



# **MySQL High Availability Solutions**

**By Pradeep Chandru**



## About Us

- Founded by a group of enthusiastic and dedicated individuals
- Hands on experience in MySQL versions 3.x, 4.0, 4.1, 5.0, & 5.1
- Experience in working with large set ups like Sify, Yahoo, Genpact, Cricinfo, BSNL, & TCS
- Have 7 Certified MySQL DBAs and a cluster certified MySQL professional with an average of 6 years of experience and 4 DBAs with an average of 3 years of experience
- Managed MySQL database driving Platforms with database volume ranging from 10GB to 6 TB



## **Our Services**

- Design and architecture services
- Query tuning and performance optimization
- Migration from any database to MySQL
- Monitoring and remote maintenance of MySQL databases
- Design, Implementation and Maintenance of MySQL high availability solutions



# **HA solutions in MySQL**



## **Why do we need HA?**

- Avoid single point of failure
- Redundancy and failover
- Scalability
- Disaster Recovery



# MySQL Replication



## MySQL Replication

- Slave applies writes from single Master
- Raw SQL (DML) is copied to Slave
- Designed for on-demand read scalability
- Cost-effective alternative to Scale-Up
- Asynchronous replication
- Statement or row-level replication support
- No special networking components
- Fully-supported via MySQL Enterprise subscription



## **... replication continued**

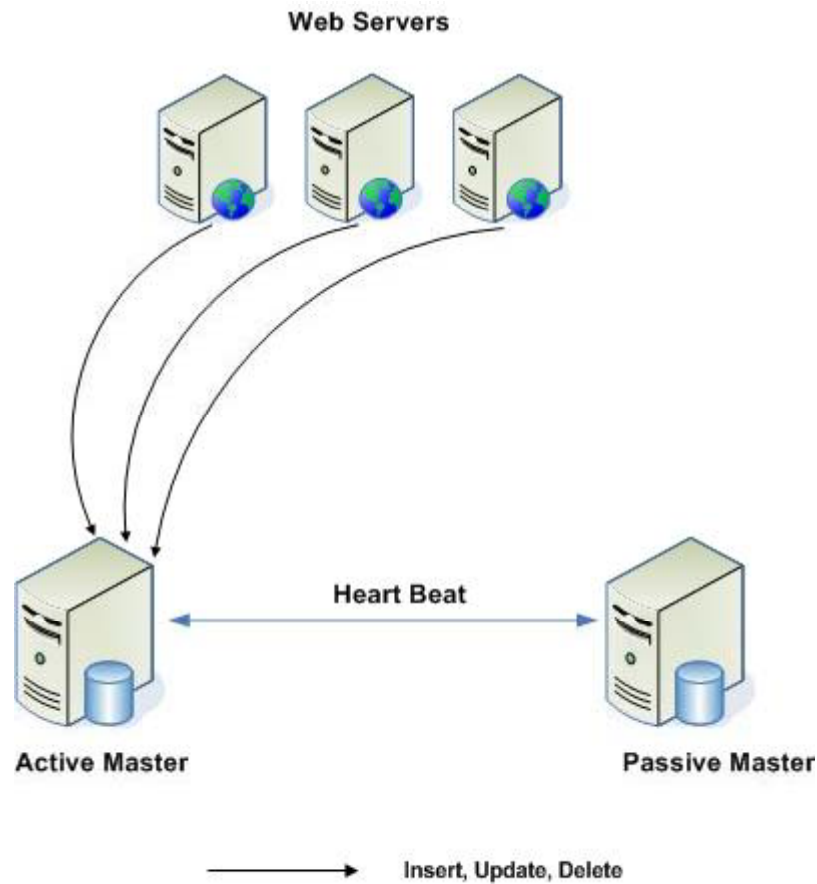
- Enterprise Monitor discovers and monitors replication topologies
- Stops if SQL errors
- Easy to manage



## **MySQL Replication**

- Master – Master Replication
- Master – Slave Replication

# Master – Master Replication

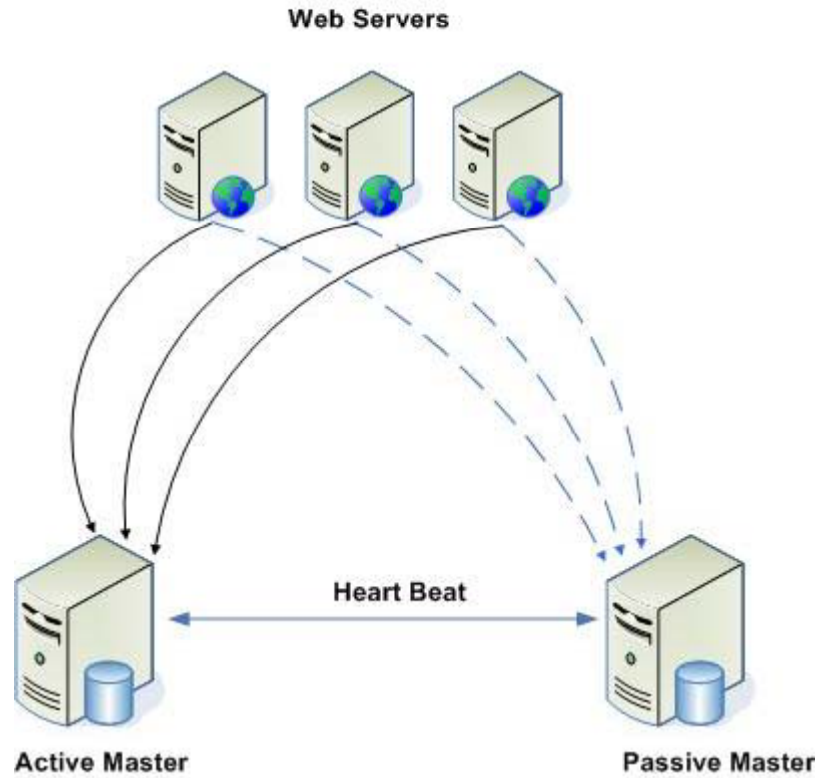




## **Advantages**

- Active Master fails, passive master will takeover
- 99.9% of High Availability
- Replication entry is skipped if server-id matches

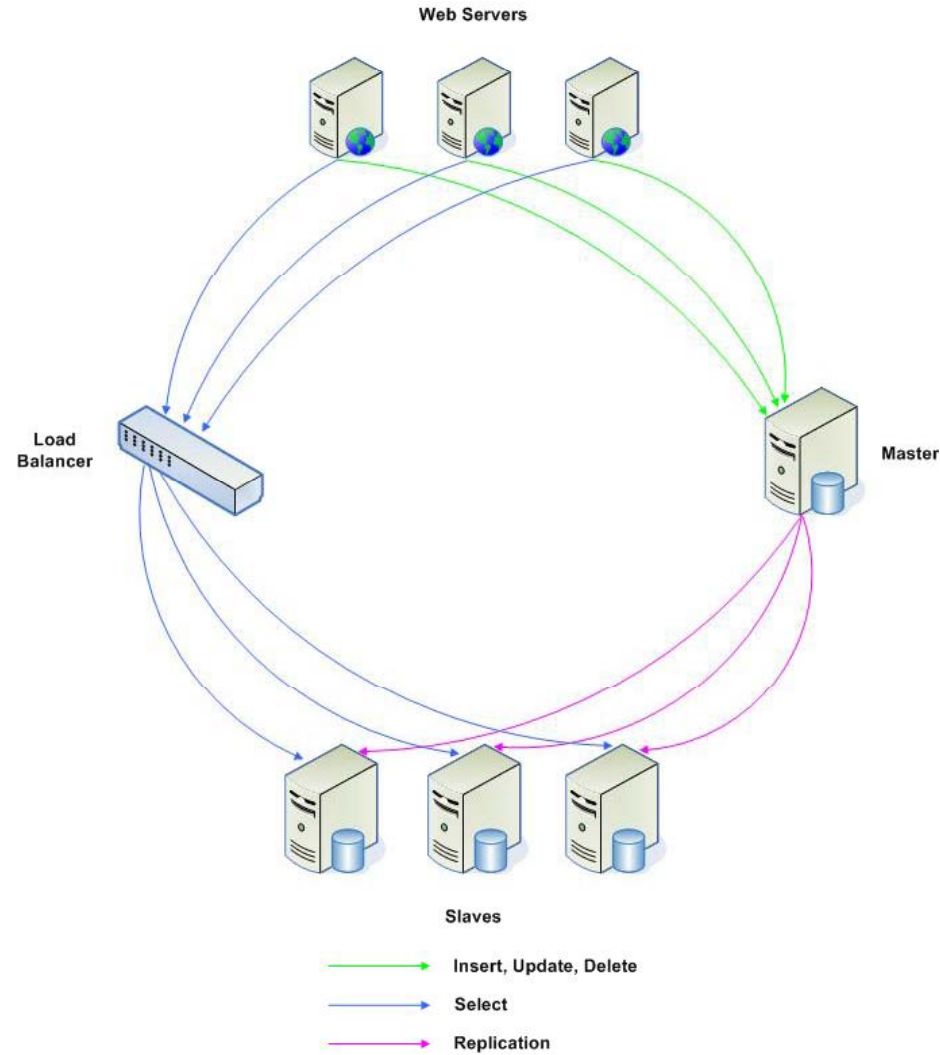
# Fail over



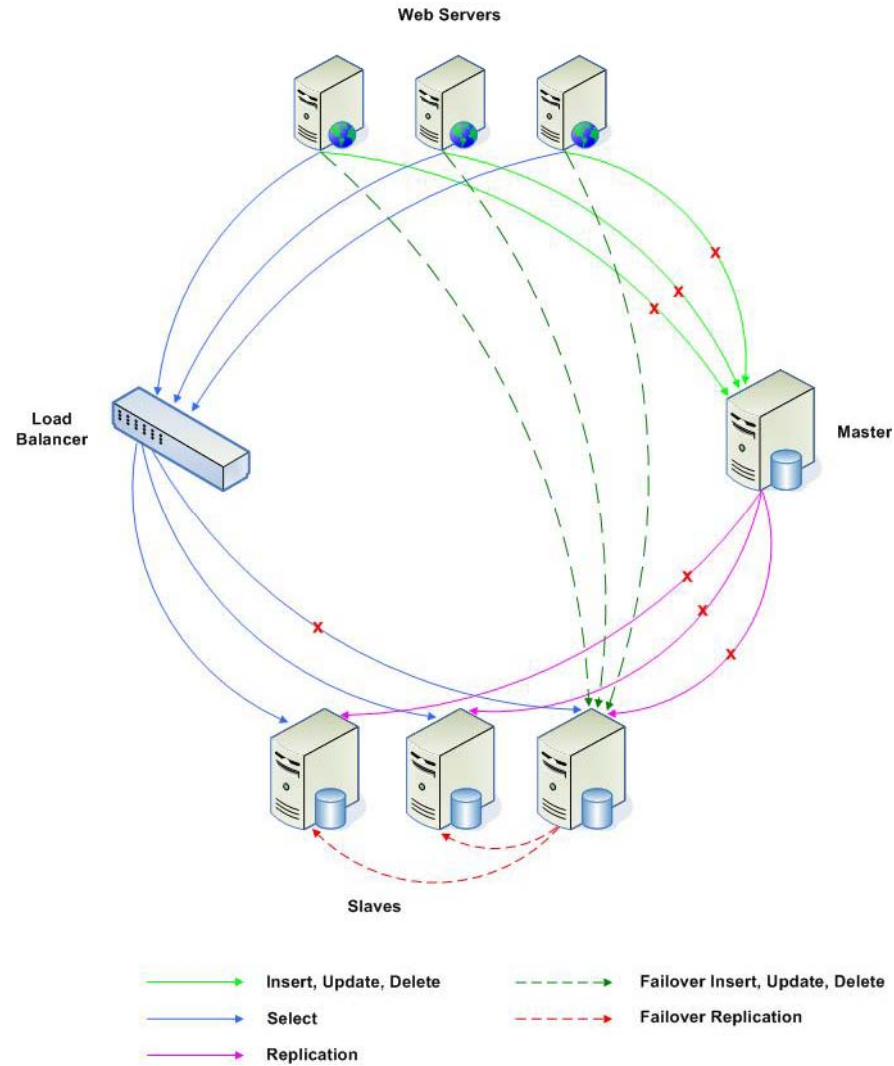
- Insert, Update, Delete
- - - - -> Failover Insert, Update, Delete



# Master – Slave Replication



# Fail over





## **Advantages**

- Write request will go to master
- Read request will go to slaves
- Active master fails, passive (slave) will takeover
- If read requests are getting increased, we can increase the slave count to scale
- 99.9% of High Availability



# MySQL Cluster



## MySQL Cluster

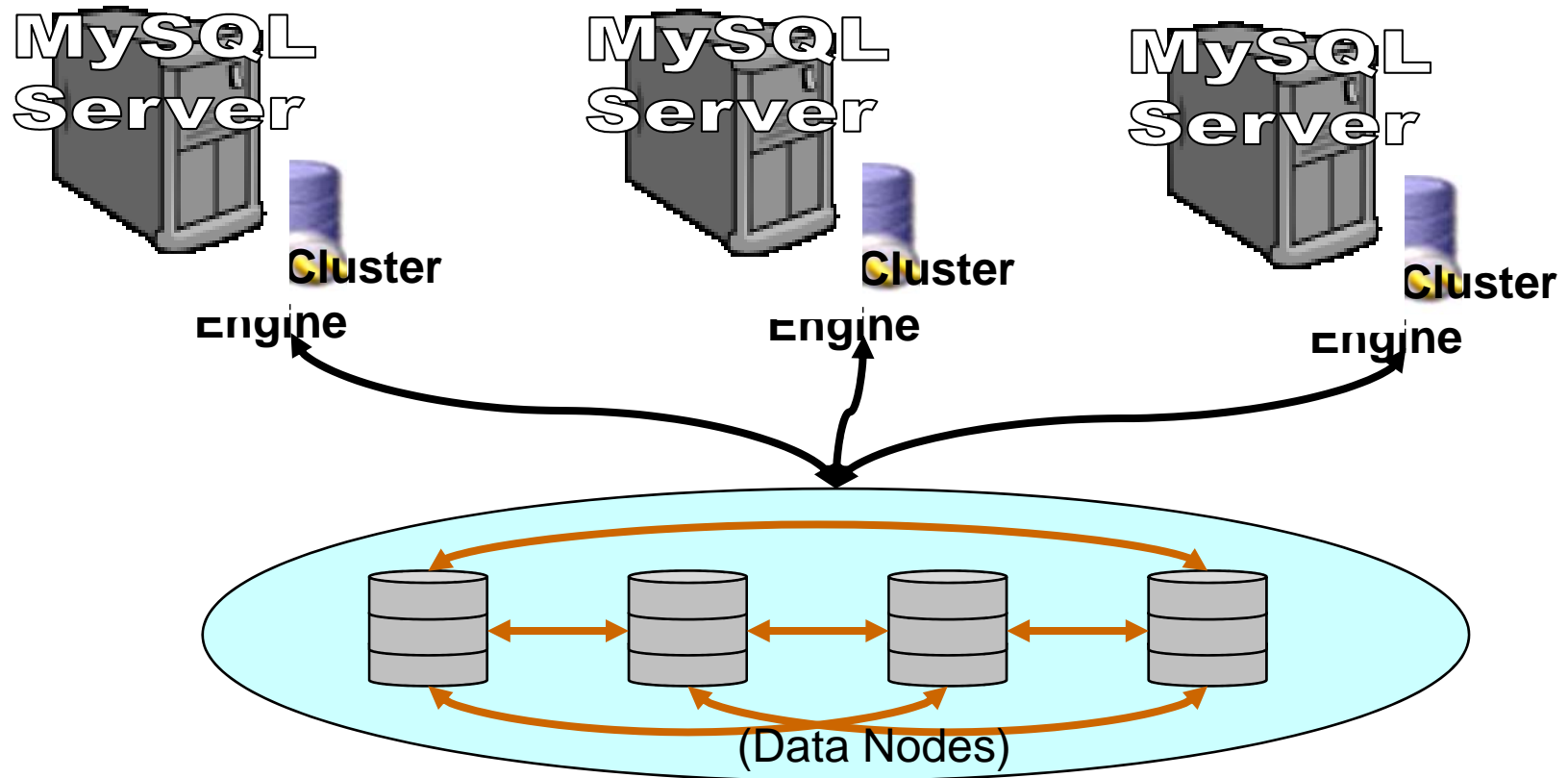
- High availability and high performance solution
- Its five 9 (99.999), 5 minutes/year downtime
- Shared nothing Architecture
- Implemented through NDB ("NDB" stands for Network Database) Storage engine
- Synchronous replication
- Fast automatic fail-over of data nodes
- Automatic partitioning
- Supports Transactions



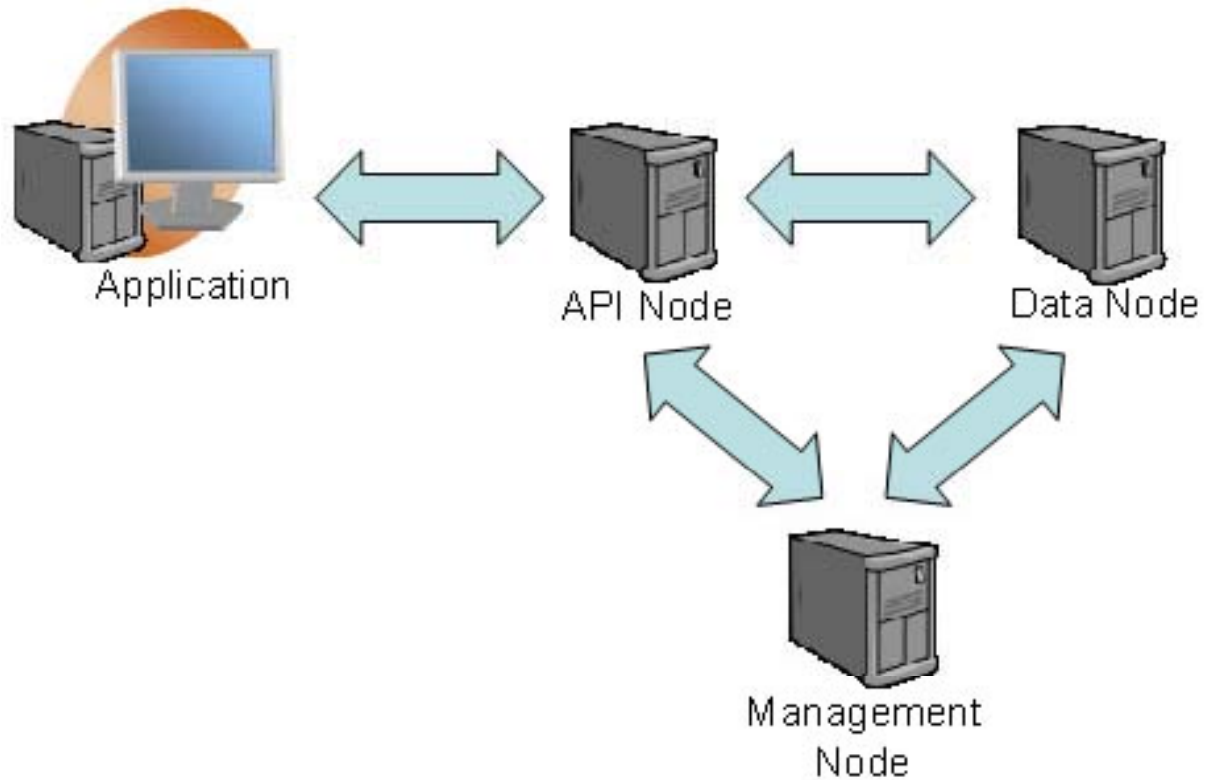
## MySQL Cluster

- In-memory indexes
- Scales reads and writes by adding 'nodes' to the cluster

# MySQL General Architecture



## Node Types



- Data Nodes
  - Storage Nodes
  - Process data
  - Ndbd daemon
  - Faster CPU , more RAM



**..continued**

- Management Node
  - Manages the configuration and control of the MySQL Cluster
  - ndb\_mgmd
  - Start the cluster
  - Need not be operational once cluster is started.
  - Low end machine



**..continued**

- API Nodes
  - Retrieve data
  - mysqld server (NDB storage engine)
  - Mediators between the end process and the data nodes



## Transporters

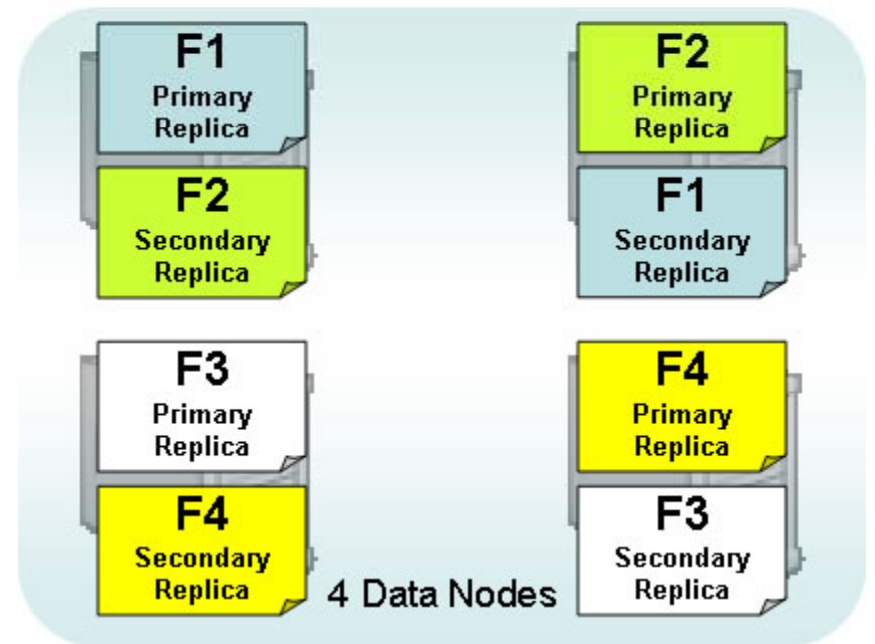
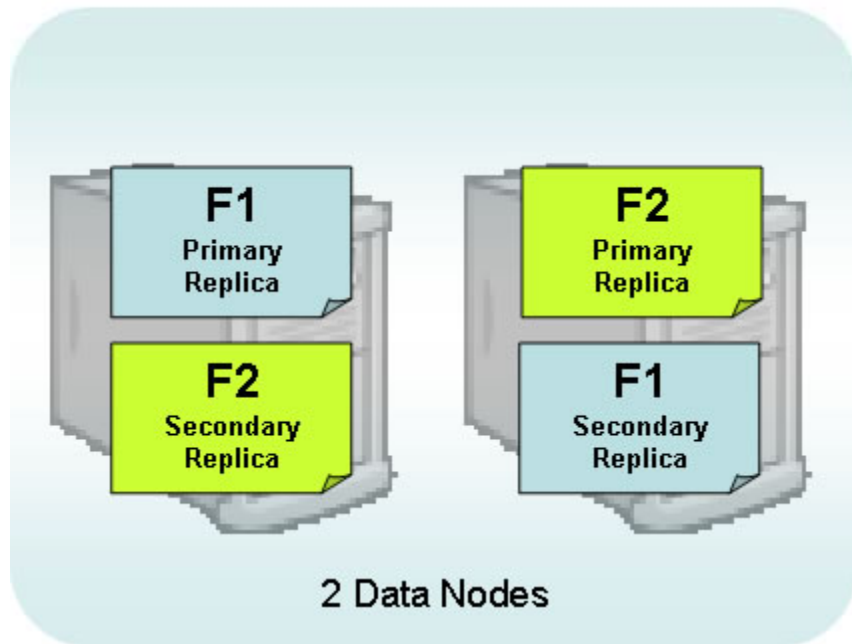
- Used to communicate between nodes
  - TCP/IP
  - Shared memory (multiple nodes in single machine)
  - Scalable Coherent Interface (SCI)



## Replicas

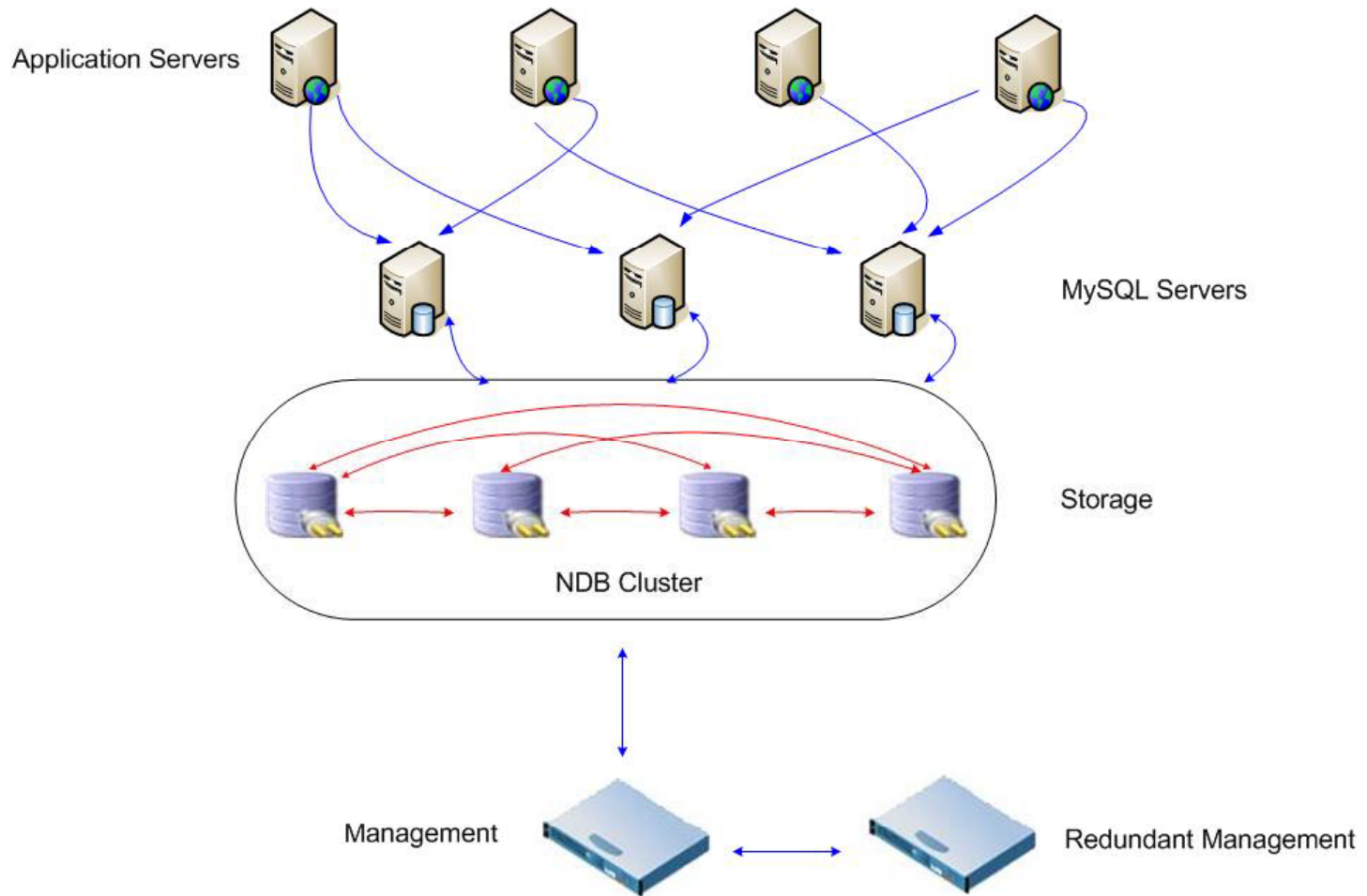
- Replica defines the amount of data redundancy
- Duplication of data
- No of data nodes =  $\max(\text{replicas})$
- Data nodes divisible by number of replicas
- Minimum one (no redundancy)
- Horizontal partitioning
- Unique data copy forms a node group.

# Fragments





## Cluster Architecture





## Before getting into cluster

- Not suitable for all query patterns (complex JOINS, range scans)
- Temporary tables are not supported
- Does not support FULLTEXT indexes
- Does not handle large transactions well. Large transactions require very large amounts of memory
- No support for foreign key constraints
- The NDBCLUSTER storage engine supports only the READ COMMITTED transaction isolation level



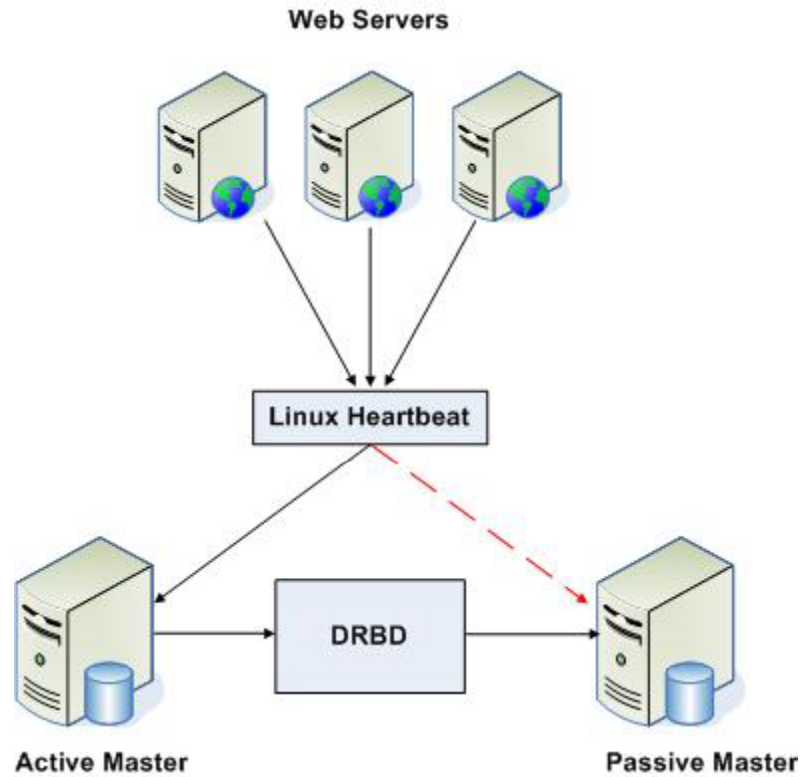
# MySQL DRBD



## **Distributed Replication Block Device**

- Runs over standard IP networks
- Distributed storage
- Similar to network RAID
- Synchronous replication
- No special networking components
- Excellent performance (blocks vs. rows of data)
- Manages inconsistencies of data during a failure
- Hides the complexity of many recovery actions
- Linux heartbeat manages fail over and virtual IPs
- Fully-supported add-on to MySQL Enterprise subscription

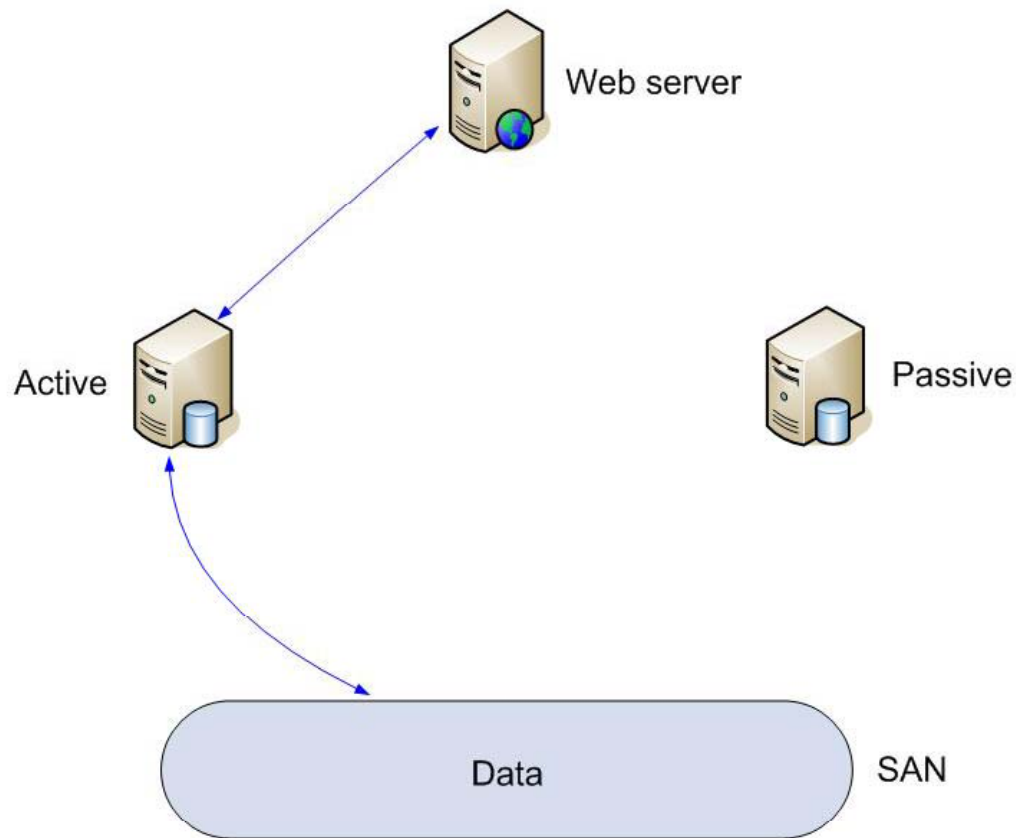
# DRBD



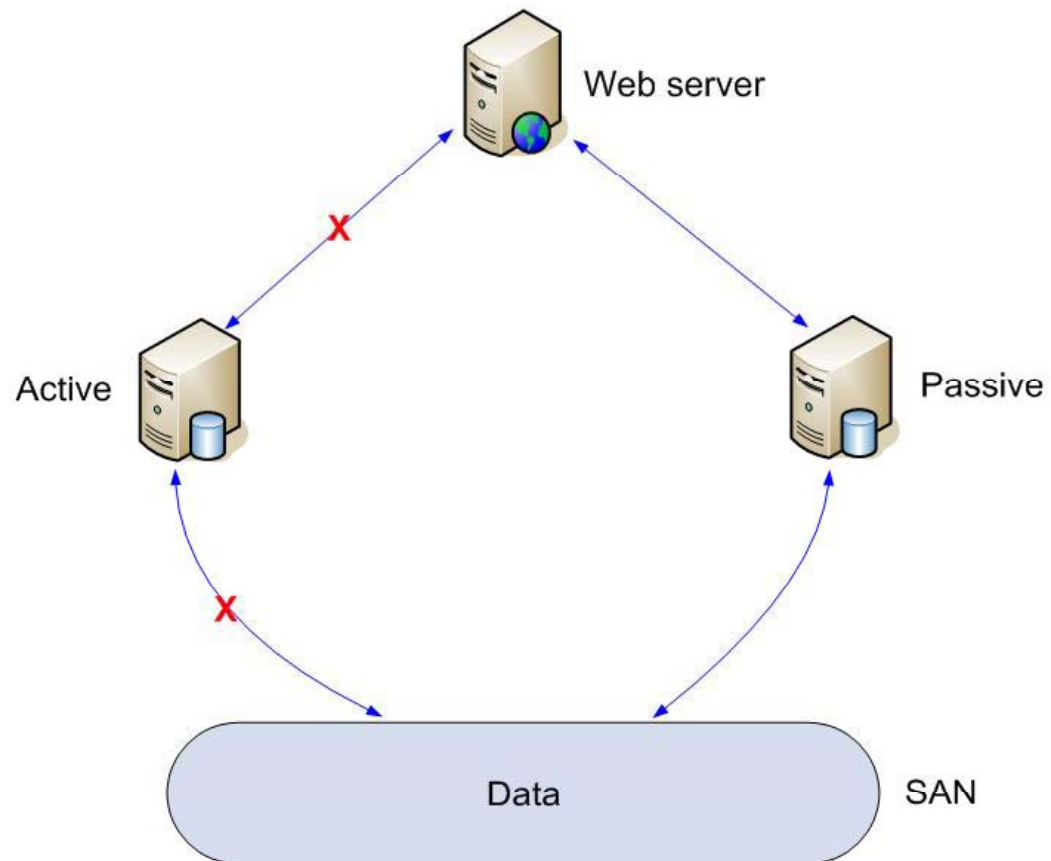
→ Insert, Update, Delete  
- - - - - Heart Beat



## High Availability using SAN



# Fail over





**Questions ?**



**Thank you!**