ISSN: 2321-7782 (Online)
Volume 2, Issue 12, December 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Database load balancing based on workload replicated partitioning in social networks

Vishwas Punjaji Badhe[1]
Student of ME (Computer),
G S Moze College of Engineering,
Savitribai Phule Pune University
Pune, Maharashtra(India)

Pradnya Velhal[2]
Assistant Professor,
G S Moze College of Engineering,
Savitribai Phule Pune University
Pune, Maharashtra(India)

Abstract: ***The Major issue of Database Load balancing is commonly observed in Social Networking Applications. The proposed system will Empower the solution for Database load balance issue by optimal workload Partitioning and effective retrieval of distributed data. The Proposed system will dynamically analyse the Multi-way relations in Multi-user operational system as well as this algorithm will effectively implement unique partitioning and selective replication method. The synchronization policy for recollection of data will be one of the additional features of algorithm. The Algorithm will be implemented on central server and accordingly central server can use the storage space of the other servers of social network system. The central server will maintain the database logs for different transactions. Ultimately the proposed system will minimizes total response time of social network system and also provides one convenient & secure way for the user storage.***

Keywords: Database Load balancing, optimal workload Partitioning, Multi-way relations in data, selective replication, synchronization policy

I. INTRODUCTION

In today's many online applications that have very elemental importance to database because whole application runs on data stored or real time data. Online social network is also a kind of application which have been increasing database availability requirement. Some database systems like Not only SQL [1] implements data partitioning and replication to achieve

Scalability and availability. This database ignores the relations among the data and leads to redundant replications and significant communication overheads during query processing, which degrades the performance [2], [3], [4], [5]. To overcome the problem of performance degradation number of approaches based on modeling the social network structure and user interactions have been proposed. These approaches emphasize on interactions between social network users via two-way relations [4] [6][7]. Whereas some operations like multicasting the feeds to all connected users are more expensive than operations that only involve bilateral interactions.

In this Survey, we claim that hyper-graphs are more suitable for modeling these multi-user operations and multi-way interactions. We can also say that performing partitioning and replication in a single phase enables more accurate cost prediction and better load balancing. In this approach modeling the span (number of activated servers) is highlighted.

we focuses on close-to-optimal query scheduling algorithm while performing replicated partitioning, and hence observe true balancing and cost estimations at the end of our replicated partitioning scheme.

In analysis, we observe that –

a) server load imbalance
b) the total number of I/O operations (read and write operations)
c) the number of servers processing a query (query span), have direct correlation with the performance of the system.

Hence, we focus on these metrics for possible improvements in query performance.

After selective partitioning and replication method for data distribution in social networks by utilizing the workload and time information. This approach values the time of interactions between users and predicts the interactions that are likely to occur in the future. We can say that simultaneous partitioning and replication (replicated partitioning) of this hyper-graph model can accurately target the objective of reducing the span of multi-user queries, subject to load balance and replication constraints.

After doing replicated partitioning, we decode the obtained result as a data-to-server mapping in hyper-graph model. This will reduces the average query span and amount of increase in I/O load due to replications by respecting to a user-provided threshold on the replication amount and by performing selective replication.

This proposed algorithm will be implemented on central server, so that it can use storage space of sub-servers for replicated partitioning and efficient retrieval of data. This central server also facilitates for events logging, by which one can see the different partitioned blocks of data going to different servers and at the time of retrieving that data how synchronization is done actually.

II. Related Work

Many of NoSQL systems use either hash-based or range-based or combination of both partitioning schemes. range-based partitioning divides key-space into ranges and each range is assigned to a server and if required this key space replicated to other servers. So this become an advantage of range-partitioning by which two consecutive keys are likely to come in the same partition, which is beneficial when range scan type queries are frequent. It generally maintains a map that stores information about which servers are responsible for which key ranges.

Hash-based partitioning simply uses the hash of data to determine the responsible server for storing that data. Consistent hash rings are a blend of range- and hash-based partitioning schemes. Many of NoSQL systems such as Cassandra [8], Dynamo [9], Voldemort [10], and Riak [11] uses either hash-based or range-based scheme.

Servers in Cassandra cluster are arranged in ring format and data stored is according to ring analogy. Individual ring is divided into several ranges and each server is responsible for one or more ranges. As new server joins Cassandra, it is assigned a new token, which determines its position on the ring and the range of data, to which this server is responsible for.

Cassandra uses Random Partitioning, which uses the MD5 hash of each row key to partition, and Ordered

Partitioning, that uses row key for a sorter order in various servers. In database partitioning data items are represented via nodes, transactions are modeled via edges. The partitioning objective is to minimize the number of distributed transactions. The major key areas of proposed social network partitioning schemes are graph-partitioning, modular-optimization and random partitioning.

There are different parameters to measured Partition qualities such as the number of internal messages or dialogs. Many times real time data is taken for test. For small partition counts, graph-based approaches are shown to perform superior, whereas for large partition counts, modular optimization algorithms perform slightly better. Replication in this scheme enforces much replication and can lead to high I/O loads [3].

III. Working Scheme

Our proposed algorithm uses social network and interactions between users. This is used to predict the user actions that are likely to occur in the near future. Using these predictions, we perform a selective replicated partition of the user data. we propose a replicated partitioning for constructed hyper-graph and interpret the partition result as a solution to the Selective Replicated Partitioning for Minimized Server I/O Problem.

We maintain log of queries and the log contains information on the timing of the events. Algorithm divide the Events into time periods and utilizing the activities in the previous periods, so the objective is identify the pattern and frequency of the activities that are likely to occur in the next period. We then partition and replicate data according to this prediction. The scheme refers more recent time activities than older one to map the predicted events and queries.

The proposed scheme contains activity hyper-graph for models the read and write requests. The hyper-graph successfully distinguishes the read and write requests performed by the same user in different time by placing separator symbol for such requests. Basically proposed scheme determines the pattern of read write requests to balance the load in distributed environment. When these chunks are partitioned and replicate at different servers in distributed environment, central log maintains the concurrent entries of transactions happening in database system.

While retrieving the necessary data on read or write request, synchronization policy take it back by their original sequence. Above mentioned read/write operations take long time to execute so minimizing the total response time, balancing data load to different distributed servers, implement unique partitioning and selective replication method effectively, are some key objectives of proposed system.

IV. Conclusion

Database load balancing using replicated workload partitioning uses hyper-graph model for partitioning can be used for data partitioning and replication in social networks. The proposed model basically focuses on multi-way interactions between users social network operations. this model can overcome the problem of latency, throughput, and scalability by achieve improvements over hash- and graph-partitioning-based approaches The basic overhead in Hash-based approaches is communication but it can work well in distributes workload and parallelism.

Hyper-graph partitioning approach can deal with maximum I/O in distributed environment. Our approach performs partitioning and replication simultaneously to reduce the number of servers processing queries while respecting load balancing and I/O load constraints under replication.

Acknowledgment

This paper work is completed successfully only because support from each and every one including family, teachers and friends. Especially, I am very thankful to those who provide me guidance and make this work complete. My acknowledgment of gratitude towards the everyone who make this work reachable.

References

1. R. Hecht and S. Jablonski, “Nosql evaluation: A use case oriented survey,” in Cloud and Service Computing (CSC), 2011 International Conference on, pp. 336 –341, dec. 2011.
2. J. M. Pujol, V. Erramilli, and P. Rodriguez, “Divide and conquer: Partitioning online social networks,” arXiv, vol. cs.NI, Jan 2009.
3. J. M. Pujol, G. Siganos, V. Erramilli, and P. Rodriguez, “Scaling online social networks without pains,” Proc of NETDB, 2009.
4. M. Yuan, D. Stein, B. Carrasco, J. M. F. da Trindade, and Y. Lu, “Partitioning social networks for fast retrieval of time-dependent queries,” in ICDE Workshops, pp. 205–212, 2012.
5. S. Kadambi, J. Chen, B. F. Cooper, D. Lomax, R. Ramakrishnan, A. Silberstein, E. Tam, and H. Garcia-Molina, “Where in the world is my data?,” PVLDB, vol. 4, no. 11, pp. 1040–1050, 2011.
6. J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez, “The little engine (s) that could: scaling online social networks,” ACM SIGCOMM Computer Communication Review, vol. 40, no. 4, pp. 375–386, 2010.
7. C. Curino, E. Jones, Y. Zhang, and S. Madden, “Schism: a workload-driven approach to database replication and partitioning,” Proc. VLDB Endow., vol. 3, pp. 48–57, Sept. 2010.
8. A. Lakshman and P. Malik, “Cassandra: a decentralized structured storage system,” SIGOPS Oper. Syst. Rev., vol. 44, pp. 35–40 Apr. 2010.
9. G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: Amazons highly available key-value store.,” in Proc. 21st ACM SIGOPS, pp. 205–220, 2007.
10. “Project voldemort.” http://project-voldemort.com, 2012.
11. “Riak.” http://basho.com/products/riak-overview/, 2012.

AUTHOR(S) PROFILE

Mr. Vishwas Punjaji Badhe, Received his BE degree (2012) from Department of Computer Engineering of Savitribai Phule Pune University (Maharashtra, India), he is now Student of ME (Computer) in G. S. Moze College of Engineering of Savitribai Phule Pune University. He is also working as a lecturer (Since 2013) in Y. B. patil Polytechnic of MSBTE Board, Mumbai, (Maharashtra, India).

Prof. Pradnya Velhal, Received his BE degree (2005) from Govt. college of Engineering, Karad (Maharashtra, India), she has also completed MTech. (Computer) from College of Engineering, Bharati Vidyapith. Now she is working as a Assistant Professor in G. S. Moze College of Engineering of Savitribai Phule Pune University (Maharashtra, India).