Enabling Database As A Service

Relational Cloud

Mavez Singh Dabas

***Database research group, VIT University, Tamil Nadu, India**

***Abstract*—The aim of this research is to shed lights on the concept of cloud computing and hence incorporating a new design of the latest field of "database as a service" (DBAAS) called Relational Cloud. A DBAAS ensures to provide and move much of the operational overhead of issuing scaling, performance, issues, backup, privacy, etc from the actual database users to the service operators, offering lower overhead and implementing cost to the user. Few efforts could be observed in the form of Amazon RDS and Microsoft SQL Azure, but are stricken by the very basic of these three difficulties: Efficient Multitenancy, Elastic scalability and Database privacy. The prominent features of the Relational cloud includes: (1) A work oriented and overloading or normal approach to multitenancy and itslocation on the cloud with respect to the server and its accessing and security. (2) Using a different data structure i.e. graph oriented partitioning of data system to achieve near-linear elastic scale out for very complex processing happening. (3) A security scheme or protocol that is self-generating and self-optimizing in nature that enables SQL queries to run over few set of encryption algorithms and data is encrypted and (4) most importantly monitoring the pattern of the query andhow data is being accessed and providing a self-optimizing system.**

I.INTRODUCTION

This concept of Database as a service is attractive for few reasons (i) due to economies of scale, the hardware issues and the cost of energy incurred by the user end are more towards the less or comparatively lower, (ii) the cost offered by this service is more likely to be proportional or at par with the usage of the service by the user and hence giving a better and more sophisticated environment for using the useful service obtained, (iii) by centralization and providing automatic many of the database management tasks, the new and old operational costs are reduced well.

Multi-tenancy and its use in the are the major concerns for the firms which are not in the track to use cloud. These compilation of security adds up to a greater level when different users share the same physical location and also the instance of the database is also shared commonly between them. The search for a database where there is single instance offered to multiple users. Another overhead occurs when for the provider of service by giving instance per user is ineffective and inefficient use of the available resources. Much needed because the end user initially has no pre-decided idea of data's size, growth rate or processing capabilities it requires.

Elastic Scalability tends to focus on the fact that database as a service must provide and must be able to handle and accommodate the overhead of the databases of different and variable sizes, processing speeds and nature. The main reasonable problem surfaces when the workload exceeds the handling ability and capacity of the single working machine, this leads to the overburdened, more traffic affected, redundant nature and slow responsiveness.

Privacy is also a tremendous job needed to be incorporated as the instances of database is shared between multiple users and hence could lead to mixing or voiding data integrity. Since in the cloud providing database as a service is very much driven by the concepts of lack of security and reduced level of reliability on the multiple users using the technology. In general terms, frontend nodes and backend nodes are being monitored by the administrator nodes, so the involvement of client nodes is incorporated through adding two sided security layer which monitors every transaction based on the particular service being provided to a particular user by administrator and a self-optimizing and self-generating security protocol.

II. DEVELOPMENT AND ARCHITECTURE

Since the whole idea is to provide the database as a service over the cloud, so the characteristics of the cloud needed to be handled are elasticity and flexibility in the implementation. Now there are database systems that are to some level flexible and elastic but through very strict configuration and are not dynamically free. If the desired design of the service and database could be handled as per the norms of cloud, the "pay for the service" idea will be well incorporated and will be leading to an established multi-tier end system. As per the current design database as a service uses the pre-existing database management system as the fundamental query to process the backend queries and storage nodes. The issue arises here is the overload of the which can be well handled by the implementing the concept of dynamic changing in terms of increasing overhead. Now the key part is to avoid the mixing of different sets of users to mix up their data into a common physical location or database. Another aspect is to enable a cryptographically embedded layer of data security and applying it on front end in order to provide data security and data privacy. All the communications and transactions happening with the relational cloud must be monitored with this layer and a security check is incorporated to check for credibility of the user and the authentication of the transactions. Data that is stored in the cloud must be resistant towards the network failure and system failure especially physical level failures. A common used solution is to maintain the multiple copies of the data and store them at multiple locations that are accessed by a common server enabled with embedded layer of encryption.

III. PRE-EXISTING STATUS AND TRANSACTION MANAGEMENT

Few components of the relational cloud has been developed and in all this development process modularity is the key to use a component at several places irrespective of their development environment. Few attempts have been observed in recent past to integrate them as a single unit before offering them as a service to public or private cloud. Now the main area to deal with is elastic and scalable transactions and keeping the functionality consistent. Scalability is necessary to involve because it will work even when the multiple updating of data or transactions is taking place so a variable level of consistency is very much needed this level of consistency should be reflected upon all the partitions and the copies maintained at different locations.

IV. DATABASE PARTITIONING

Database partitioning is needed in order to extend a database to multiple stations and to ensure and initiate work load balancing on all the nodes in place of placing the entire databases. The current technology incorporates the partitioning technique OLTP and web workloads but another alternative can be used as OLAP. It tends to provide multi-dimensional queries. OLAP was conceived for Management Information Systems and Decision Support Systems but is still widely underused.With the constant growth of data analysis and business intelligence applications (now even in small business) understanding OLAP nuances and benefits is a must if you want provide valid and useful analytics to management. After the partitioning is over the derivation of the decision tree is done which in turn will generalize the portioning done with the implementation of the predicates. If the solution with respect to the derived predicates is not found the system again uses the hash table generated look up tables to monitor the particular partitioning schema. This way is very much efficient and towards more independent from the layout and design of the conceptual schema and key constraints leading to discovery of inter related and suppressed relations among the data and hence providing an efficient way to partition the databases having more than one many to many relationships like social networking scenarios. The difficulty arises is due to the presence of graph data structure that leads to the population of nodes from N nodes to N^2 edges for a database with N- tuple definition.

Since graphs are efficient and ordered data structure but we need a structure with self-replication and self-sustaining environment with very less runtime and non-redundant in nature. So other alternative could be used is partitioning based on hash table concept. These can provide the unlimited and with proper algorithmic implementation is well secure and monitored and leading to less access time.

V. CONCLUSION

According to this research the need to meet the three difficulties of multi-tenancy, elastic scalability and security issues is met by the ideas discussed along with incorporating a very much effective technology of OLAP to partition the data to bring out the hidden relations among the data and using hash table in place of graph based structure.

REFERENCES

[1] Relational Cloud : Database As A Service, Carlo Curino, Evan P.C. Jones, NirmeshMalviya, Sam Madden, Eugene Wu, HariBalkakrishanan, Raluca Ada

[2] Phoenix : A Relational storage Component for the Cloud, Davi E.M. arnaut, Rebecca Schroedar and Carmem S. Hara,

[3] Relational Database, Virtualization and the Cloud, Maximillian Ahrens, Gustavo Alonso

[4] Querying Combined Cloud Based and Relational Database, Minpeng Zhu

[5] Towards Transactional Data Management Over the Cloud, GauravKulakarni, Atlanta, USA.