IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-9, Sept. 2018 E-ISSN: 2347-2693

Parametric Analysis of Cloud Data Partitioning Techniques: Review Paper

Kiranjit Kaur[1*], Vijay Laxmi[2]

[1]University College of Computer Applications, Guru Kashi University, Talwandi Sabo, Punjab, India
[2]University College of Computer Applications, Guru Kashi University, Talwandi Sabo, Punjab, India

[*]*Corresponding Author: gill_kiran2004@yahoo.com, Tel.: 9592200574*

Available online at: www.ijcseonline.org

Accepted: 28/Sept./2018, Published: 30/Sept./2018

***Abstract*—** Technology makes life easier but at the same time generating bundles of data which is difficult to manage in traditional data stores. To manage this huge data, new data stores called NoSQL came into existence, they resolve the problem of data management by using partitioning. This paper discusses different partitioning techniques named horizontal, Vertical and Workload Driven Partitioning. Focus of this paper is to compare these partitioning techniques on the bases of important parameters named communication cost, complexity of search, quality and scalability. It provides the result on the basis of analysis which helps to choose the relevant partitioning technique.

***Keywords*—**Horizontal partitioning, Vertical partitioning, Workload Driven partitioning, Communication cost, Complexity of search, Quality, Scalibility.

I. INTRODUCTION

As the technology is reaching at its peak, we are generating bundles of data which is mixture of structured and unstructured data. Cloud computing provides us the service of storing this huge data which we can access 24 hours * 7 days with the power of internet. Use of cloud technology not only reduce the cost but also improves security [9].

In this review paper, the focus is on analyzing different partitioning techniques used for managing this huge data. As we all know data is stored in the different databases like Access, SQL, FoxPro and Oracle. In early days these traditional databases are used by researchers. As the technology grows bundles of data generated from different social sites like Facebook, Twitter, LinkedIn and others. It was difficult for traditional database systems to manage this data so new data store named NoSQL cloud data stores was developed.

To ensure better performance, these stores use partitioning techniques to store data. Partitioning of data increases its scalability, efficiency and availability. There were different partitioning techniques available [1]:
1.1 Horizontal Partitioning
1.2 Vertical Partitioning
1.3 Workload Driven Partitioning

1.1 Horizontal Partitioning

It is a technique in which data is partitioned row wise on the bases of some specific criteria and each row contain data of all the columns. These partitions are stored on different machines. This technique do the static partitioning means the partitions are fixed and they do not change.
Criteria used for this partitioning:

1.1.1 On the bases of range of keys: Key value must be chosen in such a way that there is no overlapping and no gap between them.

1.1.2 Hash function: Hash function is applied to the primary key. The number of partitions made depends upon the value of data after applying the hash function.

1.1.3 Schema partition: In this partition similar rows are kept on the same partition. Similarity between the rows can be calculated using cosine or jacquard similarity method.

1.2 Vertical partitioning

In this technique data is partitioned on the basis of columns where set of columns are stored on different partitions. This technique is more useful when the input data is column family cloud data. Column family is a data in which columns may or may not contain sub columns. In vertical we store different column families to different partitions with primary key included in all partitions.

1.3 Workload Driven Partitioning

This is the most useful technique in these. It takes data generated from web applications, analyze this data and do partitions according to the data access patterns. These

partitions are reformed as the data access pattern changes. It improves the scalability of data.

In this paper Section I contain the introduction of different partitioning techniques. Section II describes the related work of different partitioning techniques. Section III describes different parameters related to partitioning. Section IV is the graphical analysis of these parameters with respect to different partitioning techniques. Section V concludes research work with future directions.

II. LITERATURE REVIEW

Researchers have implemented various data stores and partition techniques to upgrade the scalability of transactions for web applications.

2.1 Horizontal partitioning: - Most of the NoSQL and NewSQL data stores implement some sort of horizontal partitioning [2], which stores set of rows/records into different partitions which may be located on different machines. The most common horizontal-partitioning strategies are range partitioning [2] and consistent hashing [2] and Schema Partitioning.

2.1.1 Range Partitioning: Range partitioning do partition based on range of keys. These partitions are stored on different server where each server is responsible for the storage and read/write handling of a specific range of keys. The advantage of this approach is the effective processing of requested queries. This approach has the problem of load-balancing. Cassandra [5], HBase, BerkeleyDB and MongoDB cloud data stores implement range partitioning.

2.1.2 Hashing Partitioning: In consistent hashing, the dataset is represented as a ring. The ring is divided into a number of ranges equal to the number of available nodes, and each node is mapped to a point on the ring. DynamoDB [6], CouchDB, VoltDB, and Clustrix cloud data stores implement consistent hashing.

2.1.3 Schema level partitioning: The Schema Level partitioning scheme [1] is a static partitioning scheme which is designed to improve the performance of ElasTras [7]. It is derived from the TPC-C schema, so it is called as Schema Level partitioning. In the schema level [4], similar rows of tables are located on a single partition which decreases distributed transactions.

2.2 Vertical Partitioning: - Vertical partitioning [2] store sets of columns into different segments and distributing them accordingly on different servers. For example, vertical partitioning segments contain predefined groups of columns; therefore, data stores from the column-family category can provide vertical partitioning in addition to horizontal partitioning. This partition improves privacy and security of data [3].

2.3 Scalable Workload-Driven Partitioning: Scalable workload-driven partitioning [1] is not static or dynamic partitioning scheme. It lies between static and dynamic partitioning scheme. In this partitioning, the transaction logs and the data access patterns [8] are analyzed. This analysis is performed periodically and the partitions are formed based on data access patterns. Once the partitions are formed, they may change in future, based on data access patterns. The advantage of using this partitioning scheme is partitions are formed after performing an analysis. Therefore the least number of distributed transactions occur and reorganization of application data is not frequent. Thus the cost is also minimized.

III. PARAMETRIC ANALYSIS

In this part important parameters are analyzed on different types of partitioning techniques. Communication cost, Complexity of search [10], Quality and scalability [9] are the major parameters on the basis of which analysis of different partitioning is performed.

3.1 Communication cost: This cost is measured in terms of time. It is measured by the time taken from user request to reach the desired partitions which contain the requested data.

3.2 Complexity of search: It is measured by the time taken for the selection of true data from the partition. Search complexity increases as the search time increases. Complexity of search is measured in terms of low/moderate/high.

3.3 Qualitative: This parameter measure the quality of the data organized in the partition. It is measured with the help of precision and recall. High value of precision and recall indicates healthy arrangement of data.

3.4 Scalability: it is measured in terms of throughput and response time. A partitioning is said to be highly scalable if it optimized both the factors.

IV. GRAPHICAL ANALYSIS

This paper provides graphical analysis of these parameters which is performed using values low/moderate/high.

Table 1: Values of parameters in different partitions

Parameters/ Partitions	Horizontal	Vertical	Workload Driven
Communication Cost	3	2	1
Complexity of Search	2	2	1
Qualitative	1	2	3
Scalability	1	2	3

In the table 1 the number represent as 1: low, 2: moderate and 3: high value.

4.1 Communication Cost

Communication cost is low for workload driven partitioning because in this partitions are reformed as the workload changes. Its moderate in case of vertical partitioning and high in case of horizontal partitioning because in these types partitions are fixed, once formed do not change. Low value of partition is considered as the best value in communication cost.

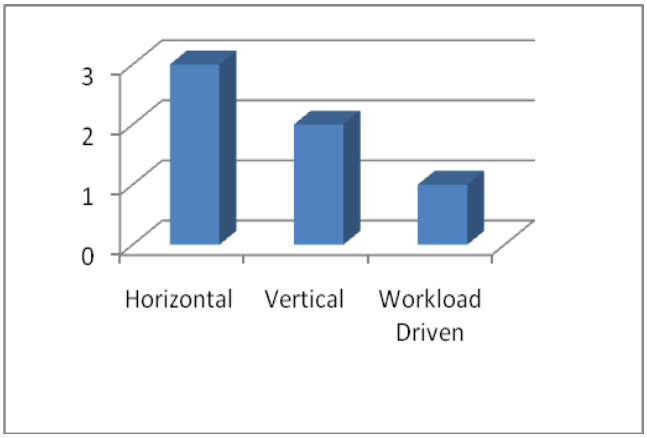

Fig 1: Communication cost for partitions

4.2 Complexity of Search

Complexity of search is low for workload driven partition and moderate for both horizontal and vertical partitioning. Partition which provides low value of complexity of search is considered as the best one.

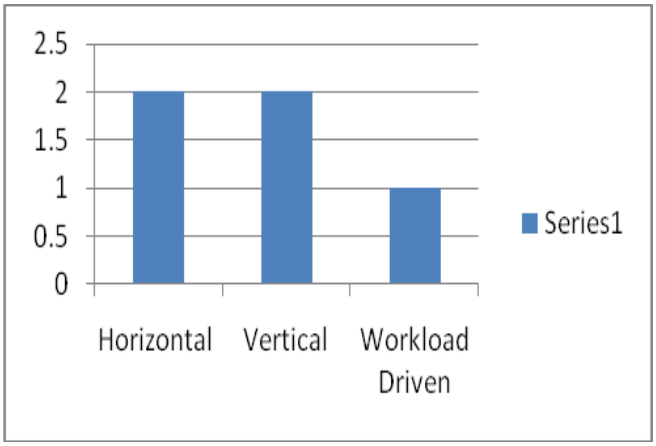

Fig 2: Complexity of search for partitions

4.3 Quality

Qualitative parameter high value indicates the healthy arrangement of data in a partition. Workload driven partition store data on the bases of data access patterns which ensures that related stored together and when the data pattern changes, partitions are reformed accordingly. Its value is low and moderate in horizontal and vertical partition respectively.

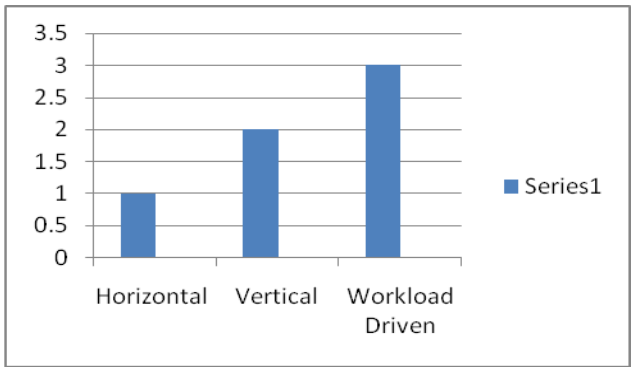

Fig 3: Quality Measure for partitions

Scalability is measured in terms of time and throughput. As per analysis workload driven approach provides highly scalable transactions as compared to horizontal and vertical partitions respectively.

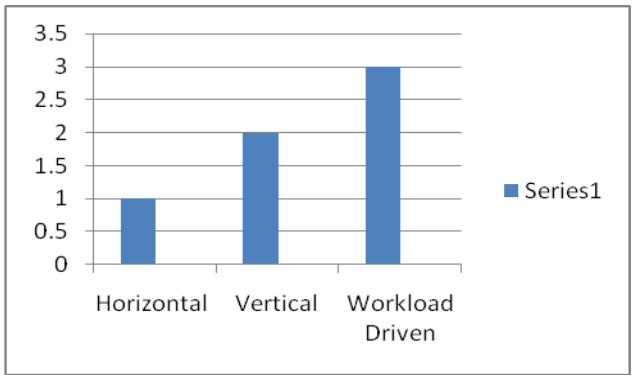

Fig 4: Scalability for partitions

V. CONCLUSION AND FUTURE SCOPE

In this paper, we focus on different partitioning techniques used for storing cloud data and analysis important parameters for checking the performance of partitioning techniques. On the basis of analysis, we conclude that workload driven partition is the best technique for partitioning cloud data. Currently it's implemented using Genetic algorithm, in future it's having lot of scope by combining machine learning with it.

REFERENCES

[1]. S. Ahirrao, R. Ingle, "Scalable transactions in cloud data stores", Journal of Cloud Computing: a Springer Open journal, 2015.

[2]. K. Grolinger et al, "Data Management in cloud environments: NoSQL and NewSQL data stores", Journal of Cloud Computing: a Springer Open journal, 2013.

[3]. K. Jens et al, "On the performance of Query Rewriting in Vertically Distributed Cloud Databases", Springer: Innovative Approaches and Solutions in Advanced Intelligent Systems, Vol. 648, pp. 59-73, 2016.

[4]. D. Agarwal et al, "Database Scalability, Elasticity and Autonomy in the Clouds", Springer: Database Systems for Advanced Applications, Vol 6587, pp 2-15, 2012.

[5]. A. Lakshman, P. Malik, "Cassandra: A decentralized structured storage system", ACM SIGOPS Operating System Review, Vol. 44, Issue 2, pp. 35-40, 2010.
[6]. G. Decandia et al, "Dynamo: Amazon's highly available key value store", in the proceedings of the 21st ACM Symposium on Operating System Principles, ACM, New York, pp 205-220, 2007.
[7]. S. Das et al, "Elastrans: An elastic transactional data store in the cloud", in the proceedings of the 1st USENIX workshop on hot topics on cloud computing, USENIX Association, Berkeley, CA, pp 1-5, 2013.
[8]. W. Vogels, "Data access patterns in the amazon.com technology platform", in the proceedings of the 33rd International conference on Very Large Data Bases, VLDB Endowment, 2007.
[9]. K. Kaur, V. Laxmi, "Partitioning techniques in Cloud Data Storage: Review paper", International journal of advanced research in computer science, Vol. 8, No. 5, May-June 2017.
[10]. Vanderlei et al, "A cooperative classification mechanism for search and reterival software components", in the proceedings of the 2017 ACM symposium on applied computing, pp 866-871, 2007.

Authors Profile

kiranjit kaur pursed Bachelor of Computer Applications from Panjab University, Chandigarh in 2001, Master of Science (IT) from Panjab University , Chandigarh in year 2003 and done M.Phil(CS) from Alagappa University , Tamilnaidu in 2008. She is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computer Applications , Guru Nanak College Moga affiliated to Panjab University Chandigarh. She has published more than 10 research papers in reputed international journals and conferences. Her main research work focuses on Big Data Storage, He has 13 years of teaching experience and 3 years of Research Experience.

Vijay Laxmi has done Ph.D. in computers and currently working as a Dean and Professor in Department of Computer Applications , Guru Kashi University of Talwandi Sabo. She acts as research guide for 2 Ph.D and 25 M.Phil/M.Tech students. She has published more than 62 research papers in reputed international journals and conferences. Her main research work focuses on Cloud Computing, Clustering, classification, Data Mining and knowledge discovery.

