

How to reduce MySQL ibdata when you're out of space

Vložil/a [cm3l1k1](#) [1], 27 Únor, 2013 - 14:15

- [GNU/Linux a BSD](#) [2]
- [Networks & Protocols](#) [3]

You can easily jump into troubles if you're using MySQL InnoDB storage engine in its default configuration (which is most used setup). It is absurd that there is not built-in function to compact ibdata file. This is something that I don't fully understand, but you simply can't delete space/transactions in this file and only acceptable (not reasonable) answer which I have found has been related to performance impact when file needs to be expanded...
The only way how you can reduce size of ibdata is database/table export which you will import back. BUT you're already out of space so how to do it?

// Imagine this scenario

Server A 1.1.1.1 - MySQL server, out of space
Server B 2.2.2.2 - standard Linux box with free space

You will see how powerful [netcat](#) [4] can be :]

// Reducing ibdata

1] Kill all daemons which can use MySQL

(Apache, application, etc.) on server A

2] Dump MySQL database

If you're not using `innodb_file_per_table = 1` dump all data and send output via network to our server B.

On server B will listen netcat on port 12345

```
[2.2.2.2]$ nc -l -p 12345 > /var/enough-space/mysql-export-all.sql
```

On server A we will export all databases to stdout which is handled by netcat and sent to our server B on port 12345 (use parameters to authorize this if needed)

```
[1.1.1.1]$ mysqldump --extended-insert --all-databases --add-drop-database  
--disable-keys --flush-privileges --quick --routines --triggers --skip-lock-tables  
--force | nc 2.2.2.2 12345
```

...it will take some time, depend on size of your database. Netcat will stop listening when transfer is completed

3] Delete all files inside datadir on server A

[usually /var/lib/mysql/]

```
[1.1.1.1]$ rm -rf /var/lib/mysql/ && mkdir /var/lib/mysql/
```

4] Initialize MySQL on server A

Update your /etc/my.cnf with option

```
innodb_file_per_table = 1
```

which will create innodb file for each table.

```
[1.1.1.1]$ sudo -u mysql mysql_install_db
```

```
[1.1.1.1]$ chown -R mysql:mysql /var/lib/mysql
```

```
# Start MySQL
```

```
[1.1.1.1]$ /etc/init.d/mysqld start
```

5] Create FIFO file and import database

At this moment we can't simply transfer SQL file back and import it into server, because size of file together with size of newly imported data from file will fill the disk again. So we will create special file which is working in same way as fifo buffer to which we will send stdout of netcat.

```
# Create FIFO
```

```
[1.1.1.1]$ mknod /var/fifo-buffer p
```

```
# Netcat will listen on port 54321 and send everything to FIFO
```

```
[1.1.1.1]$ nc -l -p 54321 > /var/fifo-buffer
```

```
# IN NEXT TERMINAL WINDOW ON SAME SERVER
```

```
# Go into mysql console
```

```
[1.1.1.1]$ mysql
```

```
mysql> SET FOREIGN_KEY_CHECKS=0;
```

```
mysql> SOURCE /var/fifo-buffer;           # during this step mysql waiting on data from  
buffer
```

```
mysql> SET FOREIGN_KEY_CHECKS=1;
```

```
mysql>
```

```
mysql> quit
```

```
# Now we will send data from server B to server A
```

```
[2.2.2.2]$ cat /var/enough-space/mysql-export-all.sql | nc 1.1.1.1 54321
```

As usual netcat will stop listening when transfer is completed and you're done!!

Few tips

First of all please consider if you can't move forward and use [Percona MySQL](#) [5] which is back-compatible with MySQL, brings a lot of new features, high performance, better memory usage, more scalability on many cores and last but not least evolution replacement of InnoDB called [XtraDB](#) [6].

You can also move second ibdata file to different disk when reached defined size in standard data home dir.

example /etc/my.cnf

```
innodb_data_file_path = ibdata1:200M:/disk2/ibdata2:50M:autoextend
```

This will create 200M file in default data home dir (innodb_data_home_dir), when this size is reached next 50M ibdata file will be created on /disk2/ibdata2 and autoextended when needed. This should be local disk or SAN. In case of NFS you will hit performance bottleneck.

References

InnoDB Configuration - <http://dev.mysql.com/doc/refman/5.0/en/innodb-configuration.html> [7]
Percona XtraDB Storage Engine for MySQL - <http://www.percona.com/software/percona-xtradb> [6]
Percona Server Feature Comparison -
<http://www.percona.com/software/percona-server/feature-comparison> [8]
Percona Server with XtraDB - <http://www.percona.com/software/percona-server> [9]

URL článku: <https://security-portal.cz/blog/how-reduce-mysql-ibdata-when-youre-out-space>

Odkazy:

[1] <https://security-portal.cz/users/cm3l1k1>
[2] <https://security-portal.cz/category/tagy/gnu/linux-bsd>
[3] <https://security-portal.cz/category/tagy/networks-protocols>
[4] <http://netcat.sourceforge.net/>
[5] <http://www.percona.com/>
[6] <http://www.percona.com/software/percona-xtradb>
[7] <http://dev.mysql.com/doc/refman/5.0/en/innodb-configuration.html>
[8] <http://www.percona.com/software/percona-server/feature-comparison>
[9] <http://www.percona.com/software/percona-server>