

## Cloud Algebra: An Innovative Approach for Managing Resources, Services and Big Data on Clouds

Mohd Abdul Ahad<sup>1,\*</sup>, Gautami Tripathi<sup>1</sup> and Sara Paiva<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, SEST, Jamia Hamdard, New Delhi-110062, India

<sup>2</sup>Instituto Politécnico de Viana do Castelo, Portugal

### Abstract

In the current era of technological advancements, cloud computing is considered as one of the most promising computing paradigms. It is cost-effective, energy efficient, scalable and is location independent. In simple terms, a cloud computing technology provides various computing tools, facilities and mechanisms as a service to the end user. A user can opt for using these services as a pay-per-use model. This technology is indeed highly cost-effective, environment friendly and is a preferred option for those who don't want to spend much on infrastructures, platforms or physical space required for setting up the enterprise. Cloud computing technology provides services like Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS), Computing-as-a-Service (CaaS) etc. Recently a new service called DaaS (Data-as-a-Service) has also emerged in which data is provided as a service to the users. This paper proposes a concept of creating algebra for the cloud computing environment called as cloud algebra (CA). Using the proposed cloud algebra we can perform several basic mathematical functions like addition, deletion, union, intersection and other aggregate functions on clouds directly. This means that using the proposed cloud algebra, two or more clouds can be added. One cloud may be joined with other clouds and two or more clouds can be compared and so on. Furthermore, the data stored on the clouds can also be effectively and efficiently managed using the proposed cloud algebra. Along with the data, the cloud services and resources can also be managed in a much effective and efficient manner using the proposed cloud algebra.

**Keywords:** Energy Efficiency, Cloud computing, big data, cloud query language, Cloud Algebra, Pilot Computer

Received on 19 September 2019, accepted on 30 November 2019, published on 14 January 2020

Copyright © 2020 Mohd Abdul Ahad *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.13-7-2018.162687

---

\*Corresponding author Email: [itsmeahad@gmail.com](mailto:itsmeahad@gmail.com)

## 1. Introduction

Cloud computing has brought a revolutionary transformation in the way processes and businesses use IT resources. With the advent of cloud computing the major computing services like storage, servers, intelligence, analytics etc can be delivered over internet on demand. The special features and benefits provided by the cloud infrastructure helps businesses and processed to utilize the IT resources and infrastructure in an efficient way. Table 1 shows some of the major benefits of Cloud computing.

Table 1. Benefits of Cloud Computing

S.No	Benefits of Cloud Computing Approach	Description
1	Cost Effective	There is no need to spend huge amount on hardware and software requirements and maintain data centers. The resources can be used as and when required paying on usage basis
2	High Performance	Since the data and computation capacities can be geographically distributed, it can facilitate parallel processing and hence increase the overall performance of the system.
3	Flexibility and Speed	With cloud, we have the provision to add, remove, increase or decrease the computational facilities, storage, infrastructure and other dependencies as per the need and requirements which make it highly flexible
4	Scalability	With pay per use model, the cloud technology provides highly scalable systems as per the requirements of the end users
5	Reliability	Since the services and resources are independent and isolated, it becomes easier to identify, diagnose and rectify faults. Also, with the provision of storing data and services in redundancy, it becomes more reliable and available to the users
6	Productivity	The primary advantage of cloud technology is to provide pay-per use computational model which supports high parallelism and thus improves productivity to a much greater extent than the conventional counterparts

In a typical cloud environment, in order to maintain high availability and consistency, the data and services are stored in multiple sites or locations. These locations are interconnected using a preferred network- public or private or hybrid.

Apart from the several unprecedented benefits of the cloud computing paradigm, there exists several limitations and challenges in its adoption. Some of these are given in the section below.

The current manuscript is divided into four sections. The second section provides the literature review of the existing approaches to handle cloud data and services. The third section provides the details of the proposed cloud algebra and its architectural framework. A complete set of proposed CA commands along with their syntax and semantics is provided in this section. The fourth section provides the conclusion and discussion and identify the future trends in cloud computing.

### 1.1 Limitations and Challenges of Cloud Computing

The introduction of cloud computing has brought a new paradigm in how companies can offer their solutions that start being available in the cloud over the internet and can be accessed through an on-demand system. While with benefits, especially for small and midsize businesses, such as the lower costs (of installing a complete infrastructure locally) and faster deployment, there are also several limitations to this paradigm that are given below.

The Privacy and security of data are two of the prime reasons that arise when talking about cloud computing, due to the existing distribution of resources by many customers. Security assurance is on the side of the cloud service provider and beyond the reach of the company providing the solution. Provider dependency becomes an unavoidable reality as each one uses their specific infrastructure, hardware and software which make migration between providers quite difficult. The unavailability of the service (for technical reasons or otherwise), is the sole responsibility of the provider so those who provide the solution in the cloud can be in a compromising situation with customers who want a 24/7 functional service [17]. The lack of a standard “service level agreement (SLA)” is another limitation of a cloud solution and is responsible for the uncertainty of what features the provider has to make available, also preventing easier migration between different providers. Performance is drastically impacted at times of high load as the resources are shared among numerous customers. Network latency is also a limiting aspect to consider as speed of data transfer over the internet is lower than locally. Since applications typically transfer large volumes of data, a cloud solution can cause transmission bottleneck with a direct impact on clients. The need for data storage space scalability for some solutions may not benefit from a cloud solution as cloud service vendors

have their own database connection solutions and adding more space is not something that can be done in a quick, flexible and easy way, which may be a limitation for some solutions. Finally, given the reuse of resources by multiple clients, it is possible that one misuse can have consequences for everyone who may have the service unavailable. In cases of attack, the IP can be blocked affecting all clients of that cloud service provider [16].

## 1.2 Motivation

There are several computing scenarios in which computing services from multiple clouds are required by the users. In such situations it is imperative to have appropriate mechanisms for providing synchronization between multiple participating clouds. In case a cloud is overloaded or becomes unavailable, a similar kind of cloud can be created or cloned, and the user's requests can be serviced using that cloned cloud. Several times the users require only the services and requests appropriate for the respective task. In such cases we can have the option to choose only the needed services and resources of the multiple clouds. This can be made possible only when we are able to identify and collate the services and resources from multiple clouds. Considering these factors as a motivation, the current manuscript provides a concept of Cloud Algebra (CA) which can be used to manage cloud services and resources along with cloud metadata.

The primary concern with cloud computing is the synchronization of various data sets stored at different locations, since multiple copies of same data are available at several locations, it is crucial that the updation done on one copy must be reflected in all the other copies. In such situations, the proposed cloud algebra can play an important role. Just like the concept of relational algebra to query and process the relational databases, cloud algebra can be used to process, query and manage cloud databases as well as other cloud-based services. The proposed cloud algebra will not only work for cloud databases but will also work for cloud resources, cloud management, cloud creation, cloud deletion etc. Just like the traditional relational database management system (RDBMS), we provide the notion of Cloud Definition Language (CDL), Cloud Control Language (CCL), Cloud Manipulation Language (CML) and the notion of Cloud Query Language (CQL). A cloud computing manager or cloud administrator will be responsible for performing CDL and CCL commands, while the end users will be allowed to execute CML commands only.

## 2. Related Works

Cloud computing has been one of the most preferred area of research for computer scientists around the globe. The cloud technology has emerged as a blessing of modern-day computing technology for attaining the sustainable

computing environment in true sense. This section presents some of the significant works done in the area of cloud computing paradigm covering the aspects like privacy, security, reliability, scalability, interoperability and ease of usage. The authors in [1] presented the overview of the cloud technology. They gave a definition to "cloud computing" as a delivery of software and hardware as a service over the internet to the users. This ensures a highly cost-effective business model for industries and enterprises. The concept of public and private cloud was explained highlighting the advantages of both types of clouds. Furthermore, the paper also discussed the obstacles and opportunities with cloud computing technology. The authors in [2] discussed the architecture of cloud computing paradigm and the open research challenges associated with the technology. The paper starts with highlighting the prominent features and characteristics of cloud computing. Furthermore, the paper discussed the several commercial cloud service providers and the services offered by them. The authors in [3] provided a market based cloud architecture. Furthermore, the paper discussed the paradigms for delivering IT as a service. Finally, a comparison of commercial cloud service platforms was provided in the paper. The authors in [4] provided architecture for managing data stored in the clouds. Their proposed model was based on "three schema architecture" and "three level object oriented database" architecture. In their proposed approach, the authors discussed the three level role based approach namely data centre, cloud service provider, and user. The authors in [5] proposed the concept of cloud algebra for managing data stored on the clouds. The paper talks about four key operations on cloud databases namely "union", "intersection", "difference" and "Cartesian product". It also describes an operator named "select" for fetching the data items from the clouds. Furthermore, the authors described the handling of unstructured data using an example of graph theory and performing the different set of operations on a hypothetical data matrix. The author in [6] proposed the concept of "Big Data Algebra" for performing analytics in terms of data science and engineering. It also presented algebraic operators for "modeling, analysis and synthesis" for effectively managing the intricate big data structures and variety of data. The authors in [12] discussed the different aspects of data management in a cloud computing environment. The example of Amazon Web Services was used to show the large scale tasks and processes. The authors further highlighted the need of a new database management system for cloud environments. In [13] the different approaches for the data intensive applications across the cloud environments are discussed. The need to develop the tools to handle the data flood in a cloud environment is also highlighted. The work presented in [14] shows the various design choices for developing an efficient data management system that is highly scalable to provide services in a cloud environment. Test Algebra optimization for cloud platform is discussed in [15].

The popularity of Cloud computing has attracted more and more businesses to opt for Cloud based services. With such large volumes of data available in the cloud there is a need for efficient data management and processing techniques that can help to achieve operational excellence and easy access to data. In the literature survey presented in this section we have tried to present a comprehensive survey of the researches in the area of cloud algebra and similar approaches for data management on the cloud. Many approaches were presented in the past to handle the volumes of data available in the cloud [17], [20]. However, there was no significant work related to energy efficient and effective processing of data directly in the cloud. In this paper we have tried to address the research gaps in the previous works and presented a cloud algebra for handling data in the cloud itself. One similar approach was presented in [5] where the author introduced some algebraic operations for cloud data however there were only limited operations presented. In our approach we have proposed sixteen mathematical commands which can be extensively used for managing and processing data in the cloud.

### 3. Cloud Algebra: The Proposal

The proposal of cloud algebra aims at providing the algebraic functions for performing operations on clouds. These functions can be used to manage datasets stored on the cloud as well as managing the cloud resources and services. They can also be used to perform cloud level operations like merging two or more clouds, getting status of a cloud, finding whether two clouds are equal or not, creating a new cloud, deleting the existing cloud, giving permissions to use the clouds and so on. The proposed algebra can be executed using the proposed cloud query language (CQL). Similar to the classical structured query language SQL, CQL also contains three types of commands, Cloud Definition Language (CDL), Cloud Manipulation Language (CML), and Cloud Control Language (CCL). The CDL consists of commands and functions to operate on cloud schema. The proposed CDL commands are: createCloud(), deleteCloud(), addCloud(), createClone(), grantPermission() and revokePermission(). The CML consists of the commands and functions to operate on cloud services and resources. The proposed CML commands are: isAlive(), addCloud(), requestPermission(), select(), isEqual(), isEmpty(), isBusy(), cloudStatus(), mergeCloud(), splitCloud(). The CCL consists of commands used for performing backup and recovery mechanisms. The proposed CCL commands and functions are: createBackup(), saveCloud(), recoverCloud(). The proposed cloud query language is a kind of distributed query language which can be used to perform operations in a distributed manner. The detailed explanation of CQL is provided below:

Let C be a set denoting cloud. We define C as a collection of one or more resources (R) and services (S).

Therefore  $C = \{R, S\}$ , where  $R = \{r_1, r_2, r_3 \dots r_n\}$ , a set of resources and  $S = \{s_1, s_2, s_3, \dots, s_n\}$ , a set of services.

We propose the following mathematical operations

- Union:

Two or more cloud can be added together. The union of clouds means that the services and resources of one cloud get combined with the services and resources of the other cloud. The output of union operation can have common resources and services.

Let C1 and C2 be two distinct clouds where  $C1 = \{R1, S1\}$  and  $C2 = \{R2, S2\}$ .

Then the Union operation is defined as

$C3 = C1.Union(C2) = \{R1, S1, R2, S2\}$ : i.e. all the resource and services of C1 and C2 are combined to form a new cloud C3.

- Intersection:

In this operation, only the services and resources common to both the clouds are returned.

Let C1 and C2 be two distinct clouds where  $C1 = \{R1, S1\}$  and  $C2 = \{R2, S2\}$ . And  $R1 = \{r_1, r_2, r_3, r_4\}$ ,  $S1 = \{s_1, s_2, s_3, s_4\}$  while  $R2 = \{r_5, r_3, r_2, r_7\}$  and  $S2 = \{s_1, s_5, s_6, s_7\}$ .

Then the Intersection operation is defined as

$C3 = C1.Intersect(C2) = \{r_2, r_3, s_1\}$ : i.e. all the common resources and services of C1 and C2.

- SELECT:

Suppose we want to include only few services and resource from C1 and few services and resource from C2. This can directly be done using SELECT command, which is used to select desired services and resource from the cloud(s).

- isFree () :

It is a function which returns true or false in order to identify whether a resource or service of a cloud is free or not.

isFree(Cloud, resource(s))

isFree(Cloud, service(s))

isFree(Cloud, resource(s), service(s))

- isOccupied () :

It is a function which returns True or False in order to identify whether a resource or service of a cloud is currently occupied or not.

isOccupied(Cloud, service, resource)

- distance():

It is a function which is used to identify the distance between two clouds

C1.distance(C2)

- isAlive() :

It is a function which is used to find whether a cloud is alive or not ( healthy or not)

isAlive(Cloud)

- deleteCloud() :

It is a function which is used to delete an unhealthy cloud from the network. Deletion does not mean physically deleting the resources and services, rather just excluding the particular cloud site from the network.

deleteCloud(Cloud)

- addResource() :

This function adds is a new resource to the existing cloud network

Cloud.addResource(r1);

- addService() :

This function adds a new service to the existing cloud

Cloud.addService(s1);

- createCloud() :

This function is used create a cloud with some resources and services

createCloud(Cloudname, service(s), resource(s));

- givePermission() :

This function is used to grant permission to the user for using the services and resources of a particular cloud.

givePermission (User\_id, Cloud\_id, {Services}, {Resources})

- revokePermission() :

This function is used to revoke all permission to use a particular cloud. It is also used to revoke permission to use selective services and resources of a particular cloud.

revokePermission(user\_id, Cloud\_id);  
revokePermission(User\_id, Cloud\_id, {Services}, {Resources})

- requestPermission() :

This function is used to request for permission for using the services and resources of a particular cloud.

User1.requestPermission (Cloud\_id);  
User1.requestPermission (Cloud\_id, {Services}, {Resources});

- createClone() :

This function is used to create an exact replica of an existing cloud. When a clone of the cloud is created, the cloud with exactly same services and resources is created.

createClone(cloud\_id)

- isEqual() :

It is a function which return true when two clouds have identical number of services and resources.

C1.isEqual(C2).

- createBackup()

This function is used to create a backup of the cloud at any particular moment. Once this command is executed, a snapshot of the existing cloud is created and saved in the storage.

createBackup(cloud id)

- saveCloud()

This command is helpful when we want to save the state and properties of the existing cloud. Once this command is executed, the current state and properties of the cloud are saved.

saveCloud(cloud id)

- recoverCloud()

This command is useful in cases of any unfortunate events causing the cloud to get corrupted or become unavailable. Using this command, the most recent saved position state and properties of the cloud can be rolled back. All the data, services and resources are recovered back.

recoverCloud(cloud id)

Cloud algebra is extremely important in cases when we want to find out the following by using cloud as a service:

- Which and how many cloud sites are providing exactly the same set of services and resources?
- Which clouds are identical?
- Which and how many cloud sites are having maximum service and resource consumed?
- Which cloud site has the minimum resource and services consumed?
- To identify and perform load balancing on the existing cloud.

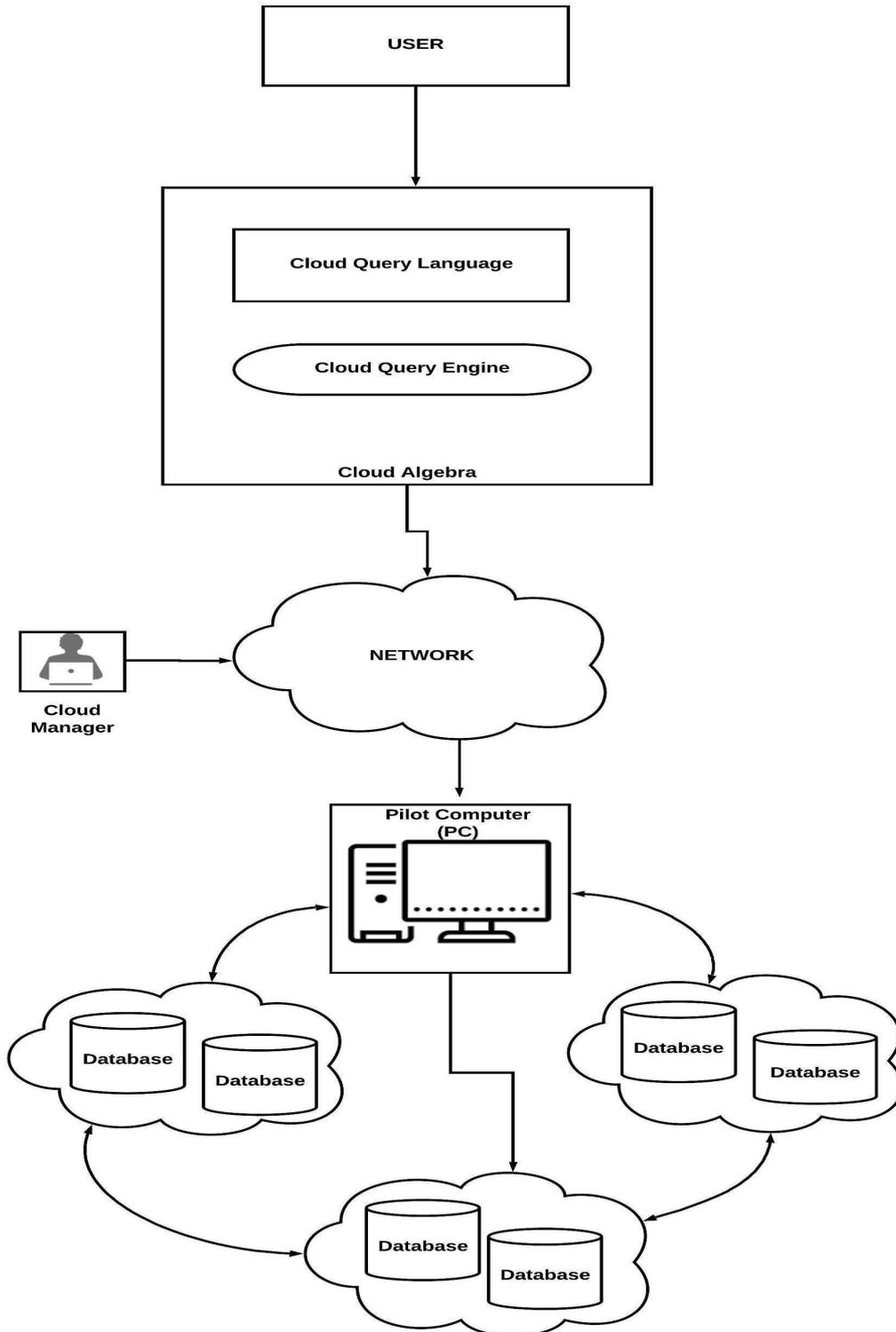
Figure 1 presents the architecture of a typical cloud database management system. Here the end user interacts with the system using the proposed cloud query language. The queries of the user are interpreted using the cloud query engine. The cloud query engine is a kind of software that interprets the user queries and performs the desired operations on the basis of those queries. The network consists of a cloud manager which is the pivot element in our proposal. The cloud manager identifies the query and selects the appropriate cloud database to execute the query. The selection of appropriate cloud depends on the following criteria.

- Distance of cloud from the user
- Existing load of the cloud
- Availability of resources and services on the cloud
- Type and nature of service or resource requested

Since cloud computing is a pay per use model, the computing, processing and storage facilities can be rented as per the needs. With cloud algebra, the users can seamlessly connect and communicate with the cloud resources and services. This is particularly important in cases of edge and fog devices and their corresponding computing paradigms. The proposed cloud algebra

facilitates the management the cloud services in an energy efficient and sustainable manner. The dependency on the

cloud manager is gradually minimized and the users can perform the desired legitimate operations effortlessly.



**Figure 1.** Architecture of Typical Cloud Database System with the proposed Cloud Algebra Unit

It can be observed from figure 1 that Cloud 1, Cloud 2, Cloud 3 and Cloud N are geographically separated cloud sites. All these clouds consist of some sets of resources and services. A user can opt for services and resources of

any of these clouds. The users also have the provision of choosing the desired services and resources from multiple clouds.

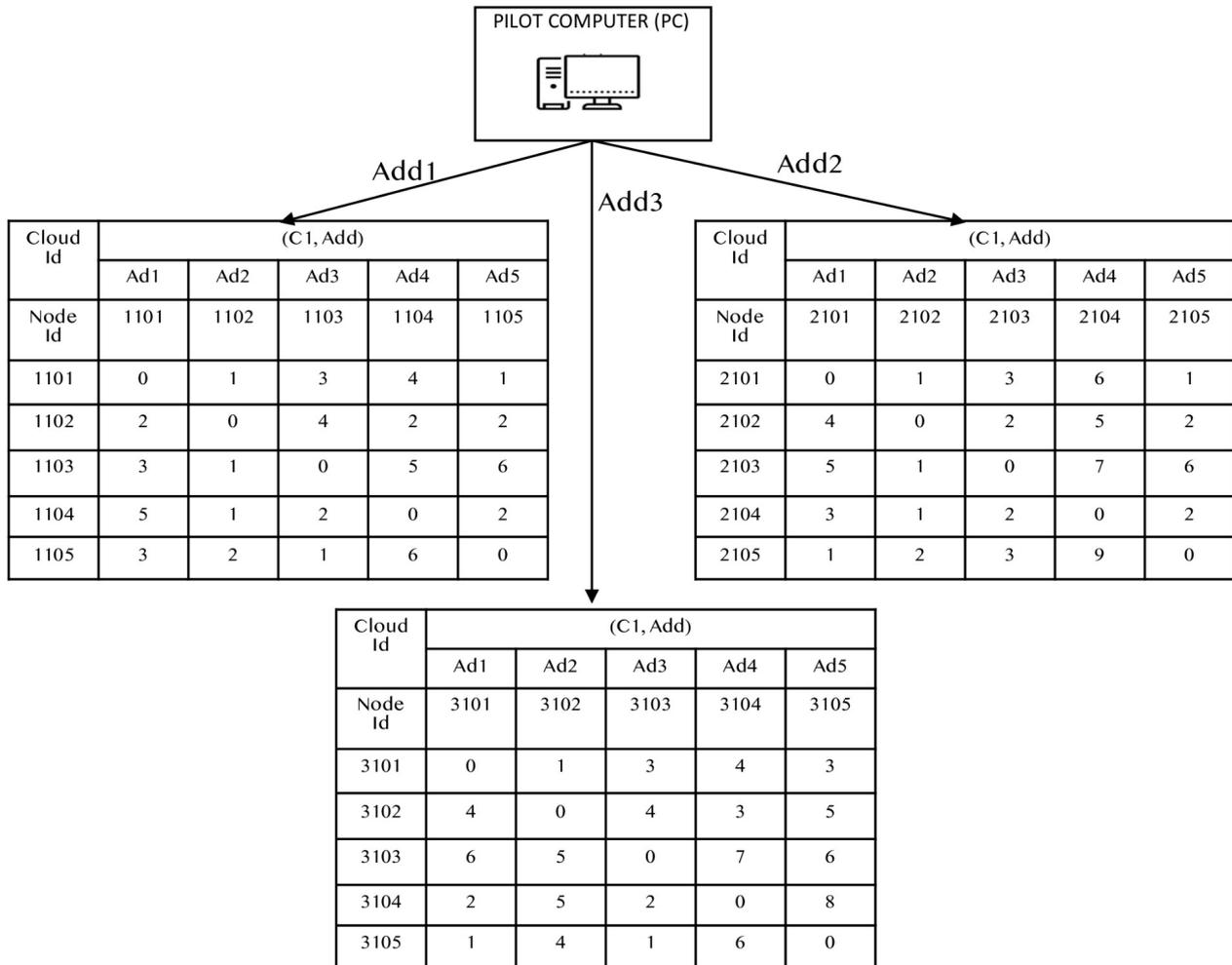
### 3.1 Working of the proposed Cloud Algebra based Model

The cloud manager analyzes the user requests and identifies the clouds for servicing the requests on the basis of the chosen resources and services. The proposed cloud algebra can be used to manage data as well as cloud schema. With the function of CML, the data stored on the cloud databases can be easily added, deleted, updated and analyzed. Similarly, higher order functions like creation of cloud, granting and revoking permission to use a cloud and its service and resources can also be done using the proposed cloud algebra. The cloud query engine acts as a parser of the queries passed to it. It understands the user queries and acts accordingly to perform the desired tasks. The difference of cloud algebra with the classical relational algebra is that the relational algebra works on structured data stored in the form of rows and columns. But the proposed cloud algebra can be used to work with structured, semi-structured or unstructured datasets. We propose to use the Atrain Distributed System to store the data on the cloud databases. ADS is a highly scalable distributed system which can be used to store large volumes of data [7]. The data within the ADS is stored in the form of Coaches. Each coach consists of larrays which may be of similar or different data types. ADS also has the provision of storing the data in a hierarchical manner wherein we can have several levels of data storage units. Since the pilot of ADS contains the metadata information about all the other coaches of the ADS. It becomes an easy task to manage the services and resources located at different geographical locations. The pilot computer (PC) contains the information like capacity of each coach, number of resources and services offered by each coach, load balancing status, distance of each coach from each other and from the pilot computer. On receiving the request from the user, the pilot computer identifies the location of the client and selects the closest cloud site to service the request(s) of the user. Since every cloud site is connected through a network to the pilot computer. The PC maintain a dynamic routing table containing the updated distances of each cloud site so as to select the optimal path for transfer of data and services from source to destination. The PC serves as a metadata repository and plays a crucial role in the selection of the most appropriate cloud for servicing the user request. It is the responsibility of the PC to identify the optimal path for the data transfer. Furthermore, the PC also ensures a

secure and privacy preserved exchanges of data and information with the help of an in-built security mechanism governed by twofish cryptographic algorithm. The pilot computer performs the task of link management wherein it stores the information of the connectivity (links) of each and every computer system connected to the multiple clouds. It implements an efficient clustering mechanism to identify the nodes (computers) which can be clustered together on the basis of several parameters like distance, load balancing, number of free resources and services etc. In order to perform the task of link management, the PC maintains a link-table consisting of the connectivity and distance information of each of the nodes within the cloud. Each node of the cloud sends regular time-stamped notification to the PC which enables the PC to identify that the node is active and running. If this notification does not reach the PC within a predefined time interval, the PC assumes that the respective node is not healthy and notifies the cloud manager to adopt corrective measures. On receiving the notification from the PC about any unhealthy node, the cloud manager identifies the problem and rectifies it. If the cloud manager is not able to rectify the problem, it delinks the respective node from the cloud and the load of the unhealthy node gets distributed among other nodes.

Cloud Id	(C1, Add)				
	Ad1	Ad2	Ad3	Ad4	Ad5
Node Id	1101	1102	1103	1104	1105
1101	0	1	3	4	1
1102	2	0	4	2	2
1103	3	1	0	5	6
1104	5	1	2	0	2
1105	3	2	1	6	0

Figure 2. An Example of Link Management Table within a cloud



**Figure 3.** Link Management of Multiple Clouds via PC

In Figure2 and 3, Node Id represents the distinct computers (nodes) in the cloud. The value in the cell represents the cost (distance) from one node to another. The value ‘0’ signifies no path or no link. The values Ad1, Ad2, Ad3, Ad4 and Ad5 represent the link address of respective nodes from the PC. And the value ‘Add’ represents the cost (distance) of cloud from the PC. These links are updated dynamically in real time on the basis of network traffic and congestions. The whole cloud ecosystem can be modeled into a problem of graph theory in order to find the optimal routes for data transfers. The PC acts as a central controller for all these routing and link management activities. Also, the PC is responsible to take necessary actions (isolate a link, delete a link) in case a link becomes inactive or is corrupted. Since the arrangement of all the nodes in the proposed system is in the form of multi-horse cart topology [7], [18-19] the complete information of the links and nodes lies in the metadata of PC.

The steps given below present the working procedure of the proposed cloud algebra.

- Step 1. The users request for the required service and resources.
- Step 2. If the user has the access privileges to the requested service/resource, it is allotted to the user through the PC from the most suitable site.
- Step 3. If the user does not have the access privilege, the cloud manager analyzes the request and the user
- Step 4. The cloud manager authenticates the user and identifies if the requested service or resource is free or not
- Step 5. If the requested service/resource is found to be free, the Cloud manager communicates with the PC to identify which copy of the service or resources is to be assigned to the user.
- Step 6. The PC on receiving the request from the cloud manager, identifies the optimal resource/service and passes its metadata information to the cloud manager
- Step 7. Finally, the cloud manager assigns the requested service/resource to the user. Once the service/resource is assigned to the particular user,

a time counter is also initiated by the cloud manager which keeps track of the time of usage by the user. Once the allotted time is expired, the service/resource is revoked from the user. If the client wants to continue using the assigned service or resource, a request needs to be sent to the cloud manager before the expiry of the allotted time.

Step 8. If the resource/service is not free, the cloud manager notifies the user to wait till the resource/service gets free.

### 3.2 Advantages of the proposed Approach

In the traditional cloud systems, the cloud manager plays an important role. All the queries, request and replies go through the cloud manager which approves and denies them with respect to the access privileges, availability of the resources, network traffic etc. Thus, there exists a major dependency on the cloud manager in case of classical cloud systems. However, in case of the proposed cloud algebra approach, the role of the cloud manager has been confined. The cloud manager comes into picture only when there is a conflict of interest between the users, authentication issues or access privilege conflicts. In all the other situations, the users can directly query the system and access the requested services/resources. Thus, the network latency can be reduced to a much larger extent. Also, the problem of dependency on cloud manager can also be reduced. Since the resources and services are directly accessible, the overall processing and data management can be greatly improved. The proposed CA promotes energy efficient and faster management of data, services and resources on the cloud along with an improved provision for access control and authentications.

## 4. Conclusion and Discussion

The proposed cloud algebra can prove to be an imperative step in attaining sustainable and environment friendly computing services. Cloud service providers offer a wide range of services to the users. The proposed cloud algebra works as an interface between the cloud service providers and the end users. The distributed and parallel query handling mechanism makes the system highly efficient and reliable. User can work very easily using the standard SQL like CQL queries to request for the required services and resources. Furthermore, the CA can also be used to manage the data stored in the cloud databases. The PC looks after the work of link management, optimal data transfer and ensure secured and privacy preserved computing environment. The proposed cloud algebra is a conceptual proposal which can be used by the programmers to build a practical SQL like query system to store, retrieve and manipulate the big data stored on the clouds. Furthermore, the cloud schema can also be manipulated using the proposed cloud algebra. Although

the concept is still in its embryonic stage, it has great potential to create a paradigm shift in managing datasets, resources and services on the cloud.

## References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50-58.
2. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1), 7-18.
3. Buyya, R., Yeo, C. S., & Venugopal, S. (2008, September). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *High Performance Computing and Communications*, 2008. HPCC'08. 10th IEEE International Conference on (pp. 5-13). Ieee.
4. Alam, M., & Shakil, K. A. (2013). Cloud database management system architecture. *UACEE International Journal of Computer Science and its Applications*, 3(1), 27-31.
5. Alam, M. (2012). Cloud Algebra for Handling Unstructured Data in Cloud Database Management System. *International Journal on Cloud Computing: Services & Architecture*, 2(6).
6. Wang, Y. (2015). Big Data Algebra: A Rigorous Approach to Big Data Analytics and Engineering. In *17th International Conference on Mathematical and Computational Methods in Science and Engineering*.
7. Biswas, R. (2015). "Atrain Distributed System"(ADS): An Infinitely Scalable Architecture for Processing Big Data of Any 4Vs. In *Computational Intelligence for Big Data Analysis* (pp. 3-54). Springer, Cham.
8. Zhu, M., & Risch, T. (2011, December). Querying combined cloud-based and relational databases. In *2011 International Conference on Cloud and Service Computing*, pp. 330-335
9. Chen, D., & He, Y. (2010). A study on secure data storage strategy in cloud computing. *Journal of Convergence Information Technology*, 5(7), 175-179.
10. Olmsted, A. (2019). Relational Algebra for Heterogeneous Cloud Data Sources. *CLOUD COMPUTING 2019*, 130.
11. Ghrist, R. (2018). Homological algebra and data. *Math. Data*, 25, 273.
12. Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. *IEEE Data Eng. Bull.*, 32(1), 3-12.
13. Sakr, S., Liu, A., Batista, D. M., & Alomari, M. (2011). A survey of large scale data management approaches in cloud environments. *IEEE Communications Surveys & Tutorials*, 13(3), 311-336.
14. Agrawal, D., El Abbadi, A., Antony, S., & Das, S. (2010, March). Data management challenges in cloud computing infrastructures. In *International Workshop on Databases in Networked Information Systems* (pp. 1-10). Springer, Berlin, Heidelberg.
15. Wu, W., Tsai, W. T., Jin, C., Qi, G., & Luo, J. (2014, April). Test-algebra execution in a cloud environment. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering* (pp. 59-69). IEEE.
16. Islam, Mohammad Manzurul; Morshed, Sarwar; Goswami, Parijat. (2013). Cloud Computing: A Survey on its

- limitations and Potential Solutions. *International Journal of Computer Science Issues*.10. 159-163.
17. Wang, Peng; Gao, Robert; Fan, Zhaoyan. (2015). Cloud Computing for Cloud Manufacturing: Benefits and Limitations. *Journal of Manufacturing Science and Engineering*. 137. 10.1115/1.4030209.
  18. Ahad, M. A., & Biswas, R. (2018). PPS-ADS: A Framework for Privacy-Preserved and Secured Distributed System Architecture for Handling Big Data. *International Journal on Advanced Science, Engineering and Information Technology*, 8(4), 1333-1342.
  19. Ahad, M. A., & Biswas, R. (2019). Request-based, secured and energy-efficient (RBSEE) architecture for handling IoT big data. *Journal of Information Science*, 45(2), 227-238.
  20. Ahad, M. A., & Biswas, R. (2018). Dynamic merging based small file storage (DM-SFS) architecture for efficiently storing small size files in hadoop. *Procedia computer science*, 132, 1626-1635.