Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.434 – 438

RESEARCH ARTICLE

Critical Analysis of Database Management Using NewSQL

Rakesh Kumar[1], Neha Gupta[2], Hitesh Maharwal[3], Shilpi Charu[4], Kusum Yadav[5]

[1,2,3,4,5]Department of Information Technology & JECRC, Jaipur, India

[1] rakeshkumar.it14@jecrc.ac.in; [2] it.nehagupta@gmail.com; [3] hiteshmaharwal.it@jecrc.ac.in;
[4] shilpicharu@gmail.com; [5] kusumyadav.it@jecrc.ac.in

Abstract— NewSQL is a new class of modern relational database management systems that is provide the same performance as NoSQL (Not Only SQL) systems and provide administrators with Atomicity, Consistency, Isolation, Durability, or ACID performance guarantees. NewSQL is a most important new database access language being touted as superior to SQL (Structured Query Language), but it is based on top of the cross database library LDBC. In this research paper, we are discussing about Introduction to NewSQL, its background, LDBC (Liberty DataBase Connectivity) and its working. Further discussing about New Architectures of NewSQL, how to write grammar in SQL, NewSQL 'jdb' (java-database) and NewSQL 'S2' (SQL II), working of NewSQL (NuoDB) and at last covering Characteristics of NewSQL Solutions. The aim of this paper is to show mainly importance of NewSQL as a database management and give the best solution for embedded in an appliance with both commercial as well as open-source offerings available.

Keywords— ACID; LDBC; NoSQL; NewSQL; SQL

I. Introduction

NewSQL [1], [2], [3] is a new database access language to be developed which is a replacement for SQL (Structured Query Language) but, it is not an extension or subset of SQL as well as not an object database language and also not an Object-Relation mapping tool. NewSQL is a new database access language which is easier to learn than SQL, elegant, consistent, well defined and standardized. NewSQL [2] was mainly designed to preserve SQL and addressing mainly existing issues with traditional online transaction processing (OLTP) systems, specifically, their scalability as well as performance features. NewSQL gives some best solutions which are software-only, while others may be embedded in an appliance with both commercial and also open-source offerings available.

II. Background

NewSQL [1] was first used by 451 Group analyst Matthew Aslett, research paper discussing the rise of new database systems as challengers to established vendors, in a 2011. Many enterprise systems that handle high-profile data such as financial and order processing systems also need to be able to scale but are unable to use NoSQL (Not Only SQL) solutions because they cannot give up strong transactional as well as consistency requirements. The only one options available for these organizations were to either purchase a most powerful single-node machine or develop custom middleware that distributes queries over traditional DBMS (Database management system) nodes. Both approaches are prohibitively more expensive and thus are not an option so, in this paper, Aslett discusses how NewSQL upstarts are poised to challenge the supremacy of commercial vendors, in particular Oracle.

III. LDBC

LDBC (Liberty DataBase Connectivity) [3] provides vendor-independent database access and it is a JDBC driver. With LDBC, your application will work on all major databases as well as you don't have to change any source code. LDBC is based on mainly ANSI-SQL, JDBC and it is radically simplifies porting of existing or new applications to new databases as well as it is also completely shields the application from database vendor specific code. The main components of Liberty DataBase Connectivity are a JDBC driver and a SQL grammar converter. The main features of LDBC (Liberty DataBase Connectivity) are providing a standard way to access a SQL database from Java.

IV. WORKING OF LDBC

Liberty DataBase Connectivity [3] comes with a JDBC driver and internally, it uses the regular vendor specific JDBC driver, but this is not visible to the application.

A. *Without LDBC driver*

The application works with one database only, because:

- The SQL is important and it is different for each databases
- Every JDBC driver support different way
- Changes are compulsory for each database
- No any time need to study all vendor documentation
- No any time to test with all databases accordingly

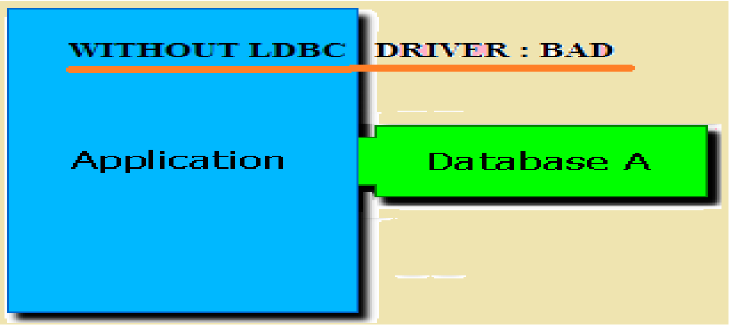

Fig. 1 without LDBC Driver

B. *With LDBC driver*

The application works with all important databases, because:

- The SQL syntax is converted to each database
- All databases behave like the same
- No any changes required to support other database
- You need to read the LDBC docs only
- LDBC supports features such as well-tested

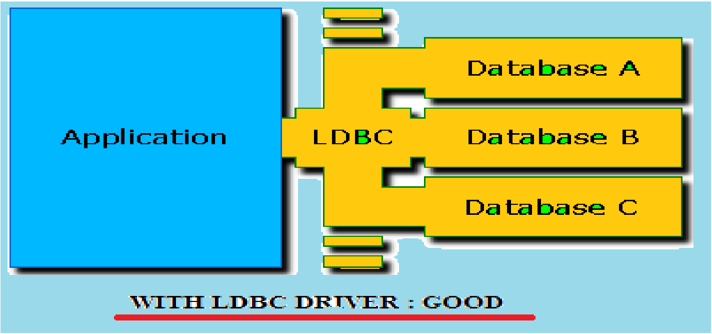

Fig. 1 With LDBC Driver

V. NEW ARCHITECTURE OF NEWSQL

A. Spanner

Spanner is a very highly scalable, globally-distributed database designed, built, and deployed at Google. Spanner provides highest level of abstraction, it is a database that shards data across many sets of Paxos state machines in data-centers spread all over the world. Spanner supports Replication which is used for global availability and geographic locality and it automatically re-shards data across machines as the amount of data or the number of servers changes, as well as it also automatically migrates data across machines to balance load and in response to failures.

B. ClustrixDB

ClustrixDB is a very important distributed peer-to-peer SQL database built for large scale, high availability and fast growing applications. ClustrixDB use mainly for run massive transaction volume and fast real-time analytics, both at the same time. ClustrixDB offers a scale-out SQL database and it is designed from ground up for the scale-out architecture of the cloud. ClustrixDB handles massive transaction volume and allows you to run real-time analytics on your live operational data without moving it into another system and also it virtually eliminates DBA operations tasks.

C. NuoDB

NuoDB [4] is a distributed database management system with support SQL implementation and ACID transactions. NuoDB provides global data management platform with guaranteed high availability, rolling upgrades, built-in data redundancy and disaster recovery. NuoDB scales elastically and it is a multi-tenant database as well as offers real-time operational intelligence also.

D. TransLattice Elastic Database

TransLattice Elastic Database (TED) is a scalable, highly available database management system, geographically distributed relational SQL database server which is built for online transaction processing (OLTP) and supports varying application loads with high-availability and exceptional performance for remote users. TransLattice Elastic Database (TED) supports some important features such as Resilience, Scalability, End user performance, Cloud migration and dramatically lower costs.

E. VoltDB

VoltDB [5] is an in-memory relational database that combines Relational, ACID-compliant SQL & the flexibility of JSON, high-velocity data ingestion, massive scalability, and real-time analytics and decisioning to enable organizations to unleash a new generation of applications that act on data at its point of maximum value. VoltDB database can be scaled in two dimensions such as scaling up and scaling out. VoltDB supports a built-in, transactional extract feature and it was designed for High Availability from the ground up.

F. SQLFire

SQLFire is an in-memory data platform delivering speed, dynamic scalability, together with the reliability as well as data management capabilities of a traditional database. Slier solves the problem of moving data into the cloud by implementing a shared-nothing architecture and it provides for continuous uptime with built in high availability as well as disaster recovery. SQLFire can receive data from any application that is able to call such as C++, C#, Java, or REST interface and it will route function calls to the relevant nodes on the cluster without interference from the caller.

G. GridGain IMDB

GridGain IMDB supports features such as local, replicated, and partitioned data sets and allows to freely cross-query between these data sets using standard SQL syntax. GridGain IMDB supports MapReduce, distributed SQL, MPP, MPI, RPC, File System, and Document API type of data processing and querying and it also provides Functional Programming (FP) APIs for most of the in-memory database operations such as cache projections, predicate-based operations, etc. GridGain IMDB concurrency is based on implementation of MultiVersion Concurrency Control. GridGain IMDB has full support for distributed transactions and supports remote client connectivity via direct TCP connectivity as well as REST protocol over HTTP.

H. Drizzle

Drizzle is a most important database optimized for Cloud infrastructure and Web applications and it is also designed for massive concurrency on modern multi-cpu architecture. Drizzle is Open source, Open community, & Open design which has client/server architecture and uses SQL as its primary command language. Drizzle is an ACID-compliant relational database that supports transactions via an MVCC design and Plugin points which support replication, storage engines, query rewrite, table functions, user-defined functions, protocol adapters, and multiple query caches.

I. MemSQL

MemSQL is a distributed, in-memory database, ACID-compliant RDBMS that most notably converts SQL into C++ through code generation. MemSQL leverages lock-free data structures, Multi-Version Concurrency Control and its query execution engine is designed for maximum speed and efficiency. MemSQL distributes multiple copies of data on separate nodes easily, both within the data center as well as across data centers, so your data is always be safe.

VI. GRAMMAR WRITING

TABLE I

SQL	**NewSQL ‘jdb’ (java-database)**	**NewSQL ‘S2’ (SQL II)**
CREATE TABLE STAFF(ID INT PRIMARY KEY, NAME VARCHER(255))	staff=new table(int id, string name, key(id))	create table staff(id int, name string, primary key(id))
INSERT INTO STAFF VALUES(1,'Hello')	staff.add(1,"Hello")	insert staff (1,'Hello')
SELECT * FROM STAFF	staff.get()	select staff

SELECT S1.ID,S2.NAME FROM
STAFF S1, STAFF S2
WHERE S1.ID=S2.ID
	s1=staff; s2=staff;
	s1.join(s2[s1.id==s2.id]).get(s1.id,s2.name)
		select s1:staff join s2:staff on s1.id==s2.id get s1.id, s2.name

UPDATE STAFF SET NAME='Hi' WHERE ID=1	staff[id==1].set(name="Hi")	update staff set id=1 where name=='Hi'
DELETE FROM STAFF WHERE ID=1	staff[id==1].delete()	delete staff where id==1
DROP TABLE STAFF	staff.drop()	drop staff

VII. WORKING OF NUODB

NuoDB [4] is a SQL (Structured Query Language) as well as ACID (Atomicity, Consistency, Isolation, Durability) compliant distributed database management system. NuoDB is a new distributed, peer-to-peer, asynchronous approach which is different from traditional shared-disk or shared-nothing architectures. NuoDB was not designed with a specific operating system (OS), network backplane as well as virtualization model in mind but it is a formal piece of software that exploits the resources it is given. NuoDB supports distributed architecture which is split into three layers such as an administrative tier, a transactional tier and a storage tier which follows a three-tiered architecture mainly.

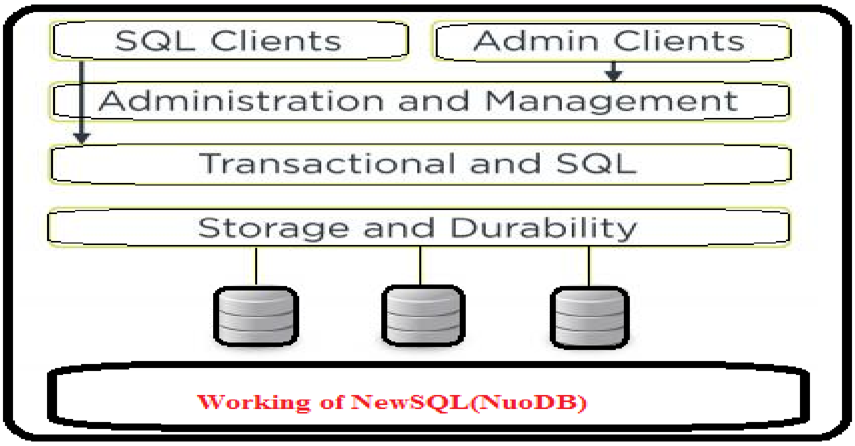

Fig. 3 Architecture of NuoDB

VIII. Characteristics of NewSQL Solutions

- NewSQL mainly provides feature of SQL which is the primary mechanism for application interaction
- NewSQL support ACID (Atomicity, Consistency, Isolation, Durability) properties for transactions
- NewSQL controls a non-locking concurrency control mechanism features which is helpful for the real-time reads will not conflict with writes
- NewSQL architecture providing more higher per-node performance than available from traditional RDBMS solutions
- NewSQL support features such as scale-out, shared-nothing architecture and capable of running on a large number of nodes without suffering bottlenecks
- NewSQL systems are approx. 50 times faster than traditional OLTP RDBMS

IX. Conclusions

NewSQL system is a new generation of information management systems, which is useful for businesses that are future planning to:

- Migrate all existing applications to adapt to new trends of data growth rapidly
- Develop new and also a powerful applications on highly scalable OLTP systems

So, NewSQL system should be considered as an alternative to NoSQL (Not Only SQL) for new all OLTP applications also. If any New OLTP system which is big and vastly used in market as I foresee, I expect we will see many NewSQL engines employing a variety of architectures in the future.

Acknowledgment

This research paper work was supported by Jaipur Engineering College and Research Centre (JECRC) Foundation, Jaipur. We are thankful to Mr. Arpit Agarwal (Director of JECRC, Jaipur) for valuable suggestions, support, motivation and encouragement. Further, we wish to thank to Dr. Prof. K. K. Agrawal (IT Department of JECRC, Jaipur) and all Faculty Member of JECRC Foundation for their time to time suggestions, proper guidelines, relevant data and technical support.

References

[1] Wikipedia URL: http://en.wikipedia.org/wiki/NewSQL
[2] Datavail URL: http://www.datavail.com/category-blog/what-is-newsql/
[3] NewSQL Source Forge URL: http://newsql.sourceforge.net/
[4] NuoDB URL: http://www.nuodb.com/
[5] VoltDB URL: http://voltdb.com/