Solaris Bare-Metal Recovery from a Specialized CD-ROM and Your Enterprise Backup Solution

Lee "Leonardo" Amatangelo – Collective Technologies
W. Curtis Preston – The Storage Group, Inc.

ABSTRACT

The bane of all system administrators is the crashing of a mission critical system. Further depression sets in when the crash has caused the boot disk to become corrupted and no longer possess the ability to boot the system. A crashed mission critical system requires immediate attention. Literally, every moment of downtime equates to lost revenue. The desire is to get the mission critical system back to its normal functioning state in the quickest amount of time. The rebuilding of a computer system that has lost its capability to boot is known as a bare-metal recovery. The system will need to be built starting from a bare disk drive up to a bootable and functioning operating system.

Not until the release of Solaris 8 did the Solaris operating system contain a utility for providing bare-metal recovery functionality. Such functionality can be found in the operating system of other UNIX variants, such as IBM AIX, HP HP-UX, and Linux. However, by making use of the following Solaris utilities: ufsdump, ufsrestore, dd, cpio, tar, format, prtvtoc, fmthard, installboot, and JumpStart, bare-metal recovery can be achieved. Furthermore, it can be achieved by using a custom-built bootable CD-ROM (or DVD-ROM) and your environment's networked enterprise backup solution, such as Legato NetWorker and Veritas NetBackup.

Introduction

There is an ever-increasing demand to have mission critical computer systems run 24 hours by 365 days a year. When a mission critical system suffers from corrupted disk data, and when the disk drive in question happens to be the boot drive, the situation becomes more involved. One cannot simply run the backup application to restore the system because a functional operating system no longer exists on the system. This type of disaster recovery is known as bare-metal recovery. (The bare-metal comes from the early days of computing when storage media was a metal core. Nowadays, storage media is commonly magnetic or optical.)

Under such dire circumstances, rebuilding a boot disk can take what seems like an inordinate amount of time. The operating system needs to be installed, standard packages (clusters) installed, customized packages installed, patches applied, kernel tuning, and finally any special or custom configurations need to be set. To perform all of these tasks could take up to a couple hours.

Even using Sun Microsystems' automated installation utility, JumpStart, can still take what might be considered too much time, because following the installation of the operating system and any customized packages, the JumpStart process still needs to apply desired patches. The application of a large set of patches can take up to a couple of hours. The process is not totally complete because following the installation of the patches, any variable data will still need to be restored to the system using the environment's backup solution.

Until Web Flash in Solaris 8, Sun Microsystems did not provide a complete disaster recovery tool as an integral part of the Solaris product. As opposed to the IBM AIX UNIX environment, which has mksysb, and the Hewlett-Packard HP-UX environment, which has Ignite-UX and make_recovery. While not providing a formal disaster recovery (cloning) tool, Sun Microsystems does provide some very useful utilities to aid in the process of disaster recovery. Specifically, these utilities are ufsdump, ufsrestore, dd, cpio, tar, format, prtvtoc, fmthard, installboot, and JumpStart. When combined, these utilities can produce a very powerful disaster recovery tool.

This paper will discuss the steps needed to perform a bare-metal recovery on a Solaris system. In so doing, this paper describes a tool that enables the system administrator to perform a timely rebuild of a crashed system to the state of its last successful backup by using a custom-built CD-ROM and the environment's networked backup solution. Currently, the tool discussed in this paper is built to handle Solaris systems within an environment that uses Legato NetWorker or Veritas NetBackup. This tool can be extended to handle other shrink-wrapped or homegrown backup products.

Background Information

This paper is a by-product from the paper, *Unleashing the Power of JumpStart: A New Technique* *for Disaster Recovery, Cloning, or Snapshotting a Solaris System*, [1] presented at the 14th Annual LISA Conference, (LISA-2000). The LISA-2000 paper and associated presentation discussed in detail how to create the "Capture and Restore Tool," (CART), a tool for capturing the image of a Solaris system onto a set of CD-ROMs, with the first CD-ROM volume in the set being bootable. Using this set of CD-ROMs, a server could be readily rebuilt onto the same hardware or cloned onto like hardware consisting of disk drives of the same or larger size as the original source system. The CART was achieved by using the same technique Sun Microsystems uses for installing the Solaris operating system from its installation CD-ROM. The technique incorporates a specialized JumpStart mechanism built in the CD-ROM.

By using the CART technique, it is possible to build a bootable CD-ROM (or DVD-ROM) that will automatically invoke a customized script whose function is to perform a bare-metal recovery using the latest filesystem backups captured for that system by the environment's backup solution. The CART restored the system's data from a set of CD-ROMs created during the capture phase. This new tool performs the bare-metal recovery by restoring data over the network from the environment's backup solution. This new ancillary tool for performing the bare-metal recovery of a Solaris system is referred to as the Bare-metal Ancillary Recovery Tool, or BART.

The concept of the BART was first announced at LISA-2000. At that time, the tool was under development and showed promising results. Since then, several requests from System Administrators and Backup & Recovery professionals have rolled in regarding the availability of this tool. At this time, we are now prepared to fully disclose the details of the BART, how it was built, and how it can be customized to address the idiosyncrasies of a particular environment.

Basic Bare-metal Recovery on a Solaris System

The use of any disaster recovery or bare-metal recovery tool does not replace the need for a complete Disaster Recovery Plan (DRP). Devising a good disaster recovery plan is hard work. It needs to be built from the ground up and it can take years to perfect. Since computer environments change constantly, the DRP must continually be tested to ensure it still works in the changed environment.

W. Curtis Preston's O'Reilly book [4] is a great resource on backup, recovery, disaster recovery, and bare-metal recovery. He walks the reader through the what, when, why, how, how many, and how often data on a system should be captured prior to the need for restoring it.

Since the topic of bare-metal recovery has been covered by other sources, specifically [3] and [4] listed in the references section, this paper will not discuss the topic in great detail. Instead, a list of pertinent steps for accomplishing bare-metal recovery on a Solaris system is provided.

Prior to the Disaster

1. Save the Volume Table of Contents, vtoc, of all disk drives on the system by using the prtvtoc utility which is very effective for capturing and saving this information to a file.
2. Save the /etc/vfstab file.
3. Save all appropriate metadata database information (which maps physical devices to logical devices). This information proves valuable when physically reconfiguring the hardware of a system.
4. Capture good backups of all filesystems on all disk drives on the system by using utilities such as: ufsdump, dd, cpio, tar, or third party backup software packages.

After the Disaster

1. Replace all defective disk drives and all other defective system components.
2. Replace the vtoc on all replaced disk drives by using the fmthard utility and the file saved in step (1) in the list of actions to take "Prior to the Disaster."
3. Boot the system to be recovered either via CD-ROM or from a boot server to place a functioning operating into memory on the system.
4. Mount the disk that is to be the new boot disk.
5. Recover the operating system to the mounted disk by using the same utility used in step (4) in the list of actions to take "Prior to the Disaster."
6. Place the boot block on the mounted disk by using the installboot command; this is a very important step else the disk will not be bootable.
7. Reboot the system.
8. Pray.

The Bare-metal Ancillary Recovery Tool – The "BART"

The Bare-metal Ancillary Recovery Tool, BART, consists of a single bootable CD-ROM. Like the CART, the BART contains a trimmed down version of the Solaris Software Installation CD-ROM. To create the BART, selected files were copied from the Solaris Installation CD-ROM to a read-writable hard disk drive. Scripts were borrowed from the CART project to accomplish the task of copying files to the "image disk." This disk drive is known as the "image disk" because once the disk is put into its desired configuration, it becomes the image that gets written to a recordable CD (CD-R or CD-WR). Since, the maximum capacity of a CD-ROM is about 650 MB, a one GB disk drive is adequate for the "image disk." The full layout, sizes, and description of the slices for the BART "image disk" are displayed in Diagram 1.

Description of the BART "Image Disk" slices

Slice s0 contains a trimmed down version of the Solaris Installation CD-ROM slice s0 and is present solely to enable the JumpStart mechanism.

Slice s1 contains the mini-root along with an adequate set of UNIX utilities. Upon booting the BART CD-ROM, slice s1 gets placed into memory and contains the mini operating system so that the system can function even though there is not anything on the disk drive(s) yet. The customized portion of the boot process accomplished through the custom JumpStart BEGIN script will eventually load the backup software package(s) into slice s1. At that point, the system will be able to access functioning backup software needed to accomplish the restore of the entire boot disk using the environment's backup solution.

Slices s2-s5 contain the boot information (bootblock) for the various hardware architectures of Sun Microsystems' products that run Solaris. The following file is also contained in these slices:

```
.SUNW-boot-redirect
```

which simply contains a single byte, the character '1', to direct the firmware boot PROM program to look for the kernel on slice 1 of the boot device.

Slice s6 is for future enhancements and could contain environment specific configuration and profile files to minimize or eliminate user interaction with the BART during a bare-metal recovery.

Lastly, slice s7 contains compressed tar files of the backup software packages, such as Legato NetWorker and Veritas NetBackup. These files get uncompressed, untarred, and placed into the appropriate directories resident in memory by the customized JumpStart BEGIN script.

Location of the Pertinent JumpStart files on the "Image Disk"

The pertinent JumpStart files that allow for the Solaris Install-like boot process to take place are located on the "Solaris Installation CD-ROM," and now the BART "image disk," in the following location (Solaris 2.6 is used in this example):

```
/s0/Solaris_2.6/Tools/Boot/usr/
    sbin/install.d/install_config
```

The pertinent standard issue JumpStart files found in this location are the following:

rules.ok: JumpStart Installation "RULES" file

install_begin: JumpStart Installation "BEGIN" script; called out by the "rules.ok" file

devsyn_finish JumpStart Installation "FINISH" script; called out by the "rules.ok" file

The BART will replace the rules.ok and the install_begin files with its own versions. It is important to name the BEGIN script the same as the BEGIN script being called out in the customized version of the rules.ok file. Similar to the CART, the BART does not need to call out a FINISH script, and thus, the devsyn_finish is removed from the "image disk." Correspondingly, the rules.ok file does not make reference to a FINISH script. Since, the customized BEGIN script ends with a reboot of the system, even if a JumpStart profile or JumpStart FINISH script were placed in the rules.ok file, they would never get invoked.

Customized JumpStart BEGIN script actions

Upon booting from the BART CD-ROM, a mini-root operating system gets placed on the target system. The BART then proceeds through the customized JumpStart BEGIN script, "bart_begin" to perform the bare-metal recovery. Some of the more salient actions performed by this BEGIN script are outlined below:

1. The BART queries the user whether to run in interactive mode or profile mode. If interactive mode, the BART further queries the use for specifics regarding the environment:
   - Legato NetWorker or Veritas NetBackup
   - Name of master backup server
   - IP Address of backup server
   - Name of target system
   - Name target system is known as by the backup server
   - IP Address of target system
2. Creates a fully writable directory, /tmp/BART_custom
3. /etc/inetd.conf file needs to be installed
4. If appropriate, uncompresses and untars "networker.tar.gz"; place an installation of Legato NetWorker under /nsr
5. If appropriate, uncompresses and untars "openv.tar.gz"; place an installation of Veritas NetBackup under /usr/openv
6. If running NetWorker, need to start daemon through /etc/rc2.d startup scripts

Slice	Partition	Contents	Size
0	a	Install Directories & Dist.	160 MB
1	b	Mini-root	40 MB
2	c	Boot Info – sun4c	1 cylinder
3	d	Boot Info – sun4m	1 cylinder
4	e	Boot Info – sun4d	1 cylinder
5	f	Boot Info – sun4u	1 cylinder
6	g	Config + Profile Files	10 MB
7	h	Compressed tar files of Backup S/W Packages	440 MB

Diagram 1: The BART "Image Disk" layout.

7. Discover Network Interfaces
8. Display network interfaces found and ask which default network interface to use
9. Display a default IP subnet netmask and ask which default IP subnet netmask to use
10. Activate specified network interface
11. Ask which BACKUP Program to use:
    a. Legato NetWorker
    b. Veritas NetBackup
12. Ask for the master backup server hostname
13. Ask for the master backup server IP Address
14. Possibly use nslookup to gethostbyaddr then display IP Address or "Don't Know IP Address"
15. If NetBackup, then
    a. ask for client hostname which the backup server knows the client
    b. bp.conf file format:

```
SERVER = $master_name
     (received from query)
CLIENT_NAME = $client_name
     (received from query)
```

    c. Place the master_name, client_name, client_name_server_knows_as information into the /etc/hosts file
16. If NetWorker, then
    a. query for NetWorker server name and build /tmp/startup
17. Have we been running the save_root_vtoc.sh? If YES,
    a. Recover from backup vfstab → /tmp
    b. Recover file to /tmp
    c. Format disk drive with file

    If NO,
    a. Query for disk partitioning information
18. Partitions the disk(s) appropriately using the target system's disk partition table.
19. Creates filesystems on the partitions.
20. Utilizes the local Backup software to restore from the enterprise backup server data for all the filesystems on the disk(s).
21. Places appropriate bootblock on the boot disk drive.
22. Reboots the system.

Requirements of the BART

1. Saved copies of volume table of contents, vtoc, for all disks on the target system.
2. A saved copy of the target system's /etc/vfstab file.
3. A good backup of all data for the target system exists in the environment's backup solution.
4. Run the save_root_vtoc.sh on all hosts in the environment everyday and save the info.
5. A CD-ROM drive exists on the target system.
6. The target system can access the network.
7. The target system is known by the environment's backup server.

How the BART Functions in the Networked Environment

Diagram 2 depicts how the BART works in a networked environment. The steps to rebuild or clone a target system involve the following:

1. Place the BART CD-ROM in the CD-ROM drive of the target system indicated by step (1) in the diagram and boot the system via the CD-ROM.
2. On the console of the target system, the BART will query the user whether the restore should get the environment information via interactive mode or from reading a profile. If the user selects the profile mode for data entry, several options are available. The next query is for the user to enter the full path to where the profile can be found; typically this will be either the floppy drive (2a) or an exported filesystem from an NFS BART profile server (2b). A secondary CD-ROM device could also be used, but most systems do not possess more than one.
3. By gleaning information from the interactive mode or the profile mode, the target system will

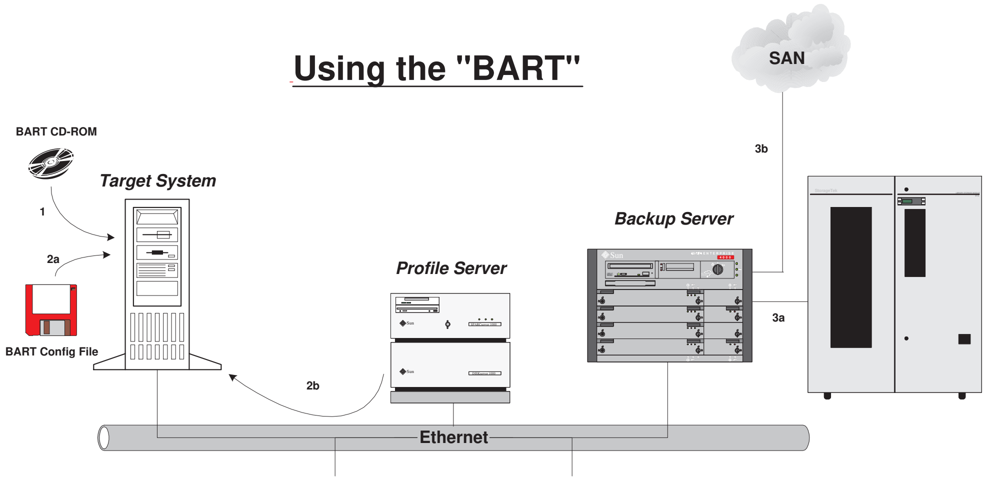

Diagram 2: How BART works in a networked environment.

be able to locate the appropriate backup server in the environment. At this point, BART does not care whether the data comes from direct attached storage (3a) to the backup server or from a Storage Area Network, SAN, (3b).

Testing the BART

The authors have used and customized the BART at various Fortune 500 clients, especially W. Curtis Preston, who has been involved in designing and implementing their Enterprise Backup & Recovery and Enterprise Disaster Recovery solutions. The BART has been a proven success at the clients where it has been employed.

Limitations of the BART

1. Currently implemented with Legato NetWorker and Veritas NetBackup. The tool can also be developed to handle other backup software products as well.
2. Currently, only handles Solaris operating system due to the special JumpStart feature.

Conclusion

For large enterprise sites with elaborate disaster recovery plans that include data mirrored to remote locations, the BART may not prove to be of value or of need. However, for the small to medium sized enterprise where budget constraints have not allowed for the desirable disaster recovery plan, or for the large site that does not have elaborate disaster recovery plans implemented, this tool may very well prove to be a life saver.

Similar to the CART, the BART can also be implemented on a networked JumpStart Server. However, the BART was specifically developed for the consultant who specializes in Backup and Recovery administration. By building the BART on a bootable CD-ROM, this professional can use it at any client site without having to setup or get familiar with an existent JumpStart server in the client's environment.

Resources

The following freeware products from Joerg Schilling [9] were used in the development and implementation of the BART:

cdrecord: A program for creating single/multiple session CD-R on a SunOS, Solaris, Linux, *BSD/SGI, HP-UX, AIX, NeXT-Step, or Apple-Rhapsody system.

sformat: A program to format/analyze/repair SCSI hard disks on a SunOS, Solaris, or Linux system.

scg: A driver to send any SCSI command to any SCSI device on a SunOS or Solaris system.

fbk: A driver to mount a file containing a filesystem; (File simulates Block device on Solaris).

mkisofs: Puts files in ISO-9660 format.

A "Smart and Friendly" CD-RW 426 Deluxe CD-Recorder was used in the development and final implementation. There were not any issues encountered with the installation or use of the cdrecord products or with the use of the "Smart and Friendly" CD-RW 426 Deluxe CD-Recorder. Both of these products receive a high endorsement from the authors.

Other CD-R recording hardware and software products (i.e., Young Minds, Inc., HyCD, Gear to name a few) could have been integrated into the CART as well. However, the price of cdrecord and its associated products could not be beat.

Also, of great value to the development of the BART is the following freeware product from Matthew R. Green:

mksunbootcd: combines filesystems for Sun Microsystems computers for creating bootable compact disc images.

Acknowledgements

We thank The Storage Group, Inc. for proposing the initial concept of the BART and for providing the equipment necessary to design, build, and test it.

We thank the "Publications Group" of Collective Technologies for providing editing expertise for this paper.

We thank Adelaida Esquivel for also providing editing expertise for this paper.

We thank Joerg Schilling whose freeware products were indispensable in the creation and final product of the BART due to their ease of installation, use, and an unbeatable cost.

Author Information

Lee "Leonardo" Amatangelo was graduated from the University of California, Irvine in 1983 with a B.S. in Molecular Biology and in 1985 with a B.A. in Anthropology. He has been working in the computer industry since 1981. Currently, he is a systems management consultant specializing in Solaris and disaster recovery for Collective Technologies. He can be reached via email at leonardo@colltech.com or lamat@earthlink.net and by physical mail at Collective Technologies, 9433 Bee Caves Road, Building III, Austin, TX 78733.

W. Curtis Preston is the President of The Storage Group, Inc. (http://www.thestoragegroup.com), and has been specializing in storage for over seven years and has designed, implemented, and audited enterprise-wide backup and recovery systems for many Fortune 500 and e-commerce companies. His O'Reilly & Associates book, UNIX Backup & Recovery, has sold over 20,000 copies, and he writes a regular column for UnixReview online and SysAdmin magazine. Curtis is also the webmaster for backupcentral.com, and can be reached at curtis@thestoragegroup.com.

References

[1] Amatangelo, Lee "Leonardo," "Unleashing the Power of JumpStart: A New Technique for Disaster Recovery, Cloning, or Snapshotting a Solaris System," *LISA XIV Conference Proceedings*, 2000.

[2] Kasper, P. A. and A. I. McClellan, *Automating Solaris Installations – Custom JumpStart Guide,* SunSoft, Prentice Hall, 1995.

[3] Nemeth, E., G. Snyder, S. Seebass, and T. Hein, *UNIX System Administration Handbook*, Second Edition, Chapter 9, Prentice Hall, 1995.

[4] Preston, W. Curtis, *Unix Backup & Recovery*, O'Reilly and Associates, Inc., 1999.

[5] Sun Microsystems, *Solaris 2.6 – Solaris Advanced Installation Guide*, Revision A, Mountain View, CA, Part No. 802-5740-10, August 1997.

[6] Sun Microsystems, *Solaris 8 – Advanced Installation Guide,* Mountain View, CA, Part No. 806-0957-10, February, 2000.

[7] Zuberi, A., "JumpStart in a Nutshell," *Inside Solaris*, Chapter 1, February, 1999,

[8] http://www.fadden.com/cdrfaq/faq00.html#[0-1] .

[9] http://www.fokus.gmd.de/research/cc/glone/employees/joerg_schilling/private/ .

[10] http://www.smartandfriendly.com/ .

[11] http://www.ymi.com/ .