



Backup and Recovery Methods for Linux on the System i Platform

*Erwin Earley
System i Technology Center
Linux Center of Competency*

August 2006

| |
|----------|
| Contents |
|----------|

| | |
|----|---|
| 2 | Overview |
| 3 | Storage Space as part of existing i5/OS Backup Strategy |
| 4 | i5/OS Based Solutions |
| 9 | Linux Based Solutions |
| 10 | Backup/Restore Solution Comparison |
| 10 | Combining Approaches – A Flexible Solution |

Overview

As the use of Linux continues to gain momentum in the enterprise space, the Information Technology (IT) manager is faced with the need to implement backup and recovery plans for their Linux servers. Additionally, some enterprises have the need to implement High Availability (HA) solutions as well as Disaster Recovery (DR) solutions for their critical Linux infrastructure services and related data.

This paper will discuss a number of solutions that can be used for backup and recovery, HA, and DR of Linux partitions running on the System i platform. Both open source as well as i5/OS based solutions will be presented along with information on what solutions should be implemented based on the needs of the enterprise. It needs to be understood that there is no “best-fit” approach that will meet the needs of all enterprises. What solution, or solutions, should be implemented can only be determined once a firm understanding of the requirements of the enterprise have been ascertained. That understanding, along with the background provided in this paper should serve as the basis for the implementation of robust solutions for the enterprise customer.

The solutions presented in this paper can be broken down into three broad categories:

- **i5/OS based solutions:** i5/OS based solutions are those solutions that rely on i5/OS capabilities, such as save/restore, BRMS, and Tivoli Storage Manager, to backup the Linux partition.
- **Linux based solutions:** These solutions center around using Linux tools and utilities to provide file-level backup. Information in this section will focus on leveraging the power of virtualization to build robust/flexible backup solutions. These solutions include performing file-level backup to a tape device and using the Integrated File System (IFS) as the target of backup solutions.
- **High Availability and Disaster Recovery:** These solutions center on using open-source solutions, such as RSYNC, and DRBD to backup data to secondary servers to provide both high availability and disaster recovery solutions.

Storage Space as part of existing i5/OS Backup Strategy

One method of backing-up a Linux partition on System i platform is to backup the entire Network Storage Space (Virtual Disk) and restore the Storage Space when files need to be restored. This method has the advantage that it uses standard i5/OS commands (sav/rst) and provides for a complete backup of the Linux instance (which provides a good foundation for Disaster Recovery solutions). This method also provides for a rapid return to production of the Linux partition since the Linux partition will only be down while the storage space is saved to a save file. However, this method imposes a number of restrictions primarily in the area of server availability (the Linux instance must be varied of while the backup is made); as well as space requirements (additional storage is required in the IFS to restore the storage space and the space will be taken up as long as the Linux server is active). With this approach the entire Linux instance is saved (and restored) as a single object. It provides a good foundation for restoration of the entire Linux system; however, additional steps (such as copy of files from one “disk” to another “disk”) would need to be taken for restoration of individual files. Here is one way that this method could work:

- Step 1** Vary off the Network Server that the Virtual disk is attached to (this will take the Linux instance down and it will be unavailable to the users)
- Step 2** Create a save file of the storage space
- Step 3** Vary the Network Server back on
- Step 4** Write the save file out to tape.

For the restore, the steps would look as follows:

- Step 1** Read the save file from tape back into the IFS
- Step 2** Restore the storage file from the save file
- Step 3** Link the storage file to the Network Server
- Step 4** In Linux re-scan the Virtual SCSI bus to see the new disk
- Step 5** Mount the virtual disk to access the files
- Step 6** Use standard Linux commands to copy the file(s) required from the restored virtual disk

While the above method will work for some Linux environments, more robust methods are possible that leverage the flexibility of virtualization that i5/OS provides. The following pages will outline these methods

Highlights

File-Level Backup/Restore via
the Integrated File System

i5/OS Based Solutions

Like most other aspects of Linux, the utilities and tools within the operating system provide simple functions that can be put together to implement a variety of solutions. Linux provides a number of utilities and tools within the distributions available for the System i platform (Red Hat Enterprise Linux 4 and Novell's SUSE LINUX Enterprise Server 9) that can be used together for providing file-level backup capability. These tools include:

- **find:** this tool can be used to locate files and directories based on a large number of characteristics including file name, file size, modification date/time, file type, etc.
- **tar:** this tool can be used to take a large number of files and combine them in a single file archive
- **gzip/gunzip:** These two tools are used to compress and uncompress files
- **cpio:** This tool can be used to write a save-set out to a tape device or file.

One can readily see from the above list that these tools could be used together to locate (find) files that meet a backup criteria (such as files modified since last backup), build an archive of those files (tar), compress that archive to save space (gzip) and then write the resulting archive out to a save-set (cpio). The goal of implementing a Linux-centric backup solution would be to combine these tools with the flexibility provided by virtualization to implement the best backup solution for your enterprise

File-Level Backup/Restore via the Integrated File System

This backup solution utilizes tools within Linux to backup files and then write the resulting save-set out to the Integrated File System (IFS) where it can then be part of the normal save/rst activities performed within i5/OS. This solution has the advantage of using standard Linux utilities and requiring little in the way of additional skills on behalf of the i5/OS administrator. Additionally, this method has the advantage that the Linux operating system can remain running while the backup is being performed. This method, unlike the method of saving the entire storage space, provides an easier methodology for save and restore of files.

In this method, scripts are implemented within Linux that will perform the actual backup of files to the IFS. One script would be

responsible for locating the files to be backed-up, building the save-set and writing it out to the IFS. This “backup” script could be scheduled to automatically execute (via cron – the Linux command scheduler) at a given time each day, thus freeing the system administrator from having to manually perform the backup operation.

This method usually results in a number of scripts being developed:

- **backup:** Finds the files that have changed over a given period of time, generates a save-set of those files, and writes the files out to the IFS
- **list:** Retrieves a save-set from the IFS and lists the contents (files) from the save-set
- **restore:** Retrieves a save-set from the IFS and retrieves a file from that save set and writes it to the Linux file system

There are a number of advantages and disadvantages to this approach. On the plus side, the size of the backup set is limited to the size of the files that have changed over the specified backup period. Additionally, since the backup method relies on the IFS to store the backup sets, the backup can be easily integrated into the existing i5/OS backup scheme. This method does; however, require that Linux scripts be developed for the backup/restore process.

Typically, the Linux scripts used to backup and restore files to the IFS are executed from the Linux command Line; however, a GUI front end can be implemented to interface/execute the scripts. As an example, a Front End to the list and restore process could be implemented within the WebMin tool:

The screenshot shows the 'Custom Commands' section of the WebMin interface. At the top, there are two links: 'Create a new custom command.' and 'Create a new file editor.'. Below these are two forms:

- List Files from Backup Set:** This form has a dropdown menu for 'Name of backup set' with the value 'linux.backup.Oct05'. It includes a text input field for 'i5/OS QSECOFR Password' and an 'Edit command' link.
- Restore Files From Backup Set:** This form has a dropdown menu for 'Name of Backup Set' with the value 'linux.backup.Oct05'. It includes a text input field for 'Name of File to Restore', a radio button selection for 'Overwrite if file exists' (with 'No' selected), a text input field for 'i5/OS QSECOFR Password', and an 'Edit command' link.

At the bottom of the screenshot, the same two links from the top are repeated: 'Create a new custom command.' and 'Create a new file editor.'.

Highlights

File-Level Backup/Restore via
Network Storage Space

With this simple interface it would be possible to list files from a specified save-set and then use cut-and-paste to copy the name of the file to be restored and paste it into the restore function.

File-Level Backup/Restore via Network Storage Space

Another method that could be implemented for file-level backup would be the backup of the entire Network Storage Space (Virtual Disk) and restoring it to another Linux partition when a file is needed. Remember that the virtual disk is an IFS object and represents the entire Linux instance (operating system, programs, and data files). Since it is an IFS object it can be saved/restored like any other object within the IFS. Let's take a closer look at this approach by first looking at the steps that would be followed to backup the storage space:

Step 1 Vary off the Network Server (The Network Server cannot be active while the storage space is saved).

Step 2 Unlink the storage space from the Network Server (this is to avoid the storage space being linked to the Network Server when it is restored)

Step 3 Write the storage space out to a savefile

Step 4 Relink the storage space to back to the production network server

Step 5 Vary on the Network Server

Step 6 Write the savefile from step 3 out to tape

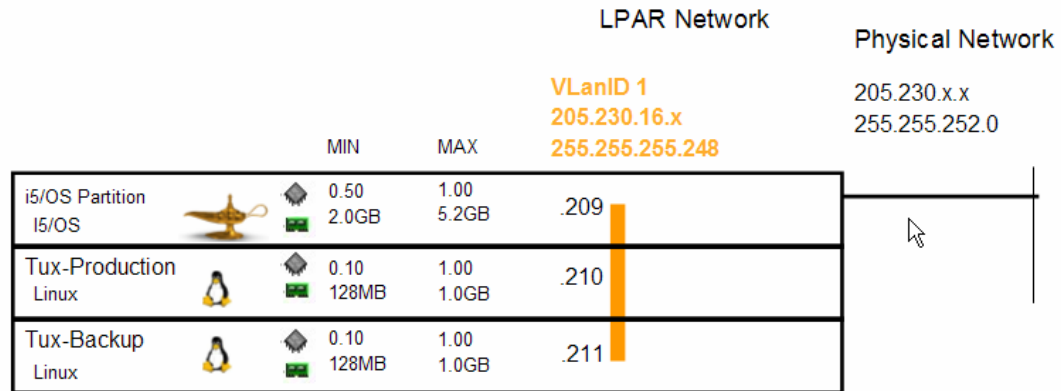
The above steps will generate a complete copy of the virtual disk that represents the Linux instance and can be used as the basis for restoration of files

NOTE: BRMS could be used to perform the above functions. Likewise a CL script could be generated in i5/OS to automate the above process.

NOTE: If multiple storage spaces are used for the Linux instance they should all be backed up at the same time – especially if a file system spanning mechanism such as LVM or Software Raid has been used.

Now that a backup of the Linux instance has been made it can be used as the basis for restoration of files. For the restore of files, a Linux partition should be created that during normal operations will have no system resources. When restore of files is required,

resources will be allocated to the partition. The following diagram is an example of the partition layout that a typical system may have to support this backup method:



With this method, a virtual LAN is established between the production Linux partition and the partition used to restore the files. The idea here is that the saved virtual disk will be restored to the backup partition and then the files that need to be restored will be copied to the production partition across the virtual LAN. The intent with this approach is to minimize the impact on the overall System i as well as the production Linux partition.

The following steps outline how files would be restored from a previously saved copied for the virtual disk:

- Step 1** Use Dynamic LPAR (via the HMC) to free-up processor and memory resources from active partitions. These resources will be used by the backup Linux partition. Typically a 10th of a processor and 512MB of memory is sufficient for this function.
- Step 2** Restore the savefile that contains the storage space.
- Step 3** Restore the storage space from the savefile. On the restore command, ensure that you give a unique name for the storage space.
- Step 4** Link the restored storage space to the Network Server for the backup Linux partition
- Step 5** Vary on the Network Server for the backup Linux partition.
- Step 6** Retrieve the file from the backup server. This can be accomplished from the production Linux partition using Linux commands such as the 'cp' command for copy.
- Step 7** Vary off the backup Linux partition.

Step 8 Unlink the storage space from the Network Server for the backup Linux partition

Step 9 Delete the storage space

Step 10 Use Dynamic LPAR to move the processor and memory resources from the backup Linux partition back to the production partitions.

NOTE: A CL script could be implemented in i5/OS to perform steps 2 through 5.

NOTE: The method used to restore the files from the backup Linux partition to the production Linux partition is dependent on what type of file is being restored. As an example, many files can simply be transferred via Linux commands across a network application (such as nfs, ftp, rcp. For some files, like files from a file server that have implemented ACLs, a method that will maintain those attributes (such as a client that has been configured to maintain ACLs) on the copy will need to be used.

This method leverages virtualization by providing an environment where additional system resources are only used for the time it takes to perform the restore function. However, this method does require that during the restore process a second copy of the virtual disk for the Linux instance is resident within the Integrated File System.

Tivoli Storage Manager

Tivoli Storage Manager could also be used to perform file-level backup/restore of files from a Linux instance. The Tivoli Storage Manager client and server component is available for Linux running in a partition on the System i platform. With TSM, files can either be pushed to a TSM server or could be pulled from the TSM server.

The ability to implement the TSM client in Linux is a benefit to those environments that have implemented TSM as their backup/restore method and wish to incorporate TSM into that approach. Usage of TSM as a solution for Linux file-level backup will require that the TSM server be running on another system within the enterprise that the Linux instance has network connectivity to.

Linux Based Solutions

File-level backup of Linux files can be completely contained within the Linux instance. Similar to the File Level Backup to the Integrated File System method outlined above, Linux based solutions will write

their save-sets directly out to tape (instead of out to the IFS). Remember that when Linux is using virtual storage hosted from an i5/OS partition, Linux also has access to the tape and CD/DVD devices allocated to the i5/OS partition.

Access from Linux to an i5/OS tape device is available when the tape device is varied off in the i5/OS partition. Once the tape is varied off, Linux utilities (such as tar) can write files (or save sets) out to the tape and can restore files from those tapes. It should be noted that using this virtual tape approach, Linux does not support tape libraries – therefore only the mounted tape from a tape library would be accessible to Linux. In addition to providing virtual access to the i5/OS attached tape drive, a Linux instance can also be given access to a tape drive by dynamically moving the tape drive from one partition to another provided that the partitions have been properly configured for the associated storage controller resource.

Deploying Linux-based solutions for file-level backup will require Linux skills in the areas of system management as well as shell scripting. For those environments that have deployed Linux on other platforms and have Linux skills in-house, the usage of Linux-based solutions is a good fit as the existing skill set (as well as any in-house developed scripts) can be directly implemented in the Linux on System i environment. For those environments that are primarily i5/OS based; where Linux is being implemented as an appliance device, the i5/OS centric methods discussed earlier should be considered.

NOTE: The format that Linux uses for tapes is different from that used by i5/OS and the same tape cannot be used to store files from both operating systems.

Backup/Restore Solution Comparison

| Solution | Advantage | Disadvantage |
|---|--|---|
| Storage Space backed up as part of existing i5/OS backup strategy | Provides for fast backup Leverages existing backup procedures | Linux partition needs to be shutdown Does not provide for file-level backup/restore |
| File Level Backup to the Integrated File system | Can be performed while Linux is active | Requires Linux scripts to be developed to drive the backup/restore process |
| File Level Backup via Network Storage Spaces | Almost completely driven from i5/OS. Simple Linux copy commands used to restore files | Linux must be shutdown Requires additional storage in IFS to store second copy of Virtual Disk |
| Linux Based Solutions | Provides for a complete Linux solution | Requires Linux skill to implement Requires two sets of tapes to be maintained Linux can only access single tape from virtual tape libraries |
| Tivoli Storage Manager | Allows for file-level backup while Linux is active Does not require additional storage in IFS | Requires a TSM server within the enterprise Requires external network connectivity from Linux to the TSM server |

Combining Approaches – A Flexible Solution

Different approaches could be used to provide for both a full system backup as well as incremental file-level backup. As an example, on a weekly basis the entire storage space that represents the Linux instance could be backed up though a save of the storage space. Then on a daily basis, individual files that have been modified could be backed up through either a backup of those files to the IFS (as described earlier in the File-Level Backup to the Integrated File System section) or through Linux based solutions. Then while a file needs to be restored it can either be restored from the incremental file-level save(s) (if it has changed within the last week) or from the full save of the storage space. Additionally, if needed, a complete restore of Linux could be performed from the weekly save of the storage space.

Highlights

High Availability and Disaster Recovery

High Availability and Disaster Recovery

A number of methods and tools are available for those environments that wish to implement High Availability (HA) and/or Disaster Recovery (DR) solutions for their Linux partitions. These solutions can be as simple as a save of the storage spaces that represent Linux to the usage of open source solutions.

In the open source space, a number of tools are available for implementing both HA and DR solutions. These include the following:

- **RSYNC:** RSYNC can be used to replicate files between Linux instances on an event driven basis.
- **DRBD:** DRBD is a device replication utility that intercepts all data being written to disk and causes that data to be written to a secondary disk device (typically on another Linux system). DRBD includes fail-over utilities and is one method for implementing an HA solution.
- **Linux Virtual Server Project:** This is a collection of tools for implementing Linux clusters as well as HA solutions.

Disclaimers and trademarks

IBM Corporation 1994-2006. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

| | | |
|----------------------|------------|-----------|
| AS/400 | eServer | i5/OS |
| AS/400e | iSeries | OS/400 |
| Blue Gene | IBM | System i5 |
| e-business on demand | IBM (logo) | System i |

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.