

**Distributed DBMS Capabilities of MYSQL**

[Abstract 3](#_Toc373187103)

[Introduction 3](#_Toc373187104)

[Mysql Open Source Database Engine & Distributed Database 3](#_Toc373187105)

[Mysql Functionalities for Distributed Databases 3](#_Toc373187106)

[Benefits of Utilizing Mysql Server in Distributed Databases 9](#_Toc373187107)

[Application of Using Mysql in Distributed Database Systems 9](#_Toc373187108)

[Distributed Client Server Web Applications 9](#_Toc373187109)

[Privately Owned Client Projects 9](#_Toc373187110)

[Multiple Sites Located Data Warehouses 10](#_Toc373187111)

[Future of Mysql Server in Distributed Databases Systems 10](#_Toc373187112)

[Conclusion 10](#_Toc373187113)

[References: 10](#_Toc373187114)

[List of Abbreviations and Acronyms 10](#_Toc373187115)

# Abstract

Database Application can be divided into three categories, Stand alone applications, Client Server Database Applications and Distributed Database Applications. **This report exclusively deals with the Managing Distributed Database Applications by using MYSQL open source database storage engine.** These types of distributed applications can be managed by using or extending the existing capabilities of MYSQL. Mysql uses the phenomenon of Clustering to achieve the goal of Query Processing over Distributed Network.

# Introduction

# Mysql Open Source Database Engine & Distributed Database

Mysql Database Engine is a data storage module which is equivalent to Microsoft’s SQL server but not limited to use for particular type of operating system like Microsoft’s Operating Systems. And as the term describes it as Open Source, it’s free to download and use.

Let’s have a look at the common concepts involved in understanding of distributed database concepts. Database can be defined as collection of information on single machine or on a client server distributed network. The concept of distributed database is specifically related to database sites which are distributed over the interconnected network of systems.

Distributed Database as described by M. Tamer and Patrick Valduriez as a collection of two or more logically inter-related database systems which are spread over the connected network. This database is made available to the users located at different locations exactly as it is provided on a single server. Figure below shows the example of distributed database and it’s explanation in deep.

## Mysql Functionalities for Distributed Databases

**Mysql gives us a way to make the operations on distributed databases through NDB (Network Database) Mysql Clustering. This NDB clustering is a auto-sharding for Mysql DBMS.**  It is designed for distributed, ACID architecture. Let’s examining clustering in deep, **clustering in Mysql uses a set of the various processes that are to be carried out in single step.** **This term of carrying out the multiple changes at any particular database node through single processes is specifically named as atomic commit.** If this atomic commit succeeds, changes are reflected into the database whereas if it couldn’t, changes are reversed.

**Mysql Clustering is developed on the basis of shared nothing architecture.** Mysql clustering includes some of the advanced features like failover, recovery of nodes, data replication in synchronous commits and no-single point failure concepts.

Mysql **cluster is developed in such a way to provide maximum functionalities for distributed node architecture** with no single point failure. These multiple nodes can be efficiently distributed across machines & regions for ensuring un-interrupted availability if in case any node failure occurs. Clustering in Mysql uses a storage engine which consist set of nodes to store related data. These nodes are also known as storage nodes and anyone can access them using query language with MYSQL database. There are three types of nodes in Mysql clustering as shown in a figure below.

Application 1

Application 1

Application 1

MySQL Cluster

MySQL

Server

MySQL

Server

MySQL

Server

DB Node

DB Node

DB Node

DB Node

**Architecture of Nodes**

DB nodes as shown in the figure are the primary nodes of system. These nodes act as database storage modules. They make data available to the network if in any case, any node failure occurs. These nodes also handle the database transactions.

Mysql Server nodes as shown in figure are the MySQL servers which access the storage nodes. And the number of Mysql Clusters can be multiple. Storage nodes handle all of the transactions executed by the MySQL server it’s whose result can be seen through all of the Mysql Servers connected to network servers as shown. As soon as the operations/ transactions are executed on MySQL server, it’s results are reflected to applications through Mysql cluster.

**System Architecture for High Availability**

* All clustered nodes (storage nodes) are connected to Mysql servers. **In case of failure of any clustered node, Mysql server can navigate through the network and can easily make use of another storage node for execution of transactions.**
* **Storage nodes store the copy of data on multiple nodes. If any storage node encounters failure, then always another storage node is available for information storage.**
* Management servers are designed in such a way that, they can be started, restarted or stopped without affecting ongoing process of execution of storage nodes. Use of these management servers is to send configuration info to database & server nodes. These management servers can be stopped or killed if storage nodes are running.

Developing the system in such a way makes it reliable & highly available as it minimizes the single points of failure. Working of any single node can be stopped without affecting the complete system as a whole. So, applications can go on working even if any storage node is down. Benefit of this system is that, data is replicated among storage nodes synchronously which leads to extremely low fail-over times if any case of node failure occurs. Multiple hosts are also created by storage nodes for ensuring that, Mysql cluster is operating if there is any hardware failure. Design of nodes is done using a term shared-nothing. This shared nothing is an architecture for nodes-cluster-application data and information sharing in terms of hard drive and memory storage. Every storage node have it’s own disk storage and memory storage. The option of sharing the disk and memory is also available when there is several storage nodes are running on the same computer.

There are some of the concepts given below which needs to be understood to understand the working of Mysql Clusters

**Example Configuration**

Mysql cluster is highly configurable because of which, we can distribute it for applying to particular situation. Number of computers running on it can be changed and we can also change the number of clusters, number of storage nodes, number of applications and other things. While considering any database stored on any node**, Mysql cluster will consider the database tables as a horizontally divided partitions.** If all of the nodes in a group fails to provide the database instance, or any group of nodes fails at any time, their file systems will be cleared and their partitions / database instances are lost. **It’s all up to the cluster designers to develop the back up nodes at each step for processing.**

**Synchronous Replication of Databases**

Information in the database is replicated on multiple storage nodes which are in the same group of nodes. Then, the number of copies is decided at the start of Mysql Cluster by DBA. Number of replicas can be kept from one through four. One means no replicated instances (4 copies).

While performing transactions, data is copied synchronously from all of the connected nodes means, the each update of database record is done at all of the nodes at each transaction processing. (Spreading of transaction updates). At the commit process of any transaction, it (Transaction committing node) sends the request to all the connected nodes and at the point when all the nodes responds to be in a ready state, it will mark that transaction as a committed one which processes the application as a success of transaction.

Synchronous Replication of Database refers to the fact that, if any node fails during transaction processing or updates processing, failover time of any node and the takeover time of any node would be less than single second as all database replicas already exist. It is important for Mysql cluster to confirm whether the database is in consistent state to achieve this consistency, transaction is aborted at the point when any storage node fails at the transaction processing and the node crash information is given to application for restarting the failed transaction as an appropriate one.

**Detection of Node Failure**

Failed nodes can be detected in two ways according to Mikael Ronstrom & Lars Thalmann in their Mysql clustering research. Those are communication loss & heartbeat failure respectively. Both of them **uses a message sending protocol and it also uses a network partitioning tool for ensuring nodes failure.** Network partitioning tool makes it sure that, how many nodes are remaining there for continuing the Mysql Cluster nodes running process. There might come a condition like, there are couple of parts of the cluster which already lost the interconnectivity between them. In such a case, both of the clusters cannot be kept alive which may cause database inconsistency. It is the responsibility of Mysql cluster to ensure that applications will remain available for automatically selecting a part of cluster which will continue its execution. Let’s have a look at both of the failure methods for a while

**Communication Loss**

Mysql clustered nodes are joined through the communication protocols. Currently TCP, OES, Shared Memory are all used. In this, all the storage nodes communicate with each other and application node with storage node. If any node fails, and all other nodes are informed immediately who jointly makes a decision about the failed node as failed. Failed nodes are designed in a way to automatically restart or connect to Mysql cluster like new node without affecting the application. This is the fastest way for determining the node failure according to Mikael Ronstrom & Lars Thalmann.

**Heartbeat Failure**

Some of the failures can’t be analyzed by communication loss. Some examples like disk problems, processor exhaustion and memory problems (Common problems) because of these failures, node may work improperly without disconnecting the communication network. Heartbeat protocol method is used for such kind of node failure. In this type of protocol, all of the storage nodes in network are kept in a logical circle. Each one storage node transforms heartbeat signals for next node in circle. If at any point, storage node is unable to send more than two consecutive signals (heartbeat signals), following storage node would consider the previous one as a non-working node/ failed node.

**Failure Order Determination**

Whenever multiple nodes encounters failure at a time, 2pc commit failure protocol is used for determination which determines orders of nodes failure for ensuring their safe restart process.

According to Laura Bocchi, Lucian Wischik, the concept of two phase commit protocol (Also known as 2PC) is a type of atomic protocol which handles the processes which participate in atomic transactions that are being committed at the different sites. This number of different sites can be two or more in number. Processes are committed if the protocol is followed otherwise operation is reversed. This is used by Mysql Clusters for distributed database queries processing.

**Recoveries, Single Node, Multiple Node and System**

Node recovery protocols are used for automatically restarting the failed nodes by taking the data from existing nodes

In single node recovery, nodes will get restarted at the occurrence of failure by referring the backup node (backup will be provided by the node which belongs to the same group of nodes) where backup node will send the data in part-by-part manner to the failure restarting node. Whenever backup node provides data to it, it can immediately start the transaction processing.

In multiple nodes recovery, the order of storage nodes failure is first determined by a failure determination protocol and then, a master node gives instructions to one node at a time for node recovery likely resembling to single node recovery method.

System Recovery is used for recovering the whole system if it fails in any case

There are multiple failure scenarios for Mysql Clusters

**Failure of Server Nodes**

At the point of Mysql Server crash, restarting and reconnecting can be done to cluster. Mysql server restarted in this way can be connected to any storage node. Whenever the server is down, Servers of Mysql Cluster gives exactly same database services.

**Failure of Storage Nodes**

At the point of Storage node crash, all of the other nodes are convinced about it by using any of the above mentioned Systems for detections of failures. Either by Communication Loss or by Heartbeat failure.

Mysql Clusters also uses the sharing of information through 2PC which guarantees availability of data at multiple sites and at the same time, requires two replicas of data in order to ensure the consistency of data.

**Failure of Management Server Node**

Server nodes of mysql are not dependent on management servers while their execution is concerned. Management server can fail and can be restarted for any number of times without affecting the ongoing process of Mysql cluster.

**Failure of Connections**

Whenever the connections among storage nodes are not working, nodes get information about failure of storage nodes. At this point, connection loss is managed by the same method as of node failure. In this, 2PC (2Phase-commit) protocol is used for determining the nodes status whether they are unreachable or not working and after that, Mysql cluster will take care of reconnecting or reforming with the nodes. These connections also have fail-over system for handling communication failures. It depends on the types of protocols used in communications. TCP-IP protocol is likely to have 100 milliseconds fail-over time. One more protocol is Scalable Coherent Interface protocol for connections which has 100 microseconds as a failover time. This connection failover system efficiently hides the PCI card issues, cable issues, switch failures etc by routing the messages through alternative connections.

**Failure of Disks**

All of the storage nodes have their own storage of data. If at any point, it detects that, the node has been corrupted, it’s execution stops and after clearing the file system, node restarting operation is done by using node recovery protocol

Clustered Sharding in Mysql Database is the best known approach for achieving distributed database queries processing in distributed environment. Mysql clusteres makes the partitions of the tables (also known as shards) automatically. By which databases can be scaled at low cost. This sharding process is completely transparent and can be done at any node. Updating operations can be done at any node of the distributed architecture which will reflect the changes into connected nodes.

Query processing and Query optimization is achieved in Mysql-clustered network by the use of Query Optimizer.

# Benefits of Utilizing Mysql Server in Distributed Databases

Mysql Clusters can handle the failure of any data node without affecting any other nodes.

For Specifying the benefits of Mysql in distributed environment, let’s have a look at two different database engines for a while. SQL Server and Oracle database engines are used here for comparisons.

All of these three database engines stand at some particular point relating with each other in some way. As, Oracle runs on varieties of different known platforms. SQL server doesn’t need a third-party IDE as Toad for oracle for query processing. Mysql is not tied to any particular operating system like windows operating system even it don’t have the licensing process as it’s open source.

# Application of Using Mysql in Distributed Database Systems

MYSQL database engine can be used in the same way with which SQL Server and Oracle databases are working for distributed environments as described below

## Distributed Client Server Web Applications

This is the mainly focused application of distributed database using SQL Server. In this, database of System is kept at multiple sites. Consider the example of Google Database Server Which is distributed all over the world. Many of the database servers are in America while as some of them are distributed in Asia and Europe. Example of IBM Server located at New-York which accepts requests for updates from all over the world spread locations like server at Argentina, Belgium and Brazil.

## Privately Owned Client Projects

This type of Client Project may consist of Applications that are using the database from multiple locations and which are privately owned ones. Take an example of Health Care Management Project of All Scripts healthcare management company. This is also applicable for small scale projects which involve participation of database administrators and application developers team at a time.

## Multiple Sites Located Data Warehouses

In Real time system, Data Warehouses uses Oracle and SQL & MYSQL Server for data processing at multiple sites. Data Warehouses access data from different locations and automatic triggering of data updates is done for those through individual SQL Server or Oracle Database Engines from connected sites. Automatic triggering involves automated programming of network using Availability Groups or Linked Servers connection if SQL server is used & Clustering if Mysql Server is used or by using generic connectivity if Oracle server is used.

# Future of Mysql Server in Distributed Databases Systems

Latest version of Mysql 5.7 & latest version 7.3 of Mysql cluster features mainly on foreign keys and future version of it is expected to be consisting of Connection Thread Scalability. Mysql cluster 7.0 extended the single threaded data nodes of previously existing cluster to be able to perform simultaneous operations on 8 threads at the same time.

# Conclusion

MYSQL Clustering is high availability database engine which is built using a shared nothing architecture with standard SQL interface. Mysql clusters tolerate the failures of multiple storage nodes which configures itself again. Mysql clusters have a self healing capabilities. It also provides the transparence in distribution of data through applications partitioning which results in very simple model of programming for database developers to effectively include highest availability in application without dealing with the complex low-level of coding.

Looking at the Functionalities of Mysql Server and it’s benefits in distributed environment, it seems to be a best way for using this database engine in distributed network.

# References:

1. Principals of Distributed Database – M. Tamer Ozsu, Patrick Valduriez
2. A Process Calculus of Atomic Commit – Laura Bocchi, Lucian Wischik
3. <http://dev.mysql.com/>
4. [www.wikipedia.com](http://www.wikipedia.com)
5. <https://confluence.oceanobservatories.org>
6. https://confluence.oceanobservatories.org/download/attachments/16418744/mysql-cluster-technical-whitepaper.pdf

# List of Abbreviations and Acronyms

DDBS Distributed Database System

NDB Network Database

ACID Atomic, Consistent, Isolated and Durable (Properties of Database)