1. **replecation.**

Replication là việc sao chép dữ liệu từ một DB server chính(master) sang một hoặc nhiều DB server khác(slave).

* **Ý nghĩa của Replication:**

+) Khả năng sẵn sàng cao: Khi có nhiều DB server để lưu trữ dữ liệu thì khả năng đáp ứng các yêu cầu dữ liệu của người sử dụng được tăng lên. Giả sử khi có sự cố trên DB chính, hoặc do lượng truy cập nhiều thì việc cần tới các DB server slave là rất quan trọng.

+) Khả năng cân bằng tải(load balance):cân bằng lưu lượng truy xuát dữ liệu từ người dùng. Khi đó sẽ có thể truy vấn dữ liệu được từ nhiều DB server.

+) Khả năng dự phòng và sao lưu dữ liệu: Vì dữ liệu được sao chép ra các DB server slave, mà tại các DB slave thì có thể chạy các chương trình backup dữ liệu mà không ảnh hưởng tới dữ liệu của DB master. Dữ liệu được lưu ở nhiều nơi khác nhau nên tính an toàn được bảo đảm.

+) Tính phân tích dữ liệu: dữ liệu thô được tạo ra tại DB master, trong khi đó thì việc phân tích thông tin dữ liệu có thể thực hiện trên các DB slave mà không ảnh hưởng tới hiệu suất của DB master.

+) Khả năng phân tán: Trong trường hợp dữ liệu được truy xuất từ nhiều nơi, có thể từ những vùng cách rất xa DB master, thì việc tạo ra một bản sao dữ liệu của DB master tại đó thì việc truy xuất dữ liệu không phải truy xuất thẳng đến DB master, đảm bảo cho performance của hệ thống.

* **Những vấn đề mà Replication không giải quyết được:**

+) Tính toàn vẹn và đúng đắn của dữ liệu: vì là sao chép dữ liệu từ một DB master sang các DB slave nên khi có thay đổi nào từ DB master thì cũng sẽ có những thay đổi tương tự trên DB slave.

+) Replication trong MySQL là replication không đồng bộ. Nên không hữu dụng với các hệ thống yêu cầu thời gian thực đối với DB slave.

* **Các thuật ngữ dùng trong Replication:**

+) DB MySQL Master: chịu trách nhiệm về việc cập nhật, thay đổi data, tạo binlog ghi log và gửi binlog cho DB slave khi có yêu cầu từ phía slave.

+) DB MySQL Slave: lấy binlog từ phía Master, chịu trách nhiệm việc sao chép dữ liệu.

+) binary log(binlog):ghi log tất cả những j đã thực thi ở Master, làm thông tin sử dụng cho việc replication.

+) Synchronous Replication(replication đồng bộ): một transaction sẽ không được commit cho đến khi mà data đã được sao chép xong để đảm bảo đồng bộ dữ liệu giữa Master và Slave.Replication đồng bộ an toàn nhưng chậm và chỉ có trong MySQL Cluster.



+) Asynchronous replication: Sau khi các transaction đã được thực thi và commit trên DB master thì việc sao chép dữ liệu mới được thực thi. Replication trong MySQL được thực hiện sau khi mà DB master đã thực thi xong các câu query và ghi những thay đôi vào binlog.



* **Các bước thực thi Replication trong MySQL:**

B1: DB master ghi lại các record thay đổi, các dữ liệu của chúng vào binary log.

B2: DB slave lấy binary log từ DB master và copy vào relay log của DB slave.

B3: DB slave tiến hành thực thi các sự kiện trong replay log và tiến hành update các thay đổi vào data của mình.

* **Cấu hình cho Replication:** Để cài đặt Replication thì thông thường qua các bước cơ bản như sau: thiết lập tài khoản user cho replication tại mỗi server, cấu hình master và slave, thiết lập việc replication(liên quan đến việc kết nối tới Master và sao chép dữ liệu từ Master).

**+)** Thiết lập tài khoản Replication: Vì mỗi một slave phải connect đến master thông qua một tài khoản. Nên phải tạo tài khoản truy cập cho slave trên master để slave có thể kết nói với master. Cú pháp tạo tài khoản như sau:

[GRANT REPLICATION SLAVE ON \*.\* TO ‘rep\_user’@’slave-host’ IDENTIFIED BY ‘this-is-the-password’]. Trong đó \*.\* có thể là dbname, dbname.table\_name, table\_name,..

+) Cấu hình cho Master và Slave:

Cấu hình cho Master: Trong file config: my.cnf cấu hình cho master thông qua 2 giá trị là [server-id] và [log-bin].[server-id] là ID của server, mỗi server chỉ có một ID duy nhất là số trong khoảng từ 1-2^32.[log-bin] là file binary log lưu các thay đổi của DB Master.

Cấu hình cho Slave: Trong file config: my.cnf cấu hình cho slave thông qua giá trị là [server-id].

+) Tiến hành Replication: Để thực thi replication trên slave, ta dùng câu lệnh sau:

*mysql> CHANGE MASTER TO*

*-> MASTER\_HOST='master\_host\_name',*

*-> MASTER\_USER='replication\_user\_name',*

*-> MASTER\_PASSWORD='replication\_password',*

*-> MASTER\_LOG\_FILE='recorded\_log\_file\_name',*

*-> MASTER\_LOG\_POS=recorded\_log\_position;*

*MASTER\_HOST:* tên domain hoặc địa chị IP của DB Master

*MASTER\_USER, MASTER\_PASSWORD* : user và password của tài khoản mà truy cập đến Master.

*MASTER\_LOG\_FILE:* tên file log của master(binary log)

*MASTER\_LOG\_POS:* vị trí log trong file binlog

1. **DB Master.**

DB Master là DB MySQL Server mà đóng vai trò làm DB lưu trữ dữ liêu chính của hệ thống. Tất cả các việc liên quan đến thay đổi dữ liệu như là tạo mới, thay đổi, xóa dữ liệu đều được thực thi trên Master. Chúng ta có thể sử dụng một số câu lệnh sau trong việc monitoring replication trên Master:

* **Show Master Status**: cung cấp thông tin về binary log file của Master

Kết quả trả về của lệnh này là :

+---------------+----------+--------------+------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+---------------+----------+--------------+------------------+

| mysql-bin.003 | 73 | test | manual,mysql |

+---------------+----------+--------------+------------------+

* **Show Binary Log** : list ra các file binary log

Kết quả trả về là :

+---------------+-----------+

| Log\_name | File\_size |

+---------------+-----------+

| binlog.000015 | 724935 |

| binlog.000016 | 733481 |

+---------------+-----------+

* **Show binlog events** : hiển thị thông tin về các evenst được ghi trong binlog.

Kết quả trả về có dạng là :

Log\_name: slave-bin.000001

Pos: 390

Event\_type: Query

Server\_id: 2

End\_log\_pos: 476

Info: use `test`; create table t1 (a int)

* **Reset Master**: Xóa tất cả các file binary log mà được list trong file index, reset file index binary log, và tạo mới môt file binary log.

1. **DB Slave:** là nơi lưu trữ dư liệu sao chép từ DB master. Chúng ta có thể sử dụng một số câu lệnh sau để monitoring status của Slave.

* **Show Slave Status:** Hiển thị thông tin trạng thái của slave như là: Các thread đang chạy(Slave IO thread, Slave SQL thread), thời gian trễ của slave, vị trí mà replication trong file binary log,…
* **Start Slave [thread\_types]:** khởi động các slave thread.[thread\_types] có thể là SQL\_Thread hoặc IO\_Thread. Nếu không có [thread\_types] thì mặc định là cả 2 bị dừng.
* **Stop Slave [thread\_types]:** stop các thread slave: [thread\_types] có thể là SQL\_Thread hoặc IO\_thread. Nếu không có [thread\_types] thì mặc định là cả 2 bị dừng.
* **Reset Slave:** xóa các thông tin trong relay log.