A Web-Based Backup/Restore Method for Intel-based PC's

Tyler Barnett – Lexmark International
Kyle McPeek – Aerotek Inc., on behalf of Lexmark International
Larry S. Lile – Aerotek Inc.
Ray Hyatt, Jr. – Aerotek Inc.

ABSTRACT

This paper describes two similar user-initiated methods to backup and restore PC workstations over an ethernet network. The first method concerns "PC Hardware All Alike," or the backup and restoring of sets of identical disk images across many identical systems. The second method is the backup and restoring of "PC Hardware All Different," which manages unique disk images on individual systems.

This paper describes the installation and customization of a "controlling" unix partition on each client system PC platform. The purpose of the unix partition is to quickly backup and restore Windows and OS/2 disk images on a separate partition on the client system hard drive.

The web page interface is used to schedule the image backups and restores. The unix server that supports the backup and restore process is described, along with the scripts that control the backup and restore process. Usage statistics are generated that can quantify the cost benefits of the backup and restore method.

Background

Lexmark's core business is printing solutions, which are primarily inkjet and laser printers, and the product line changes rapidly. Testing is required of a large amount of printer drivers and networking utilities, all of which are translated to numerous foreign languages. This software testing requires the testers to have at their disposal a client PC platform capable of Win31, Win95, WinNT, and OS/2 to support all the versions of drivers and utilities.[1] This testing mission is loosely referred to as NLS, or National Language Support. From now on, "NLS" is equivalent to the backup and restore mission of "PC Hardware All Alike."

The Problem

Once a printer driver or network utility is installed and tested on a PC, further installation of yet another revision of drivers or utilities would not reflect the customer situation. The "now corrupt" system could possibly mask installation problems, or cause other problems due to the old drivers or utilities still on the hard drive. Based on experience, it is virtually impossible to reliably and economically "uncorrupt" a system by removing all possible installed DLL's, EXE's, DRV's, and help files. The real solution is providing a quick method of installing and configuring a fresh OS on demand.

Two years ago, our organization was directed to support over 20 foreign languages on all products. Installation of a foreign language operating system contains its own challenges, and usually the tester is not conversant in that language. We were asked to setup the infrastructure for an NLS Lab that would support these languages during on-site visits by teams of reviewers from various foreign countries. These "native" reviewers would use these systems during their on-site visit.

A simple system was needed that foreign reviewers could use, without getting the sysadmin involved during an on-site workshop. This web backup and restore method allows the reviewer to quickly restore any previously installed language/OS combination, and be assured it is correct. This also gives our Test team the ability to quickly return the client system to a known state at any time.

Supported Hardware/Software

A decision was made to only support PC's with SCSI and ethernet. Token-ring network cards and Microchannel systems are not supported, because neither is supported by FreeBSD.[2] Since FreeBSD can be installed everywhere without additional licensing fees, and had proven reliable in several other large projects, it was our choice for this project.

This backup and restore method could be easily adapted to other versions of unix by making changes in the scripts for different device names, and other

[1] Printer drivers and utilities are installed on the client PC during normal printer installation.

[2] http://www.freebsd.org/handbook/handbook11.html .

minor changes. Users familiar with unix and perl should be able to use our scripts with minimal effort.

History

The first NLS requirement was for UK English, French, German, Spanish, and Italian to be each loaded onto five separate P-75 systems. Only Windows 3.1 and OS/2 2.1 were supported. OS/2 Boot Manager was used to switch between bootable partitions. Rebuilding the systems was done manually. This was deemed too labor intensive after one NLS workshop.

The first try at duplicating SCSI drives used a small C program with DOS BIOS calls to read a 512 byte sector, and write it onto an identical drive on the SCSI chain. This proved unusable because a 2 GB drive took over 24 hours to duplicate. This was a dead end, but directly copying the drive was still in our minds.

A bootable DOS diskette was built that logged onto an OS/2 server as an IBM LAN Manager client.[3] It allowed an entire 200 MB C: partition to be backed up onto a server drive using the DOS XCOPY program, and restored as well. Several language workshops using Win31 and OS/2 were conducted using this method.

The number of client systems was increased to 10, two for each language. The ethernet hub was discarded in favor of a 10 MBit ethernet switch, because packet collisions during multiple restores stretched a restore to several hours.

The matrix grew to six languages with the addition of Brazilian Portuguese, and three operating systems with the addition of Windows 95. The introduction of Windows 95 broke our XCOPY method with its long filenames and registry.

We purchased a commercial solution called CSCDUPE[4] that would read/write sectors from a SCSI drive to/from a DOS filehandle on a NETBIOS server. This software was marketed primarily for burning CD-ROM's, and requires the loading of an ASPI SCSI DOS driver in CONFIG.SYS, and running CSCDUPE on top of it.

Entire 200 MB disk images were saved without compression on several 2 GB server drives. A restore of a 200 MB C: partition took 45 minutes across a 10 MBit ethernet segment on a P-75 using an Adaptec 2940 and a SCSI-2 drive.

Several more language workshops (including Win95 testing) were performed with this CSCDUPE method. However, CSCDUPE is entirely menu-driven and prone to input errors. A better method was needed that freed the sysadmin from this task.

As an experiment, a FreeBSD[5] boot diskette was constructed that NFS mounted a FreeBSD file server. Then the dd command was successfully used to backup and restore a DOS partition from one client system to another, with the image stored on the FreeBSD server. Several iterations of this method were explored, including attempts at BOOTP using a boot EPROM on the ethernet card, and a DOS boot diskette that invoked a BOOTP of FreeBSD.[6]

The final method implemented was a separate 200 MB FreeBSD partition created on each client system, while Booteasy[7] was used to boot either FreeBSD or the DOS partition with the target OS/language. This method does not require boot diskettes.

An ethernet switch was installed with a 100 MBit uplink to the server, which provided 24 client 10 Mbit downlinks.[8] The Concatenated Disk Driver (ccd)[9] on the FreeBSD server striped data on two wide SCSI drives to increase throughput, and gzip/gunzip was used on the client system to compress/decompress the images stored on the server.

The final benchmark on P-166 client systems using Adaptec 2940UW, a wide SCSI drive containing a 200 MB DOS partition, and 10 Mbit ethernet was 7.5 minutes for a restore.

The biggest improvement was the web page interface that allowed scheduling of image backups and restores by the user, without getting the sysadmin involved. This web page is served by Apache[10] on the FreeBSD server holding the images.

NLS User Perspective of a Backup

The NLS tester schedules a backup on the Web page using a few mouse clicks to select their hardware system name, desired operating system, and foreign language. A password is needed to perform a backup, which keeps an inexperienced user from overwriting existing backup images. The Web page instructs the user to boot their client system into FreeBSD Booteasy, which briefly asks the user for two choices: "`F1 for BSD`," or "`F2 for DOS`." After pressing F1, there is no further user interaction. The user will later receive a message from FreeBSD on the client system that the backup is complete.

NLS User Perspective of a Restore

A restore is the same as the backup process, but without needing a password. The user schedules the desired restore on the same Web page, and boots the client system into FreeBSD using "`F1 for BSD`."

[3] We had to use 2.88 MB diskette drives and diskettes to hold the OS/2 boot diskette and networking.

[4] http://www.corpsys.com .

[5] http://www.freebsd.org .

[6] http://www.freebsd.org/tutorials/disklessx/disklessx.html .

[7] http://www.freebsd.org/tutorials/multios/multios.html .

[8] http://www.3com.com/products/switches.html . The ethernet switch we use is called a Linkswitch 1000.

[9] http://www.freebsd.org/handbook/handbook50.html .

[10] http://www.apache.org .

The user will later receive a message on the client system that the restore is complete, at which time the user reboots the client system, presses "`F2 for DOS`," and boots into their desired Windows or OS/2 environment. This provides a guaranteed clean install in a fraction of the time normally taken by ordinary media installation. Currently, a 400 MB image backup or restore takes less than 11 minutes.

NLS Client System Description and Configuration

Our current platform is a Dell GxPro Pro-200, with Adaptec 2940UW[11] SCSI and ultra-wide drives because we wanted the fastest drives possible at the time. The FreeBSD kernel supports Adaptec tag-queuing for better performance.[12] The internal 3Com 905X ethernet is used at 10 Mbit, although 100 Mbit is available on the Dell platform.

The heart of the backup/restore method is the installation of FreeBSD in a separate physical partition on each client system hard drive. This partition has been standardized at 200 MB, with root, var, and swap of 32 MB each, and the remainder devoted to usr.

The SCSI drive is first low-level formatted with the Adaptec ROM utility, and then a DOS diskette is booted to fix any network card EEPROM settings. A FreeBSD boot diskette is loaded, and we FTP install a "Kernel Developer" system, which consists of all binaries (except Xfree86) and all kernel sources. The Booteasy boot manager is also installed.

Once the FreeBSD operating system is booted for the first time, a custom kernel can be configured based on the GENERIC configuration file supplied.[13] This custom kernel is needed for activating the Adaptec tag-queuing support, and removing device drivers that don't need to be probed, and waste time during boot. Here are the lines in the kernel configuration file necessary to be added or changed; see Figure 1. The FreeBSD sysconfig file is modified to make the client system an NFS client, and the hostname, and IP addressing is configured; see Figure 2. Above is an example of an Intel Pro/100B ethernet card entry.

An entry is made in fstab to NFS mount the FreeBSD server that supports the client:

```
server:/array /array nfs rw 0 1
```

An entry is made in rc.local that checks for any backup or restore requests at the very end of booting, and tells the user what it's doing:

```
[-f /array/bin/task ] && \
              /array/bin/task
```

If the script task does not find any backup or restore work to be done, `task` does some housekeeping (explained later), exits normally, and the familiar `login:` prompt appears on the client system.

If the client system is booted without a scheduled backup or restore, then it needs to be shutdown cleanly sometime later. To address this problem, a user id called "shutdown" is created. Then the "shutdown" id is setup to run a simple shell script:

```
shutdown:*:0:0:Quick Shutdown:
         /home/shutdown:/sbin/scram
```

To shut the client system down cleanly, the users are told to login with the id "shutdown," and the script `/sbin/scram` is run at login time:

```
#!/bin/sh
/sbin/shutdown -r now
sleep 60
```

Finally, the FreeBSD root password is set to keep users from changing anything, the client system BIOS setup password is enabled, and the "CTRL-A" boot message is disabled on the Adaptec ROM setup menu.

Once the client system is configured with FreeBSD, then a DOS diskette is booted, and DOS FDISK is used to set the desired size of the user partitions (C: and D:), and both are FAT-formatted. At this point, the completed system is ready to be duplicated on the rest of the NLS system inventory.

In summary, the client system hard drive is organized as below:

```
config      kernel  root on sd0    # default was IDE
controller  ahc0                   # Adaptec 2940 driver
options     "AHC_TAGENABLE"        # Adaptec support
options     "AHC_SCBPAGING_ENABLE" # for speed
```

Figure 1: Customization for Adaptec.

```
hostname="your.domain.com"
network_interfaces="fxp0 lo0"
ifconfig_fxp0="inet 123.456.789.012 netmask 255.255.255.0"
nfs_client=YES
```

Figure 2: FreeBSD sysconfig file changes.

[11] http://www.adaptec.com/work/AHA2940UWOF.html.

[12] http://www.freebsd.org/releases/2.0.5R/notes.html. Only release 2.0.5 and beyond has tag queuing.

[13] Found in /usr/src/sys/i386/conf. Copy GENERIC to your own file, and edit it to suit.

- Booteasy boot manager – 512 bytes at first drive sector
- Partition table
- FreeBSD – 200 MB partition
- C: drive – 400 MB physical partition
- D: drive – 500 MB logical partition

The method used on the 35 NLS systems was to completely build and test a single client system per the above. Then this system was booted onto the network with a DOS boot floppy, and CSCDUPE was used to backup the FreeBSD image across the network to our server running Samba.[14] This uncompressed 200 MB image is then downloaded onto the other 34 client systems in the same manner; the `sysconfig` and `hosts` files are configured on each system.

The CSCDUPE backup and multiple restore is done from the first sector on the drive, through the first sector of the C: partition. This means that Booteasy, the partition table, and the FreeBSD partition are all in the image. The CSCDUPE restore requires 45 minutes for this 200 MB image, but several can be started at once. All that remains to be done on each client system after a CSCDUPE image restore, is to assign D: as a logical drive with DOS FDISK (since the D: partition is not copied), and format the C: and D: partitions as FAT. The individual client systems are now ready to be turned over to the test community for installing operating systems.

Server System Description and Configuration

The server platform is based on an Intel Venus Pro-200 motherboard,[15] with 128 MB of RAM, Adaptec 2940/3940, and two Intel 100 MBit ethernet cards. This FreeBSD server supports 90+ client systems.

The disk space mounted under `/array` on the server is composed of eight 9 GB SCSI drives (72 GB), which formats to 66 GB. To keep from having to mount and keep track of these drives, the Concatenated Disk Driver support is used in FreeBSD. Below is the line that need to be added to the kernel configuration file:

```
pseudo-device ccd 4
        # ccd driver support
```

In addition, the following changes need to be made in `sysconfig`, in addition to standard customization changes:

```
nfs_client=YES
nfs_server=YES
gateway=YES
```

Our eight 9 GB drives are configured in `ccd.conf` (yours may be different):

```
ccd0 32 none /dev/sd8e /dev/sd12e
     /dev/sd9e /dev/sd13e /dev/sd10e
     /dev/sd14e /dev/sd11e /dev/sd15e
```

We mount this ccd array in `fstab`:

```
/dev/ccd0c /array ufs rw 1 2
```

Verify the below lines are still in /etc/rc to startup the disk array:

```
# Configure ccd devices:
if [ -f /etc/ccd.conf ]
      ccdconfig -C
```

All that remains is to keep the image server's filesystem backed up via DLT tape to preserve the `/array` directory structure, the created disk images, and associated scripts. A drawback with CCD is that any drive that fails ruins the entire data array, and this has already happened once. However, a good feature of CCD is data striping across multiple disk drives, which raises the throughput over that achieved by a single drive. Our CCD array of eight SCSI-2 drives Bonnie benchmarks[16] at 9.5 MB/sec, which is about equivalent to the theoretical throughput of a single 100 Mbit ethernet interface.

The server contains two Intel EtherExpress Pro/100 TX PCI ethernet cards,[17] one which interfaces to the private class-C network of 35 NLS "PC Hardware All Alike," and the other which interfaces to the rest of the client systems covered by the "PC Hardware All Different" Web page. The server also is an IP gateway, which allows the 35 NLS systems to obtain access to the rest of the network, while isolating them from IP, Novell, Netbios, and Appletalk traffic on the rest of the network.

The server runs a DHCP daemon, which is not part of the FreeBSD distribution.[18] This daemon listens for DHCP requests from the NLS client systems, and assigns them a unique IP address on the private network each time they are booted into Windows or OS/2. The DHCP daemon assigns each client system the same IP address configured in the FreeBSD partition on that system.

Users setup each OS to obtain the IP configuration via DHCP, which keeps down configuration errors, especially over a large number of possible images. This also allows multiple copies of an OS/language to be running on different client systems at the same time for any special debugging efforts.

The Apache web server is running on this server system, and it supplies the two different Web pages to users. Apache installation or configuration will not be covered here.

[14] http://lake.canberra.edu.au/pub/samba/ .

[15] http://developer.intel.com/design/motherbd/vs/index.htm .

[16] http://www.freebsd.org/ports/benchmarks.html .

[17] http://www.intel.com/network/doc/6285/index.htm .

[18] http://www.isc.org/isc/dhcp.html .

The server also runs Samba, which supplies the protocol used to duplicate each client system with CSCDUPE, and this protocol is used during the Language Workshops for file transfers and loading new code releases.

Backup Process Description

Once the user initiates FreeBSD with "`F1`," FreeBSD boots normally and NFS mounts the host server filesystem, which we have defined on each client system as `/array`. Just before FreeBSD exits `rc.local`, it runs a Perl[19] script from the NFS mounted directory called `/array/bin/task`, which does the following:

- Obtains the `$hostname` of the client system, and the `$date` (for other uses later).
- Opens (for append) a log file on `/array/log` that is used for statistical purposes.
- Re-makes the hard drive device id, in case the user changed the C: partition size.
- Checks for the presence of some information files about each client system, and if not present, `task` runs and records the output of commands `dmesg`, `fdisk`, and `disklabel`. These three files are stored into: `/array/machines/$hostname/info/`, and are backed up when the entire `/array` structure is backed up. In case of disaster, all the setup information about a client system can be obtained from them.
- Runs the `ntpdate` command to set the time on the system, in case it has drifted.
- Finally, `task` checks for the presence of a file: `/array/machines/$hostname/backup`, which contains the name of the desired image to be backed up, i.e., `uk_win.dd.gz`. This file was created by user interaction with the Web interface, and will be explained later.
- The `task` script then prints a line to the client system console, letting the user know that the backup has started, and issues the following command:

```
‘dd if=/dev/sd0s2c | gzip -1 > \
/array/machines/$hostname/\
images/$image‘;
```

- The `dd` utility reads the data from the client system hard drive (the C: partition is specified as `sd0s2c` above).
- The `dd` data is compressed by `gzip` on the client system, using the fastest rate option.[20]
- The image file is transferred across the network, and stored on the server disk array at: `/array/machines/hostname/images/$image`, where `$image` is a unique filename that that associates it with the OS and language.
- After the backup is complete, throughput numbers are displayed and recorded in the log file, which is then closed.
- The stored image on the server now has a unique filename that associates it with the OS and language.
- Finally, the client system is shutdown.

Compression efficiency of a user disk image varies with the OS and support applications installed on the 400 MB C: partition. Our average compression ratio, based on 63 NLS images (Win31, Win95, WinNT, and OS/2) is almost exactly 4:1.

To improve compression of the client system image, zeros are stored in all unused areas of the C: partition. Gzip'ed zeros consume very little space in the stored server image. The Norton WIPEINFO program is used on the C: drive after it has been initially formatted, but before the user begins the construction of an OS on the drive.[21] During a trial, we were able to store a 400 MB disk image of all zeros in about 150 KB.

Restore Process Description

The user schedules a restore on the Web page, and is reminded that the proper language keyboard needs to be plugged into the client system. Once the user boots into FreeBSD with "`F1`," FreeBSD NFS mounts `/array`. All of the previous steps described above for a backup are done by `/array/bin/task`, with these differences:

- The `task` script prints a line to the client system console, letting the user know that the restore has started, and issues the following command:

```
‘gunzip < /array/machines/\
$hostname/images/$image | \
dd bs=27136 of=/dev/sd0s2c‘;
```

- The desired disk image is read from the server across the network and is uncompressed by `gunzip` on the client system.
- The `dd` utility takes the output pipe from `gunzip`, and writes the data to the client system hard drive (the C: partition is specified above).
- After the restore is complete, throughput numbers are displayed on the client system and recorded in the log file, which is then closed.
- Finally, the client system is shutdown and rebooted.

[19] Redundant footnote. If names Wall, Christiansen, or Schwartz don't ring a bell, seek professional help.

[20] We settled on the "-1" option, the fastest compression method. See "man gzip."

[21] http:/www.symantec.com/nu/fs_nu8win.html. WIPEINFO.EXE is part of Norton Utilities 8.0.

NLS Web Page Description – "PC Hardware All Alike"

This Web page is named `nlslab2.html`. This page has pulldown menus to select each client system, the Languages, and the Operating System desired. It includes a radio button to select either Backup or Restore, and demands a password for Backups.

The web page fields can be edited to add/remove Machines, Languages, and Operating Systems. Combinations of these fields become the unique filenames used for the images.

CGI scripts called from web page buttons perform other useful functions:

- `nlsaction.cgi` – Main cgi script for the page.
- `nlsview.cgi` – View the scheduled backup and restores in process.
- `nlsimages.cgi` – View the available images stored, and their size/dates.
- `nlsstats.cgi` – View backup and restore statistics.

NLS Lab Performance Observations

The bottleneck seems to be how fast the Dell client system can read and write the hard drive. However, the compressed partition traffic may still consume considerable network bandwidth, and plugging multiple client systems into a 10 Mbit ethernet hub can cause an unacceptable collision rate. Initial trials with ethernet hubs proved unsatisfactory when more than one backup or restore operation was done simultaneously. Presently, an ethernet switch is used which has a 100 Mbit uplink port for server data, and in theory should be able to handle ten client systems at 10 Mbit. Each client system occupies it's own 10 Mbit switched ethernet port, while several network printers are on a hub attached to the 10 Mbit switch. More testing and verification is being done in this area to find the best infrastructure performance.

The Adaptec SCB paging in the FreeBSD kernel for tag-queuing disk drives speeds disk transfers significantly.[22] An incident occurred that slowed our restore process by a factor of four, and SCB paging was found accidentally turned off.[23] Slower SCSI or IDE drives can be used, with lower performance expectations.

An interesting aspect of this method is that the majority of the processor-intensive work (compression and uncompression of images) is distributed among the client systems, and the file server is only lightly loaded serving NFS to them.

Web Page Description – "PC Hardware All Different"

Separate from the NLS infrastructure realm, this newer Web page is popular with the rest of the Test and Development users that want this backup/restore capability, but who are not in the foreign language testing business. This method uses many of the same concepts as the NLS web page for "PC Hardware All Alike."

This web page is really a redirected CGI script called `index.cgi`. This method is different from the one for "PC Hardware All Alike," in that it searches a directory called `/array/people` to find all users, and then sorts and lists the client systems assigned to them.

Each client system is then listed on a separate line by hostname, with pulldowns to select Backup/Restore, a password field, a text-entry field to enter a new image name, and a Submit button. The CGI script behind this page is `action.cgi`, and processes the requests from this page.

Groups of client systems may be assigned to a user. Each user has an assigned password, and each system has a password, either of which will allow the scheduling of a backup or restore. Because a larger number of people are associated with this web page, demanding a password for both backups and restores reduces the possibilities for user error.

The hostname on each line of the web page is also a link to a maintenance page for each client system, and is driven by the CGI script `machine.cgi`. This script provides information about the owner, and statistics about the backups/restores performed on that system. Through another script named `maint.cgi`, the owner is allowed to delete an existing image for that client system, or cancel a scheduled backup/restore that has not started.

CGI scripts hidden behind radio buttons perform other functions:

- `viewlab.cgi` – View the scheduled backup/restores in process.
- `statlab.cgi` – View the overall backup and restore statistics.

"PC Hardware All Different" Client System Duplication

The method to install these client systems uses a FreeBSD boot diskette to perform an FTP installation, and then the system is manually configured. FTP installation is used because all client systems have ethernet, and we can get the latest FreeBSD release from our FTP server. FreeBSD supports a number of popular ethernet cards, and good results have been obtained with Intel Pro/100B's and 3Com chipsets, at both 10 and 100 Mbit rates.

[22] Refer to FreeBSD SCB paging description by Justin T. Gibbs in /usr/src/sys/i386/scsi/aic7xxx.c.

[23] NT 4.0 turns off this Adaptec feature when it loads.

The minimum SCSI drive size has been 1 GB, which leaves 800 MB for the C: partition.[24] This remaining space can be used any way the user wants, and filesystem types can be any format, such as FAT, NTFS, HPFS, etc.

Usage Statistics, System Cost, and ROI

Usage statistics are available from both web pages. When the statistics button is pressed, a perl script parses through the log files kept on the server, and presents them to the requester. The statistics show the users getting the most benefit from the system, and perhaps which users need to drop off this strategy. These scripts are:

- `nlsstats.cgi` – NLS "PC Hardware All Alike" statistics
- `statlab.cgi` – "PC Hardware All Different" statistics

The 35 Dell client systems described here cost over $100K. The server to support them was pieced together for about $20K. This backup and restore method may not suit everyone. This is an on-demand system, and users expect restores within minutes of scheduling them. Using tape, CD-ROM, boot diskettes, or other slower means of image restore would not be acceptable in our case. We decline to bring aboard users that just want to hold the contents of their office PC in case it crashes. If they actively use it to test software however, this method is usually their best choice.

Security

Our biggest problem is when a user is inattentive to the pulldown selections. The worst case scenario is when a user overwrites an image on the server, due to an error specifying a backup. Passwords are needed to backup an image on the NLS page, but are not necessary for restores. However, passwords are required for any operation on the "PC Hardware All Different" web page.

Support Software

This paper, all the scripts described within, and any other documentation necessary to duplicate this backup/restore system, will be available for anonymous FTP from `heathers.stdio.com/pub/lisa97/`.

Author Information

Tyler Barnett joined IBM in 1977 as a Field Engineer in Louisville, KY. He left the University of Kentucky in 1989 with an BSEE. He is now in the software test organization at Lexmark International, where he is responsible for sysadmin, printer performance, and whitebox testing. His mail address is: Lexmark International, B035-3-2D4, 740 New Circle Road, Lexington, KY 40511. Reach him by email at <tbarnett@lexmark.com> or <n4ty@stdio.com>.

[24] We use 1 GB Seagate ST31055W, and 2 GB Seagate ST32171W. Both discontinued and inexpensive.