Disabling Write Barrier

1. Mount the device:

```
$ mkdir test
$ sudo mount /dev/sda1 -o nobarrier test
$ sudo chown -R yourUsername:yourUsername test
```

You need to change `/dev/sda1` to the partition name of your device and `yourUsername` to your user name. Note that we have turned off the **write barrier** option (`-o nobarrier`) to mitigate the overhead of *fsync()*. The detailed reasons are as follows:

A **write barrier** is a kernel mechanism used to ensure that file system metadata is correctly written and ordered on persistent storage, even when storage devices with volatile write caches lose power. File systems with write barriers enabled also ensure that data transmitted via fsync() is persistent throughout a power loss. However, enabling write barriers incurs a substantial performance penalty for some applications. Specifically, applications that use fsync() heavily or create and delete many small files will likely run much slower. For devices with non-volatile, battery-backed write caches and those with write-caching disabled, you can safely disable write barriers at mount time using the **`-o nobarrier`** option for mount.

2. You can check the mounted device and see **nobarrier** option with the below command:

```
$ mount
...
/dev/sda1 on /home/mijin/test type ext4 (rw,relatime,nobarrier,data=ordered)
...
```

3. Specify the path of the mounted device directory to the MySQL server by modifying *my.cnf*.

```
$ vi my.cnf
...
# If you want to change the path of the data directory, update below variable:
datadir=/home/mijin/test
...
# If you want to change the path of the log directory, update below variable:
innodb_log_group_home_dir=/home/mijin/xxx
...
```

4. Run the MySQL server in your MySQL directory:

```
$ ./bin/mysqld_safe --defaults-file=/home/mijin/my.cnf
```

Enabling/Disabling the Doublewrite Buffer

When enabled (the default), InnoDB stores all data twice, first to the doublewrite buffer, then to the actual data files. This variable can be turned off with `--skip-innodb-doublewrite` for benchmarks or cases when top performance is needed rather than concern for data integrity or possible failures. Or you can modify the *my.cnf* file and change the value of `innodb_doublewrite` as you want:

```
$ vi my.cnf

#
# The MySQL database server configuration file.
#
[client]
user    = root
port    = 3306
socket  = /tmp/mysql.sock

...

# Doublewrite buffer ON or OFF
innodb_doublewrite=ON
```