Technical Report

Best Practices Guidelines for MySQL

Karthikeyan Nagalingam, NetApp
December 2010 | TR-3657

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

This document describes best practices guidelines for running MySQL databases on NetApp® storage systems with a Linux® platform. This document reflects the work done by NetApp, by MySQL, and by NetApp engineers at various joint customer sites. It is a starting reference point that describes the bare minimum guidelines for deploying MySQL on NetApp storage systems.

MySQL Enterprise is one of the world's most popular open source databases. A key strength of the MySQL Enterprise database is its excellent performance and scalability, making it well suited to serve as a back end for Web sites, data warehousing, and other data-intensive applications in the enterprise environment.

NetApp provides the ideal storage platform for MySQL Enterprise databases that require high availability and high performance. NetApp storage not only supports but excels at protecting and serving data with all the major storage protocols, including NFS, FCP, and iSCSI. In support of enterprise data systems, NetApp storage systems provide superior, cost-effective data protection, backup and recovery, availability, and administration through NetApp tools and features that include the following:

- Fast, reliable backups using Snapshot™, SnapVault®, and NearStore® technologies
- Cloning and database refresh tools
- Disk redundancy through RAID and RAID-DP®
- Storage system redundancy through the use of cluster technology
- Extensive array of disaster recovery tools

NetApp storage technologies combined with MySQL Enterprise provide creative, tailored solutions based on customers' performance and business requirements.

This guide assumes basic understanding of `technology and operation of NetApp products. It presents options and recommendations for planning, deployment, and operation of NetApp products to maximize their effectiveness.

2 NETAPP STORAGE SYSTEM CONFIGURATION

2.1 NETWORK SETTINGS

When configuring network interfaces for new systems, it's best to run the setup command to automatically bring up the interfaces and update the `/etc/rc` and `/etc/hosts` files. The setup command requires a reboot to take effect.

However, if a system is in production and cannot be rebooted, network interfaces can be configured by using the `ifconfig` command. If a NIC is currently online and needs to be reconfigured, it must first be brought down. To minimize downtime on that interface, a series of commands can be entered on a single command line, separated by a semicolon (;).

Example:

```
filer>ifconfig e0a down;ifconfig e0a 'hostname'-e0a mediatype auto netmask
255.255.255.0 partner e0a
```

When configuring or reconfiguring NICs or VIFs in a cluster, it is imperative to include the appropriate `partner <interface> name` or VIF name in the configuration of the cluster partner's NIC or VIF to provide fault tolerance in the event of cluster takeover. Consult your NetApp support representative for assistance. A NIC or VIF used by a database should not be reconfigured while the database is active. Doing so can result in a database crash.

```
nvfail on
```

```
fcp.enable on
iscsi.enable on
nfs.v3.enable on
nfs.tcp.enable on
nfs.tcp.recvwindowsize 65536
nfs.tcp.xfersize 65536
iscsi.iswt.max_ios_per_session 128
iscsi.iswt.tcp_window_size 131400
iscsi.max_connections_per_session 16
```

2.2 ETHERNET—GIGABIT ETHERNET, AUTO NEGOTIATION, AND FULL DUPLEX

Any database using NetApp storage must use Gigabit Ethernet on both storage systems and database servers.

NetApp Gigabit II, III, and IV cards are designed to auto-negotiate interface configurations and are able to intelligently self-configure themselves if the auto negotiation process fails. For this reason, NetApp recommends that Gigabit Ethernet links on clients, switches, and NetApp systems be left in their default auto-negotiation state, unless the link is not established, performance is poor, or other conditions arise that might warrant further troubleshooting.

Flow control should by default be set to "full" on the storage system in its `/etc/rc` file, by including the following entry (assuming that the Ethernet interface is e5): `ifconfig e5 flowcontrol full`

If the output of the `ifstat –a` command does not show full flow control, then the switch port must also be configured to support it. (The `ifconfig` command on the storage system always shows the requested setting; `ifstat` shows what flow control was actually negotiated with the switch.)

2.3 VOLUME, AGGREGATE SETUP, AND OPTIONS

DATABASES

The structure of the volumes used to store a database should be driven by backup, restore, and mirroring requirements.

A single database instance should not be hosted on multiple unclustered storage systems, because a database with sections on multiple storage systems makes maintenance that requires storage system downtime—even for short periods—hard to schedule and increases the impact of downtime. If a single database instance must be spread across several separate storage systems for performance, care should be taken during planning to minimize the impact of storage system maintenance or backup.

AGGREGATES AND FLEXVOL VOLUMES OR TRADITIONAL VOLUMES

With Data ONTAP® 7G, NetApp supports pooling a large number of disks into an aggregate and then building virtual volumes (FlexVol® volumes) on top of those disks. These virtual volumes have many benefits for MySQL database environments.

For MySQL databases, NetApp recommends that you pool all of your disks into a single large aggregate and use FlexVol volumes for your database data files and log files as described in "Recommended Volumes for MySQL Database Files and Log Files." This provides the benefit of much simpler administration, particularly for growing and reducing volume sizes without affecting performance.

VOLUME SIZE

Although the maximum supported volume size on a NetApp system is 16TB, NetApp recommends that the size of a volume be smaller for the following reasons:

- Reduced per volume backup time
- Individual grouping of Snapshot copies, qtrees, and so on.
- Improved security and manageability through data separation
- Reduced risk from administrative mistakes, hardware failures, and so on

Consider issues such as backup windows and q-tree management in selecting the size of volumes.

RECOMMENDED VOLUMES FOR MYSQL DATABASE FILES AND LOG FILES

Table 1 lists the storage layouts that are adequate for most scenarios based on our testing. The general recommendation is to have a single aggregate across a maximum number of disk spindles and that contains all the flexible volumes for database components.

Table 1) Flexible volumes and aggregates.

Aggregate	Flexible Volumes
Database config files	Dedicated FlexVol volume
Transaction log files	Dedicated FlexVol volume
Database binary Log	Dedicated FlexVol volume
Data files	Dedicated FlexVol volume

2.4 RAID GROUP SIZE

When reconstruction rate (the time required to rebuild a disk after a failure) is an important factor, smaller RAID groups should be used. In this section, the optimum RAID group sizes are recommended based on RAID-DP.

RAID-DP

With the release of Data ONTAP 6.5, Double Parity RAID, or RAID-DP, was introduced. With RAID-DP, each RAID group is allocated an additional parity disk. Given this additional protection, the likelihood of data loss due to a double-disk failure has been nearly eliminated, and therefore larger RAID group sizes can be supported.

NetApp recommends the default RAID group size of 16 for RAID-DP.

2.5 FLEXVOL

NetApp FlexVol technology delivers true storage virtualization solutions that can lower overhead and capital expenses, reduce disruption and risk, and provide the flexibility to adapt quickly and easily to the dynamic needs of the enterprise. FlexVol technology pools storage resources automatically and enables you to create multiple flexible volumes on a large pool of disks. When the MySQL database grows on the fly, you can increase the volume size.

2.6 FLEXCLONE

NetApp FlexClone® technology enables true cloning—instant replication of data volumes and data sets without requiring additional storage space at the time of creation. Each cloned volume is a transparent, virtual copy that you can use for essential enterprise operations, such as testing and bug fixing, platform and upgrade checks, multiple simulations against large data sets, remote office testing and staging, and market-specific product variations. In MySQL disaster recovery solutions, cloned volumes are used for fast recovery of customer data.

2.7 FLEXCACHE

FlexCache™ has the ability to distribute files to remote locations without the need for continuous hands-on management. NetApp systems deployed in remote offices automatically replicate, store, and serve the files or file portions that are requested by remote users without the need for any replication software or scripts. In addition, the same storage systems can be used to store data created and accessed locally.

FlexCache automatically manages the consistency between its locally stored files and the central storage system. Users don't wait for the next replication window to transfer changes from their remote location back to the central storage system. FlexCache simplifies an organization's file distribution processes and thus reduces overall data management and WAN bandwidth costs.

NetApp Storage Accelerator systems are a family of storage systems that host only FlexCache volumes. Although FlexCache software can be added to existing FAS/V-Series systems running NFS, customers will find this new product more cost effective in deployments where only the caching function is required of the storage systems. NetApp Storage Accelerator systems support both standalone and active-active configurations.

2.8 SNAPSHOT AND SNAPRESTORE

NetApp strongly recommends using Snapshot and SnapRestore® for MySQL database backup and restore operations. Snapshot provides a point-in-time copy of the entire database in seconds without incurring any performance penalty, while SnapRestore can instantly restore an entire database to a point in time in the past.

For Snapshot copies to be effectively used with MySQL databases, they must be coordinated with the "flush tables with read locks" and "unlock tables" facility. For this reason, NetApp recommends that automatic Snapshot copies be turned off for volumes that are storing data files for a MySQL database.

To turn off automatic Snapshot copies on a volume, enter:

```
vol options <volname> nosnap on
```

If you want to make the `.snapshot` directory invisible to clients, enter:

```
vol options <volname> nosnapdir on
```

With automatic Snapshot copies disabled, regular Snapshot copies are created as part of the MySQL backup process when the database is in a consistent state.

2.9 SNAP RESERVE

Using `snap reserve` on a volume sets aside a part of the volume for the exclusive use of Snapshot copies.

Note: When using `snap reserve`, Snapshot copies may consume more space than allocated, but the user files may not consume the reserved space.

To see the snap reserve size on a volume, enter:

```
snap reserve
```

To set the volume snap reserve size (the default is 20%), enter:

```
snap reserve <volume> <percentage>
```

Do not use a percent sign (%) when specifying the percentage.

Snap reserve should be adjusted to reserve slightly more space than the Snapshot copies of a volume consume at their peak. The peak Snapshot copy size can be determined by monitoring a system over a few days when activity is high.

You can change snap reserve at any time. Do not raise the snap reserve to a level that exceeds free space on the volume; otherwise, client machines may abruptly run out of storage space.

NetApp recommends that you observe the amount of snap reserve being consumed by Snapshot copies frequently. Do not allow the amount of space consumed to exceed the snap reserve. If the snap reserve is exceeded, consider increasing the percentage of the snap reserve, or delete it.

Snapshot copies until the amount of space consumed is less than 100%. NetApp DataFabric® Manager can aid in this monitoring.

2.10 STORAGE SYSTEM OPTIONS

FILE ACCESS TIME UPDATE

File access time update is another option that can improve access time. If an application does not require or depend on maintaining accurate access times for files, this option can be disabled. Use this option only if the application generates heavy read I/O traffic.

To disable file access time updates, enter:

```
vol options <volname> no_atime_update on
```

3 MYSQL WITH LINUX OPERATING SYSTEMS

3.1 RECOMMENDED LINUX VERSIONS

Table 2) Recommended Linux versions.

OS	MySQL version	FCP	iSCSI	NFS
SUSE 10	5.0.52	✓	✓	✓
RHEL 5	5.0.52	✓	✓	✓
RHEL 4	5.0.52/4.1.22	✓	✓	✓
SUSE SP 2 and 3	5.0.52/4.1.22	✓	✓	✓
Windows® 2003	5.0.52	✓	✓	N/A

3.2 LINUX SETTINGS

ENLARGING A CLIENT'S TRANSPORT SOCKET BUFFERS

Enlarging the transport socket buffers that Linux uses for NFS traffic helps reduce resource contention on the client, reduces performance variance, and improves maximum data and operation throughput. In future releases of the client, the following procedure will not be necessary, because the client will automatically choose an optimal socket buffer size.

1. Log in as `root` on the client and enter:

   `cd into /proc/sys/net/core`

```
echo 262143 > rmem_max
echo 262143 > wmem_max
echo 262143 > rmem_default
echo 262143 > wmem_default
```

2. Remount the NFS file systems on the client.

   This is especially useful for NFS over UDP and when using Gigabit Ethernet. Consider adding this to a system startup script that runs before the system mounts NFS file systems. The recommended size (262,143 bytes) is the largest safe socket buffer size that NetApp has tested. On clients with 16MB of memory or less, leave the default socket buffer size setting to conserve memory.

   Red Hat versions after 7.2 contain a file named `/etc/sysctl.conf` where changes such as this can be added so that they are executed after every system reboot.

3. Add the following lines to the `/etc/sysctl.conf` file on these Red Hat systems:

```
net.core.rmem_max = 262143
net.core.wmem_max = 262143
net.core.rmem_default = 262143
net.core.wmem_default = 262143
kernel.shmmax=2147483648
kernel.shmall=2147483648
kernel.msgmni=2048
kernel.msgmax=65536
kernel.sem=250 32000 32 1024
fs.file-max=65536
```

OTHER TCP ENHANCEMENTS

The following settings can help reduce the amount of work that clients and storage systems do when running NFS over TCP:

```
echo 0 > /proc/sys/net/ipv4/tcp_sack
```

```
echo 0 > /proc/sys/net/ipv4/tcp_timestamps
```

These operations disable optional features of TCP to save a little processing time and network bandwidth.

When building kernels, make sure that `CONFIG_SYNCOOKIES` is disabled. SYN cookies slow down TCP connections by adding extra processing on both ends of the socket. Some Linux distributors provide kernels with SYN cookies enabled. Linux 2.2 and 2.4 kernels support large TCP windows (RFC 1323) by default. No modification is required to enable large TCP windows.

3.3 LINUX NETWORKING—FULL DUPLEX AND AUTO-NEGOTIATION

Most network interface cards use auto-negotiation to obtain the fastest settings allowed by the card and the switch port to which it attaches. Chipset incompatibilities may sometimes result in constant renegotiation, negotiating half duplex, or a slow speed. When diagnosing a network problem, make sure that the Ethernet settings are as expected before looking for other problems. Avoid hard coding the settings to solve auto-negotiation problems, because it only masks a deeper problem. Switch and card vendors should be able to help resolve these problems.

3.4 LINUX NETWORKING—GIGABIT ETHERNET NETWORK ADAPTERS

If Linux servers are using high-performance networking (gigabit or faster), be sure to provide enough CPU and memory bandwidth to handle the interrupt and data rate. The NFS client software and the gigabit driver reduce the resources available to the application, so make sure that resources are adequate. Most gigabit cards that support 64-bit PCI or higher should be able provide good performance.

Any database that uses NetApp storage should use Gigabit Ethernet on both the storage system and the database server to achieve optimal performance.

NetApp has found that the following Gigabit Ethernet cards work well with Linux:

- **SysKonnect:** The SysKonnect SK-98XX series cards work very well with Linux and support single- and dual-fiber and copper interfaces for better performance and availability. A mature driver for this card exists in the 2.4 kernel source distribution.
- **Broadcom:** Many cards and switches use this chipset, including the ubiquitous 3Com solutions.

  This provides a high probability of compatibility between network switches and Linux clients. The driver software for this chipset appeared in the 2.4.19 Linux kernel and is included in Red Hat distributions with earlier 2.4 kernels. Make sure that the chipset firmware is up to date.
- **AceNIC Tigon II:** Several cards, such as the NetGear GA620T, use this chipset, but none are still being manufactured. A mature and actively maintained driver for this chipset exists in the kernel source distribution.
- **Intel® EEPro/1000:** This appears to be the fastest gigabit card available for systems based on Intel, but the card's driver software is included only in recent kernel source distributions (2.4.20 and later) and may be somewhat unstable. The card's driver software for earlier kernels can be found on the Intel Web site. There are reports that the jumbo frame MTU for Intel cards is only 8998 bytes, not the standard 9000 bytes.

3.5 LINUX NETWORKING—JUMBO FRAMES WITH GBE

All of the cards just described support the jumbo frames option of Gigabit Ethernet. Using jumbo frames can improve performance in environments where Linux NFS clients and NetApp systems are together on an unrouted network. Consult the command reference for each switch to make sure that it is capable of handling jumbo frames. There are some known problems in Linux drivers and the networking layer when using the maximum frame size (9000 bytes). If unexpected performance slowdowns occur when using jumbo frames, try reducing the MTU to 8960 bytes.

3.6 LINUX NFS PROTOCOL

MOUNT OPTIONS

For background on NFS and a brief explanation of what the various NFS client mount options do, see NetApp Technical Report 3183: Using the Linux NFS Client with Network Appliance Storage.

```
LINUX: rw, hard, nointr, rsize=32768, wsize=32768, bg, vers=3, proto=tcp
```

Note: To disable the lock in the NetApp storage system for MySQL, add the `nolock` option with these options. Enabling and disabling this option depends on the MySQL deployment in your environment.

NETAPP STORAGE SETTINGS

1. NFS-specific option `nfs.tcp.recwindowsize`:

   BTCPPE-FILER-21*> options nfs.tcp.recvwindowsize 262144

2. Create a volume for NFS by entering:

   BTCPPE-FILER-22> vol create MySQLDataRH5U1 aggr1_22 50g

```
Creation of volume 'MySQLDataRH5U1' with size 50g on containing aggregate
'aggr1_22' has completed.
```

3. Check the security by entering:

   BTCPPE-FILER-22> qtree status MySQLDataRH5U1

```
Volume   Tree     Style Oplocks  Status
-------- -------- ----- -------- ---------
MySQLDataRH5U1           unix  enabled  normal
```

   If the security style is `ntfs` or `mixed`, change it:

   qtree security /vol/MySQLDataRH5U1 unix

4. Export the volume with write permission by entering:

   BTCPPE-FILER-22> exportfs -p "rw,anon=0" /vol/MySQLDataRH5U1

   BTCPPE-FILER-22> exportfs

```
/vol/vol0/home  -sec=sys,rw,root=10.73.68.51,nosuid
/vol/vol0       -sec=sys,rw,anon=0
/vol/vfiler1    -sec=sys,rw,root=10.73.68.51,nosuid
/vol/MySQLDataRH5U1     -sec=sys,rw,anon=0
```

LINUX SETTINGS

1. Create the folders required for mysql database, mysql binary by entering:

   mkdir /var/lib/mysqldata; mkdir /var/lib/mysqlbin

2. Mount the file system and check the writeable permission by entering:

```
[root@zmanda ~]# mount -t nfs -o
rw,bg,hard,rsize=32768,wsize=32768,vers=3,nointr,timeo=600,proto=tcp
10.73.68.22:/vol/MySQLDataRH5U1 /var/lib/mysqldata

[root@zmanda ~]# mkdir /var/lib/mysqldata/testfolder;  ls -ld
/var/lib/mysqldata/testfolder
```

```
drw-rw-r-- 2 root root 4096 Mar 31 18:04 /var/lib/mysqldata/testfolder
```

3. Update `/etc/fstab` for the auto-mounting during reboot by entering:

```
[root@zmanda scripts]# df -HT
```

```
Filesystem    Type     Size   Used  Avail Use% Mounted on
/dev/sda3     ext3      53G   7.3G    43G  15% /
/dev/sda1     ext3     104M    19M    80M  19% /boot
tmpfs        tmpfs     2.1G      0   2.1G   0% /dev/shm
10.173.68.23:/vol/mysqldatabase
nfs     541G   479G    62G  89% /var/lib/mysqldata
```

3.7 FCP AND ISCSI PROTOCOL FOR MYSQL

For information about FCP and iSCSI protocols for MySQL, see http://now.netapp.com/NOW/knowledge/docs/hba/iscsi/linux/iscsi_linux_hu_3.0/pdfs/setup.pdf.

For information about setting up FCP and iSCSI for MySQL, see the appendix, “FCP and iSCSI Protocols for MySQL.”

4 MYSQL DATABASE SETTINGS

Table 3 describes the recommended configuration changes for the MySQL.

Table 3) MySQL configuration changes.

mysqld Option	Value	Default Value	Description
Max_connections	1600	100	Maximum number of connections.
Max_connect_errors	10	512	Number of interrupted connections from a host after which connections are blocked.
Table_cache	2048		Number of open tables for all threads.
sort_buffer_size	64K		Buffer size for each thread that needs to sort results. Increase for faster ORDER BY and GROUP BY operations.
Binlog_cache_size	1M		Size of cache to hold SQL statements for the binary log during a transaction. Increase for large, multiple-statement transactions.
Join_buffer_size	1M		Buffer size for joins that do not use indexes and require a full table scan.
thread_cache_size	16		Number of server threads to cache for reuse.
thread_stack	64k	192k	Per-thread stack size.
query_cache_size	0		Amount of memory allocated for caching query results. 0 disables query cache.

mysqld Option	Value	Default Value	Description
ft_min_word_len	4		Minimum length of the word to be included in a full-text index.
Tmp_table_size	64M		Maximum size of in-memory temp tables. Increase for GROUP BY.
innodb_data_file_path	ibdata1: 100M\ :autoext end		Path to innodb data files and sizes. Increase size from 10M to 100M.
innodb_buffer_pool_size	1800M		Number of bytes of the memory buffer to cache data and indexes.
innodb_additional_mem_pool_size	20M		Number of bytes of the memory pool used to store data dictionary information and data structures. Default is 1M.
innodb_log_file_size	900M		Size of each log file in log group. Default is 5MB.
innodb_flush_log_at_trx_commit	2	1	Determines how log buffer is written to disk. Value of 2 means that data is flushed on every commit, but no disk flush.
innodb_lock_wait_timeout	300	50	Time in seconds to wait on a lock before being rolled back.
innodb_locks_unsafe_for_binlog	1	0 (disabled)	Determines whether or not next-key locking in search and index scans is enabled.
innodb_thread_concurrency	1000	20	Number of threads currently inside the innodb engine. Default varies with MySQL version. For MySQL v5.0.17c.
innodb_concurrency_tickets	500		Number of “free tickets” to allow reentry into the innodb engine.

5 BACKUP, RESTORE, AND DISASTER RECOVERY

5.1 HOW TO BACK UP DATA FROM A NETAPP SYSTEM

Data that is stored on a NetApp system can be backed up to online storage, near-line storage, or tape. The protocol used to access data when a backup is occurring must always be considered. When NFS is used to access data, Snapshot and SnapMirror® can be used and will always result in consistent copies of the file system. They must coordinate with the state of the MySQL database for database consistency.

With Fibre Channel or iSCSI protocol, Snapshot copies and SnapMirror commands must always be coordinated with the server. The file system on the server must be blocked and all data flushed to the storage system before invoking the `Snapshot` command.

Data can be backed up on the same NetApp storage system, to another NetApp storage system, to a NearStore system, or to a tape storage device. Tape storage devices can be directly attached to an appliance, or they can be attached to an Ethernet or Fibre Channel network, and the appliance can be backed up over the network to the tape device.

Possible methods for backing up data on NetApp systems include using:

- Automated Snapshot copies to create online backups

- Scripts on the server that `rsh` to the NetApp system to invoke Snapshot copies to create online backups
- SnapMirror to mirror data to another storage system or NearStore system
- SnapVault to vault data to another NetApp storage system or NearStore system
- Server operating system-level commands to copy data to create backups
- NDMP commands to back up data to a NetApp storage system or a NearStore system
- NDMP commands to back up data to a tape storage device
- Third-party backup tools to back up the storage system or NearStore system to tape or other storage devices

5.2 CREATING ONLINE BACKUPS BY USING SNAPSHOT COPIES

NetApp Snapshot technology makes efficient use of storage by storing only block-level changes between the creation of each successive Snapshot copy. Because the Snapshot process is virtually instantaneous, backups are fast and simple. Snapshot copies can be automatically scheduled, they can be called from a script running on a server, or they can be created by using SnapDrive® or SnapManager®.

Data ONTAP includes a scheduler to automate Snapshot backups. Use automatic Snapshot copies to back up nonapplication data, such as home directories.

Database and other application data should be backed up when the application is in its read only mode. For MySQL databases, this means placing the database tablespaces into read only mode before creating a Snapshot copy.

NetApp recommends using Snapshot copies for backing up Oracle® databases.

No performance penalty is incurred in creating a Snapshot copy. NetApp recommends turning off the automatic Snapshot scheduler.

For information about online MySQL backup using Snapshot copies, see http://media.netapp.com/documents/tr-3601.pdf

RECOVERING INDIVIDUAL FILES FROM A SNAPSHOT COPY

Individual files and directories can be recovered from a Snapshot copy by using native commands on the server, such as the UNIX® `mount` the `snapshot copy` and `cp` command, or by dragging and dropping in Microsoft® Windows. Data can also be recovered by using the single-file SnapRestore option. Use the method that works most quickly.

5.3 RECOVERING DATA USING SNAPRESTORE

SnapRestore quickly restores a file system to an earlier state preserved by a Snapshot copy. SnapRestore can be used to recover an entire volume of data or individual files within that volume.

When using SnapRestore to restore a volume of data, the data in that volume should belong to a single application. Otherwise operation of other applications may be adversely affected.

The single-file option of SnapRestore allows individual files to be selected for restore without restoring all of the data in a volume.

Note: The file being restored by using SnapRestore cannot exist anywhere in the active file system. If it does, the appliance will silently turn the single-file SnapRestore into a copy operation. This may result in the single-file SnapRestore taking much longer than expected (normally the command executes in a fraction of a second) and also requires that sufficient free space exist in the active file system.

NetApp recommends using SnapRestore to instantaneously restore a MySQL database. SnapRestore can restore the entire volume to a point in time in the past or it can restore a single file. It is advantageous to use SnapRestore on a volume level because the entire volume can be restored in minutes, and this reduces downtime while performing MySQL database recovery. When using SnapRestore on a volume level, NetApp recommends storing the MySQL binary log files, Update log files on a separate volume from the main data file volume and use SnapRestore only on the volume that contains the MySQL data files.

5.4 CONSOLIDATING BACKUPS WITH SNAPMIRROR

SnapMirror mirrors data from a single volume or qtree to one or more remote NetApp systems simultaneously. It continually updates the mirrored data to keep it current and available.

SnapMirror is an especially useful tool to deal with shrinking backup windows on primary systems.

SnapMirror can be used to continuously mirror data from primary storage systems to dedicated near-line storage systems. Backup operations are transferred to systems where tape backups can run all day long without interrupting the primary storage. Because backup operations are not occurring on production systems, backup windows are no longer a concern.

5.5 CREATING A DISASTER RECOVERY SITE WITH SNAPMIRROR

SnapMirror continually updates mirrored data to keep it current and available. SnapMirror is the correct tool to use to create disaster recovery sites. Volumes can be mirrored asynchronously or synchronously to systems at a disaster recovery facility. Application servers should be mirrored to this facility as well.

If the DR facility needs to be made operational, applications can be switched over to the servers at the DR site and all application traffic directed to these servers until the primary site is recovered.

When the primary site is back on line, SnapMirror can be used to transfer data efficiently back to the production storage systems. After the production site takes over, normal application operation begins again, and SnapMirror transfers to the DR facility can resume without requiring a second baseline transfer.

5.6 NDMP AND NATIVE TAPE BACKUP AND RECOVERY

The Network Data Management Protocol (NDMP) is an open standard for centralized control of enterprise-wide data management. The NDMP architecture allows backup application vendors to control native backup and recovery facilities in NetApp and other file servers by providing a common interface between backup applications and file servers.

NDMP separates the control and data flow of a backup or recovery operation into separate conversations. This allows greater flexibility in configuring the environment used to protect the data on NetApp systems. Because the conversations are separate, they can originate from different locations, as well as be directed to different locations, resulting in extremely flexible NDMP-based topologies.

If an operator does not specify a Snapshot copy when performing a native or NDMP backup operation, Data ONTAP creates one before proceeding. This Snapshot copy is deleted when the backup is complete. If a file system contains FCP data, a Snapshot copy that was created at a point in time when the data was consistent should always be specified. This is ideally done in script by quiescing an application or placing it in hot backup mode before creating the Snapshot copy. After Snapshot copy creation, normal application operation can resume, and tape backup of the Snapshot copy can occur at any convenient time.

When attaching an appliance to a Fibre Channel SAN for tape backup, it is necessary to first make sure that NetApp certifies the hardware and software in use. A complete list of certified configurations is

available on the NetApp data protection portal. Redundant links to Fibre Channel switches and tape libraries are not currently supported by NetApp in a Fibre Channel tape SAN. Furthermore, a separate host bus adapter must be used in the storage system for tape backup. This adapter must be attached to a separate Fibre Channel switch that contains only storage systems, NearStore appliances, and certified tape libraries and tape drives.

The backup server must either communicate with the tape library via NDMP or have library robotic control attached directly to the backup server.

5.7 USING TAPE DRIVES WITH NETAPP SYSTEMS

NetApp storage systems and NearStore systems support backup and recovery from local, Fibre Channel, and Gigabit Ethernet SAN-attached tape drives. Support for most existing tape drives is included, as well as a method for tape vendors to dynamically add support for new devices. In addition, the RMT protocol is fully supported, allowing backup and recovery to any capable system. Backup images are written by using a derivative of the BSD dump stream format, allowing full file system backups as well as nine levels of differential backups.

5.8 ONLINE BACKUP AND RECOVERY BEST PRACTICES

The MySQL Enterprise Edition does not have the facility of online backup for many engines. We have two solutions for achieving online backup in those cases by using NetApp's suite of storage software. These solutions have the following advantages:

- Speedy backups and restores result in a great reduction in backup time requirements.
- Backups can be made more frequently because they are faster.
- Use of Snapshot copies to create backups—Snapshot creation time does not depend on database size.
- It's easy to recover a particular file, directory, or volume from an online backup.
- Disaster recovery is quicker with online mirroring and restores.
- Data availability is higher because of the high speed of data recovery.

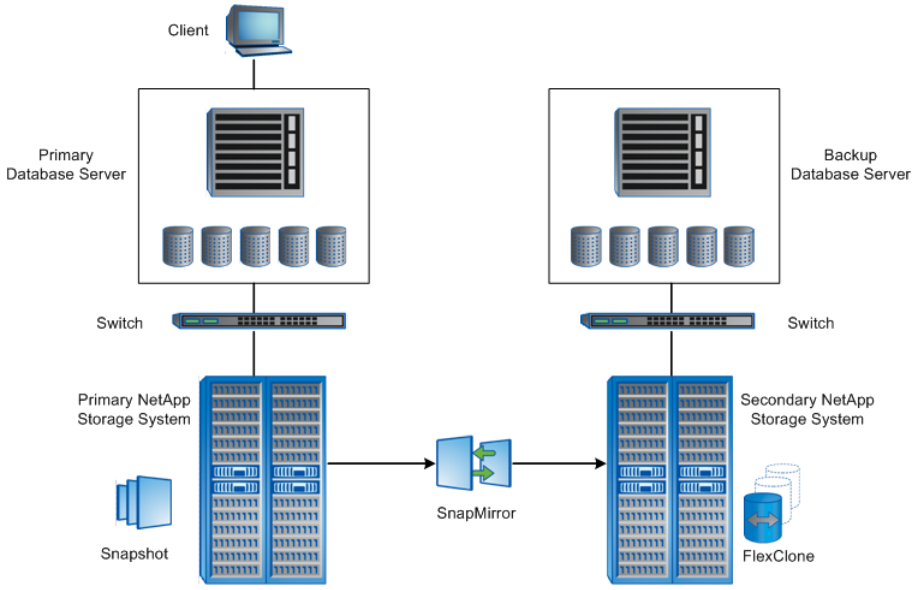
The following are two configurations for online backups for MySQL. The provided scripts can be customized to automate the creation of Snapshot copies and backups.

- Solution A
- Solution B

SOLUTION A

Figure 1 illustrates one of the online backup solutions for MySQL.

Figure 1) Online backups for MySQL (solution A).



Solution Overview

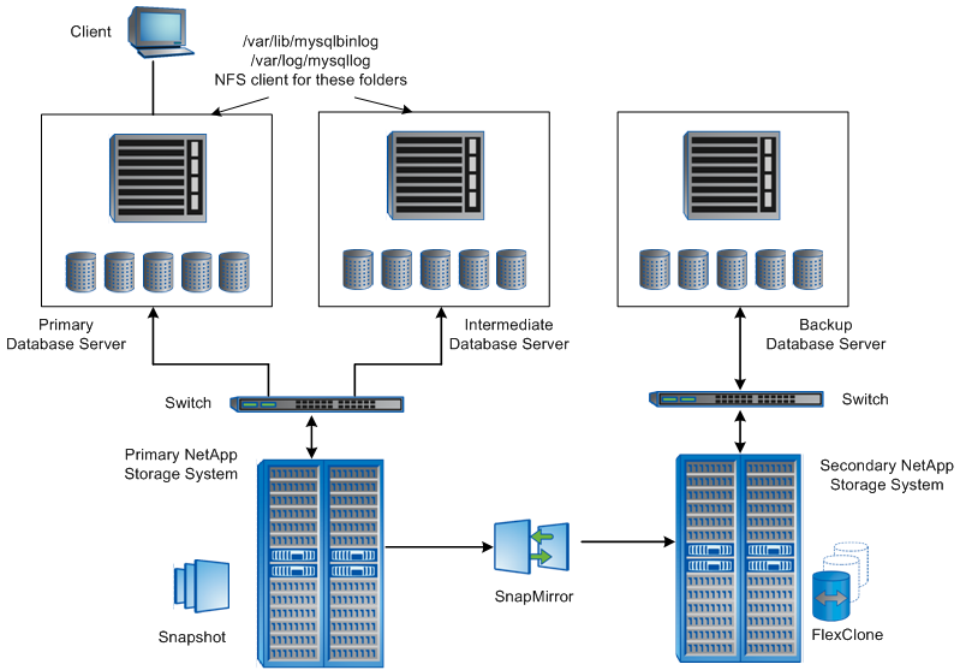
1. The primary database server responds to the clients to access data by using client programs like PHP, Perl, and so on.
2. The primary database server accesses the three LUNs from the primary NetApp storage system through the iSCSI protocol. You can download the iSCSI software initiator from the NOW™ (NetApp on the Web) site.
3. Each LUN serves a separate task, such as one for MySQL `datadir`, one for MySQL `binlog`, and one for MySQL `updatelog`.
4. The daily backup Perl script takes the Snapshot copy of the whole database when the database I/O is low.
5. NetApp SnapMirror mirrors the volume between the primary and secondary storage systems. SnapMirror does the mirror based on the volume update and maintains the last two Snapshot copies of the volume that includes the daily backup Snapshot copies.
6. The NetApp FlexClone technology creates the clone volume based on the daily backup Snapshot in the secondary storage system, which acts like a flexible volume.
7. The backup database server accesses the cloned volumes through the iSCSI protocol for `datadir`, `binlog`, and `mysqlupdate` and maps them to the proper folders.

SOLUTION B

The configuration shown in Figure 2 provides a backup solution while providing almost 100% availability of the primary MySQL database. This solution employs an intermediate database that takes care of the backup process by using the binlog and update-log file features of MySQL. This provides continuous availability of the primary database without the need to put the tablespaces in read-only mode.

The intermediate database instance syncs with the primary database by applying binlog or update-log and is available for backups by using Snapshot and SnapMirror technology.

Figure 2) Online backups for MySQL (solution B).



In this architecture, the settings for the primary NetApp storage system and the secondary NetApp storage system are the same in as previous solution.

1. Using the full backup Snapshot image of the primary database server, create the base database in the intermediate database server. This is a one-time activity.
2. The `binlog` and `update-log` folders are accessible from both the primary and intermediate servers.
3. The logs need to be applied to the intermediate database to keep in sync with the primary.
4. In this solution, the daily and hourly backup scripts make use of this intermediate database.

   For details, see http://media.netapp.com/documents/tr_3601.pdf.

5.9 ZMANDA RECOVERY MANAGER WITH NETAPP SNAPSHOT TECHNOLOGY—MYSQL

NetApp Snapshot and Zmanda Recovery Manager can be used to back up and restore a MySQL database for NetApp storage systems. The Zmanda Recovery Manager covers the following topics:

- Infrastructure required to integrate Zmanda Recovery Manager with a NetApp storage system
- Backing up a MySQL database by using Zmanda Recovery Manager with a NetApp Snapshot plug-in
- Restoring a MySQL database by using Zmanda Recovery Manager

For details, see http://media.netapp.com/documents/tr-3656.pdf.

6 NETAPP SNAP CREATOR FRAMEWORK AND THE MYSQL PLUG-IN

MySQL is a popular database alternative to Oracle Database and SQL Server®. The MySQL module supports both Windows and UNIX systems running MySQL 5.x. The MySQL module uses Net-MySQL to communicate with the database.

For more information about Snap Creator and its server-agent architecture, see the SnapCreator 3.2 Installation and Administration Guide.

This section describes the following:

1. Installation and setup of Snap Creator Framework for MySQL
2. Backup using Snap Creator Framework
3. Restore/Recovery using Snap Creator Framework
4. Cloning using Snap Creator Framework

6.1 INSTALLATION AND SETUP OF SNAP CREATOR FRAMEWORK FOR MYSQL

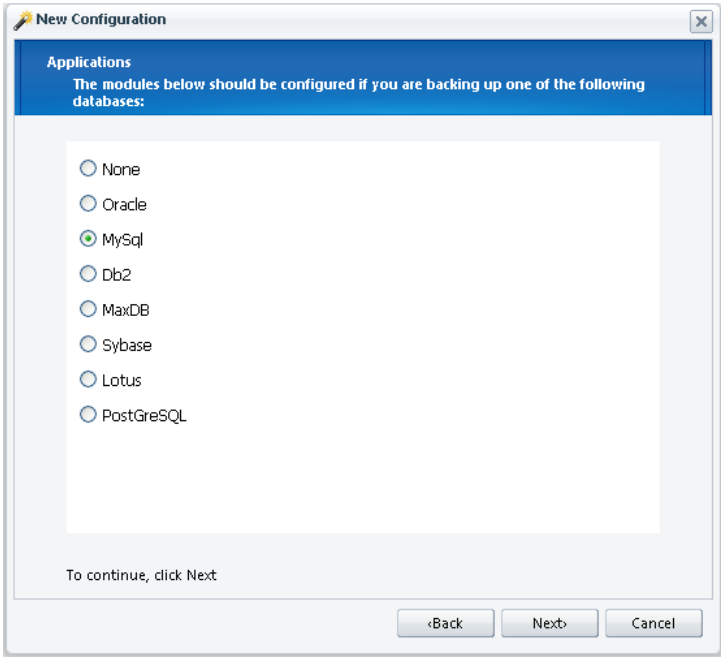
This section describes how to set up the MySQL server to use Snap Creator. For the installation and basic setup of Snap Creator, see the SnapCreator 3.2 Installation and Administration Guide.

Creating the Snap Creator configuration file:

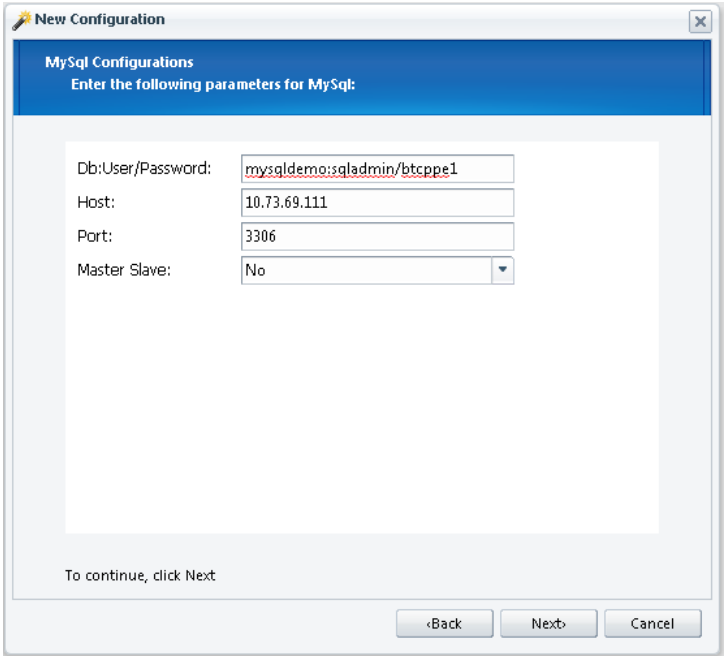
A MySQL specific configuration file is required to perform the backup or restore of MySQL database. The configuration file can be set up using the GUI or CLI.

USING GUI

1. In the SnapCreator GUI, select the MySQL option from the New Configuration wizard.



2. Enter the required credentials and server parameters for the MySQl database.



USING CLI

Use a text editor such as VI to edit the configuration file. Snap Creator configuration file has different sections. The following example shows the minimum required fields for each section. The default values have been changed to those specific to MySQL. When editing the configuration file, attempt to follow the same order as the sections in the default configuration file.

Basic Configuration

Table 4) Basic configuration.

Parameter	Settings (Example)	Description
SNAME	mysqldemo	Your Snapshot copy naming convention should be unique. Snapshot copies on NetApp are deleted according to the naming convention and retention policy used.
VOLUMES	VOLUMES=10.73.69.181:mysqldata	This is the list of source appliances and volumes you want to create a Snapshot copy of, that is, filer1:vol1,vol2,vol3; filer2:vol1;filer3:vol2,vol3.
NTAP_USERS	NTAP_USERS=10.73.69.181:root/btcppe1	This is the list of appliances and their corresponding user names/passwords, that is, filer1:joe/password1;filer2:bob/password2;filer3:ken/password3. **Note**: If you want to use protected passwords, first run **`./snapcreator --cryptpasswd`** and then save the scrambled password in the config file.
NTAP_SNAPSHOT_RETENTIONS	daily:8	This setting determines the number of NetApp Snapshot copies you want to retain for a given policy, that is, daily:7,weekly:4,monthly:1.
NTAP_SNAPSHOT_RETENTION_AGE	1	This setting (in days) allows you to define a retention age for Snapshot copies. If configured, Snapshot copies are deleted only if there are more than the number defined in NTAP_SNAPSHOT_ RETENTIONS and if they are older than the retention age (in days).

NetApp and Additional Plug-ins Options

Table 5) NetApp and additional plug-ins.

Parameter	Settings (Example)	Description
NTAP_SNAPSHOT_RETENTIONS	daily:8	This setting determines the number of NetApp Snapshot copies you want to retain for a given policy, that is, daily:7,weekly:4,monthly:1.
NTAP_SNAPSHOT_RETENTION_AGE	1	This setting (in days) allows you to define a retention age for Snapshot copies. If configured, Snapshot copies are deleted only if there are more than the number defined in NTAP_SNAPSHOT_ RETENTIONS and if they are older than the retention age (in days).
APP_IGNORE_ERROR	Y	The database cannot be access when corrupted. But Snap Creator needs to access the database for restore. You can ignore the application error by setting APP_IGNORE_ERROR=Y to continue the restoration

MYSQL Settings

Table 6) MySQl settings.

Parameter	Settings (Example)	Description
APP_NAME	mysql	The application name.
MYSQL_DATABASES	mysqldemo:sqladmin/btcppe1	A list of MySQL database(s) and the username/password i.e., mysqldb1:user1/pwd1;mysqldb2:user2/pwd2
HOST	10.73.69.111	Name of the host where the databases are located, i.e., localhost/hostname/IP
PORTS	mysqldemo:3306	A list of database(s) and the ports they are listening on, i.e., mysqldb1:3306;mysqldb2:3307
MASTER_SLAVE	N	If the database(s) is (are) part of the MASTER SLAVE environment.

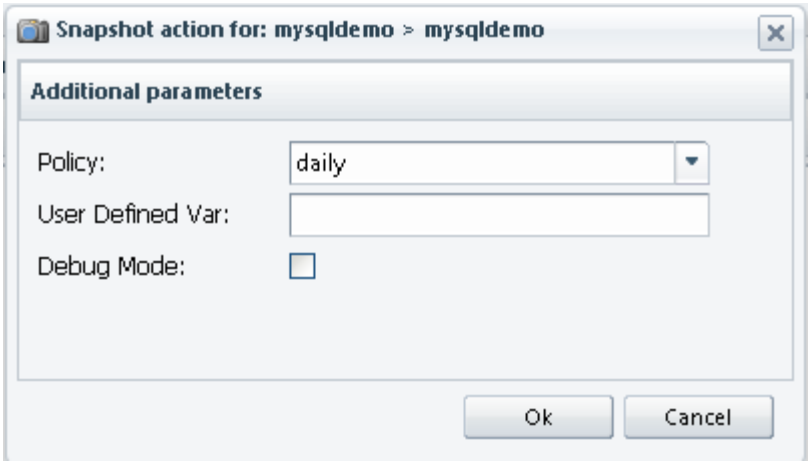
6.2 PERFORMING BACKUP USING SNAP CREATOR AND THE MYSQL PLUG-IN

After you create your configuration file, you can take a backup on the NetApp storage system using Snap Creator. When the `snapcreator` command is executed with the `snap` action, the following operations happen in the background:

1. Snap Creator calls the MySQL plug-in via the Snap Creator configuration file and executes a 'flush tables with read lock" command on the MySQL server to quiesce the database.
2. Snap Creator does an inventory of all the configured volumes by using the ONTAP API calls on the NetApp storage system.
3. Snap Creator creates Snapshot copies for all configured volumes. It also checks that Snapshot copies for all volumes are done successfully.
4. Snap Creator applies the Snapshot management policies and deletes the expired Snapshot copies from the NetApp storage system.
5. Snap Creator sends alerts, if configured in the config file, showing the status of the backup operation.
6. Snap Creator unquiesces the database using the **`unlock tables`** command.

USING GUI

1. Start the backup for MySQL



USING CLI

To use Snap Creator CLI to create the backup of the database, do the following:

`snapcreator --profile [profile_name] --action [action_type] --policy [ policy name] --config [config_file] --verbose`

Where:

- `Profile_name` is the name of your profile or folder that contains your config file.
- `action_type` is the operation you want to perform.
  - `snap` creates a Snapshot copy.
  - `snaplist` lists all Snapshot copies created by Snap Creator.
  - `restore` enters an interactive restore mode.
  - `delete` enters an interactive menu to delete Snapshot copies.
- `policy_name` is the name of the backup policy defined in your config file. The `policy_name` value is determined by the value of `NTAP_SNAPSHOT_RETENTIONS`.
- `config_name` is the name of the configuration file that you wish to run. This field is needed if you have multiple configuration files in the profile directory.
- `verbose` shows the Snap Creator operation on the console and is optional. Snap Creator will save the output to the logs regardless of whether you specify this option.

For example, to create a consistent Snapshot copy for a profile named “mysqldemo” with the policy “daily” and a config file named “mysqldemoconfig,” you would run the following command:

`snapcreator --profile mysqldemo --action snap --policy daily –config mysqldemoconfig --verbose`

1. Quiesce the database (to prevent writes to the MySQL database).

```
########## Application quiesce ##########
[Tue Nov 16 15:02:09 2010] INFO: Quiescing databases
[Tue Nov 16 15:02:09 2010] INFO: Quiescing database mysqldemo
[Tue Nov 16 15:02:09 2010] DEBUG: Connection to mysqldemo successfully established
[Tue Nov 16 15:02:09 2010] DEBUG: Executing sql command 'flush tables with read lock' for
database mysqldemo
[Tue Nov 16 15:02:09 2010] DEBUG: Executing sql command 'flush logs' for database mysqldemo
[Tue Nov 16 15:02:09 2010] INFO: Quiescing database mysqldemo finished successfully
[Tue Nov 16 15:02:09 2010] INFO: Quiescing databases finished successfully
```

2. Check the Snapshot inventory.
3. Create a Snapshot copy.

```
[Tue Nov 16 15:02:10 2010] INFO: Creating NetApp Snapshot for mysqldata on 10.73.69.181
[Tue Nov 16 15:02:10 2010] DEBUG: ZAPI
<snapshot-create>
        <snapshot>mysqldemo-daily_20101116150209</snapshot>
        <volume>mysqldata</volume>
</snapshot-create>

[Tue Nov 16 15:02:11 2010] INFO: NetApp Snapshot Create of mysqldemo-daily_20101116150209 on
10.73.69.181:mysqldata Completed Successfully
```

4. Unquiesce the database.

```
[Tue Nov 16 15:02:11 2010] INFO: Unquiescing databases
[Tue Nov 16 15:02:11 2010] INFO: Unquiescing database mysqldemo
[Tue Nov 16 15:02:11 2010] DEBUG: Connection to mysqldemo established successfully
[Tue Nov 16 15:02:11 2010] DEBUG: Executing sql command 'unlock tables' for database
mysqldemo
[Tue Nov 16 15:02:11 2010] DEBUG: Disconnecting from database mysqldemo
[Tue Nov 16 15:02:11 2010] INFO: Unquiescing database mysqldemo finished successfully
[Tue Nov 16 15:02:11 2010] INFO: Unquiescing databases finished successfully
```

After the Snapshot copy is created, you can list or view it using Snap Creator. To list the Snapshot copy created for the profile "mysqldemo," run the following command:

```
[root@node1  ]# ./snapcreator --profile mysqldemo --action snaplist
########## NetApp Snap Creator Framework Snapshot (Primary) List for 10.73.69.181:mysqldata
#########

### Snapshot Name ###                                ### Snapshot Timestamp ###
mysqldemo-daily_20101115183654                           Nov 15 2010 18:36:29
mysqldemo-daily_20101115190707                           Nov 15 2010 19:06:42
mysqldemo-daily_20101116143244                           Nov 16 2010 14:32:02
mysqldemo-daily_20101116150209                           Nov 16 2010 15:01:28
```

6.3 PERFORMING RESTORE USING SNAP CREATOR AND THE MYSQL PLUG-IN

Any production database is subject to data corruption either by a malicious process or a user error. To revert to a point in time prior to data corruption, you will need to restore the database using the latest available good Snapshot copy. Snap Creator can be used to restore the MySQL database using the Snapshot copies from the NetApp storage system. The Snap Creator restore process is an interactive process that requires user input. The restore process will only list the Snapshot copies for individual volumes that are taken through Snap Creator. This means any other Snapshot copies either taken manually or through some other process on the NetApp storage system will be ignored.

To restore the profile named "mysqldemo" using an earlier Snapshot copy taken through Snap Creator, you would run the following command:

```
snapcreator --profile mysqldemo --action restore --policy daily --verbose
```

The following occurs during a restore operation through Snap Creator:

1. Snap Creator interactive restore menu prompts the restore of the configured volumes.

```
### You have chosen to do a snap restore on one or more volumes for the Config: mysqldemo Policy: daily ###
Are you sure you want to continue (y|n)? y
[Mon Nov 15 18:53:27 2010] INFO: Checking API version on 10.73.69.181

### Volume Menu for 10.73.69.181 ###
01. mysqldata
Select a volume for snapshot restore (enter a number, "n" for next filer, "c" to continue, or "q" to quit): 01
```

   After restoring all volumes, be sure to select "c" to continue instead of "q" to quit which immediately exit Snap Creator without processing any post restore commands that you may have configured.

2. Snap Creator will list the Snapshot copies on the NetApp storage system and the user can select the appropriate Snapshot copy to restore the configured volumes.

```
########## Gathering Information for 10.73.69.181:mysqldata ##########
[Mon Nov 15 18:53:30 2010] INFO: Performing NetApp Snapshot Inventory for mysqldata on 10.73.69.181
[Mon Nov 15 18:53:30 2010] INFO: NetApp Snapshot Inventory of mysqldata on 10.73.69.181 completed Successfully

### Snapshot Menu for 10.73.69.181:mysqldata ###
01. mysqldemo-daily_20101115185304 (Nov 15 2010 18:52:39)
02. mysqldemo-daily_20101115183654 (Nov 15 2010 18:36:29)
Select a snapshot for restore (enter a number or "q" to quit): 02
```

3. Snap Creator prompts for the type of restore. NetApp recommends using Volume Restore.

```
### Restore Menu for 10.73.69.181:mysqldata snapshot mysqldemo-daily_20101115183654 ###
01. Volume Restore
02. File Restore
Select a restore type (enter a number, or "q" to quit): 01
```

4. After all the volumes are restored, Snap Creator will call the POST commands to restart the database. When restoring all volumes, be sure to select "c" to continue instead of "q" to quit which immediately exit Snap Creator without processing any post restore commands that you may have configured.

6.4 SINGLE DATABASE RESTORE WITH SNAP CREATOR

If you are using a block-based protocol such as FCP or iSCSI, you need to install SnapDrive to access single databases within a Snapshot copy. For instruction on how to install

If you are running MySQL on a file-based protocol like NFS, then accessing a Snapshot copy for single file restore is even easier. Each NetApp volume contains a snapshot directory called `.snapshot`, which has a list of Snapshot copy names. Browsing this directory allows you to access the point-in-time files taken during the Snapshot copy.

For example, if you browsed to the `/mnt/domdata` volume mapped on a Linux server, you can browse to the `.snapshot` directory and execute an **`ls`** command, which might return a similar list like the following:

```
[root@node1 .snapshot]# pwd
/usr/local/mysql/.snapshot
[root@node1 .snapshot]# ls -ltr
total 20
drwxr-xr-x 12 root root 4096 Jul 30 23:58 btcppe182(0118056828)_ossvdest.9
drwxr-xr-x 12 root root 4096 Nov 15 14:12 mysqldemo-daily_20101116150209
drwxr-xr-x 12 root root 4096 Nov 15 14:12 mysqldemo-daily_20101116143244
drwxr-xr-x 12 root root 4096 Nov 15 14:12 mysqldemo-daily_20101115190707
drwxr-xr-x 12 root root 4096 Nov 15 14:12 mysqldemo-daily_20101115183654
[root@node1 .snapshot]#
```

Each of the listed subdirectories is a read-only point-in-time Snapshot copy, which can be browsed with NFS just like a file system. Because the files are read-only, you might need to copy the file to an alternate location if you need to change any data. To access the needed file, you can browse the file system at this point as if it is the production file system.

Cloning

We can do either volume/LUN clone for the MySQL database volume. The volume clone flow for MySQL is as follows:

1. Quiesce the databse
2. Check the Snapshot inventory
3. Create the Snapshot copy
4. Create a clone based on the newly created Snapshot copy
5. Unquiesce the database

We don't have the direct cloning of MySQL database to new name in the same database server or new database server. You can create a clone of the MySQL database with a new name in the same database server or a new database server.

To clone the existing MySQL databse in the same database server, do the following:

1. Mount/map the cloned voume to the temp folder.
   `[root@node1 src]# df -h`

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda10             17G   16G   12M 100% /
tmpfs                 4.0G  1.9G  2.1G  48% /dev/shm
10.73.69.181:/vol/mysqldata
        40G  2.9G   38G   8% /usr/local/mysql
10.73.69.181:/vol/oraclepoc/data
              56G   14G   43G  24% /orapocdata
10.73.69.181:/vol/oraclepoc/fra
              56G   14G   43G  24% /orapocfra
10.73.69.181:/vol/cl_mysqldata_20101116150209
               40G  3.0G   38G   8% /usr/local/mysqlnew
[root@node1 src]#
```

2. Rename the existing database folder in the cloned volume.

[root@node1 src]#cd /usr/local/mysqlnew/data

[root@node1 src]#mv mysqldemo mysqldemonew

[root@node1 src]# mysql -u sqladmin -h 10.73.69.111 -pbtcppe1 --port=3306

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.1.46sp1-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| db2                |
| mysql              |
| mysqldemo          |
| mysqldemonew       |
| oowmysql           |
+--------------------+
6 rows in set (0.03 sec)

mysql>
```

3. Create a soft link from the newly created database folder to the old database name in the existing database mount point.

```
ln -s /usr/local/mysqlnew/data/mysqldemonew /usr/local/mysql/data/mysqldemonew
```

4. Grant the required access to the new database(`mysqldemonew`).

[root@node1 src]# mysql -u sqladmin -h localhost -pbtcppe1 --port=3306

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.1.46sp1-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> grant all on mysqldemonew.* to sqladmin@'node1.btcppe.netapp.com';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

5. Check the new database.

[root@node1 data]# mysql -u sqladmin -h localhost -pbtcppe1 --port=3306

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.1.46sp1-log Source distribution
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mysqldemonew
Database changed
mysql>
mysql> create table atest (sl int, name varchar(40));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into atest values(1,'testuser');
Query OK, 1 row affected (0.01 sec)

mysql> insert into atest values(2,'testuser2');
Query OK, 1 row affected (0.00 sec)
mysql> select * from atest;
+------+-----------+
| sl   | name      |
```

```
+------+-----------+
|    1 | testuser  |
|    2 | testuser2 |
+------+-----------+
2 rows in set (0.00 sec)

mysql> show tables;
+------------------------+
| Tables_in_mysqldemonew |
+------------------------+
| atest                  |
| crashmysql1            |
| crashmysql10           |
| crashmysql100          |
| crashmysql101          |
+------------------------+
5 rows in set (0.01 sec)
```

6. To clone the existing MySQL database do the following:
   a. Mount the cloned volume to the MySQL database "datadir" location.
   b. Grant access to the database

7 QUESTIONS AND ANSWERS

How can I get exclusive control over MySQL?

When restoring a database or performing other system-wide maintenance, you may want to have exclusive control over the server to block all other users from having access. You can do this by stopping the MySQL server and then restarting it:

```
mysqld-nt --enable-named-pipe \
--socket=MySQL_temp --skip-networking
```

On Windows systems that are not NT-based, named pipes are not available, and therefore you must use TCP/IP. In this situation you may want to start the server by using the `--port` option with a port other than 3306. You can use any nonprivileged port that is not used by other services.

The `--skip-networking` option prevents users from accessing the server via TCP/IP and the socket file or named pipes. It permits connections only from the local host. This still allows access from a console physically on the server or through a secure shell connection. This option with a different socket name prevents outsiders and API scripts on the server from being able to access the database until you're able to restore the data.

Once you have the server under your exclusive control, you can proceed without worrying about other users trying to access the server while you're recovering data or performing other maintenance. To access the server from the mysql client, you can use the following example command:

```
mysql -u root -pmypwd --socket=/tmp/mysql_temp.sock
```

To restore a `mysqldump` file while access is restricted, you can use the following example command:

```
mysql -u root -pmypwd --socket=/tmp/mysql_tmp.sock \
< /var/backup/20050420.sql
```

When you finish working on the server, you can give access to other users again by restarting the server as you normally would, but without the `--skip-networking` option.

If MySQL is running on Linux kernel 2.6 with a two-CPU machine, can MySQL be configured to run on only one of the CPUs?

Starting with Linux kernel 2.6, the `taskset` command, from the `schedutils` package, can be used to do this. To check to see if the package is installed on your Linux server, enter:

```
rpm -q schedutils
```

```
schedutils-1.4.0-4
```

To install `schedutils`, enter:

```
yum -y install schedutils
```

Once this package is installed, to set the process for MySQL to run on the first CPU, first use the `ps` command to determine the process ID (PID):

```
ps aux | grep mysql
```

Using the PID from the results (for example, 1000), you can then enter the following command to limit the MySQL process to the first CPU:

```
taskset -p 0x1 1000
```

You have to do this for each MySQL process. To limit all processes associated with the MySQL server, enter:

```
taskset 0x1 mysqld_safe
```

```
taskset 0x1 mysqld
```

What can I set up point-in-time recovery?

You can set up a server to back up data regularly by using utilities such as `mysqldump`, but they do not allow you to recover data lost since the last backup. To be able to recover data lost from any point in time, you need to set up binary logging.

Binary logging records in a binary file all SQL transactions executed and attempted on the server. The contents of the binary log file can be extracted by using the `mysqlbinlog` utility, so that the SQL statements can easily be rerun. To enable binary logging, add the following line to your server's options file (that is, `/etc/my.cnf` or `c:\my.ini`, depending on your system) in the [mysqld] group:

```
log-bin = /var/log/mysql/bin.log
```

The exact path to use depends on your file system and your preferences. You may have to create the directory where logs are to be stored and change the ownership and permissions for the directory on your file system accordingly. NetApp recommends that the path where you back up data and keep log files should be on a separate hard drive for added safety. The suffix log in the file name above will be replaced automatically with a six-digit number. You can limit binary logging to specific databases, or you can omit certain databases from the log. After making the entry to your options file, restart the MySQL server for it to take effect.

In addition to making regular backups with something like `mysqldump`, you may want to synchronize the starting of binary logs with the backups. This can be done by flushing the logs when you run a backup. You can add something like the following line to `cron` or a similar scheduling utility:

```
mysqladmin -u root -pmypwd flush-logs
```

In this line, the `mysqladmin` utility is used to flush the server's logs. You could instead use the mysql client to execute the `FLUSH LOGS` statement.

How can I recover all of my data up to now?

With daily backups and the binary logs running, it is a simple, although sometimes tedious, procedure to recover data on a MySQL server. If you do not have binary logs running, then you may not be able to recover the data since the last backup

The following example describes how to recover data by using binary logs. Suppose that your data is backed up each day at midnight using `mysqldump`, and that on a certain day (say, at 10 a.m. on May 20, 2008) your data is lost and you want to recover it. To begin, you may want to stop the MySQL server and then restart it in such a way that you are the only user to have access to it. This can be done by entering:

```
mysqld-nt --enable-named-pipe \
--socket=MySQL_restore --skip-networking
```

On Windows systems that are not NT-based, named pipes are not available, and therefore you must use TCP/IP. In this situation you may want to start the server by using the `--port` option with a port other than 3306. You can use any nonprivileged port that is not used by other services.

The `--skip-networking` option prevents users from accessing the server via TCP/IP and the socket file or named pipes. It permits connections only from the local host. This still allows access from a console that is physically on the server or through a secure shell connection. This option with a different socket name prevents outsiders and API scripts on the server from being able to access the database until you are able to restore the data.

Once you have the server under your exclusive control, you can proceed without worrying about more problems occurring from users trying to access the server while you are recovering the data. The first step is to restore the dump file from the previous night's backup:

```
mysql -u root -pmypwd --socket=/tmp/mysql_restore.sock \
< /var/backup/20080520.sql
```

The names of the paths and files will be different for your server. The line here restores the data as of the start of the day. To restore the transactions since the dump file was created, you use the `mysqlbinlog` utility. If you are flushing the logs each day when making your nightly backups, you would restore the entire binary log file from the command line:

```
mysqlbinlog /var/log/mysql/bin.123456 \
   | mysql -u root -pmypwd \
          --socket=/tmp/mysql_restore.sock
```

In this example, the results of `mysqlbinlog` are piped (notice the vertical bar) to the MySQL client for processing. When the processing is completed, restart the MySQL server without the temporary socket file and network restriction.

These steps restore the data as it stood at the time of the previous backup, and then all SQL statements that were entered since the logs were flushed are reentered. To determine the name of the latest binary log file, you can get a list of the directory that contains the logs and look for the file with the largest numeric suffix. If you have restarted the server, you must subtract 1 from the suffix of the log name because restarting starts a new log just like flushing the logs.

How does mysqlhotcopy work?

The `mysqlhotcopy` utility is a Perl script that uses several basic system and SQL commands to back up a database. Specifically, it locks and flushes the tables, makes a copy, and then unlocks the tables. Although this is the fastest method available for backing up a MySQL database, it is limited to backing up only those databases that reside on the same machine on which it is executed.

The `mysqlhotcopy` function can be executed to back up one database, a number of databases, or only those databases that match a name specified by a regular expression. In this section, the syntax involved with each scenario is provided, followed by a few examples.

- Using `mysqlhotcopy` to back up just one database:

  `mysqlhotcopy [options] db_name /path/to/new_directory`

- Using `mysqlhotcopy` to back up several databases:

  `%>mysqlhotcopy [options] db_name_1 ... db_name_n /path/to/new_directory`

- Using `mysqlhotcopy` to back up only those tables in a given database that match a regular expression:

  `mysqlhotcopy [options] db_name./regex/`

- The options can be viewed by executing the following command:

  `mysqlhotcopy --help`

Examples:

Experiment with `mysqlhotcopy` by backing up the widgets database to the `/usr/mysql/backups/` directory path. Execute the following command:

`mysqlhotcopy -u root -p widgets /usr/mysql/backups`

As a second example, assume that the widgets database contains the tables "`products2000`", "`products2001`", "`clientele2000`", and "`clientele2001`", with the four digits at the end of each name representing the year that data represents. The administrator wants to back up only those tables relative to the year 2000:

`mysqlhotcopy -u root -p widgets./^.+('2000')$/ /usr/mysql/backups`

In this example, the regular expression `/^.+('2000')$/` tells `mysqlhotcopy` to back up only those tables ending with the `2000` string.

How can I avoid inserting duplicate rows from a dump file?

When restoring a dump file generated by `mysqldump`, there are sometimes problems with duplicate rows. This can happen if a dump file is being restored over existing data while attempting to restore deleted rows. A problem arises when running the restore because a mutliple row `INSERT` is stopped; an error message like the following is displayed:

```
ERROR 1062 (23000): Duplicate entry '100' for key 1
```

The result of this error depends on the storage engine being used. When a duplicate key error occurs while performing a multirow insert into a MyISAM table, all rows up to the row that caused the duplicate key error are inserted into the table. When such an error occurs while performing a multirow insert into an InnoDB table, the statement is aborted and none of the rows are inserted into the table. This is because MyISAM is not able to roll back the portion of the statement that has already been processed. InnoDB is a transactional storage engine, so it is able to roll back the statement and therefore avoid inserting any of the rows.

Starting with `mysqldump` version 10.9, distribution 4.1.12, the `--insert-ignore` option may be given when creating the dump file. This adds the `IGNORE` keyword to the `INSERT` statements. The `IGNORE`

keyword tells MySQL to skip duplicate rows, to supress error messages, and to continue inserting rows of data where the script would have otherwise been aborted.

If you have an earlier version of `mysqldump` and don't want to upgrade, you can edit the dump file instead. One way is to use a programming language like Perl. Enter the following from the command line:

```
perl -p -i.bak -e 's/INSERT INTO/INSERT IGNORE INTO/;' backup.sql
```

This replaces all occurences of `INSERT INTO` with `INSERT IGNORE INTO` in the dump file called `backup.sql`. If you don't have Perl installed on your server, you could use the `replace` utility, which comes with the standard distribution of MySQL. For example:

```
replace 'INSERT INTO' 'INSERT IGNORE INTO' -- backup.sql
```

Before using either of these methods, you may want to make sure that your data does not contain the string `INSERT INTO`, because that will be modified along with the DDL statements.

Why is the Backup Screen Disabled in MySQL Administrator?

MySQL Administrator can be used to perform database backups. It produces text backups that are compatible with the `mysqldump` utility.

When a user first clicks the Backup menu item, the Backup Project screen is disabled and dimmed.

The Backup Project screen is disabled unless there is an active project. Either choose an existing backup project from the list at the bottom left corner of the MySQL Administrator screen or click the New Project button at the bottom right side of the MySQL Administrator screen.

How do I restore a dump file?

If you have created a backup of your database by using `mysqldump`, you can use the created dump file to restore the data. The result of a dump file is a simple text file. If you open the file with a text editor, you see `CREATE DATABASE` statements for each database and `CREATE TABLE` statements for each table. These are followed by an `INSERT` statement for each row of data.

To stop `mysqldump` from entering `CREATE DATABASE` statements in the dump file, add the `--no-create-db` option. The `--no-create-info` option prevents the `CREATE TABLE` statements from being generated. To compress the various separate `INSERT SQL` statements into one `INSERT` statement with multiple values, use the `--extended-insert` option.

After the dump file is created, you can edit it manually to remove any statements that you don't want to run. An easier method is to restore a database to a temporary location on a workstation and then selectively move it to your live server and data directory.

By executing a dump file with a client like MySQL, the databases and tables can be recreated and the data inserted as follows:

```
mysql -u root -ppassword < /tmp/backup.sql
```

In this example, the standard input redirect (that is, the less-than sign) is used instead of the standard output redirect. It tells MySQL to accept input from the `backup.sql` dump file in the `/tmp` directory and not from the keyboard. This means that you do not enter the MySQL monitor and instead remain at the command line.

You can redirect the standard output to a file so that you have a record on disk of any errors or warnings produced in case you want to examine them later. You can do this by adding `> /tmp/output`.

8 APPENDIX: FCP AND ISCSI PROTOCOL SETTINGS FOR MYSQL

8.1 NETAPP STORAGE AND LINUX SERVER SETTINGS FOR ISCSI

NETAPP STORAGE SETTINGS

1. NFS tcpwindowsize option setting:

   BTCPPE-FILER-21*> options nfs.tcp.recvwindowsize 262144

2. Install iscsi-initiator-utils.
3. Find the iSCSI initiator name in Linux by entering:

   [root@zmanda 101]# iscsi-iname

   ```
   iqn.1994-05.com.redhat:3b93bdd7ccf
   ```

4. Add or modify the following parameters in `/etc/iscsi/initiatorname.iscsi`:

   `InitiatorName=iqn.1994-05.com.redhat:3b93bdd7ccf`

   `InitiatorAlias=RHEL5U1`

5. Check the iSCSI license in the storage system and start the iSCSI service by entering:

   BTCPPE-FILER-22> license

   ```
   Iscsi xxxxxxx
   ```

   ```
   BTCPPE-FILER-22> iscsi start
   ```

   ```
   Tue Apr  1 04:05:39 PDT [BTCPPE-FILER-22: iscsi.service.startup:info]: iSCSI
   service startup
   iSCSI service started
   BTCPPE-FILER-22> iscsi status
   iSCSI service is running
   ```

6. Get the target name from the storage system by entering:

   BTCPPE-FILER-22> iscsi nodename

   ```
   iSCSI target nodename: iqn.1992-08.com.netapp:sn.118053173
   ```

7. Add or modify the following parameter in `/etc/iscsi/iscsid.conf`:

   ```
   DiscoveryAddress=10.73.68.22:3260
   TargetName=iqn.1992-08.com.netapp:sn.118053173
   ```

8. Create the igroup in the storage system by entering:

   BTCPPE-FILER-22> igroup create -i -t linux iscsimysql

9. Add the Linux node with the igroup by entering:

   BTCPPE-FILER-22> igroup add iscsimysql iqn.1994-05.com.redhat:3b93bdd7ccf

   BTCPPE-FILER-22> igroup show

   ```
   iscsimysql (iSCSI) (ostype: linux):
   iqn.1994-05.com.redhat:3b93bdd7ccf (not logged in)
   ```

10. Restart the iSCSI client service from Linux by entering:

    [root@zmanda ~]# /etc/init.d/iscsi restart

```
Stopping iSCSI daemon: /etc/init.d/iscsi: line 33: 19441 Killed
/etc/init.d/iscsid stop
iscsid dead but pid file exists                            [  OK  ]
Turning off network shutdown. Starting iSCSI daemon:       [  OK  ]
                                                           [  OK  ]
Setting up iSCSI targets: Login session [iface: default, target: iqn.1992-
08.com.netapp:sn.118053173, portal: 10.73.68.22,3260]
                                                           [  OK  ]
```

You can see the login session while restarting iSCSI in Linux.

11. Check the igroup in the storage system to check the login session by entering:

    BTCPPE-FILER-22> igroup show

```
iscsimysql (iSCSI) (ostype: linux):
iqn.1994-05.com.redhat:3b93bdd7ccf (logged in on: e0a)
```

12. Enable the iSCSI login while rebooting by using **chkconfig iscsi**:

```
discover the target name
```

13. Create a volume for the LUN by entering:

    BTCPPE-FILER-22> vol create mysqliscsi aggr1_22 30g

```
Creation of volume 'mysqliscsi' with size 30g on containing aggregate
'aggr1_22' has completed.
```

14. Create a LUN by entering:

    BTCPPE-FILER-22> lun create -s 23g -t linux /vol/mysqliscsi/lun1

    BTCPPE-FILER-22> lun show

```
/vol/mysqliscsi/lun1          23g (24696061952)   (r/w, online)
```

15. Map the LUN with the igroup by entering:

    BTCPPE-FILER-22> lun map /vol/mysqliscsi/lun1 iscsimysql 5

    BTCPPE-FILER-22> lun show

```
/vol/mysqliscsi/lun1          23g (24696061952)   (r/w, online, mapped)
```

The number 5 should be a unique number.

LINUX SERVER SETTINGS

1. Check the LUN availability in Linux by entering:

   [root@zmanda ~]# /etc/init.d/iscsi restart

```
Logout session [sid: 1, target: iqn.1992-08.com.netapp:sn.118053173, portal:
10.73.68.22,3260]
Stopping iSCSI daemon: /etc/init.d/iscsi: line 33: 19522 Killed
/etc/init.d/iscsid stop
iscsid dead but pid file exists                            [  OK  ]
Turning off network shutdown. Starting iSCSI daemon:       [  OK  ]
                                                           [  OK  ]
Setting up iSCSI targets: Login session [iface: default, target: iqn.1992-
08.com.netapp:sn.118053173, portal: 10.73.68.22,3260]
                                                           [  OK  ]
[root@zmanda ~]# fdisk -l

Disk /dev/sda: 146.5 GB, 146576244736 bytes
```

```
255 heads, 63 sectors/track, 17820 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14         536     4200997+  82  Linux swap / Solaris
/dev/sda3             537        7064    52436160   83  Linux
/dev/sda4            7065       17819    86389537+   f  W95 Ext'd (LBA)
/dev/sda5            7065        9674    20964793+  83  Linux
/dev/sda6            9675       10197     4200966   82  Linux swap / Solaris
/dev/sda7           10198       10210      104391   83  Linux

Disk /dev/sdb: 24.6 GB, 24696061952 bytes
64 heads, 32 sectors/track, 23552 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdb doesn't contain a valid partition table
```

2. Create the partition and file system on the newly created LUN by entering:

[root@zmanda ~]# fdisk /dev/sdb

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-23552, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-23552, default 23552):
Using default value 23552
Save the partition table.
[root@zmanda ~]# partprobe
[root@zmanda ~]# mkfs.ext3 /dev/sdb1
```

3. Mount the `/dev/sdb1` file system in the `/var/lib/mysqldata` folder and start the MySQL server.

8.2 NETAPP STORAGE AND LINUX SERVER SETTINGS FOR FCP

1. The HBA drivers are loaded in the Linux server. If they are not loaded, download them from http://support.qlogic.com/support/drivers_software.aspx?id=m10#.

[root@zmanda ~]# lsmod | grep -i ql

```
qla2xxx              1008556  0
qla2xxx_conf          335368  1
intermodule            37764  2 qla2xxx,qla2xxx_conf
scsi_mod              186489  9
ib_iser,iscsi_tcp,libiscsi,scsi_transport_iscsi,qla2xxx,sg,libata,aacraid,sd_mod
```

2. Get the driver information (World Wide Port Name) for the HBA card by entering:

[root@zmanda ~]# cat /proc/scsi/qla2xxx/3

```
SCSI Device Information:
scsi-qla1-adapter-node=200100e08bbb976c;
scsi-qla1-adapter-port=210100e08bbb976c;
```

3. Set the priv into advanced mode in storage by entering:

BTCPPE-FILER-22> priv set advanced

4. Offline the Fibre Channel Adapter 0.

BTCPPE-FILER-22*> fcadmin config -d 0c

Change the 0c adapter to target mode

BTCPPE-FILER-22*> fcadmin config -t target 0c

```
Wed Apr  2 01:01:07 PDT [BTCPPE-FILER-22: fci.config.state:info]: Fibre channel
initiator adapter 0c is in the PENDING (target) state.
A reboot is required for the new adapter configuration to take effect.
```

5. Reboot the storage system by entering:

   BTCPPE-FILER-22*> reboot -t 0

6. Set the CF mode to be same in both storage systems by entering:

   BTCPPE-FILER-22*> fcp set cfmode single_image

7. Start the FCP services in both storage systems by entering:

   BTCPPE-FILER-22*> fcp start

```
Wed Apr  2 02:15:16 PDT [BTCPPE-FILER-22: fcp.service.startup:info]: FCP service
startup
Wed Apr  2 02:15:16 PDT [BTCPPE-FILER-22: scsitarget.ispfct.onlining:notice]:
Onlining Fibre Channel target adapter 0c.
BTCPPE-FILER-22*> Wed Apr  2 02:15:16 PDT [BTCPPE-FILER-22:
scsitarget.ispfct.linkUp:notice]: Link up on Fibre Channel target adapter 0c.
Wed Apr  2 02:15:16 PDT [BTCPPE-FILER-22: scsitarget.ispfct.lipReset:notice]: FCP
Target 0c: LIP Reset from AL_PA 0x0 (WWPN unknown)
```

8. Create a volume of size 40g in the `privault` aggregate by entering:

   vol create fcpmysql privault 40g

9. Create a LUN of size 38G in the `fcpmysql` volume by entering:

   BTCPPE-FILER-22> lun create -s 38g -t linux /vol/fcpmysql/lun1

```
lun create: created a LUN of size:   38.0g (40806383616)
```

10. Get the WWPN by entering:

    BTCPPE-FILER-22> fcp show initiator

```
Initiators connected on adapter 0d:
Portname                 Group
21:00:00:e0:8b:9b:97:6c
```

11. Create an igroup of Linux type for FCP with the WWPN number by entering:

    BTCPPE-FILER-22> igroup create -t linux -f fcpgroup 21:00:00:e0:8b:9b:97:6c

12. Check the FCP connection status, whether the Linux server logged in through the 0d port, by entering:

    BTCPPE-FILER-22> igroup show

```
fcpgroup (FCP) (ostype: linux):
21:00:00:e0:8b:9b:97:6c (logged in on: 0d)
iscsimysql (iSCSI) (ostype: linux):
iqn.1994-05.com.redhat:3b93bdd7ccf (logged in on: e0a)
```

    If the WWPN of the Linux server is not logged in, reload the driver in Linux and then check again.

13. Map the LUN with the FCP igroup by entering:

```
BTCPPE-FILER-22> lun map /vol/fcpmysql/lun1 fcpgroup 7
```

14. Either reload the HBA driver (**modprobe –vr qla2xxx**) or reboot the Linux box to view the new LUN in Linux.
15. Create the partition and file system on the new LUN by entering:

```
fdisk /dev/sdc'

mkfs.ext3 /dev/sdc1
```

16. Check whether the LUN mapped and online to the igroup in the storage controller by entering:

```
BTCPPE-FILER-22> lun show
```

```
/vol/fcpmysql/lun1          38.0g (40806383616)   (r/w, online, mapped)
```

17. Check the new LUN by mounting on the Linux server using the `mount` command:

```
[root@zmanda ~]# mount /dev/sdc1 /var/lib/mysql

[root@zmanda ~]# df -HT
```

```
Filesystem    Type     Size   Used  Avail Use% Mounted on
/dev/sda3     ext3      53G   6.4G    43G  13% /
/dev/sda1     ext3     104M    19M    80M  19% /boot
tmpfs        tmpfs     2.1G      0   2.1G   0% /dev/shm
/dev/sdc1     ext3      41G   185M    38G   1% /var/lib/mysql
```

9 ACKNOWLEDGEMENTS

The authors would like to thank the following individuals for their direction and guidance in creating this paper:

Daniel Morgan, Director, Solution Technologies, NetApp

Uday Shet, Manager, NB PPE (Solutions Engineering), NetApp

Michelle Nguyen, Senior Product Manager, Enterprise Applications, NetApp

Michael B. Flannery, Field Technology Lead, NetApp

Dean Brock, Database Performance Engineer, NetApp

10 REFERENCES

- Online MySQL Backup Using NetApp Snapshot Technology: http://media.netapp.com/documents/tr-3601.pdf
- MySQL Backup and Restore Using Zmanda Recovery Manager and NetApp Snapshot Technology: http://media.netapp.com/documents/tr-3656.pdf
- NetApp Best Practice Guidelines for Oracle: http://media.netapp.com/documents/tr-3369.pdf
- MySQL Enterprise Edition: http://enterprise.mysql.com

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

www.netapp.com