



EMC NETWORKER SCRIPT-BASED BACKUP



Upanshu Singhal
Consultant Engineer
EMC Corporation
Upanshu.singhal@emc.com

Asif Khan
QA Lead
EMC Corporation
Asif.khan@emc.com

Satyam Chikkam
QA lead
EMC Corporation
chikkam.satyam@emc.com

Kiran Dudala
QA Lead
EMC Corporation
Kiran.dudala@emc.com

Abstract	3
Introduction.....	5
What is NetWorker Software?	5
NetWorker building blocks	6
NetWorker architecture concepts	7
NetWorker support for Database applications	8
NetWorker module advantages.....	9
NetWorker Unassisted Modules.....	9
How can we back up non-supported application backups?	10
NetWorker PRE/POST command method.....	10
NetWorker supported script based approach	11
Performing backups using script-based approach.....	13
How to choose a scripting language	13
Identify the backup and recovery methods provided by database application – MySQL.....	13
Making Delimited-Text File Backups	14
Making Backups with mysqldump or mysqlhotcopy.....	14
Making Incremental Backups by Enabling the Binary Log.....	14
Making Backups Using Replication Slaves.....	15
Perform Recovery using MYSQL commands	15
Example of what happens if we use just file system backup of MySQL DB files.....	15
mysqldump and mysqlhotcopy	15
Full and incremental methods of database application	16
MySQL database backup using Script with NetWorker	16
Guide for using MySQL backup script integrated with NetWorker	17
MySQL requirements	17
Binary logging to be enabled in my.cnf file for Incremental backup	17
NetWorker requirements	17
NetWorker save group and NetWorker client resource to be configured per the given guidelines	17
Operating System requirements.....	17
NetWorker client – Linux Server.....	17
Initial POC was done with MySQL database application on Linux Server.....	17
NetWorker server – Any Operating System supported by NetWorker	17
NetWorker storage node – Any Operating System supported by NetWorker.....	17
Full and incremental backup	18
Recovery	22
Types of Recovery	22
MySQL Recover commands	22
Procedure for Data Recovery.....	23
What we get free as part of script-based approach	23
Challenges.....	24
Conclusion.....	25

Disclaimer: The views, processes, or methodologies published in this article are those of the author. They do not necessarily reflect EMC Corporation's views, processes, or methodologies.

Abstract

EMC NetWorker® is the best backup and recovery software for a heterogeneous environment because of its support for large number of operating systems, database applications, number of backup devices, and more. Sometime customers do not consider it in spite of its large application support. The reason? NetWorker does not provide support for database applications such as MySQL, PostGreSQL or a latest version of an existing database application. However, NetWorker can actually perform scheduled hot backups for these applications with a simple script or an executable binary. Customers can perform the recovery with the existing NetWorker framework without any additional script or binary.

NetWorker provides a method to execute any script or commands before and after any backup. This is called Pre and Post script method in NetWorker and is executed during a scheduled backup. Pre script method prepare the database application for hot backup and post script method can be used for sending backup completion status to database application and cleanup of some temporary information generated during backup. Pre and post methods can even enable users to perform snapshot-based backups of a database application, resulting in less disruption to continuous database I/Os and faster backup window.

This Knowledge Sharing article will take you on the journey of performing non-supported database application backup and recovery using NetWorker, using MySQL databases as an example.

This article will provide an introduction to:

- Choosing a scripting language
- Describing the backup procedure suggested by database application – MySQL
- Developing script for full and incremental backup
- Configuring scheduled NetWorker backup with script
- Restoring backup using NetWorker command line or User interface
- Restoring data to database application using database application methods – MySQL
- Snapshot-based backup method with NetWorker pre and post methods

Benefits of this article:

- Help customers perform backup and recovery for database applications not supported as integrated NetWorker module
- Help professional services write a customized solution for customers for the application like MySQL without waiting on engineering to deliver an integrated module
- Single point of reference to configure backup and recovery using NetWorker pre and post methods
- Self-learning with very minimal effort and time required of others

- Additional procedures can be updated to this document instead of re-writing the new procedures in a different document. All changes to the document can be maintained by versions of this document in order to track the changes.
- Document can easily be integrated into the Procedure Generator Tools and can be made available via the Web

While Engineering personnel are the intended audience for this article, it can also be helpful to end users as a “Quick user guide” for quick solutions on non-supported database applications.

Introduction

What is NetWorker Software?

NetWorker is a network backup and recovery application that reproduces online filesystem data at a protected location and maintains location- and obsolescence-tracking information about the data. NetWorker can then re-create the data if the online version is inadvertently changed, lost, or corrupted. NetWorker features include a storage management application that directs high performance writes to a range of storage devices, either local or remote. Supported storage devices include:

- Tape devices (including autochangers)
- Optical disk devices (including autochangers)
- Online disks
- Data Domain[®]
- Avamar[®]

As data is written to storage, NetWorker tracks the data by volume and by the specific location where it is written on the volume. This speeds up retrieval when required. An administrator can configure NetWorker so that during backup, data is directed onto specific pools of volumes. Pools can be configured based on a variety of data characteristics—data from similar clients can be written to the same pool, or data backed up at the same level. After data is written to a volume, NetWorker maintains information about the age of the data and automatically limits its recoverability, even eventually recycling the volume, based on administrator-defined policies.

When data recovery is needed, users and administrators can browse an index of recoverable files, create reports that describe the status of data or the contents of the volumes, and recover data to a user-specified point in time. In response to a recovery request, the NetWorker storage management system locates the volume that contains the data and either directs a device to automatically mount the appropriate volume, or sends a message requesting the volume by name. Data can be browsed on a different platform using command line interfaces (though not with GUI). With NetWorker 6.0 and later, most data can also be recovered across different platforms.

If a site experiences a disaster, NetWorker can re-create all NetWorker client and server filesystems to their original structure (as they existed at the time they were written to storage).

NetWorker building blocks

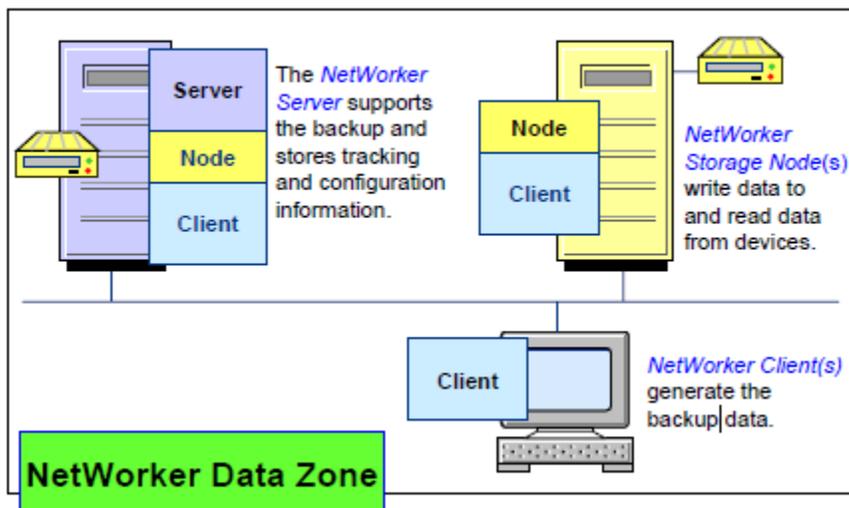
NetWorker consists of 4 tier architecture:

- NetWorker Management Console (NMC) server
 - Monitor and Configure NetWorker server
 - Collect reporting data and generates reports
 - Can connect to one or more NetWorker server
- NetWorker Server
 - Single point of control, can be managed by one or more NMC Servers
 - Hosts all NetWorker internal databases
 - Start scheduled jobs and sends on-demand notifications
 - Controls all target devices
- NetWorker Storage Node
 - Directs backup data to a target device
 - Can accept local data to avoid TCP network transfers or data over TCP network
- NetWorker client
 - Runs agent and start jobs as requested by NetWorker server
 - Jobs can also be started manually
 - Application modules are installed on top of NetWorker client
- NMC Server, NW Server, NW storage node and NW client do not have to be on the same version
 - Recommended that Storage node is same version as NW server
 - Long term compatibility for obsolete clients
- Command line interface
 - Administrators can use number of commands to monitor and configure NetWorker.
 - Nsradmin command line program performs all the configuration functions that are found with NMC Server.
 - Save/recover commands can be executed from command line to backup and recover data.
 - An administrator can initiate all NetWorker functions either from within NMC server or from the command line.
- Some NetWorker supported operating systems
 - Microsoft Windows
 - Sun Solaris
 - RedHat Linux
 - IBM AIX
 - HP-UX and Tru64

NetWorker architecture concepts

- Co-existence
 - Each of 4 components can co-exist
 - Small environment can have NMC, NW server, and NW storage node on a single system
 - For large environments, it is recommended to have NMC server separate and use NW server for server tasks only. Use NW storage nodes for actual data movement.
- RPC over TCP
 - NetWorker uses RPC over TCP for most inter-process communication
 - NW does not use static TCP port numbers except for general listener: 7937-7938
 - All other processes are registered in RPC portmapper upon startup and can be anywhere in allowed port range
 - NW can use existing portmapper if one exists or internal one (embedded in nsrexecd)
 - RPC sessions are authenticated upon creation
 - GSS RPC in NW 7.3 and higher
 - AUTH LGTO in NW 6.0 and higher
 - AUTH UNIX in pre NW 6.0
- Direct communication model
 - Each NW process contacts target process or daemon directly thus avoiding bottleneck of tunnel-typed communications
- NetWorker supports IPv4 and IPv6 protocols

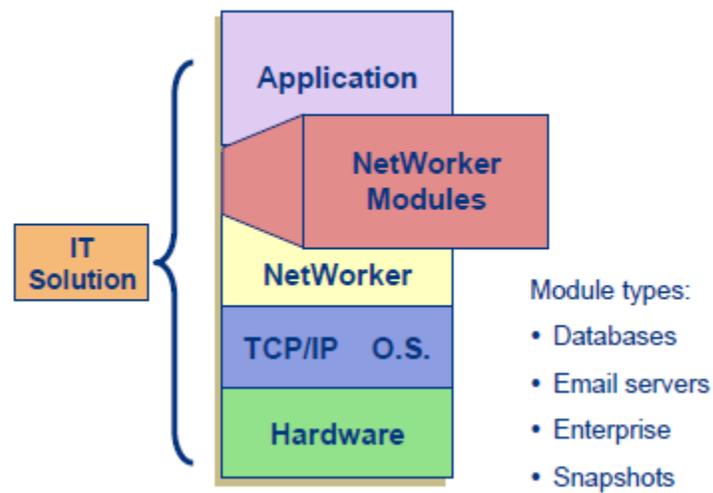
NetWorker Hosts



NetWorker support for Database applications

- NetWorker provides support for several database applications, few of them are:
 - Microsoft Exchange Server
 - Microsoft SQL Server
 - Microsoft SharePoint Server
 - ORACLE
 - IBM Lotus Notes
 - IBM Sybase
 - DB2
 - SAP
- All applications are supported by a separate NetWorker module
- NetWorker Application Modules are generally installed on top of NetWorker client
- NetWorker application module provide integration for online backup of target application
- NetWorker module Application integration is done according to vendor specifications
 - Examples:
 - Oracle: Requires that media manager layer (NW) implements specific APIs and registers itself with Oracle
 - Exchange: Requires that backup vendor uses API for VSS provided by OS
- Backup configuration is application specific and can be configured using NMC GUI Wizards
- Recovery is application-specific and data can be configured using NetWorker module GUI

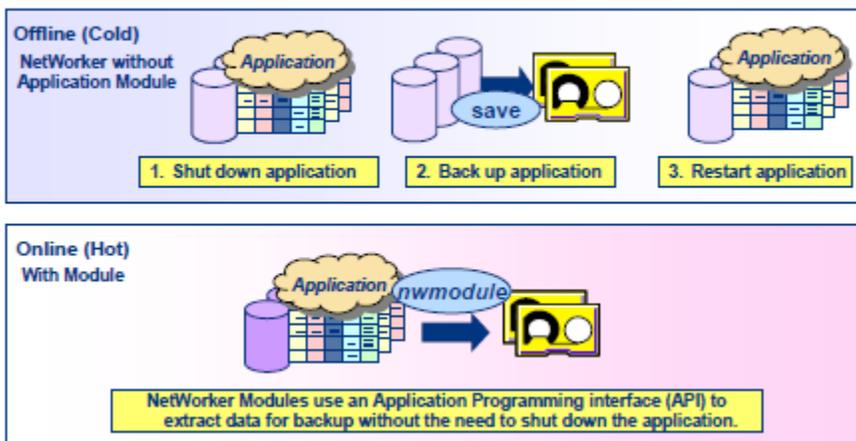
The NetWorker Module Solution



NetWorker module advantages

- An application can be backed up offline or online. NetWorker alone can back up closed applications as flat files (called a cold backup). Procedure for cold backup is:
 - The application is shut down prior to being backed up.
 - The application data is backed up as flat files
 - The application is restarted after the backup completes.
- NetWorker modules support online or hot backup and do not require taking down the application prior to backing it up.
 - Module uses application APIs to extract data for backup without the need to shut down the application. The application remains open for transactions during backup.
 - Some of the interfaces for online backups are
 - RMAN for ORACLE
 - Microsoft Volume Shadow Copy Service for Microsoft Applications

NetWorker Module Advantage: Online Backups



NetWorker Unassisted Modules

EMC NetWorker backup and recovery solution does not provide NetWorker module for few database applications, e.g. MYSQL database application. The support is not provided for few reasons, such as:

- Less customer base for the particular database application
- Lack of ask from customers
- A very new application or application version which has less adoption
- High development cost against the revenue.

This article will give detailed information about how to perform backup and recovery using NetWorker for Databases which are not available as NetWorker modules.

How can we back up non-supported application backups?

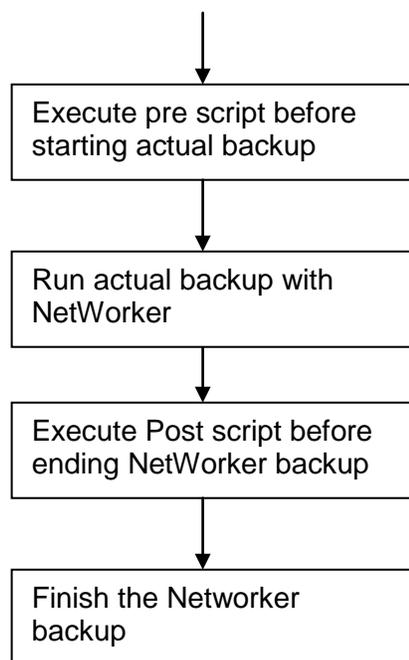
NetWorker has a feature called Pre and Post script method which can help execute certain commands before backing up files and after completion of backup. Additionally, this method can be exploited to back up a database application which is not available as NetWorker module.

NetWorker PRE/POST command method

A standard program on NetWorker clients—"savenpc"—will execute a named script prior to the client backup (pre-processing) and execute another named script after the client backup (post-processing). The scripts must be named in a configuration file resident on the client named /nsr/res/groupname.res where groupname is the name of the group to which the client belongs on the server. The format of the groupname.res file is:

```
type: savenpc;  
precmd: /somedir/some_pre_script;  
postcmd: /somedir/some_post_script;  
timeout: "4:00am";
```

This timeout value is optional and if backup has not completed by the given time, savenpc will then execute the post command. Work flow with savenpc is:



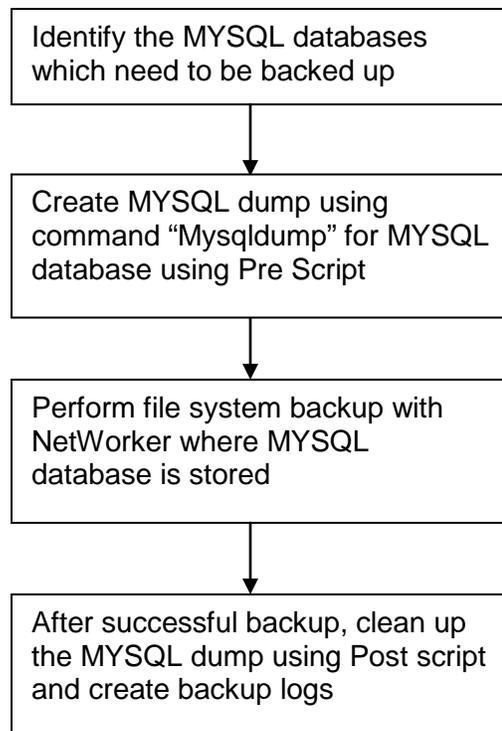
NetWorker Pre/Post Script FlowChart

NetWorker-supported script-based approach

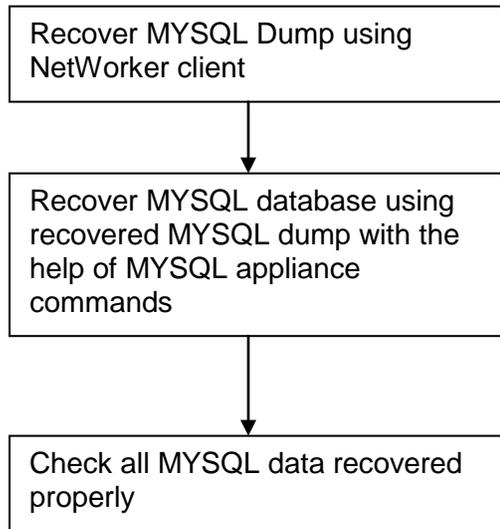
NetWorker can perform backup of a database application via a script as well. For example, a script was developed to perform hotline backup for MySQL databases using a script with NetWorker standard framework.

Upon execution of MySQL backup script, NetWorker identifies the databases which need to be backed up to a target device. Backup script uses MYSQL commands to create a consistent database dump in a file or pipes of MySQL databases. NetWorker will back up the file and clean up the dump after backup completion.

NetWorker provides support for recovery of MySQL databases. In order to restore MySQL databases, user is required to first restore the databases from NetWorker using NetWorker GUI or “recover” command. Once the MySQL databases are recovered from NetWorker, MySQL restore commands are used to actually restore the databases to MySQL Server.



MYSQL Backup script Flowchart



MYSQL Recovery script Flowchart

Performing backups using script-based approach

NetWorker supports a script-based approach to performing online backups and recoveries for any database or application with the help of that particular database or application appliance. For example:

- MySQL database application
- Microsoft Exchange Server

How to choose a scripting language

To perform any online backups for database or application, we need to freeze that particular database or application for some time to bring the databases to a consistent format to back up the database.

To identify the scripting language, we should know what platform on which we are about to run the script. For example:

- If backup needs to be done only on Windows system, then a batch file can be developed using Windows Powershell, system commands, and database-specific commands
- If backup needs to be done only on Unix system, then shell scripting can be chosen to develop the script. This shell script can use system commands and database application commands for all the tasks.
- If backup needs to be done on Windows and Unix both then a scripting language such as Perl or TCL/TK or Python can be chosen for script. But, these scripting languages require a learning curve and also a script engine on the host.

Identify the backup and recovery methods provided by Database application – MySQL

Making Backups by Copying Table Files

For storage engines that represent each table using its own files, tables can be backed up by copying those files. For example, MyISAM tables are stored as files, so it is easy to do a backup by copying files (*.frm, *.MYD, and *.MYI files). To get a consistent backup, stop the server or lock and flush the relevant tables

```
LOCK TABLES tbl_list READ;  
FLUSH TABLES tbl_list;
```

You need only a read lock; this enables other clients to continue to query the tables while you are making a copy of the files in the database directory. The FLUSH TABLES statement is needed to ensure that all active index pages are written to disk before you start the backup. See LOCK TABLES and UNLOCK

TABLES Syntax, and FLUSH Syntax. You can also create a binary backup simply by copying all table files, as long as the server isn't updating anything. The `mysqlhotcopy` script uses this method. Note that table file copying methods do not work if your database contains InnoDB tables. `mysqlhotcopy` does not work for InnoDB tables because InnoDB does not necessarily store table contents in database directories. Also, even if the server is not actively updating data, InnoDB may still have modified data cached in memory and not flushed to disk.

Making Delimited-Text File Backups

To create a text file containing a table's data, you can use `SELECT * INTO OUTFILE 'file_name' FROM tbl_name`. The file is created on the MySQL server host, not the client host. For this statement, the output file cannot already exist because permitting files to be overwritten constitutes a security risk. See `SELECT Syntax`. This method works for any kind of data file, but saves only table data, not the table structure. Another way to create text data files (along with files containing `CREATE TABLE` statements for the backed up tables) is to use `mysqldump` with the `--tab` option. To reload a delimited-text data file, use `LOAD DATA INFILE` or `mysqlimport`.

Making Backups with `mysqldump` or `mysqlhotcopy`

The `mysqldump` program and the `mysqlhotcopy` script can make backups. `mysqldump` is more general because it can back up all kinds of tables. `mysqlhotcopy` works only with select storage engines. For InnoDB tables, it is possible to perform an online backup that takes no locks on tables using the `--single-transaction` option to `mysqldump`.

Making Incremental Backups by Enabling the Binary Log

MySQL supports incremental backups. You must start the server with the `--log-bin` option to enable binary logging. The binary log files provide the information you need to replicate changes to the database that are made subsequent to the point at which you performed a backup. At the moment you want to make an incremental backup (containing all changes that happened since the last full or incremental backup), you should rotate the binary log by using `FLUSH LOGS`. Once done, you need to copy to the backup location all binary logs which range from the one of the moment of the last full or incremental backup to the last, but one. These binary logs are the incremental backup; at restore time. The next time you do a full backup, you should also rotate the binary log using `FLUSH LOGS`, `mysqldump --flush-logs`, or `mysqlhotcopy --flushlog`. See `mysqldump`, and `mysqlhotcopy`.

Making Backups Using Replication Slaves

If you have performance problems with your master server while making backups, one strategy that can help is to set up replication and perform backups on the slave rather than on the master. If you are backing up a slave replication server, you should back up its master.info and relay-log.info files when you back up the slave's databases, regardless of the backup method you choose. These information files are always needed to resume replication after you restore the slave's data. If your slave is replicating LOAD DATA INFILE statements, you should also back up any SQL_LOAD-* files that exist in the directory that the slave uses for this purpose. The slave needs these files to resume replication of any interrupted LOAD DATA INFILE operations. The location of this directory is the value of the --slave-load-tmpdir option. If the server was not started with that option, the directory location is the value of the tmpdir system variable.

Perform Recovery using MYSQL commands

MySQL provided the commands below to perform recovery using MYSQL dumps.

If we performed FULL backup for MYSQL dumps then we can use the following command to recover the mysql database.

```
mysql db1 < dump.sql
```

If we recover from incremental backup, we can use the following command to recover MYSQL database from Incremental MYSQL dump.

```
mysqlbinlog binlog_files | mysql -u root -p
```

Example of what happens if we use just file system backup of MySQL DB files

If we want to perform online backup for MYSQL database using NetWorker file system backup of MYSQL database, then backup will not allow until we freeze MYSQL database and generate MYSQL dump for that database. Hence, we need some script-based approach to perform all pre activities before starting online backups for MYSQL databases.

mysqldump and mysqlhotcopy

Mysqldump

- Simple command for each backup and restore
- Allows you to chose what to back up and not to backup
- Can be used on Windows and UNIX
- Not an online solution
- Poor performance for large databases

Mysqhotcopy

- Fastest method for database or single table backup
- Quick database backup using LOCK TABLE, FLUSH TABLES and cp or scp
- Can be run on the same machine where databases are residing
- Can backup only MyISAM and ARCHIVE tables
- Runs only on UNIX and Netware
- It is a Perl script

Full and incremental methods of database application

FULL backup can be done by using mysqldump command. Once we generate MYSQL dump for database, we can backup that MYSQL dump using NetWorker.

MYSQL is providing circular logging support to perform incremental backup. If we want to perform Incremental backup, we need to enable circular logging in MYSQL. This will allow us to perform Incremental backup for MYSQL databases.

Once circular logging is enabled, we can flush the MYSQL database logs using “Flush logs”. Using this command enables logs to be flushed to a specific file system location. NetWorker will perform incremental backup of file system location.

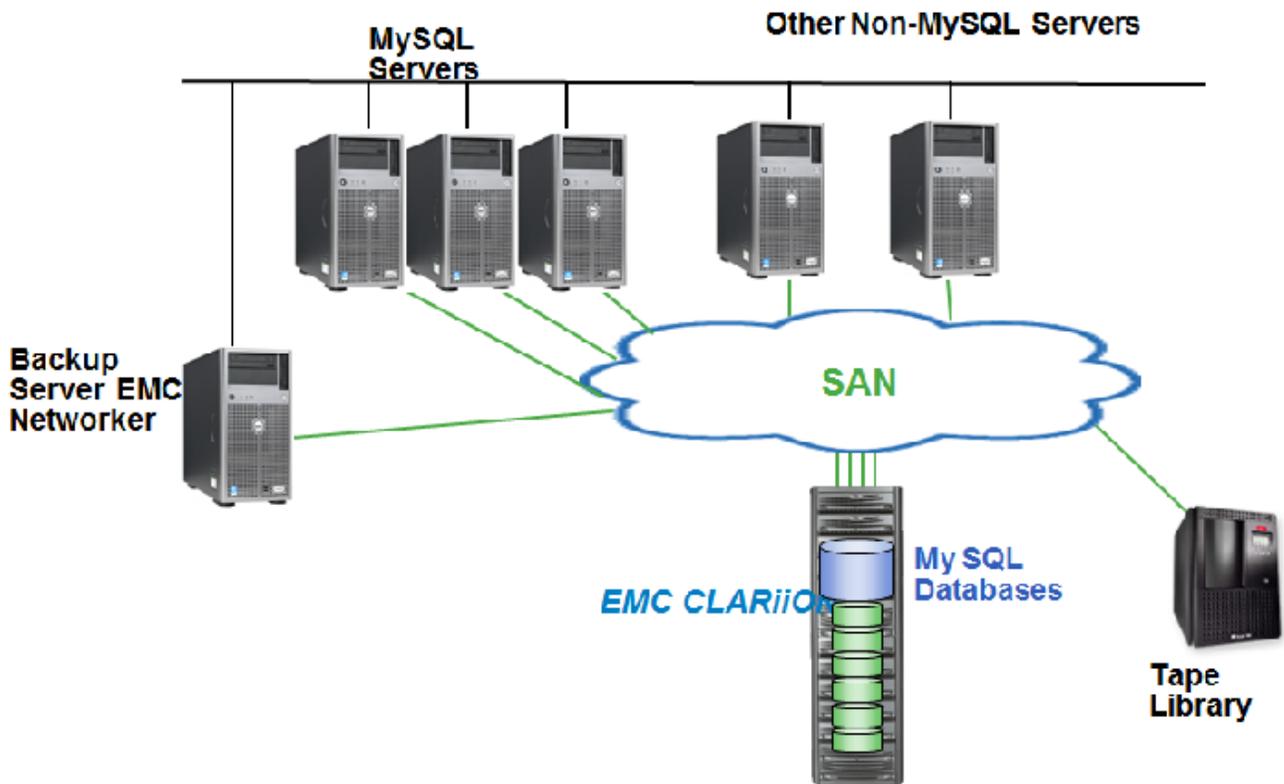
MySQL database backup using Script with NetWorker

During the course of providing support of MySQL databases backup and recovery, a POC was done with developing a script integrated with NetWorker to backup MySQL databases on Linux hosts. The script, developed using Shell scripting method on Linux, is responsible for the following tasks on execution:

- Identify whether to perform FULL or Incremental backup
- Identify all the databases residing in MySQL database application
- Create the list of databases which the user selected for backup via a configuration file
- Create consistent copies of MySQL databases using MySQL backup command “*mysqldump*” and dump the data in a flat file or system pipes. For example:
 - *mysqldump --database TestPWD TestDB >/tmp/mysql.fifo*
- Start the NetWorker backup, where input file is MySQL dump file or through system pipes. For example:
 - *save -s nwserver -f /tmp/.rawasm -N MySQL-TestDB /tmp/mysql.fifo*
- Once the backup completes, clean up the system with the dump file

- For incremental backup, the script uses MySQL commands to perform the backup of binary log files which are configured as part of MySQL backup configuration.

A high level system architecture diagram is given below:



Guide for using MySQL backup script integrated with NetWorker

- 1.1.1 MySQL requirements
- 1.1.2 Binary logging to be enabled in my.cnf file for Incremental backup
- 1.1.3 NetWorker requirements
- 1.1.4 NetWorker save group and NetWorker client resource to be configured per the given guidelines
- 1.1.5 Operating System requirements
- 1.1.6 NetWorker client – Linux Server
- 1.1.7 Initial POC was done with MySQL database application on Linux Server
- 1.1.8 NetWorker server – Any Operating System supported by NetWorker
- 1.1.9 NetWorker storage node – Any Operating System supported by NetWorker

Full and incremental backup

Operational Sequence

1. Create two save groups in the NetWorker console using NMC GUI for Full and Incremental backups.

The screenshot shows the 'Group Properties - MySQL Backup Group' dialog box with the 'Setup' tab selected. The dialog is divided into several sections:

- Identity:** Name: MySQL Backup Group, Comment: Test - 185
- Clones:** Clones: , Clone pool: Default Clone
- Setup:** Start time: 3:33, Autostart: Enabled
- Status:** Status: idle, Last start: , Last end:
- Output:** Printer:
- Snapshots:** Snapshot: , Snapshot Policy: Daily, Snapshot Pool: Default

Buttons at the bottom: ? (help), OK, Reset, Cancel.

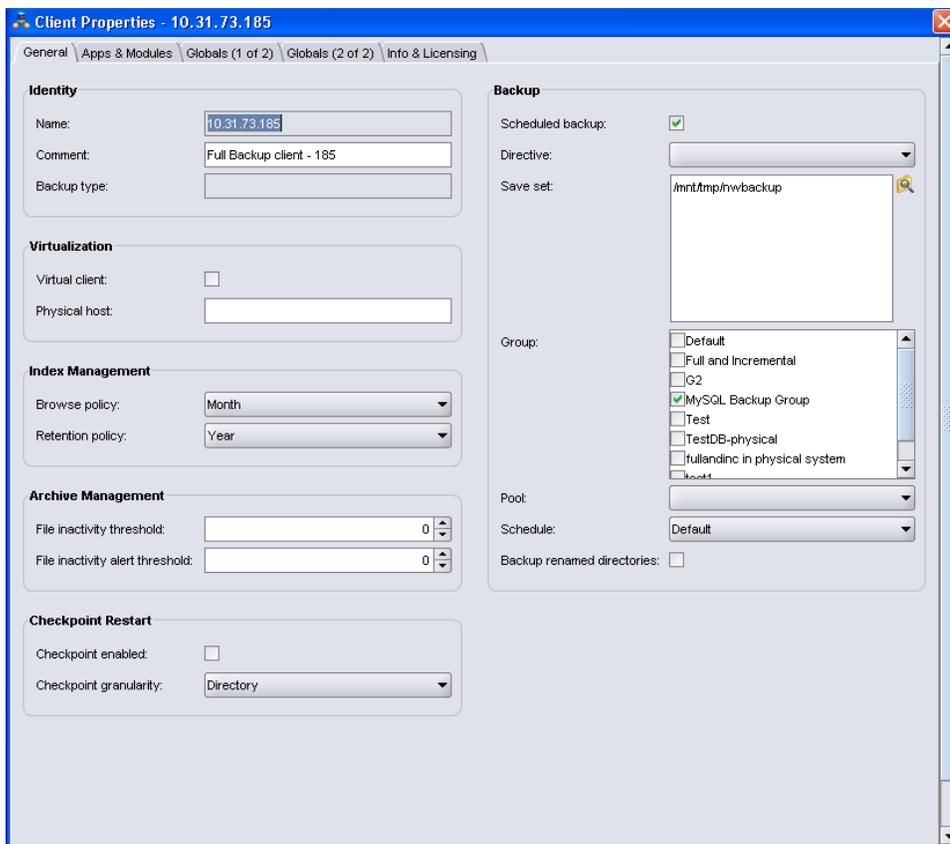
The screenshot shows the 'Group Properties - MySQL Backup Group' dialog box with the 'Advanced' tab selected. The dialog is divided into several sections:

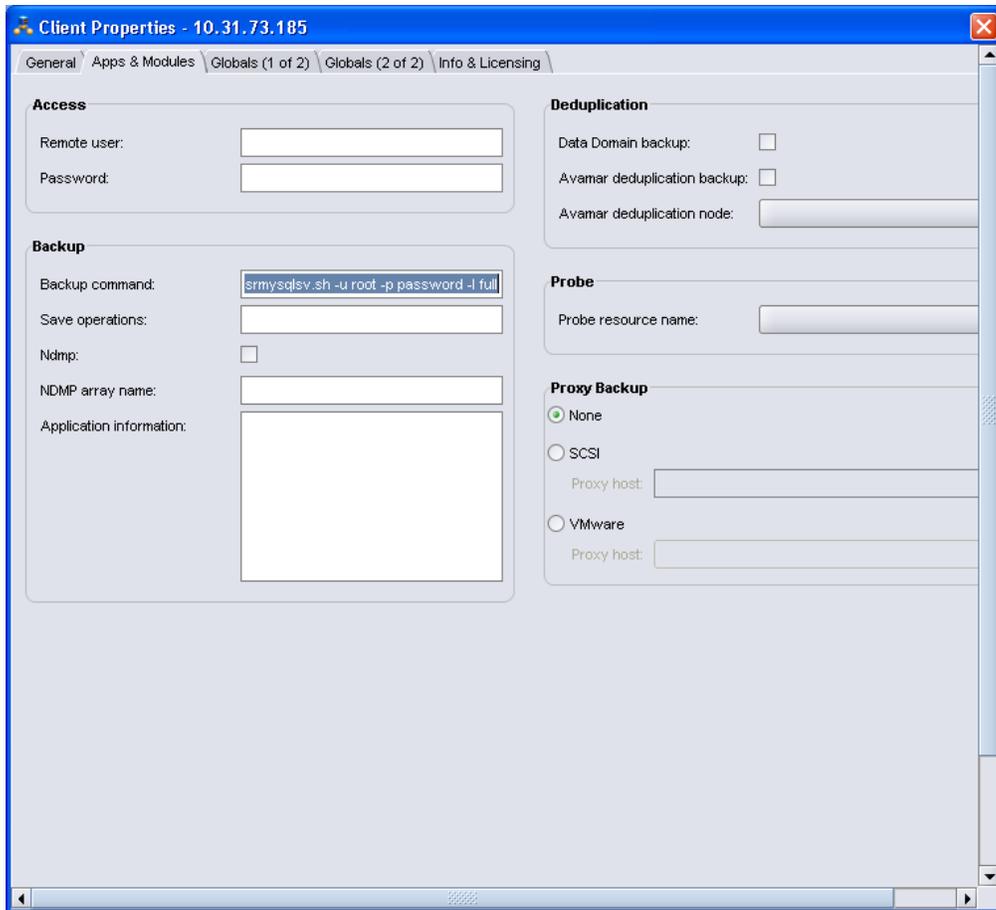
- Client Overrides:** Schedule: , Level: , Interval: 24:00, Force incremental: , Schedule time: , Expiration time: , File inactivity threshold: 30, File inactivity alert threshold: 30
- Probe:** Probe based group: , Probe interval: 60, Probe start time: 0:00, Probe end time: 23:59, Probe success criteria: all, Time since successful backup: 0, Time of the last successful backup:
- Configuration:** Autorestart: Enabled, Restart window: 12:00, Success threshold: Warning, Client retries: 1, Inactivity timeout: 30, Savegrp parallelism: 0, Soft runtime limit: 0, Hard runtime limit: 0, Options: No Monitor, No index save, No save, Index only, Verbose, Estimate, Preview, Manual contact, Timestamp: none

Buttons at the bottom: ? (help), OK, Reset, Cancel.

Sample client resource for Full backup

- **Saveset Location:** The location provided in the Save set here should be the absolute path of the location where the mysql dump file to be created or the Incremental Logs created, would be copied to, in the case of Full and Incremental backup, respectively. For this example, assume the save set location is **/home/nw_backup**
 - Backup command: `nsmysqlsv.sh -u USERNAME -z PASSWORD -I full`
 - The script mentioned above should reside in the location **'/usr/bin'** to enable NetWorker to locate it for execution
 - Group Name: Full backup
 - Type of backup (select in advance section of group): Full





For incremental backup, “Backup command” should have *“nsmysqlsv.sh -u root -p password -I incr”*, where -I is backup level incremental.

2. In the monitoring section, execute the Group – ‘Full Backup’
3. Upon execution, the script mentioned in the backup command is called and the execution continues as follows:

Full Backup

Prerequisites

- The list of databases to be excluded is to be provided in the file ‘exclude.list’
- This file resides in the same location as the script
- User name/Password: To be mentioned at the command line

Work-flow

- The script first creates a log file in the location mentioned in LOG_DIR, with a specific time-stamped name

- The Backup directory path is fetched from the argument list that is passed to the script when it is called from NW. The Backup directory is the last argument of the argument list.
 - This is also the Save set path that is provided in the Group in NetWorker
 - In the above mentioned Backup directory, a new directory for Full Backup is created with a specific format with the timestamp
 - The list of all the databases is fetched
 - From the file mentioned above, the exclude list is populated with the list of databases that are to be excluded from the backup process
 - Iterating through the list of all databases and checking if the current database is to be excluded or not, it is chosen for the dump procedure.
 - *Mysqldump* is called with the login credentials to create a dump file of the current database in the iteration. The output is redirected to a file name called 'database.sql' where database is the current db in iteration.
 - Once all the databases in the list have been traversed and mysqldump called for all of them, NetWorker SAVE is called with the argument list fetched in the beginning of the script
 - This call has the following arguments passed to it:
 - *-s Server* Specifies which machine to use as the NetWorker server
 - *-g Group* This option is used to denote the group of the save
 - *-c client-name* Specifies the client name for starting the save session
 - *Save set path*
 - Logging is ended and the log file is closed
 - Location of the Log File and the Backup is mentioned in the log file
4. Changes are made to the table and new data is added
 5. In the monitoring section, execute the Group – 'Incr Backup'
 6. On execution, the script mentioned in the backup command is called and the execution continues as follows:

Incremental Backup

Pre-requisites

- Binary logging to be enabled in my.cnf file

Work-flow

- The script first creates a log file in the location mentioned in LOG_DIR, with a specific time-stamped name
- The Backup directory is fetched from the argument list that is passed to the script when it is called from NW. The Backup directory is the last argument of the argument list.
- This is also the Save set path that is provided in the Group in NetWorker
- In the above mentioned Backup directory, a new directory for Incremental Backup is created with a specific format with the timestamp
- The Binary log files are first flushed using the command 'flush logs'. They are then copied from the BIN_LOG_DIR to the Backup directory mentioned in the Save set in NetWorker.
- NetWorker SAVE command is called to back up the new Incremental directory that is created, using the argument list fetched from the command line

Recovery

Types of Recovery

There are two recoveries that have to be performed:

- Recover from Full data
- Recover from Incremental data

MySQL Recover commands

After recovery of data using NetWorker, the following MySQL commands should be used to replay the Full backup and dump database and Incremental backup Dump database

Full backup replay:

```
mysql --user=root --password=password Database < Database.sql
```

Incremental backup replay:

```
/usr/bin/mysqlbinlog <mysql-binarylog.000001> | mysql --user=root --password=password  
<Database name>
```

Procedure for Data Recovery

- Delete the database
- Run 'cd /home/nw_backup'
- Type 'recover'
- In recover prompt, Type 'ls'. The following output will be displayed:
 - FBK_2011-08-25 INCBK_2011-08-25_20
- In recover prompt, Type 'add FBK_2011-08-25'
- In recover prompt, Type 'recover'. The recovery is complete and data is recovered to /home/nw_backup
- Exit from recover prompt and run the following command to restore the Full backup dump file
 - mysql --user=root --password=password EMC_DB < FBK_2011-08-25/EMC_DB.sql
- Next, restore data from Incremental backup
- Type 'recover' prompt, Type 'ls'. The following output will be displayed:
 - FBK_2011-08-25 INCBK_2011-08-25_20
- In recover prompt, Type 'add INCBK_2011-08-25_20'
- In recover prompt, Type 'recover'. The recovery is complete and data is recovered to /home/nw_backup
- Exit from recover prompt and run the following command to replay the Incremental backup dump file
 - /usr/bin/mysqlbinlog INCBK_2011-08-25_20/mysql-bin.000001 | mysql --user=root --password=password EMC_DB
- The dump files are replayed and the database is restored completely.

What we get free as part of script-based approach

The NetWorker script-based approach provides a few features without extra cost and without writing code from the customer end. This means that following the NetWorker script-based approach, customers can perform backup of any type of databases which are not directly supported by any NetWorker module. Using the script-based approach, users also get certain NetWorker features such as LAN-Free backups, data deduplication-based backup using Data Domain or Avamar, and so on.

Please refer to the NetWorker Admin Guide to configure LAN-Free backups and Data Domain- / Avamar-based deduplication.

Challenges

There are different categories of challenges a customer could face with the above approach.

- EMC challenges
 - The POC above does not provide any official support on the backup methods used via backup script.
 - Security: MySQL User name and password to be defined in plain text in NetWorker client resource for backup script.
 - Though NetWorker server and client communication is secure and NMC GUI is always under NetWorker administrator control.

- Customer challenges
 - Backup time is large as *mysqldump* command takes a significant amount of time in dumping the MySQL databases. If databases are too large, the backup window will be very large.
 - Customer can use exclude list to filter the databases which they do not want to back up
 - Using system pipes in the script can reduce the time of creating dump as data will be returned to system pipes and not on the disk.
 - Disk space is required for dump file and it would be about the same as database size.
 - Customer can use exclude list to filter the databases which they do not want to back up
 - Using system pipes in the script can reduce the time of creating dump as data will be returned to system pipes and not on the disk.
 - Customer needs to identify backup and recovery methods for a given database application.
 - Customer needs to learn scripting to use this method.
 - Database recoveries are flat files and customer needs to use MySQL commands by themselves to restore files.

Conclusion

NetWorker software is capable of providing support for backup and recovery of database applications which are not directly supported by any NetWorker module. Using the script method, customer can develop their own backup and recovery module and integrate with NetWorker. Customers would get all the benefits with their own script without adding anything specific in the script, e.g.

- Support for a large number of operating systems
- Support for a large number of backup targets
 - Disk as backup target
 - Data Domain
 - Avamar
 - Tape libraries
- Automatic support for LAN Free backup
- Other NetWorker features, such as:
 - Cloning of backup
 - Backup Staging
 - Disaster Recovery

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.