Unleashing the Power of JumpStart: A New Technique for Disaster Recovery, Cloning, or Snapshotting a Solaris System

Lee "Leonardo" Amatangelo – Collective Technologies

ABSTRACT

Due to the demand of 24x7 coverage by present day data centers, the need for a proven disaster recovery plan is a must. To assist in providing disaster recovery for systems running Sun Microsystems' Solaris 2.x (SunOS 5.x) operating system, a tool was developed which captures the image of the system to one or more volumes of optical media with the first volume being bootable. The optical media used by this tool is CD (Compact Disc) with hooks put in place for DVD (Digital Video Disc or Digital Versatile Disc).

This tool was developed with the following objectives in mind: (1) no magnetic media; (2) bootable media; (3) minimal user interaction; (4) handle multiple volume sets; (5) handle environments that do not use a Network Information Service (NIS or NIS+); and, (6) handle environments that do not use Network File System (NFS). The power of this tool is made possible by utilizing two special features present in the Solaris operating system, namely the installboot utility and the JumpStart mechanism (as implemented on the Solaris 2.x Install CD).

The complete process of capturing and restoring a Solaris system's image to bootable optical media involves five phases: (1) pre-imaging preparation; (2) setup of target host; (3) capture image of target host; (4) burn image to media; and, (5) restore image to target host. Each of these phases is controlled by one master Bourne shell script. The overall tool is implemented in 26 Bourne shell scripts controlled by the 5 master scripts. This tool, known as the "CART" (Capture And Recovery Tool), was placed on a mobile cart and consisted of the following components: UltraSparc 10, internal CD-ROM drive, internal floppy drive, 256 MB memory, 2 network interface cards (NIC), keyboard, mouse, monitor, external SCSI 18 GB disk drive, external SCSI CD-RW, and external SCSI DLT.

This tool provides the following functions: (1) bare-metal recovery; (2) capture a snapshot of a system to optical media; (3) cloning a system; and, (4) rollout multiple clones of a system via optical media. Using the principles discussed in this paper and creating additional Bourne shell scripts, the CART has been modified to provide the following additional outputs: (1) make copies of the Solaris Install CD; (2) make customized Solaris Install CD (essentially a customized JumpStart from CD); and, (3) a specialized bootable CD for disaster recovery that assists third party Backup and Recovery utilities (Legato Networker and Veritas NetBackup).

Introduction

With the increase of e-commerce on the Internet, there is an ever-increasing demand to have mission critical computer systems run 24 hours by 7 days for long periods between scheduled down times. When a mission critical system suffers from corrupted disk data, either software or hardware induced, the disk data will need to be restored. Software induced data corruption include application malfunctions that render data inaccessible or a user accidentally entering a data destructive keystroke such as rm -r. Hardware induced data corruption include physical disk crashes. In the case of physically damaged disk drives, the disk drive of course will need to be replaced prior to the restoration of software and data.

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 there is no longer a viable operating system running (nor present) on the system. 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 install utility, JumpStart, can take what might be considered too much time, because following the installation of the operating system and any customized packages, the Jumpstart process will still need to initiate the installation of the desired patch sets. The application of a large set of patches can take up to a couple of hours.

Due to the lengthy time it sometimes takes to patch a system, Hewlett-Packard's automated install utility for the HP-UX operating system, "Ignite-UX," can prove to be more efficient time-wise to install a system. When Ignite-UX is finished, the system will be complete with operating system, applications, patches, kernel modifications, and any specialized configurations. Ignite-UX uses the concept of saving an entire system image of a benchmark host, known as a "golden image," and then restores the image during the automated rollout. The Ignite-UX process essentially performs a bit-by-bit copy, and therefore, is much faster than the JumpStart process, which builds the target system from the ground up by installing the operating system, software packages, patches, and any additional applications and files desired.

Taking the best of both the JumpStart and Ignite-UX worlds, a tool for providing disaster recovery, cloning, or snapshotting has been created specifically for Solaris systems.

The Capture And Restore Tool – The "CART"

During its evolution, this tool passed through a client site that needed to maintain the utmost in security. As such, the client site did not run network services to ease the administration nightmare of maintaining the numerous users, hosts, and system administrative files (usually kept in the /etc directory). Thus, this client did not use Network Information Services, NIS, nor its more secure brother, NIS+. This client also did not use the Network File System, NFS, to share resources over the network for fear that such a service was not secure enough for the very confidential information that was stored on the network. In order to accommodate this client, the tool was made to be self-contained and lived on a mobile cart so that the tool could be easily negotiated through the data center. The system hosting the tool will be referred to as the "control host" while the system to be imaged will be referred to as the "target host." The two hosts will talk to each other via a directly connected network cable.

This mobile tool physically consists of the following components: UltraSparc 10, internal CD-ROM drive, internal floppy drive, 256 MB memory, 2 network interface cards (NIC), keyboard, mouse, monitor, external SCSI 18 GB disk drive, external SCSI CD-RW, and external SCSI DLT. Additionally, the system has a network cable attached to one of the NICs and an RS-232 serial cable attached to serial Port B. The RS-232 cable was a convenience added to handle headless hosts via the 'tip' utility.

The main purpose of the tool discussed in this paper is to capture and restore system images. Furthermore, the tool physically resides on a mobile cart. Because of these two reasons, the tool became known as, and will be referred to during the rest of this paper, as the "Capture And Restore Tool," or simply the CART. Figure 1 diagrammatically depicts the physical layout of the CART.

Two Special Solaris Features

The secret and power of the CART is brought about by two very special features found in the Solaris 2.x operating system. These two features are specifically the installboot utility and the JumpStart mechanism. The former facilitates the placing of a bootblock on storage media while the latter facilitates a way to customize the resultant boot process. These two features are further explored below.

How to Make Solaris Storage Media Bootable

One of the objectives of the CART was to restore a system via bootable optical media. The first question to ask is, "Is it possible to make bootable media under the Solaris 2.x operating system?" The answer is a resounding YES! The solution is to use the *installboot*(1M) utility. The installboot utility, uniquely found in the UNIX command sets distributed in Sun Microsystems' Solaris 1.x (SunOS 4.x) and Solaris 2.x (SunOS 5.x) operating systems, provides the capability to install a bootblock on various types of storage media. Furthermore, the bootblock can be placed on any of the slices (partitions) on the target media.

The second question to ask is, "Can a bootblock be placed on optical media?" Again, the answer is a resounding YES! As a matter of fact, the bootblock can be placed on various types of bootable media, such as a floppy diskette, hard disk drive, CD-R, CD-RW, CD-ROM, DVD-R, DVD-RW, DVD+RW, DVD-ROM, and DVD-RAM. When the bootblock program resides in the boot area of a disk partition, the bootblock program will load the *boot*(1M) program, also known as ufsboot. The ufs boot objects are platform-dependent, and reside in the /usr/platform/'uname -i'/lib/fs/ufs directory, where 'uname -i' gets the correct platform-name. The full command syntax for installboot is as follows:

```
installboot bootblock raw-disk-device
```

where:

bootblock is the name of the bootblock code.

raw-disk-device is the name of the disk device onto which the bootblock code is to be installed; it must be a character device which is readable and writable. Naming conventions for a SCSI or IPI drive are of the format: c?t?d?s?. Naming conventions for an IDE drive are of the format: c?d?s?.

For a thorough explanation of the installboot utility, see the Man Pages on *installboot*(1M). In short, the installboot utility was used to install the necessary bootblocks so that the CART could handle the various Sun Microsystems platform architectures. More on how this was accomplished later.

A Brief Discussion on JumpStart

There are plenty of books on the subject of JumpStart [1, 3, 6], and as such, the scope of this paper is not to teach JumpStart. However, a brief discussion of JumpStart will help elucidate its role and importance as implemented in the CART. The JumpStart feature was introduced by Sun Microsystems and SunSoft in 1992 and first appeared in the Solaris 2.x operating system [3]. The JumpStart feature is primarily used to install multiple client hosts over the network simultaneously with the intent to lessen this tedious and time-consuming task. The various hosts can have their own unique configuration if desired or groups of hosts can have like configurations. JumpStart is a utility to help facilitate the ease of rolling out the Solaris operating system to new hosts (or existent hosts to be upgraded) in an environment and is especially useful for rolling out a large number of installs across an enterprise [2, 5].

Typically, JumpStart is used over a network to allow the automated software installation rollout of multiple hosts simultaneously. Once configured, the JumpStart mechanism is very much hands off. To facilitate the JumpStart mechanism over a network, a networked JumpStart server is needed. But JumpStart servers are not the only place that one will find the implementation of JumpStart. Sun Microsystems has also incorporated the JumpStart mechanism on the bootable installation media (i.e., the Solaris 2.x Install CD) as the means for installing the Solaris operating system.

The JumpStart mechanism has four main parts: (1) a BEGIN script which performs pre-processing actions to take place prior to the software installation; (2) a PROFILE which defines how Solaris is to be installed on a particular client; (3) a FINISH script which performs any post-processing actions following the software installation; and, (4) a rules file (rules.ok) that dictates which BEGIN script, PROFILE, and FINISH script to use.

Because JumpStart offers the following three characteristics: (1) can be configured to have minimal user interaction; (2) can be placed on optical media; and, (3) can be invoked seamlessly following a boot, JumpStart was selected hands down as the method of choice for the CART.

As will be described in the next section, the CART ended up customizing JumpStart in a completely different way and was able to unleash the true power of JumpStart!

The "CART" in a Nutshell

To make use of the existent JumpStart technology as implemented on the Solaris Install CD, the CART performs the following steps: (1) partitions a hard disk drive (which will be referred to as the "image disk") to have slices that mimic (size and location) the Solaris Install CD; (2) creates filesystems on the slices; (3) copies select contents of the Solaris Install CD to the corresponding slices on the "image disk"; (4) opens up the permissions on the directory and files where the JumpStart mechanism resides; (5) replaces the standard issue JumpStart BEGIN script, PROFILE, FINISH script, and RULES.OK file with its own highly customized versions; and, (6) installs the appropriate platform-specific bootblock that corresponds to the platform of the "target host." See Figure 1 for a quick overview of CART's hardware configuration.

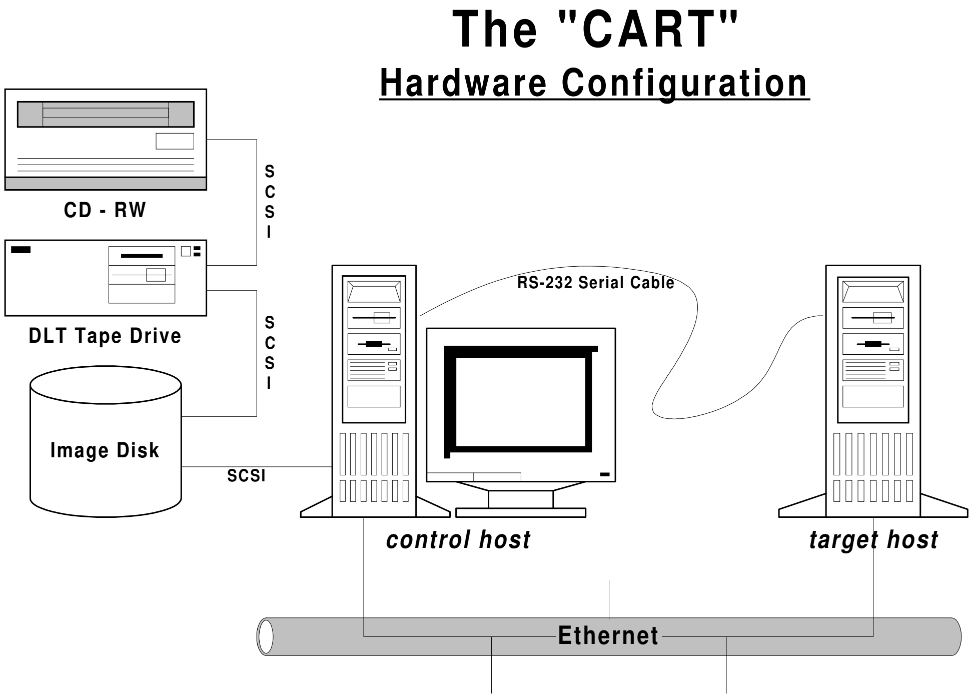

Figure 1: CART hardware configuration.

At this point, if the contents of the ''image disk'' is burned to optical media, that optical media will be bootable and will go through an automated install based on instructions in the customized JumpStart files, specifically the customized BEGIN script and the RULES.OK file. The customized BEGIN script ends with a system call to reboot the system. This reboot essentially causes the normal JumpStart process of loading packages, patches, etc. not to occur. In short, the CART makes use of the Solaris Install CD JumpStart mechanism only as a means to provide the automatic running of a script following the boot from optical media. It was decided not to reinvent the wheel but instead to modify an existent proven technology. The steps just described are what truly give the CART its power.

This technique can be used for producing a wide range of useful applications, some of which will be discussed later in this paper. The final implementation of the CART did not use a PROFILE or a FINISH script. However, a customized FINISH script could be implemented to initiate backup software to restore variable data from the latest backup of the system.

After the ''image disk'' has been prepared successfully, the next task of the CART is to capture the entire system image of the ''target host.'' To accomplish this task, a private network is setup between the ''control host'' and the ''target host'' with NFS mounts established between the ''target host'' and the ''image disk'' (located on the ''control host''). Next, the ''target host'' is interrogated to determine its number of disk drives. The partition table of each disk drive is logged capturing the slices used and the sizes of those slices. The CART then methodically runs through each disk drive and captures an image of each slice. As the image for each slice is captured, the CART logs the size of the slice image to a file. The size of each slice is captured for two reasons. First, to determine if an individual slice image will be able to fit on a single optical media volume. This is important in the case when saving to CD because a single volume can only hold 650 MB. The second reason is to determine how many optical media volumes will be required to save the entire system image of the ''target host.''

Once the entire system image of the ''target host'' is captured, which typically will be comprised of several individual slice images, the entire system image must be burned to the optical media of choice. The log containing the size information of the individual slice images is used to determine which slices get placed on which volume. The user of the tool will be informed how many volumes will be required to burn the entire system image and as each volume is burned the user is requested to remove the volume and insert the next volume.

Once all of the optical media volumes have been successfully burned, the result will be a snapshot of the entire ''target system'' captured onto a set of optical media, with the first volume being bootable and having the smarts to recreate the ''target host.''

The last step to complete this eventful saga is to perform the restore of the image from the bootable media, either to the original ''target host'' or to an identical host with as large or larger disk drives. The restore takes place at the Open Boot PROM (OBP) level by booting from the optical media.

The Five Phases of the "CART"

The capture and restore of a Solaris system image using the CART involves five distinct phases.

Phase 1, the pre-imaging preparation of "image disk" phase, prepares a buffer area (hard disk space) known as the ''image disk.'' The ''image disk'' will have appropriate slices (partitions) and filesystems created for holding select directories and files of the Solaris Install CD, as shown in Table 2. These select directories and files contain the JumpStart mechanism and the bootblocks for the various Sun Microsystems architectures. In addition, the ''image disk'' will have a separate slice to hold log files during the capture phase and another slice to receive the compressed filesystem images from the target host.

Phase 2, the setup of target host phase, establishes a private network connection and NFS mounts between the ''target host'' and the ''control host'' that has access to the ''image disk.''

Phase 3, the capture image of target host phase, determines the number of disks on the target host, the number of partitions on each disk, and then proceeds to capture (andufsdump) compress (compress or gzip) each filesystem found. All disks, partitions, and compressed image file sizes are logged during the process. These logs will be used during the following ''burn'' and ''restore'' phases.

Phase 4, the burn image to optical media phase, creates a bootable volume and any additional volumes required for holding the complete system image. If the optical media is CD-R and the entire set of compressed images from the target host exceeds 650 MB, then the appropriate number of additional CD-R volumes will be created. If the optical media is DVD-R and the entire set of compressed images from the target host exceeds 3.9 GB, then the appropriate number of additional DVD-R volumes will be created.

Phase 5, the restore image phase, involves restoring the system image to the same or equivalent target host via bootable media. The boot process of the restore proceeds through a highly customized JumpStart process located on the bootable media itself. If the entire system image is contained on more than one volume, then an additional CD drive or DVD drive (depending on which optical media is being used) will need to be attached to the host receiving the image.

Commands for the Five "CART" Phases

The commands to invoke each of the five phases of the CART are listed below:

Phase 1: Pre-Imaging Preparation of "Image Disk". The command for this phase is entered on the "control host" and can be run prior to connecting the "control host" to the "target host":

```
prepare_image_disk
```

Phase 2: Setup of Target Host. The command for this phase is entered on the "target host" after installing the "setup" floppy diskette, CD, or DVD:

```
setup
```

Phase 3: Capture Image of Target Host. The command for this phase is forked on the "target host" by the Setup phase (if the setup process is successful):

```
capture_system_image
```

Phase 4: Burn Image to Optical Media. The command for this phase is entered on the "control host":

```
burn_system_image
```

Phase 5: Restore Image. The command entered at the open boot PROM 'ok' prompt on the original "target host" or a similar host that is to receive the image:

```
ok boot cdrom
```

or

```
ok boot dvd
```

(The second option currently being developed by Sun Microsystems as a standard option in the OBP.)

A Closer Look at the Solaris Install CD

The Solaris Install CD was reverse-engineered to determine how it boots and goes through its JumpStart process. The first area looked at was the logical layout of the Solaris Install CD itself. Table 1 reports this layout, specifically for the Solaris 2.6 Install CD.

The first volume in the set of media containing the system image of the target host will emulate the Solaris Install CD layout above with three minor modifications. First, "slice 0" is trimmed down to contain only the bare essentials to have the CD boot and run through the customized JumpStart process. Second, a seventh slice, "slice 6," is added to the CD. This slice will contain the log files generated during the image capture phase. Third, an eighth slice, "slice 7," is added to the CD to hold some (if not all) of the compressed image files from the "target host." The emulation of the Solaris Install CD with these modifications is captured on the "image disk" and is reported in Table 2.

Slice	Partition	Contents	Size
0	a	Installation and Distribution	600 MB
1	b	Miniroot	40 MB
2	c	Boot Info, sun4c*	1 cyl.
3	d	Boot Info, sun4m*	1 cyl.
4	e	Boot Info, sun4d*	1 cyl.
5	f	Boot Info, sun4u*	1 cyl.

Table 1: Solaris Install CD Layout.

* *Note that these slices contain the boot information (bootblock) for the various hardware architectures of Sun Microsystems' products that run Solaris. Also contained in these slices is the file .SUNW-boot-redirect which 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	Part.	Contents	Size
0	a	Installation and Distribution	200 MB
1	b	Miniroot	40 MB
2	c	BootInfo – sun4c *	1 cyl
3	d	BootInfo – sun4m *	1 cyl
4	e	BootInfo – sun4d *	1 cyl
5	f	BootInfo – sun4u *	1 cyl
6	g	Log Files	11 MB
7	h	Compressed Image Files	Rem'dr

Table 2: Image Disk layout.

* *The contents of these slices are as described in the note in Table 1.*

Also note that the CART implemented an 18 GB disk drive for the "image disk." The size of the "image disk (or disks)" is determined by the size of disk space on the target host.

Location of the JumpStart files on the Solaris Install CD and the "CART"

During the "pre-imaging preparation" (Phase 1), select portions of the Solaris Install CD are copied to the "image disk." Two specific portions are copied: (1) all of "slice 1," which contains the miniroot (the operating system); and, (2) parts of "slice 0," which contains the pertinent files to allow the JumpStart mechanism to function. The location for these files is as follows:

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

The pertinent standard issue JumpStart files are the following:

rules.ok Install JumpStart "RULES" file

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

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

How the "CART" Customized JumpStart

Since, the JumpStart files (listed in the section above) now exist on the "image disk," and the tool runs as 'root' user, the permissions on the location and the files themselves can be modified. Once the permissions are changed, the files can be modified (replaced) with highly customized versions, which is precisely the approach taken by the CART to highly customize the restore (Phase 5).

The customization involved replacing the "rules.ok" file and adding a custom JumpStart BEGIN script, named "restore_system_image_begin," to the JumpStart location on the "image disk." The CART did not use the standard issue JumpStart files "install_begin" and "devsyn_finish." Furthermore, the CART did not need a custom JumpStart FINISH script at all. All the actions the CART needed to accomplish were performed in the custom BEGIN script "restore_system_image_begin." These actions included verifying all of the appropriate disks are present, the disks then get partitioned, filesystems get created, and one by one, the filesystem images get restored. As additional volumes containing the system image are needed, the user is instructed to remove the current volume and insert the next. When all of the filesystems are restored and an appropriate bootblock has been installed, the customized BEGIN script reboots the system. When the "target host" finishes rebooting, it will look like the original host. The reboot at the end of the customized BEGIN script also has another side effect, it aborts the rest of the JumpStart process. Thus, the system does not try to load the distribution, packages, nor patches. Instead, the completed process looks more like an Ignite-UX "golden image" restore.

Functional Details of the Five "CART" Phases

For those who want to know the nitty-gritty details of the entire CART process, the functional details for all five phases are itemized below.

Phase 1: Pre-Imaging Preparation of the "Image Disk"

1. Requests user to perform physical setup
2. Unmounts the "image disk"
3. Partitions the "image disk"
4. Creates filesystems
5. Makes mount points for mounting the "image disk"
6. Mounts the "image disk"
7. Places proper permissions on the "image disk"
8. Copies the contents of the "Solaris Install CD" to the "image disk"
9. Removes the "lost+found" directories from slices s0-s7
10. Installs the various architecture bootblocks
11. Opens the permissions on the JumpStart locations in slice s0 on the "image disk"
12. Creates additional directories with proper permissions in the miniroot filesystem (slice s1) of the "image disk"
13. Share the "image disk" and other local shares for the NFS mounts from the target host

Phase 2: Setup of Target Host

1. Creates the work directory on the target host
2. Saves the original /etc/hosts file
3. Reassigns a temporary IP Address
4. Determines the network interface type
5. Saves the original network interface settings
6. Creates a virtual network interface
7. Establishes a network connection between the target host and the master control host
8. Logs all changes so the original configuration can be re-established when complete
9. Forks the execution of the "capture image" script

Phase 3: Capture Image of Target Host

1. Determines the physical device name of the optical media (CD or DVD) drive
2. Saves /etc/vfstab file information
3. Determines the number of disk drives on the system using format
4. Saves the partition information of each disk drive using prtvtoc
5. Performs ufsdumps of the individual filesystems
6. Compresses the filesystem dump files
7. Calculates the number of media volumes that will be required to hold the entire image (compressed dump files) of the system
8. Places the appropriate bootblock on the "image disk"
9. Copies the customized JumpStart BEGIN script to the JumpStart location on slice s0 of the "image disk"
10. Copies the customized "rules.ok" file to the JumpStart location on slice s0 of the "image disk"
11. Restores the original /etc/hosts file
12. Restores the original network interface settings

Phase 4: Burn Image to Optical Media

1. Kills the volume management if it is running
2. Checks that the "image disk" is mounted
3. Reads the number of volumes to create from the log files
4. Puts files in ISO-9660 format
5. Creates (burns) the bootable first volume
6. Writes a label on the first volume indicating Volume 1 of 'x' number of volumes and target host name
7. Creates (burns) all additional volumes
8. Writes a label on each volume indicating Volume 'y' of 'x' number of volumes and target host name

Phase 5: Restore Image

1. The target host to receive the image is booted from the first volume of the optical media set

2. Customized JumpStart located on the first volume of the optical media automatically gets initiated
3. Mounts slice 6 of the optical media to /log_dir
4. Uses 'fmthard' to recreate the VTOC of the original disks to the target host disks
5. Creates new filesystems on the appropriate slices of the new disks to match the filesystems from the original disks
6. Mounts one at a time the intended slices on the new disk to the mount point /mnt
7. Uncompresses and ufsrestores the compressed image files
8. Places the pertinent bootblock on the boot device

Handling the CD Mounts During Installation

This paper would be remiss if it did not describe the manner in which the CART mounts the bootable first volume upon a restore of the image (Phase 5). The mounting of the bootable CD is identical to how a Solaris Install CD gets mounted during an install. This information is important because it elucidates one of the limitations of the CART, that being that if the CART produces a multi-volume set during the burning of the media (Phase 3), then an additional CD drive will be required during the restore of the image (Phase 5).

The details on how the contents of the bootable CD get mounted at boot time, is shown in Table 3.

/tmp	/tmp
/proc	/proc
/devices	/tmp/devices
/dev	/tmp/dev
/	/devices/pci@1f,0/pci@1,1/ide@3/atapicd@2,0:b
/cdrom	/devices/pci@1f,0/pci@1,1/ide@3/atapicd@2,0:a
/dev/fd	fd

Table 3: The boot-time mounts of the CART bootable CD.

Note: The information in this table is for an Enterprise 250. The / and /cdrom mounts will vary depending upon the hardware platform being used.

Table 3 reports that the miniroot (slice b on the optical media) represented by the line ending with a ":b," is being mounted to the / root mount point. The miniroot is the operating system being used for the restoration (in the case of the CART) or the installation (in the case of the Solaris Install CD). Since the operating system must always be present, the first volume must always be mounted during the entire restore (or install) process. Thus, a second media (CD or DVD) drive must be attached to the "target host" in order to restore a system image contained on a multi-volume set.

When booting from a CART generated bootable first volume or from a Solaris Install CD, the CD will get mounted as in Table 3.

Optical Media

The optical media used by the CART is CD and DVD.

CD technology is attractive because it is inexpensive. DVD technology is attractive because it can hold much larger capacities.

After speaking with several vendors [10, 11, 12, 14], the same message was echoed that there is really only one vendor, Pioneer, that provides a DVD writer. The vendors that provide CD/DVD recording solutions (hardware & software) use the DVD writer specifications from Pioneer.

In terms of the writable media, there is a dramatic increase in cost going from CD to DVD: ~$1 per 650 MB CD vs. ~$35 per 3.95 Billion Bytes DVD (which is not really 3.95 GB in the computer sense of the word but rather in the true mathematical sense of the word – which is a marketing ploy; see the note under the "DVD Technology Capacity and Compatibility" section below).

So far, Young Minds, Inc., (YMI), a leading CD recording technology vendor, is the only vendor I found working on a DVD solution for UNIX. In early 2000, YMI was awaiting hardware and specifications from Pioneer. Furthermore, the current CD solution from YMI is scalable to handle DVD; all that is required is a DVD writer and a PROM update to the YMI CD Studio hardware.

Unlike the CD technologies, where the CD-R, CD-ROM, and CD-RW versions all have the same capacity, each version of DVD technology has a different capacity. See tables below. The varying capacities of DVD technology will become a source of confusion for the consumer primarily because the various DVD technologies will not be compatible with one another. Currently, for non-video applications, the DVD technology market still needs to sort itself out. On the other hand, the CD technology market is well established.

To learn more about CD and DVD technology, turn to the web [7, 8].

CD Technology Capacity and Compatibility

Currently, there are four CD versions: CD-R, CD-RAM, CD-ROM, and CD-RW. All have the same capacity and are compatible with one another:

Type	Capacity
CD-R	650 MB
CD-RAM	650 MB
CD-ROM	650 MB
CD-RW	650 MB

DVD Technology Capacity and Compatibility

The DVD standards committee is still trying to decide on the DVD standards [7, 14]. Currently, there are five DVD versions: DVD-R, DVD-RAM, DVD-ROM, DVD-RW, and DVD+RW. These versions have varying capacities, and thus, are mostly incompatible with one another:

Type	Capacity
DVD-R	3.95-4.7 Billion bytes
DVD-RAM	2.6-4.7 Billion bytes
DVD-ROM	4.7 Billion bytes
DVD-RW	4.7 Billion bytes
DVD+RW	3.0-4.7 Billion bytes

As an aside, these capacities are truly in billions (1,000,000,000) of bytes and not a Gigabyte (GB) in the computer sense of the word, where a GB is defined as 1024 x 1024 x 1024 or 1,074,790,400 bytes.

Where Do We Go From Here

There are a couple of areas that the author would like to explore and develop in the very near future. First, modify the CART to work in a networked environment. Second, develop similar tools for other *NIX operating systems.

Implementing the "CART" in a Networked Environment

To implement the CART in a networked environment running NIS or NIS+ and NFS would have been much simpler. The "control host" could be placed on the network, and if a naming service is running, then all hosts will know about each other. Three types of CART servers can be placed on the network: (1) a standalone CART server; (2) a JumpStart server providing only CART functionality; and, (3) a JumpStart server providing both CART functionality and JumpStart functionality. A method for altering the BEGIN script to perform the restore (clone) can be implemented fairly easily. Thus, the JumpStart server can be used to perform site-specific JumpStarts as well as the CART method of imaging. Instead of maintaining the images of "target hosts" on optical media, the images can be stored on network disk space and accessed through NFS. Again, this would be an entirely different way to unleash the power of JumpStart.

Implementing the "CART" in Other *NIX Operating Systems

One of the obvious next steps is to investigate implementing this tool in other *NIX operating systems such as HP-UX, IRIX, AIX, Digital UNIX, FreeBSD, BSDI, and Linux. The first operating system to tackle on the list will be HP-UX, since it already has the fully developed Ignite-UX utility.

Limitations of the "CART"

1. The biggest limitation to the CART is encountered when the target host contains relatively large filesystems (greater than 1 GB) and CD-R is chosen as the optical media to store the system image. The limitation lies in the maximum carrying capacity of a CD-R which is 650 MB. By using compression utilities (compress or gzip) in this tool, and assuming about 60% compression efficiency, the maximum filesystem size that can be captured is about 1000 MB or 1 GB. So, when capturing to CD the system image of a target host that contains at least one filesystem larger than 1 GB, the CART will report and error. The error is due to the resultant compressed file for the filesystem exceeding the carrying capacity of the CD. However, there still is hope for systems containing filesystems larger than 1 GB. The system image could and should be captured to DVD-R, since the carrying capacity of a DVD-R media is about 4 GB. Again, assuming about 60% compression efficiency, a filesystem on the order of about 7 GB can be imaged.
2. A system image saved to a multi-volume set requires an additional optical media drive for the restore operation to work.
3. The system being restored must not only have the same kernel architecture but also the same platform architecture as the system that was imaged. As an example, even though an Enterprise 250 and an Enterprise 450 have the same kernel architecture (sun4u), they have different hardware platforms. When the image from one is restored onto the other, the resultant system will not be bootable.
4. When restoring a system or cloning to a like system, the disk drives in the host being restored must be as big or bigger than the disk drives in the original host imaged.
5. Currently, the CART cannot capture system images of target hosts that have mirrors or RAID volumes (created by Veritas Volume Manager or Solstice Disksuite) on disk drives larger than 1 GB. When such logical volumes exist, the entire disk must be captured using the dd utility to ensure the private region on the disk is captured. By having to capture the entire disk as one image file, even after compressing, the resultant file would not fit on one CD volume. This limitation goes away if the CART is implemented in a networked environment.
6. Currently, the CART is only implemented for the Solaris operating system.

Additional By-Products That Arose From the "CART" or Its Technology

Several by-products resulted from the development of this tool. The possibilities are wide open for many more. The more useful of these by-products are identified below and how they were accomplished.

Make Copies of the Solaris Install CD. How many times have you not been able to locate your Solaris Install media because there were too few to begin with and they are all checked out (or locked in the office of the other system administrator)? Well, now you can make as many copies as you like (keeping in mind copyright laws). Make a slight modification to the Bourne shell script used during Phase 1 that copies select portions of the Solaris Install CD to the "image disk," to have the script copy the entire CD. (Actually, during the initial stages of developing the CART, the original script copied the entire Solaris Install CD to the "image disk" anyway.) Following the completion of the copy, go straight to the burn phase (Phase 3). While the "image disk" is in this state, as many copies that are desired can be burned off, one at a time. Using the CART configuration described earlier in this paper, it took about 30 minutes to copy the entire Solaris Install CD to the "image disk" and about five minutes to burn off a copy.

Make A Customized Solaris Install CD or CD Set. Instead of having the generic Install CD as provided by Sun Microsystems, place some of your Customized JumpStart configurations on the CD. This essentially makes your JumpStart come off of the CD instead of a JumpStart server. Who would want to do this? How about an experienced JumpStart administrator who finds him/herself in a non-networked environment and is faced with a task of rolling out a large number of host installs or upgrades. If there are several additional packages and tarballs needed for your intended JumpStart, additional CDs may be required. By using the same technology incorporated into the CART, having additional CD volumes will not be a problem. However, just like with the CART, an additional CD drive will be needed to perform the JumpStart from CDs.

Make A Specialized Bootable CD. An industry authority on UNIX Backup and Recovery, (W. Curtis Preston) has asked to have a bootable CD built that will contain enough of an operating system to allow third party backup and recovery software (Legato Networker and Veritas NetBackup) to run. Very customized scripts automatically invoked on the CD will prompt the user for certain information. Based on the information provided, the backup software selected will proceed to perform backups over the network to restore a complete system image. The intent of making the CD bootable is for the situation of a boot disk crash. For such situations, this specialized tool will perform a bare-metal recovery (disaster recovery on a virgin disk drive) and will result in bringing the system to the state of its last good backup. The development of this tool is currently in progress and looks quite promising.

Resources

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

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. No issue arose with the use of this CD-RW device during the development and use of the CART.

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. 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 author.

Acknowledgements

First, I thank the *Collective Intellect* of Collective Technologies who provided an invaluable resource and wealth of knowledge on many areas during the development of the CART.

Second, I thank Joerg Schilling. Indispensable in the creation and final product of the CART were several shareware products provided by Joerg Schilling.

Third, I thank Adelaida Esquivel, Senior Computer Programmer, who spent countless hours supporting the process of writing and provided the proofreading of this paper.

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 and by

physical mail at Collective Technologies, 9433 Bee Caves Road, Building III, Austin, TX 78733.

References

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

[2] Heiss, J., ‘‘Enterprise Rollouts with JumpStart,’’ *LISA XIII Conference Proceedings*, 1999.

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

[4] Nemeth, E., Snyder, G., Seebass, S., & Hein, T., *UNIX System Administration Handbook*, 2nd Edition, Prentice Hall, 1995, Ch. 9.

[5] Shaddock, M. E., Mitchell, M. C., & Harrison, H. E., ‘‘How to Upgrade 1500 Workstations on Saturday, and Still Have Time to Mow the Yard on Sunday,’’ *LISA IX Conference Proceedings*, 1995.

[6] Zuberi, A., ‘‘JumpStart in a Nutshell,’’ Inside Solaris, February 1999, Ch. 1

[7] http:/dvddemystified.com/dvdfaq.html .

[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.gearcd.com/ .

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

[12] http://www.pioneerusa.com/ .

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

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