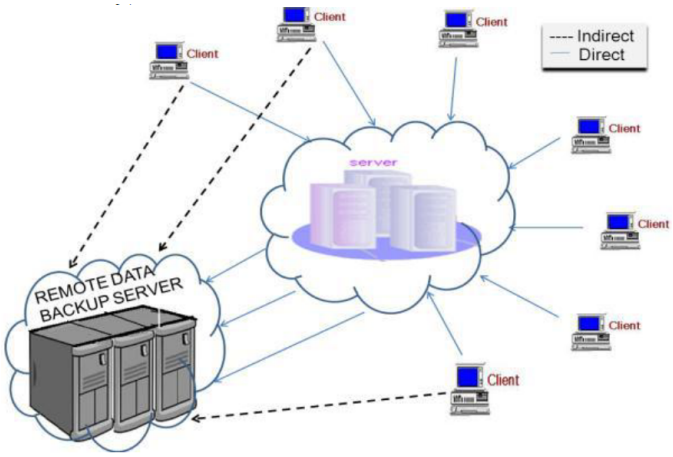


Fig. 8: A structural schematic of a remote failsafe server [6].

data transmission is unavailable or the primary cloud is non-functional in providing customers access to the requested data. Clients are permitted to access the files from the secondary repository, as indicated in Fig. 8, if they cannot access data from the default location .

This mechanism, in order to be deemed fit for deployment in a particular industrial scenario, must satisfy the following requirements:

1) Authentication and access controls
2) Dynamic server role-switching
3) Consistency
4) Data encryption standards
5) Minimal-cost solution
6) Minimal retrieval time

All of these above requirements represent a certain aspect or avenue of vulnerability for a data-centric industry. It is essential to address all these concerns in close context of the actual deployment situation to best judge if all the foreseeable precautions have been taken in order to safely store and backup data in a real-time deployed use case.

*1) **Authentication and access controls**:* Various customers can access the cloud using various user credentials or following any verification protocols. They can freely choose to store their confidential and essential data to the cloud. In order to ensure data privacy and ownership, only the authorized users should have access to his personal information and be able to read, write, or conduct any other operations on it. These authentication and access control details must be maintained by all images of the central server.

*2) **Dynamic server role-switching**:* Servers must be relocated to the virtual world in order to gain back access to lost data. Data from the primary server is sent to a new server; however, the client is not intimated of the changed location. The clients get the data in the exact similar experience as earlier, so that it gives the customers and other intermediaries transparency regarding information related to the migrated server while the data is being transferred to an alternate server.

*3) **Consistency**:* The properties of dependability must be present in the remote cloud. Because each client depends on the primary cloud for every individual data in cloud computing and the reason for this is that the primary data server keeps whole of the data by default, the cloud and distant fallback cloud must have trustworthy functionality. This requires that both servers are capable of instantly incorporating data to clients whenever needed, whether from the primary cloud or a distant server.

*4) **Data encryption standards**:* The consumer's data is securely kept in a centralized database. Its peripheral repository should also adhere to such security. Data should be completely safeguarded in remote repositories so that no third parties or other clients may access them or harm them, whether on purpose or accidentally.

*5) **Minimal-cost solution**:* When constructing the framework for the default primary server and the corresponding alternate site, the cost of implementing the distant server and its recovery and backup technique are also crucial. Small businesses should be able to sustain the costs and resource demands, and major industries should spend as little money as feasible, thus the cost of setting up the peripheral system and implementing its approach must be as low as possible.

*6) **Minimal retrieval time**:* Since the peripheral data retrieval site is not close to the main data source and its clients, the data recovery procedure takes some time while retrieving data from it. Because of this, the time required for such a process must be as short as feasible so that the querying node may receive the results with minimal time lag without having to worry about how distant the offsite repository is from the consumer.

### *E. High Security Distribution and Rake Technology (HS-DRT)*

The HS-DRT is a ground-breaking idea for file recovery that takes use of a quick and efficient ultra-widely dispersed data transmission system [14]. It is made up of of the following components:

- Data Center
- Supervising Server
- Storage points/nodes; made of different end-point, consumer-grade devices. They share a encrypted communication channel with both the servers.

The actual recovery pipeline is comprised of two sequences, backup and recovery.

*1) **Backup Sequence**:* When the storage system gets the data that it needs to safeguard, it partitions and unorganizes the data, splits it into some independent chunks. These are then duplicated to fulfill the needed recovery rate in accordance with the predetermined and static parameters. The Data Center distributes them in a random sequence to the client nodes after encrypting each fragmentation once more, along with metadata needed during the decoding process. This semantics is made up of fragmentation, duplication, and distribution-related data as well as cryptographic keys (both in the first and second phases).

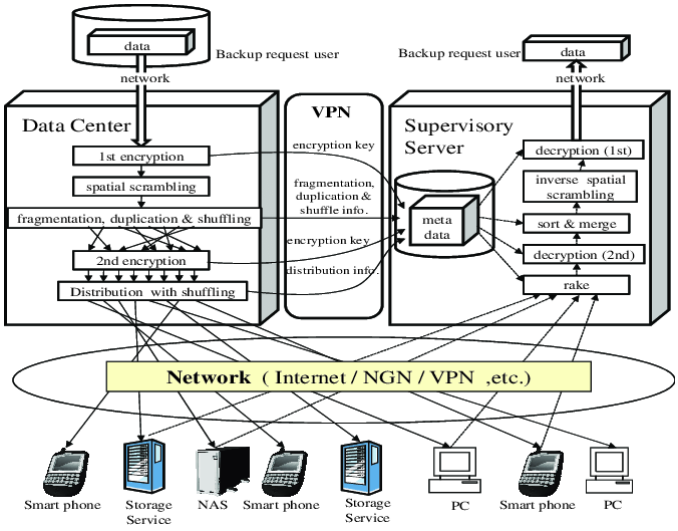


Fig. 9: High Security Distribution and Rake Technology implementation proposed by Ueno *et al.* [14].

*2) **Recovery Sequence**:* When a crisis hits or on a regular basis, the supervising server initiates the backup process. It gathers the encrypted pieces from various suitable clients, such as the rake reception process, and then decrypts, merges, and organizes them in reverse order to finish the decryption. The Supervisory Server is able to restore the original data that has to be recovered thanks to these procedures.

This paradigm, however, has significant drawbacks, making it difficult to claim that it is the ideal back-up and recovery option. Which are: First, any web-centric deployments must be properly configured to use the HS-DRT engine in order to fully leverage the HS-DRT processor. The second is that the web platform's processor speed would suffer as a result of the growth in the number of duplicated images of the target data.

### *F. Parity Cloud Service Technique*

PCS essentially offers a parity-based backup solution. The PCS is really straightforward, can entirely allay users' worries about user privacy, is simple to use, costs a significant degree on the server side, and has a high enough likelihood of recovering user data. It creates a virtual image of the physical disk resource on the user's PC for the purpose of backing up private data, creates a parity group using virtual disks belonging to different users, and then saves the metadata of the parity group in cloud storage. Neither the production of parity nor the recovery of data involves the transfer of any user data. Each user just needs to back up their data to their own storage device so that they may request backup and recovery from the PCS agent program in the future.

The PCS server creates and maintains the Virtual Disk Parity Group (VDPG), which is kept in secure storage. Three main parts make up the PCS agent software: Virtual Disk Interface (VDI), Recovery Manager (RM), and Storage Manager (SM). Users can access the this virtualized resource using the storage device interface. Through VDI, users may back up their data

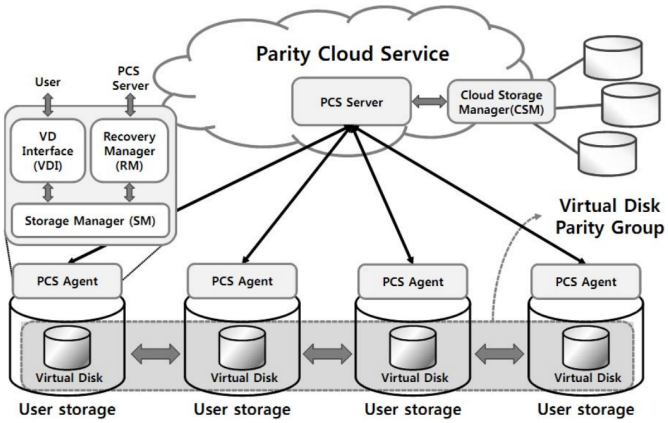


Fig. 10: PCS schematic diagram [15].

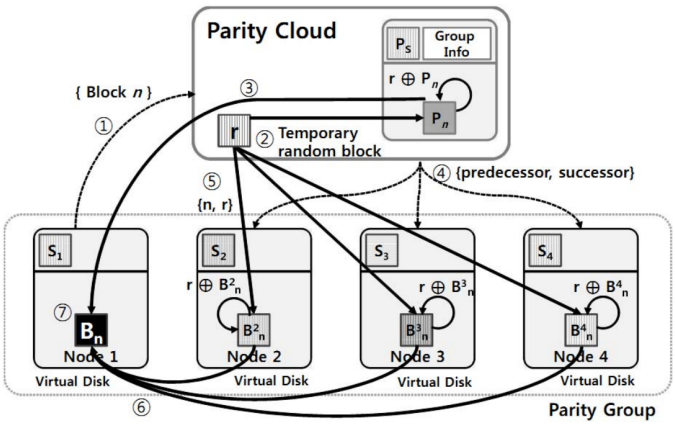


Fig. 11: The recovery process in PCS scheme [15].

to the virtual drive. Them can think of the virtual disk as a regular storage location since the VDI gives users access to normal file copy and delete operations. For recovering data, the Recovery Manager talks with the PCS server in the cloud. It reads data bits for Recovery Manager and saves the data appropriately. The Storage Manager manages the existence and logging of blocks in the virtual drive to and from each file, much like regular file-systems do. It also manages disk information and file metadata. For each virtual disk, the PCS server creates parity groups (VDPG). It keeps track of each VDPG's metadata, including the group ID, and network ID of the client nodes, and relevant data corresponding to the registered users. It also works with recovery managers to create parity blocks, which are then stored in online storage.

*1) **Parity Initialization**:* This creates the root generator block for the target disk. Each Recovery Manager assigned for this purpose receives the prompt from the respective servers. The PCS server sends the first node a transient block, generated arbitrarily, r, after sending the initialize event. The node first in the sequence creates an interim parity block after receiving the r block and delivers it to the next node in the chain of fragments. As a result, node 2 creates a transitional parity block by performing encryption operations on the the parity block with its root block and transmits it to subsequent nodes. To create the root validation block across all root blocks, the last block sent from node 4 to the PCS server is again operated on with the transient block.

*2) **Updation of Block**:* The parity creation schema is maintained by the a managing software created for this purpose. It indicates if the validation block has been produced for each of the unfragmented blocks on the target device. After initializing any block of data in the virtual disk, the bitmap is set with a stable start value. On every update operation on block, the root schema is referenced to. The manager software refers to the referenced value when a block in the first node is to be upgraded to a new block. If upon checking, the manager program finds that it is zero, an interim block is created after operation with the new block with the root node and sets the relevant value in the schema to 1. If not, the new block and the old block are XORed to create the intermediary parity block. The server also manages the root schema for every VDPG. Keep in mind that the PCS server and the data updating node may both simply change the parity block. Other nodes are not required to take part in the update procedure.

*3) **Recovery of Lost Data Blocks**:* It is feasible to recover a corrupted block of data by using the a validation node provided by the server and the ciphered blocks referenced by other nodes in the group of all validation nodes. Consider that node i's bin, the $n^{th}$ data block, is broken. Node I sends a restore message to the data server. Upon receipt of the prompt by the server, the data server tracks the sender node in the network of data blocks and reads the pertinent parity block, Pn. Then, during the recovery process, it generates a arbitrary parity block, Pr, and an intermediate arbitrary block, r. Each node creates its own encrypted block of data, E, after it receives the message by operating on each data block with its corresponding index value, and then delivers it to the node i. Keep in mind that repeating the aforesaid data block recovery technique will allow you to restore the entire virtual disk corruption. Apart from the optimal efficiency that the outlined algorithm displays above, this technique overall somehow falls short in offering ideal recovery and backup solutions because of several restrictions.

### *G. Efficient Routing Grounded on Taxonomy (ERGOT)*

A routing-oriented system for recovery operations in infrastructures that are primarily decentralized, mostly in the domain of cloud computing, is called Efficient Routing Grounded on Taxonomy [16]. The reason this strategy is chosen over others is because it totally relies on the conceptual likeliness between descriptions of its operations and requests that are received during its operation; it is not a fallback technique. Additionally, it makes use of both finer-grained and coarse-grained service configuration definitions.

ERGOT is composed of three parts. These elements consist of: 2) A SON (Semantic Overlay Network), which facilitates the aggregation of peers that have conceptually comparable service descriptions, is used to advertise structural and technical descriptions which draws its ideas using concepts from

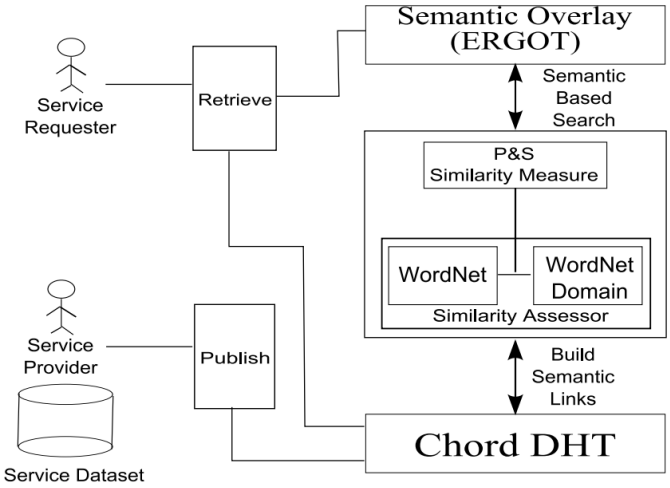


Fig. 12: ERGOT implementation schematic [16].

taxonomy. A measure of semantic similarity across service metadata, the SON is created progressively as a byproduct of product publicity through DHT.

Both the network designs for DHTs and SONs have some drawbacks. Therefore, ERGOT blends these two network theories. By creating a SON over a DHT, the ERGOT system offered conceptually-motivated query responding in DHT-based systems. A thorough analysis of the system in various network circumstances showed its effectiveness in units of search performance and real-time traffic handling in the network. However, DHT-based algorithms do not show as impressive results when used in cases of similarity measure query deployments. They only execute precise queries with logarithmic performance constraints.

### *H. Shared Backup Router Resources (SBBR)*

According to a study, one strategy essentially deals with the probable scenario of a router breakdown scenario and considerable cost reduction, with the motive of optimizing throughput and services offered, per capita (SBBR). The pivotal element it gives the transmission management architecture via multi-layer signaling is IP logical connection, which remains operational even after a router outage [17]. Some discrepancies still persist, unfortunately, between logical and physical products that might cause possible efficiency issues when considering the notion of optimizing cost-efficiency. This, in turn, demonstrates how the setup of the SBRR architecture is directly impacted by service-imposed maximal outage requirements, given there are the bare minimum resource locations distributed over a respectable coverage area. It cannot, however, combine the optimal notion with cost cutting.

### *I. Cold and Hot Backup Service Replacement Strategy*

When a potential architecture-side fault is discovered, the Cold Backup Service Replacement Strategy (CBSRS) recovery procedure is started. Its operation is only started when the above criteria is satisfied. This immensely effective backup technique is used in the Hot Backup Service Replacement Strategy (HBSRS) for resource composition in dynamic networks [18]. It resinstates the composition of the resources that is dynamically allocated based on availability and the existing constraints on the existing infrastructure, taking into account previous outages and breakdowns. The failsafe facilities always stay in the ready-to-deploy mode during execution of a program, and the first few rounds of operation results are then analyzed to improve existing code. The motive of this to ensure that deployed configuration works as intended, and unforeseen bugs are resolved at the earliest.

### *J. Rent Out the Rented Resources*

Since cloud-based resources and functionalities are often expensive for small-scale and low-turnout clients, many businesses and people are naturally drawn to those that offer more economic yet satisfactory service schemes. The model that we determined to be the lowest cost was ”Rent out the Rented Resources” [19]. The primary objective of this model is to devise methods via which the net costs incurred can be made minimal, without compromising considerable functionality. The results obtained from the study suggested that the most suitable architecture would be a multi-phase cross-cloud collaboration architecture. The three stages that would be involved are identification, pairing, and validation. This strategy is based on the predictable notion that a cloud service provider that generates its income from leased resources and then provides resources to its customers from these ones, in the form of cloud computing following virtualization. The symbiotic arrangement benefits both parties, bringing both the resource owner and the middleman their desired profit margins. It is founded on three main goals: 1) It reduces the expense of the cloud platform. 2) By lowering overheads for cloud suppliers, it offers reasonably priced cloud services to customers. 3) It provides existing businesses with financial gain by using extensive underutilised infrastructure and technology (cooperating ventures).

### *K. Linux Box Architecture for Data Backup*

This is an alternative method that centers around the idea that the most important factors to be considered are operational costs and disaster-proofing. A welcome consequence of this fact that it becomes fairly simple to change cloud service providers as and when required. The costs are set to accomodate all clients from all viable economic levels. This approach ensures that customers' reliance on ISPs and the corresponding recovery expenses are done away with. A Linux box device, a cheap hardware setup, may sufficiently perform all the required functions by synchronising data at all hierarchical levels from the client to the server vendor [4]. It comprises of a Linux program that will do on-site disc backups, for reinforcing any lost data even after the reliability assurance from the cloud vendor. The program will use a encrypted and intrusion-resistant route to transmit data from and to the cloud, periodically update firmware and information, and ensure that the on-site image does not have any considerable time-lag at any point of time. The data transmission will

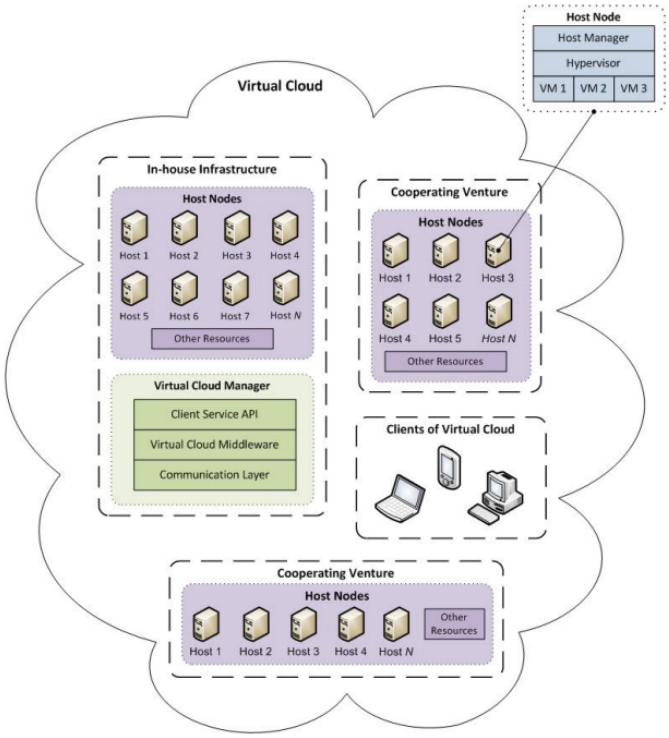


Fig. 13: The cloud architecture as proposed by Malik *et al.* [19].

be monitored and guarded. Following a successful login, the application uses in-flight cryptography and IP Security to secure the link. The application then communicates directly with the cloud system environment's application stack and initiates a voluntary and immediate complete backup. It merely synchronizes the incremental data on the cloud with local site database on an eventual inspection. The restriction we discovered is that a customer can backup both the data and the entire virtual machine itself, which uses up traffic because it backs up the full virtual machine each time a restore operation is performed.

## III. Case Study: Data Backup in the Oracle Database

### A. Overview

Each of the recovery techniques offered by Oracle has a unique quality that enables technical administrators to choose a different approach depending on their operational scenario. Cold backup and hot backup are two categories of backup technology. This approach is not commonly utilised and has only been employed in a few test systems and limited scale databases since cold backups may backup data in a consistent condition but databases cannot be used during the restore procedure, which cannot be tolerated by certain operating enterprises. Then users concentrate on hot backup, which allows databases to remain accessible during backup periods and uses hot backup to seamlessly enable transaction records. This system is adaptable and well-liked in various businesses. One sort of file is called database dynamic physical files, which includes data files, policies and protocols, and redo log files. A database has several crucial files that directly impact how it runs. These files are all backup targets since they are all necessary for restoration if a database crash occurs.

Hot backup refers to system restore procedures that may be carried out without disrupting database operations or invalidating operational systems based on databases. DBAs are in favour of this backup technique once medium- and large-scale databases are established. Oracle offers a variety of tools and ways to carry out hot backup jobs, and it may be divided into physical backup and logical backup.

### B. Operating System-centric Hot Backup

The assumption is that the data repositoryis required to operate in the stored model and that the DBA should be knowledgeable with the underlying schema of Oracle. Prior to current Oracle solutions, DBAs would always do database hot backups using OS utilities. To stabilise the condition of the page table, the DBA can run a few short queries. After that, they can perform some duplicate procedures as a cold backup. This approach is simple to use and comprehend, making it appropriate for beginning DBAs dealing with straightforward database servers. Although this secondary objective is easy to use, there are drawbacks as well. Additional rewrite logs will be visible when a large transaction is active during the restoration phase, which might boost the number of online rewrite switches and the waiting time for log file switches, impede responsiveness, and impair database functionality.

### C. Specialized Recovery Manager-based Hot Backup

Oracle Recovery Manager (RMAN) is a tool that can handle all of your data recovery and backup needs. Because of its apparent complexity and its control over carrying out crucial operations, RMAN is frequently used with caution by DBAs. The established backup and recovery procedures are conventional. RMAN is a powerful tool for implementing backup functionality. It offers a number of advantages. RMAN can also predict and locate corrupted data chunks concurrently during its backup process, and perform I/O operations alongside, keep an periodic record of all data loss prevention activities, and have built-in logging and reviewing procedures. Tablespace are not put in passive mode, therefore there is no additional redo log creation during online image stores.

RMAN may be used to perform backups whether or not a restoration catalogue is present. The recovery catalogue stores data on past backups as well as data on backups from other database iterations. Additionally, the targeted platform's configuration files may contain this data. A restoration catalogue improves backup and restoration efficiency when the database employs a lot of data files (more than 1000). Multiple backups should be preserved for important files, containing credentials or system configuration files, on various drives, tapes, and/or workstations if a backup catalogue is not being used. Without RMAN, these backups must be recoverable. When utilised, the recovery catalogue database also has to be maintained and

backed up, and because of its modest size, the DBA always exports the database to a bit stream.

### D. Backups using Data Pumping

Before Oracle 10g, DBAs would always utilise the virtual restore function, export convert data to binary file, which could export specific schema objects, tablespaces, and records. Oracle has created a new export tool called expdp, which keeps the majority of essential functionality while also adding some new, practical features [20]. This tool's quick transit speed is its standout feature. It may be used to transfer items inside a single database, move information across databases, relocate a particular tablespace to another database, and perform logical backup and recovery operations.

### E. Flashback Recovery

Previously, if a user accidentally modified data, there was no ideal option than to do a system recovery and utilise repository dumps to retrieve the original data. This procedure is tedious and time-consuming, but it may be streamlined by utilising the flashback feature. When a database encounter crashes, database recovery can be replaced with database flashback since it is simpler and faster. The system must be run in archivelog mode, the flashback function must be activated before launching the system, and the flash recovery area and retention must be set to a suitable value before flashback may be utilised.

## IV. Discussion

Data Backup is a copy or archive of important information on a device. Backup and recovery is the act of making copies of data and storing it securely in case it is lost or damaged. Then, the data is restored to the original location or a secure substitute so that it may once again be utilised in operations. In the case of a main data failure, the backup's goal is to make a duplicate of the data that can be retrieved. Primary data failures may come from hardware or software issues, data corruption, or a human error like a hostile attack (virus or malware), data deletion accident, or another human-caused incident. Data backup includes managed systems, information security integrated management systems, and corporate data backup. By doing a risk analysis, it is also possible to safeguard important corporate data in the case of a hardware failure, hacker intrusion, and many other dangers to digitally stored data.

Data security risks are increasing. Phishing, ransomware, and other malicious assaults can severely impair your productivity and result in data loss that is financially ruinous [21]. Data is not just at risk from attacks. Critical data loss might result from a simple human mistake. Data loss and power disruption are two additional effects of natural catastrophes including hurricanes, floods, and fires. Any sane disaster recovery strategy must include data backup as it is one type of disaster recovery. On a variety of media, such as magnetic tape cartridges, CDs, hard drives, or arrays, solid-state or flash drives, you may make partial or complete backups. Enterprise

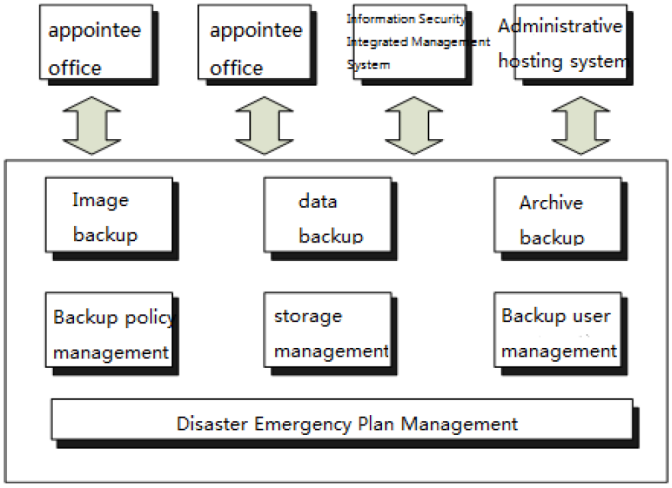


Fig. 14: Backup technique for Data Operation Access [22].

data backup and database backup are both becoming more and more common archive destinations nowadays in the cloud.

Depending on the circumstances, the technique of data backup may be full back up or incremental backup, or it may even take the form of a mix of the two methods [23]. There are two formats available for transmitting and storing backup data: file format and data stream format. While the latter is appropriate for backup storage of tiny files, the former is excellent for backup storage of vast data.

*1)* ***Full Backup:****:* This is a fundamental backup procedure that copies all your data to a different media set, like a disc, tape, or CD. Thus, all your data is made completely accessible on a single media set. Because it takes longer to complete and takes up a lot of storage space, it is frequently combined with a differential or incremental backup.

*2)* ***Incremental Backup:****:* With this procedure, just the data that has changed since your last backup process is copied. The time and date of each backup activity will be noted and tracked by a backup programme. Compared to a complete backup solution, this procedure is quicker and uses less storage space.

*3)* ***Differential Backup:****:* Differential backups are similar to incremental backups in that they transfer all modified data from a previous episode, but they also keep copying all the data that has changed since the last complete backup, which is specified in the backup's name.

The data storage and backup system include features including data storage, data backup and recovery, system backup and recovery, and application backup and recovery [24]. It is focused on the application's database, business system, and core server. The service objects, implementation procedures, and system architecture of the data storage and backup system are shown in Fig. 14:

The main distinction between backup and recovery is that the former refers to the process of restoring your database to its proper (original) condition after a failure, whilst the latter is a copy of original data that may be utilised in case of a database failure. Data, control files, log files, and archived

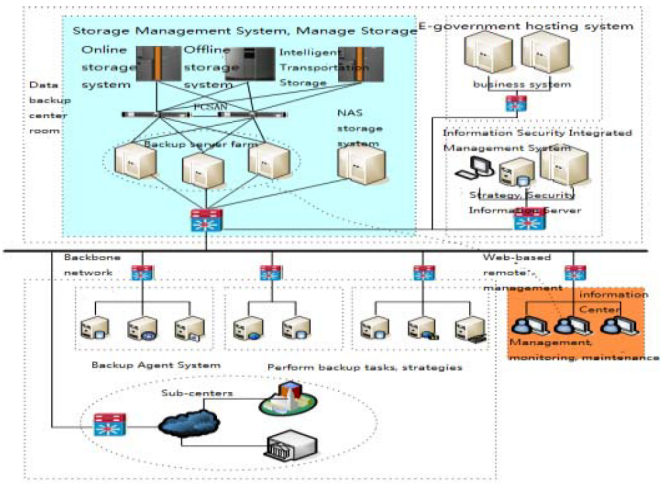


Fig. 15: Diagram of a data storage and backup system [22].

redo logs are all examples of physical database files that are included in a physical backup. It serves as the backbone of the database recovery method and is a copy of the files that hold database data in another place. Logical backups are made up of tables, procedures, views, functions, and other logical data that is taken from databases. However, as logical backups only offer structural data, they are neither advised nor helpful on their own.

In contrast, recovery aids you in returning your database to its original condition in the event of a failure. Since it enables the database to return to a consistent state after an unexpected failure, it increases database dependability. The design and development of the data storage backup system uses a two-stage backup system architecture, with an offline tape backup as a secondary backup system and network storage backup as the primary backup system. Two Fibre Channel switches, SAN storage array devices, VTLs, and tape libraries make up the data storage backup system. Figure 15 illustrates how the system is built.

The entire solution completely fulfils the client's needs for data, system, and application backup. The apparatus can rapidly restore any PC or server in the branch in a matter of seconds and may recover swiftly any equipment in the storage and server hosting facilities backup centre in a flash. This guarantees the most dependable degree of business continuity.

## REFERENCES


[1] S. Tekin, K. Bicakci, O. Mersin, G. N. Erdem, A. Canbay, and Y. Uzunay, "Optimal data backup policies for information systems subject to sudden failure," *Journal of Quality in Maintenance Engineering*, no. ahead-of-print, 2022.

[2] H. Kim, H. Y. Yeom, and Y. Son, "An efficient database backup and recovery scheme using write-ahead logging," in *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, pp. 405–413, IEEE, 2020.

[3] A. Z. Abualkishik, A. A. Alwan, and Y. Gulzar, "Disaster recovery in cloud computing systems: An overview," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020.

[4] A. A. Tamimi, R. Dawood, and L. Sadaqa, "Disaster recovery techniques in cloud computing," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 845–850, IEEE, 2019.

[5] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 27–33, 2010.

[6] K. Sharma and K. R. Singh, "Online data back-up and disaster recovery techniques in cloud computing: A review," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 5, pp. 249–254, 2012.

[7] M. A. Abd Elmonem, E. S. Nasr, and M. H. Geith, "Benefits and challenges of cloud erp systems–a systematic literature review," *Future Computing and Informatics Journal*, vol. 1, no. 1-2, pp. 1–9, 2016.

[8] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation computer systems*, vol. 28, no. 3, pp. 583–592, 2012.

[9] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud computing: An overview," in *IEEE international conference on cloud computing*, pp. 626–631, Springer, 2009.

[10] V. Javaraiah, "Backup for cloud and disaster recovery for consumers and smbs," in *2011 Fifth IEEE International Conference on Advanced Telecommunication Systems and Networks (ANTS)*, pp. 1–3, IEEE, 2011.

[11] A. Cavalli, A. Pagano, O. Aidel, C. L'Orphelin, G. Mathieu, and R. Lichwala, "Geographical failover for the egee-wlcg grid collaboration tools," in *Journal of Physics: Conference Series*, vol. 119, p. 062022, IOP Publishing, 2008.

[12] T. Singh, P. S. Sandhu, and H. S. Bhatti, "Replication of data in database systems for backup and failover-an overview," *International Journal of Computer and Communication Engineering*, vol. 2, no. 4, p. 535, 2013.

[13] P. Brouwer, "The art of data replication," *An Oracle Technical White Paper*, pp. 1–26, 2011.

[14] Y. Ueno, N. Miyaho, S. Suzuki, and K. Ichihara, "Performance evaluation of a disaster recovery system and practical network system applications," in *2010 Fifth International Conference on Systems and Networks Communications*, pp. 195–200, IEEE, 2010.

[15] C.-w. Song, S. Park, D.-w. Kim, and S. Kang, "Parity cloud service: a privacy-protected personal data recovery service," in *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 812–817, IEEE, 2011.

[16] G. Pirro, P. Trunfio, D. Talia, P. Missier, and C. Goble, "Ergot: A semantic-based system for service discovery in distributed infrastructures," in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 263–272, IEEE, 2010.

[17] E. Palkopoulou, D. A. Schupke, and T. Bauschert, "Recovery time analysis for the shared backup router resources (sbrr) architecture," in *2011 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2011.

[18] S. Suguna and A. Suhasini, "Overview of data backup and disaster recovery in cloud," in *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pp. 1–7, IEEE, 2014.

[19] S. Malik and F. Huet, "Virtual cloud: Rent out the rented resources," in *2011 International Conference for Internet Technology and Secured Transactions*, pp. 536–541, IEEE, 2011.

[20] F. Robert, "Oracle database 10g new features," *Osborne Ora-cle Press Series*, 2004.

[21] P. Prajapati and P. Shah, "A review on secure data deduplication: Cloud storage security issue," *Journal of King Saud University-Computer and Information Sciences*, 2020.

[22] Y. Zhao and N. Lu, "Research and implementation of data storage backup," in *2018 IEEE International Conference on Energy Internet (ICEI)*, pp. 181–184, IEEE, 2018.

[23] B. Zhu and B. Liu, "Research and implementation of e-government data disaster recovery in administrative units. comput," *Knowledge Technol*, vol. 21, pp. 5059–5056, 2011.

[24] J. Huang, "Research and implementation of data backup system," *Huazhong University of Science and, Technology (HUST)*, 2008.