
DEVELOPER ZONE ::

Chapter 14. High Availability, Scalability, and DRBD

Table of Contents

When using MySQL you may need to ensure the availability or scalability of your MySQL installation. Availability refers to the ability to cope with, and if necessary recover from, failures on the host, including failures of MySQL, the operating system, or the hardware. Scalability refers to the ability to spread the load of your application queries across multiple MySQL servers. As your application and usage increases, you may need to spread the queries for the application across multiple servers to improve response times.

There are a number of solutions available for solving issues of availability and scalability. The two primary solutions supported by MySQL are MySQL Replication and MySQL Cluster. Further options are available using third-party solutions such as DRBD (Distributed Replicated Block Device) and Heartbeat, and more complex scenarios can be solved through a combination of these technologies. These tools work in different ways:

- *MySQL Replication* enables statements and data from one MySQL server instance to be replicated to another MySQL server instance. Without using more complex setups, data can only be replicated from a single master server to any number of slaves. The replication is asynchronous, so the synchronization does not take place in real time, and there is no guarantee that data from the master will have been replicated to the slaves.
    - ***Advantages***
    - MySQL Replication is available on all platforms supported by MySQL, and since it isn't operating system-specific it can operate across different platforms.
    - Replication is asynchronous and can be stopped and restarted at any time, making it suitable for replicating over slower links, partial links and even across geographical boundaries.
    - Data can be replicated from one master to any number of slaves, making replication suitable in environments with heavy reads, but light writes (for example, many web applications), by spreading the load across multiple slaves.
    - ***Disadvantages***
    - Data can only be written to the master. In advanced configurations, though, you can set up a multiple-master configuration where the data is replicated around a ring configuration.
    - There is no guarantee that data on master and slaves will be consistent at a given point in time. Because replication is asynchronous there may be a small delay between data being written to the master and it being available on the

slaves. This can cause problems in applications where a write to the master must be available for a read on the slaves (for example a web application).

- ***Recommended uses***
- Scale-out solutions that require a large number of reads but fewer writes (for example, web serving).
- Logging/data analysis of live data. By replicating live data to a slave you can perform queries on the slave without affecting the operation of the master.
- Online backup (availability), where you need an active copy of the data available. You need to combine this with other tools, such as custom scripts or Heartbeat. However, because of the asynchronous architecture, the data may be incomplete.
- Offline backup. You can use replication to keep a copy of the data. By replicating the data to a slave, you take the slave down and get a reliable snapshot of the data (without MySQL running), then restart MySQL and replication to catch up. The master (and any other slaves) can be kept running during this period.

For information on setting up and configuring replication, see Chapter 15, *Replication*.

- *MySQL Cluster* is a synchronous solution that enables multiple MySQL instances to share database information. Unlike replication, data in a cluster can be read from or written to any node within the cluster, and information will be distributed to the other nodes.
  - ***Advantages***
  - Offers multiple read and write nodes for data storage.
  - Provides automatic failover between nodes. Only transaction information for the active node being used is lost in the event of a failure.
  - Data on nodes is instantaneously distributed to the other data nodes.
  - ***Disadvantages***
  - Available on a limited range of platforms.
  - Nodes within a cluster should be connected via a LAN; geographically separate nodes are not supported. However, you can replicate from one cluster to another using MySQL Replication, although the replication in this case is still asynchronous.
  - ***Recommended uses***
  - Applications that need very high availability, such as telecoms and banking.
  - Applications that require an equal or higher number of writes compared to reads.

  For information on MySQL Cluster, see Chapter 16, *MySQL Cluster*.

- *DRBD (Distributed Replicated Block Device)* is a third-party solution from Linbit supported only on Linux. DRBD creates a virtual block device (which is associated with an underlying physical block device) that can be replicated from the primary server to a secondary server. You create a filesystem on the virtual block device, and this information is then replicated, at the block level, to the secondary server.

  Because the block device, not the data you are storing on it, is being replicated the validity of the information is more reliable than with data-only replication solutions. DRBD can also ensure data integrity by only returning from a write operation on the primary server when the data has been written to the underlying physical block device on both the primary and secondary servers.
  - ***Advantages***
  - Provides high availability and data integrity across two servers in the event of hardware or system failure.
  - Ensures data integrity by enforcing write consistency on the primary and secondary nodes.
  - ***Disadvantages***
  - Only provides a method for duplicating data across the nodes. Secondary nodes cannot use the DRBD device while data is being replicated, and so the MySQL on the secondary node cannot be simultaneously active.
  - Cannot provide scalability, since secondary nodes don't have access to the secondary data.

- ***Recommended uses***
- High availability situations where concurrent access to the data is not required, but instant access to the active data in the event of a system or hardware failure is required.

For information on configuring DRBD and configuring MySQL for use with a DRBD device, see Section 14.1, "Using MySQL with DRBD for High Availability".

- *Heartbeat* is a third party software solution for Linux. It is not a data replication or synchronization solution, but a solution for monitoring servers and switching active MySQL servers automatically in the event of failure. Heartbeat needs to be combined with MySQL Replication or DRBD to provide automatic failover.

The information and suitability of the various technologies and different scenarios is summarized in the table below.

Requirements	MySQL Replication	MySQL Replication + Heartbeat	MySQL Heartbeat + DRBD	MySQL Cluster
Availability				
Automated IP failover	No	Yes	Yes	No
Automated database failover	No	No	Yes	Yes
Typical failover time	User/script-dependent	Varies	< 30 seconds	< 3 seconds
Automatic resynchronization of data	No	No	Yes	Yes
Geographic redundancy support	Yes	Yes	Yes, when combined with MySQL Replication	Yes, when combined with MySQL Replication
Scalability				
Built-in load balancing	No	No	No	Yes
Supports Read-intensive applications	Yes	Yes	Yes, when combined with MySQL Replication	Yes
Supports Write-intensive applications	No	No	Yes	Yes
Maximum number of nodes per group	One master, multiple slaves	One master, multiple slaves	One active (primary), multiple passive (secondary) nodes	255
Maximum number of slaves	Unlimited (reads only)	Unlimited (reads only)	Unlimited (reads only)	Unlimited (reads only)

Previous / Next / Up / Table of Contents

User Comments

Add your own comment.

Top / Previous / Next / Up / Table of Contents