The following paper was originally presented at the
Seventh System Administration Conference (LISA ’93)
Monterey, California, November, 1993

The Amanda Network Backup Manager

James da Silva & Olafur Guthmundsson
Department of Computer Science
University of Maryland

The Amanda Network Backup Manager

James da Silva & Ólafur Guðmundsson
– Department of Computer Science, University of Maryland

ABSTRACT

We present *Amanda*, a freely redistributable network backup manager written at the University of Maryland. Amanda is designed to make backing up large networks of data-full workstations to gigabyte tape drives automatic and efficient.

Amanda runs on top of standard Unix backup tools such as `dump` and `tar`. It takes care of balancing the backup schedule and handling any problems that arise. Amanda runs backups in parallel to insure a reasonable run time for the nightly backups, even in the presence of slow computers on the network. Tape labeling insures that the wrong tape is not overwritten. A report detailing any problems is mailed to the system administrator in the morning.

In our department, we use Amanda to back up about 35 gigabytes of data in 336 filesystems on more than 130 workstations, using a single 5 gigabyte 8mm tape drive. Nightly runs typically complete in three to four hours. Amanda is currently in daily use at sites around the world.

Motivation

Until a few years ago, the backup medium of choice for most large Unix sites was the 9 track reel-to-reel tape, while 1/4" cartridge tapes were (and still are) popular with smaller systems. Storage capacities for 9-track and cartridge tapes vary from about 40 to 200 Megabytes. These tape systems are often of smaller capacity than the disk subsystems they are backing up, requiring an operator to feed multiple tapes into the drive for a full backup of the disks.

This problem has had a big influence on large site system administration. Sites with only a few large timesharing systems or file servers can arrange backups by operators at scheduled times, but the coordination of backups of a large number of workstations on a network is more difficult. Requiring users to do their own backups to cartridge tapes doesn't work very well; even computer-literate users just don't do backups on a regular basis.

A solution that many sites have adopted is a *dataless* workstation model, in which all user data is stored on file servers, with small local disks to hold temporary files and frequently used binaries, or even a *diskless* workstation model, where the workstations have no disks at all [1]. These network organizations require fast file servers with large disks, and generate heavy network traffic.

Our department, on the other hand, has always used *datafull* workstations, where all user data, temporary files, and some binaries, are stored on the workstations. File servers only provide shared binaries. This allows the use of smaller file servers, with smaller disks. A big advantage of this model is political; users tend to want their own disks with their own data on their own desks. They don't want to deal with a central authority for space or CPU cycles, or be at the whim of some file server in the basement.

Since most file writes are local, network traffic is lower and expensive synchronous NFS file writes are avoided, improving performance [2]. With the datafull model we are able to have each fileserver support over 40 machines if needed, while in dataless and diskless environments only specialized fileservers can support more than 20 workstations. The big disadvantage is the difficulty of managing and backing up all the datafull workstations.

The arrival of inexpensive gigabyte Digital Audio Tape (DAT) and 8mm video tape technology changed the situation drastically. Affordable disks are now *smaller* than affordable tape drives, allowing the backup of many disks onto a single gigabyte tape. It is now possible to back up all the workstation disks at a site over the network onto a single 8mm tape.

With the space problem solved, the new problem is *time*. Backing up workstations one at a time over the network to tape is simply *too slow*. We found that we could not add workstations to our network backups because the nightly backup would not finish until well after the start of the next work day. Many workstations cannot produce backup data as quickly as tapes can write [3]. For example, typical backup rates (both full and incremental) on our network range between about 5% to 70% of the rated 246 KB per second of our Exabyte EXB-8200 8mm tape drives [4].

Amanda, the ''Advanced Maryland Automated Network Disk Archiver,'' was developed to solve these problems. To make the project manageable, we first built Amanda on top of the standard BSD Unix `dump` program. Amanda uses an optional *holding disk* to run multiple backups in parallel, and copies the backup images from the holding disk to tape, often as fast as the tape can stream. This version was described in [5].

More recently, we have be working on generalizing Amanda to handle backup programs other than BSD `dump`, like `tar` (and potentially PCs and Macintoshes in the future), and adding support for Kerberos-style authentication and data encryption. Meanwhile our site has grown from 10 gigabytes of data backed up with Amanda, to 35 gigabytes, and we have moved to a 5 gigabyte tape drive.

This paper concentrates on the features of Amanda as seen from the point of view of the system administrator and operators. We will touch on configuration possibilities, daily operation, restores, reported problems, backup data integrity, and have a look at the performance of Amanda at our site for the past year and a half. We conclude with a comparison of Amanda with some other free and commercial network backup systems.

Amanda Overview

Amanda is designed to back up a large network of computers (*hosts*) to a Unix host with a gigabyte or larger tape drive. The host with the tape drive, known as the *backup server host*, can optionally contain a *holding disk*, which is used as a staging area for parallel backups. While the holding disk is optional, a relatively large disk is recommended for high performance. Depending on the site, from 200 MB up to 1 GB of holding disk can be effectively used to speed up backups. Without the holding disk, backup rates are limited to the rate at which individual hosts can generate backup data sequentially.

Amanda backups are intended to be run in the middle of the night from `cron` on the backup server host. This server host communicates with Amanda programs running via `inetd` on all the hosts to be backed up, known as the *backup client hosts*. When all the night's backups are completed, a detailed mail report is sent to the system administrators.

The server host program is `amdump`, which consists of several distinct submodules that can report results to the user. `planner` is the backup cycle scheduler; it determines what level each filesystem will back up at each night. `driver` manages the nightly run and orchestrates the actual flow of backups. `dumper` communicates with each client host, and `taper` drives the tape device. On the client hosts, `amandad` is invoked (via `inetd`) by requests from the server host.

In addition to the main overnight backup program, Amanda has several auxiliary programs:

1. `amadmin` is the general purpose administrator's utility. Amadmin encapsulates a number of small functions, like database and log queries.
2. `amrestore` restores backups from Amanda tapes. It takes care of finding the right filesystem's backup on the tape and piping the backup data to the underlying restore program.
3. `amcheck` is usually run in the afternoon to make sure that everything is set up correctly for the next `amdump` run. It sends mail reporting any potential problems to the system administrators so that the problems can be fixed before the night's run. In particular, amcheck makes sure the correct tape is loaded into the tape drive, and checks for common problems on the server and all the client hosts, such as permissions problems or nonexistent filesystems.
4. `amflush` writes backup files from the holding disk onto tape. If `amdump` detects a tape error, it will still try to back up as much data as possible to a holding disk on the server host, to avoid complete failure of the nightly backups. `amflush` is run by an operator the next day after the tape problem is corrected.
5. `amlabel` writes Amanda labels onto fresh tapes.
6. `amcleanup` recovers after any crash in the middle of an `amdump` run. It is usually run at boot time, and takes care of sending the mail report so that the system administrators know that backups were interrupted.

Configuration

Amanda is organized around *configurations*. Each configuration backs up a list of filesystems to a particular tape drive using a particular schedule. Multiple configurations can co-exist on a single server host. This can be useful for separating archives from daily backups, or balancing filesystems between tape drives.

Configuration Files

The Amanda programs are driven completely by two simple files maintained by the system administrators. The configuration file, `amanda.conf`, gives settings for a number of parameters. The `disklist` file contains a one-line entry for each filesystem to be backed up.

An example `amanda.conf` file is shown in Figure 1. This file is the central control panel for all Amanda activity. A number of parameters can be controlled by the system administrator to customize the backups to taste. Some of the possibilities are discussed in more detail below.

The `disklist` file merely lists all the filesystems that are to be backed up by this Amanda configuration, like so:

```
# hostname diskdev dumptype
salty      sd0a    comp-root
salty      sd0g    comp-user
```

The host name and device name for the partition are given, followed by the *dump type* name. This name refers back to an amanda.conf definition which specifies various per-filesystem parameters.

The Backup Schedule

Amanda manages the backup schedule within the parameters set in amanda.conf. It will move up full backups to balance the size of each night's run across the whole schedule, but will never delay a full backup for balancing purposes.

The configuration files allow many styles of backup schedule to be implemented with Amanda. Some of these are:

- **Periodic Full Backups with Daily Incrementals**: This is the most common style of backup. The backup schedule is set to some number of weeks (i.e. set mincycle 2 weeks in amanda.conf). Each filesystem will normally get a full backup once within this cycle, and an incremental backup every other night. The full backups can be moved forward at Amanda's discretion to balance the schedule.
- **Periodic Archival Backups**: An Amanda configuration can be set up that does just full backups to a new tape each time. These tapes are then archived permanently. Set

  ```
  options skip-incr, no-compress
  ```

```
org "CSD"                # your organization name for reports
mailto "csd-amanda"      # the mailing list for operators at your site
dumpuser "bin"           # the user to run dumps under

inparallel 8             # maximum dumpers that will run in parallel
netusage  500            # maximum net bandwidth for Amanda, in KB per sec

mincycle  10 days        # the number of days in the normal dump cycle
tapecycle 20 days        # the number of tapes in rotation
bumpsize 10 MB           # minimum savings (threshold) to bump level 1 -> 2
bumpdays     2           # minimum days at each level
bumpmult     2           # threshold = bumpsize * (level-1)**bumpmult

tapedev "/dev/nrst8"     # the tape device
tapetype EXB-8500        # what kind of tape it is (see tapetypes below)
labelstr "^VOL[0-9][0-9]*$"      # label constraint regex: all tapes must match

diskdir "/amanda2/amanda/work"  # where the holding disk is
disksize 800 MB                 # how much space can we use on it

infofile "/usr/adm/amanda/csd/curinfo"  # database filename
logfile  "/usr/adm/amanda/csd/log"      # log filename

define tapetype EXB-8500 {    # specifies parameters of our tape drive
    length 4200 mbytes
    filemark 48 kbytes
    speed 480 kbytes
}

define dumptype comp-user {   # specifies parameters for backups
    program "DUMP"
    options compress          # compression is optional
    priority medium
}

define dumptype comp-root {
    program "DUMP"            # DUMP or GNUTAR or ...
    options compress
    priority low         # root partitions can be left for last
}
```

Figure 1: Example Configuration

in the dump type specifications to turn off incrementals and compression, and set

```
tapecycle inf
```

to tell Amanda that the tapes are never cycled.

- **Incremental Only, with external full backups**: Large timesharing hosts that are always active are best backed up by hand in single user mode during a scheduled down-time period. The daily backups can still be done with Amanda, by specifying `options skip-full` on those filesystems, and running `amadmin force` to lock the full backup position to the night the external backup is done. Thereafter Amanda will attempt to keep in sync with the external backup, and even warn the operators when the scheduled backup is due.
- **Incremental Only, with no full backups**: Some filesystems don't normally change at all relative to some reference filesystem. For example, root partitions are often derived from a site-wide standard prototype, plus small local customizations. These partitions can be installed such that incremental backups capture just the local changes. With `options no-full` in the dump type, Amanda will do incremental backups for these filesystems on each run, with no bumping (see below for a description of *bumping*).
- **Frequent Full Backups, No incrementals**: Some sites don't like to bother with incremental backups at all, instead doing full saves of all their disks each night, or as often as possible. Such a site can be run similarly to an archive configuration, with `options skip-incr` set for each disk, and `mincycle` set as low as possible given the size of the disks and the backup tape.

Automatic Incremental Bumping

Berkeley `dump` supports the concept of multiple *levels* of incremental backups, whereby a backup at level *n* backs up every file modfied since the last backup at level *n-1*. Other backup programs, such as `tar`, can be run in the same way.

The different backup levels allow a tradeoff between redundancy of data on tape, and saving tape space by only backing up recently changed files. Coming up with the right tradeoff can be a chore: experienced administrators will remember the ''Modified Tower of Hanoi algorithm'' recommended in the original Berkeley `dump` man pages.

Amanda is smart enough to only change the incremental level (known as *bumping*) for a filesystem when significant tape space would be saved by doing so. Amanda also takes care to not bump too eagerly, since having too many incremental levels makes full restores painful. Three `amanda.conf` parameters are provided for the system administrator to control how bumping is done.

- **`bumpsize`** Default: `10 MB`. The minimum savings required to trigger an automatic bump from incremental level one to level two. If Amanda determines that a level two backup will be this much less than a level one, it will do a level two.
- **`bumpmult`** Default: `2.0`. The bump multiplier. Amanda multiplies the bumpsize by this factor for each level. This prevents active filesystems from bumping too eagerly by making it harder to bump to the next level. For example, with the default bumpsize and bumpmult, the bump threshold will be 10 MB for level one, 20 MB for level two, 40 MB for level three, and so on: 80 MB, 160 MB, 320 MB, 640 MB, and finally 1280 MB savings required to bump from level eight to level nine.
- **`bumpdays`** Default: `2`. To insure redundancy in the backups, Amanda will keep filesystems at the same incremental level for at least bumpdays days, even if the bump threshold criteria are met.

Tape Management

Amanda supports the labeling of tapes to avoid overwriting active data or non-amanda tapes.

The `amlabel` command puts an Amanda label onto a fresh tape. The `tapecycle` parameter controls how many tapes are considered to be in active rotation. Normally there would be at least several more tapes in rotation than there are days in the backup cycle. This allows some slack should a machine be out of commision for several days.

Amanda labels are arbitrary names; the system administrator chooses the tape naming system. The `labelstr` configuration parameter constrains valid tape labels to a certain regular expression pattern. For example,

```
labelstr "^VOL[0-9][0-9]*$"
```

only allows labels of consisting of the prefix `VOL` followed by a number.

The `labelstr` facility can prevent two configurations using the same tape drive from overwriting each other's tapes. If each configuration uses a different label prefix, tapes from other configurations will be protected.

Daily Operation

Once Amanda is installed and configured, very little effort is required for daily operation. Adding and deleting filesystems from the backup list is as simple as editing the `disklist` file.

In addition to maintaining the `disklist`, the operators must change the tapes, handle any restore requests, read the nightly report generated after the backups complete, and deal with any problems mentioned in the reports.

Day-time Check

Since the Amanda backups are done in the middle of the night, presumably when no operators are around, it is important that possible failure modes are checked for before the run, when operators are present.

The `amcheck` program checks that the right tape is in the drive, and that there is enough room on the holding disk for proper operation. If not, it will send mail to the operators listing its complaints. `amcheck` is run from `cron` after the time the tape is normally changed, but early enough that someone can solve the problems before the run.

Figure 2 shows a sample of the amcheck mail generated when two problems occurred: the holding disk had less free space than requested in `amanda.conf`, and the wrong tape is in the tape drive. Both problems are most likely the result of an operator doing a restore from tape `VOL18` earlier in the day using the holding disk during the restore. The mail message reminds the operators to clean up after they are finished.

Reported Problems

After the nightly `amdump` run completes, mail is sent to the operators giving the details of the night's operations. Any errors are summarized at the very top of the report, with details given below. The report includes summary statistics as well as a line for each filesystem, telling of its success or failure and how it performed.

An excerpt of a nightly report is given in Figure 3. In this example, one of hosts (**`idaho`**) is down, and a filesystem on **`rath`** has developed a bad spot. Even though `dump` continues after read errors and eventually succeeds, Amanda catches the problem by scanning through the `dump` message output for anything interesting. If unknown patterns pop up, the `dump` output is displayed for the operators to deal with the problem. In this case, the filesystem in question should be reformatted and restored.

Amanda catches a number of common problems, including:

- As in the example, *disk errors* that occur during backup are brought to the operators' attention. This allows them to be detected and corrected very quickly.
- Any other *backup program errors*, such as permission problems, or even a core dump, are caught and brought to the operators' attention.
- Any *down client hosts* are identified by Amanda. Their filesystems are failed, giving them a higher priority the next run.
- Any *backups that hang* are detected; Amanda times out if no backup data is received for a certain time.
- If the *wrong tape* is in the tape drive, Amanda will not overwrite it. Instead it writes, in priority order, as many incremental backups to the holding disk as will fit. These can be put onto the next tape with the `amflush` command.

In addition to identifying problems, the report gives many vital statistics and *notes* from the various subsystems. In Figure 3 we see several notes from `planner`. Any bumps of incremental levels or promotions of full backups from later in the schedule are mentioned. In addition, we see that the operators have requested that a filesystem be forced to a full backup on this run. `planner` confirms in the report that the full backup will be done.

Restores

There are two phases to doing a restore. First, the correct tapes to restore from must be determined, and second, the data must be retrieved from the tape.

The `amadmin find` command shows the backup history for a particular filesystem. Consider the following example output:

```
date      host disk lv tape  file stat
93-09-11  rath sd0g 1  VOL2   323 OK
93-09-10  rath sd0g 1  VOL1   305 OK
93-09-09  rath sd0g 1  VOL20  262 OK
93-09-08  rath sd0g 1  VOL19  242 OK
93-09-07  rath sd0g 1  VOL18  127 OK
93-09-04  rath sd0g 0  VOL17   99 OK
```

To do a full restore of this filesystem, only tapes `VOL17` and `VOL2` need to be restored. To restore a single user file or directory, more information is needed. For example, a user might create a file on

```
From:    bin
To:      csd-amanda
Subject: CSD AMANDA PROBLEM: FIX BEFORE RUN, IF POSSIBLE

WARNING: disk space low: 552972 KB avail < 884736 KB requested.
         (please clear out cruft from /amanda2/amanda/work's partition)
ERROR: cannot overwrite active tape VOL18.
       (expecting tape VOL2 or a new tape)
```

Figure 2: Example `amcheck` report

```
From:    bin
To:      csd-amanda
Subject: CSD AMANDA MAIL REPORT FOR September 11, 1993

These dumps were to tape VOL2.
Tonight's dumps should go onto tape VOL3 or a new tape.

FAILURE AND STRANGE DUMP SUMMARY:
  idaho      sd2h lev 0 FAILED [could not connect to idaho]
  rath       sd0a lev 1 STRANGE

STATISTICS:                Total       Full      Daily
                          --------   --------   --------
Dump Time (hrs:min)         3:38       1:57       1:17   (0:12 start, 0:12 idle)
Output Size (meg)         2709.8     1796.3      913.5
Original Size (meg)       4881.7     3044.0     1837.7
Avg Compressed Size (%)     51.4       53.4       48.5
Tape Used (%)               64.9       42.8       22.1   (level:#disks ...)
Filesystems Dumped           335         26        309   (1:276 2:26 3:5 4:2)
Avg Dump Rate (k/s)         48.8       56.6       38.4
Avg Tp Write Rate (k/s)    238.1      262.1      201.8

FAILED AND STRANGE DUMP DETAILS:

/-- rath       sd0a lev 1 STRANGE
| senddump: start rath sd0a level 1 to amanda.cs.umd.edu
|   DUMP: Date of this level 1 dump: Thu Sep  9 01:38:51 1993
|   DUMP: Date of last level 0 dump: Thu Sep  2 01:58:25 1993
|   DUMP: Dumping /dev/rsd0a (/) to standard output
|   DUMP: mapping (Pass I) [regular files]
|   DUMP: mapping (Pass II) [directories]
|   DUMP: estimated 786 blocks (393KB) on 0.00 tape(s).
|   DUMP: dumping (Pass III) [directories]
|   DUMP: dumping (Pass IV) [regular files]
?   DUMP: (This should not happen)bread from /dev/rsd0a [block 6992]: ...
|   DUMP: level 1 dump on Thu Sep  9 01:38:51 1993
|   DUMP: 790 blocks (395KB) on 1 volume
|   DUMP: DUMP IS DONE
| senddump: end
\--------

NOTES:
  planner: Forcing full dump of tove:sd0a as directed.
  planner: Incremental of cortex:sd0g bumped to level 3.
  planner: Full dump of lovedog:rz9g promoted from 1 days ahead.

DUMP SUMMARY:
                                     DUMPER STATS                        TAPER STATS
  HOSTNAME   DISK    LV  ORIG-KB   OUT-KB  COMP%  MMM:SS   KB/s  MMM:SS   KB/s
  --------------------  ------------------------------------  -------------
  idaho      sd0a    1  FAILED ---------------------------------------------
  idaho      sd0h    1  FAILED ---------------------------------------------
  idaho      sd2h    0  FAILED ---------------------------------------------
  lovedog    rz3a    1      403      128   31.8    0:04   35.6    0:03   57.8
  lovedog    rz3g    3     9745     1678   17.2    1:14   22.5    0:09  192.4
```

Figure 3: Excerpt from Nightly Amanda Report

September 7 then accidentally delete it on 9th, and want it back a few days later. In this case VOL19 must be restored to get the file. The restores are done with the `amrestore` program. `amrestore` gets the proper backup off of the Amanda tape and outputs the backup image. This can be put on a staging disk (the holding disk works well for this), or piped directly to the restore program.

For example, to do a full restore of `rath`'s `sd0g` disk from `rath`, the command would be:

```
rsh amanda amrestore -p /dev/nrst8 \
        rath sd0g | restore xf -
```

where `amanda` is the Amanda tape server host.

Data Integrity

There are two major issues affecting the integrity of backup data that system administrators need to keep in mind when designing their backup system. First is the online backup problem, the second is compression.

Online Backups

The Online backup problem is well-known and has been discussed in previous LISA papers [6, 7]. As Shumway shows, it is impossible in general to insure completely correct backups on an active filesystem without operating system support. Adding, modifying, deleting, and moving files and directory trees while the backup is running can cause data to be missed, or worse, confuse the backup program into crashing or generating a corrupted output that cannot be restored.

Amanda suffers from this problem to the same extent that the underlying backup program does. If the vendor's backup program does not make system calls to lock out filesystem changes at sensitive times, then the potential for problems exists. Unfortunately, most vendors' operating systems do not have such a facility.

In practice, it turns out that the effect of this problem is small. For most filesystems on user workstations, very little is going on in the middle of the night. Since the technology to solve the problem is not yet generally available, an administrator faced with backing up dozens or hundreds of filesystems has little choice but to take the risk and do online backups.

For very active filesystems, like those on large timesharing systems or 24 hour database engines, it is probably still best to do full backups the old fashioned way, by bringing the machine down to single user mode for regularly scheduled backups. On such a system, Amanda can still be used for daily incremental backups.

Compression

Compression is completely optional in Amanda; it can be turned on or off on a per-filesystem basis.

Compression has a negative effect on the ability to restore from partially damaged backup images. The standard Unix uncompression program will be confused by the first error, causing the rest of the backup image to be lost or garbled.

For this reason, compression of data on long-term, archival backups is not recommended, as the chance of tape errors increases with long term storage. However, for tapes in a short term backup rotation, the chances of errors is small if proper care is taken of the tapes and the drive. In this situation, compression of backups is not much risk, and is worth the benefit of more than doubling the amount of data that will fit on each tape.

Turning off compression is no guarantee that errors can be recovered from. Some vendors' tape drivers will not keep reading after a medium error. A system administrator that is counting on this to work should test the hardware and software carefully. A strong magnet applied to a loop of tape somewhere in the middle of a large backup file can produce surprising results.

Backups at `CS.UMD.EDU`

Amanda's home site is the Computer Science Department of the University of Maryland at College Park. Here we have been running the parallel version of Amanda for over a year and a half, keeping statistics the entire time.

Figure 4 shows the growth in the data on the hosts being backed up by Amanda at our site. This does not include two active timesharing systems, and some of the active file server disks, which are still backed up by hand in single user mode (these non-Amanda disks add about another 8 GB to the site size).

After an initial test period from January to March, 1992, we brought all the workstations in the department onto the Amanda backups by the summer of 1992. All the growth since that time has been from bringing more data online. The plunging cost of gigabyte disk drives has had a dramatic affect on the department; the amount of data on CSD disks more than doubled, from about 15 GB in September 1992, to over 35 GB in September 1993.

We expect that other departmental level sites are seeing similar growth rates. Given the current availability of inexpensive 2 GB drives and user's insatiable demands for disk space, it seems reasonable to expect continued large increases in the amount of data system administrators are expected to back up.

Luckily, the amount of data that needs to be written to tape each night grows much more slowly. Use of compression divides the growth rate in half, and a two week backup cycle divides it again by ten. When the nightly backup reaches capacity, the backup cycle can be extended. Amanda's automatic bumping relieves the increased pressure of incremental backups in this situation.

In CSD our original 2 GB EXB-8200 became uncomfortably full in September 1992. We extended our backup cycle to three weeks, which kept us going until we brought the 5 GB EXB-8500 on-line in January 1993.

Amanda has also done a good job of holding down the backup times in the face of fast growth, as can be seen from Figure 5, which shows each of the nightly `amdump` run times. The run time has stayed for the most part in the 3 to 4 hour range. Interestingly, the variance in run times has increased considerably, with the occasional run taking more than 6 hours.

The number of short or completely failed runs have reduced, as the operators have gotten into the routine. One run in particular stands out: In August 1992 an operator added a 300 MB filesystem on a very, very slow Sun 2 with compression turned on. That disk alone took almost twelve hours to complete a full backup. Needless to say, we turned off compression for that disk the next night!

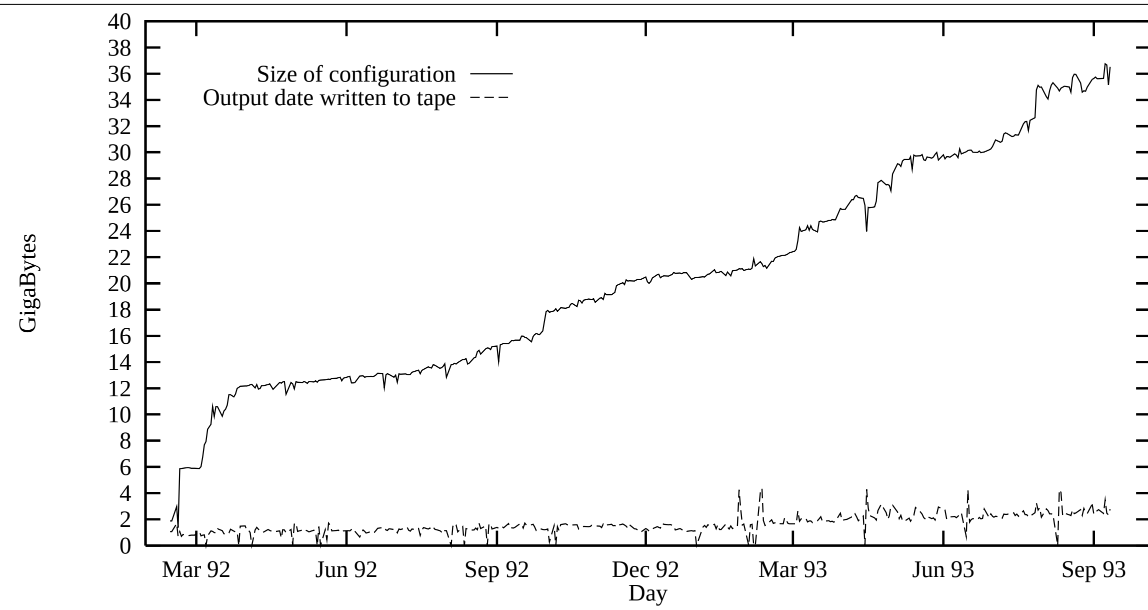

Figure 4: Nightly Amanda Backup Size at `CS.UMD.EDU`

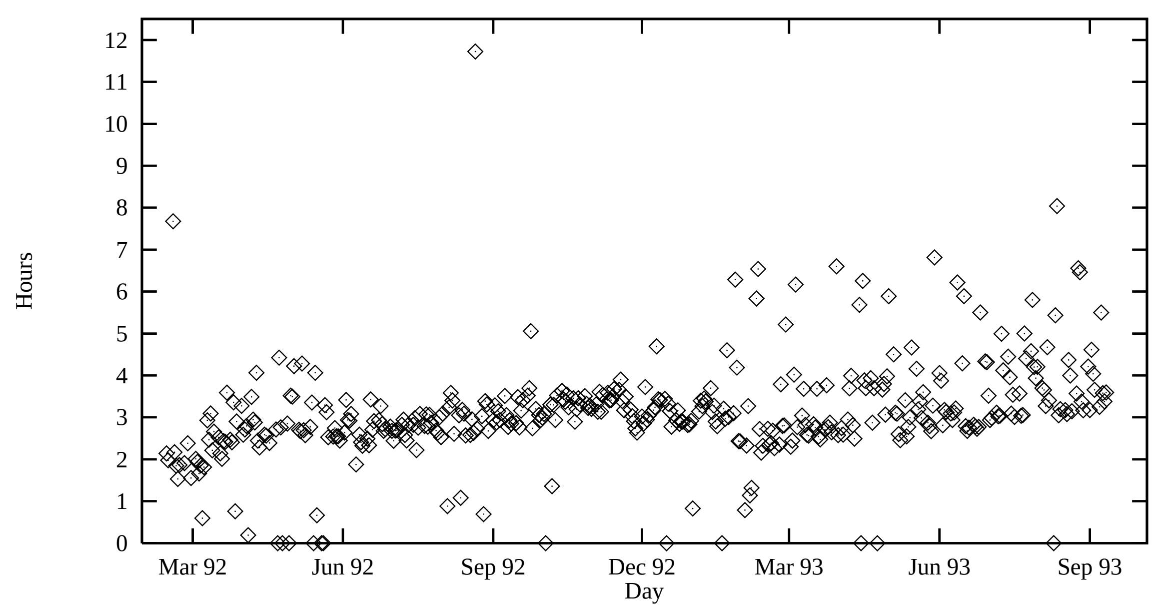

Figure 5: Nightly Amanda Run Time at `CS.UMD.EDU`

Comparisons with other Backup Systems

There are a number of systems available that perform similar functions as Amanda. This section makes no judgement, but will highlight some of the similarities and the major differences. The systems that we examined for this study that are freely distributed on the Internet are:

- Amanda from University of Maryland [5]
- Backup-2.6 from Ohio State University [1, 8]
- CUCCS Network Backup System from Carleton University (CUCCSNB) [9]
- DeeJay from Columbia University [10]

We also looked at three of the commercially available products:

- Budtool from Delta Microsystems [11]
- EpochBackup from Epoch Systems [12]
- Networker from Legato Systems [13]

All the systems above are designed to perform the same function, that is: *back up a heterogenous network of computers to large tapes, without an operator present*. The main differences are in the approach taken by the different tools. There are many different ideas about the ''Right Way'' to perform backups, and the tools reviewed have chosen different policies.

This is not a complete list of available systems but it is a good cross section. Some systems we did not look at are vendor specific and thus useless in a heterogenous network.

Approaches to Parallelism

One of the most common approaches to performing the backups in limited time is to divide the site into multiple partitions, with each one going to its own tape drive, and perform the backups in each partition sequentially. Once the partitions are in place the system should be rather stable, but some support is required to balance the load across the partitions, and to select the appropriate partition for additions. Load balancing may have to be done for both space and time.

A further advantage of this approach is that it is simple, and single tape failures affect only some of the hosts. The main disadvantage is low tape utilization due to low backup rates from hosts. Another disadvantage is that when configurations are highly loaded, operators may have to reorganize and load balance frequently.

Staging the backups to a disk is a slightly more complex approach, but it is less expensive than the one above, as it can utilize the tape better. In this scheme backups are performed at their natural speed to a holding disk, and then transferred to tape at high speed. This allows more backups to fit in each configuration. It is more reliable, as the staging disk can be used to store emergency incremental backups when there is a tape problem.

Writing multiple parallel backups to tape is the most complex approach, as this requires a special tape format. Of the systems we looked at, only Legato Networker uses this approach. This approach should outperform the other two in backup speed, but at the cost of complexity, non standard tape format, and slower restores (as the data for a particular disk will be spread out on the tape).

Backup Scheduling

The simplest way of performing backups is to always backup filesystems in the same order. In this scheme the variable is the level each filesystem is backed up at. Systems like Backup-2.6, Networker, CNCCS Network Backup and DeeJay use this method exclusively. Epoch and Budtool support this mode along with other modes. The problem with this scheduling is that tape utilization must be kept low to accommodate differences in backup sizes between nights.

A slightly more intelligent scheduling takes into account the size of the backups and moves full backups around to balance the nightly backup size.

Another approach is to perform only incremental backups using the automated system during the week and then have operators perform the full backups over the weekend. Epoch, Budtool and Amanda allow the user to specify exactly on what days full backups will be performed.

Some systems allow the system administrator to force a full backup of a set of hosts on selected days. Other options are to skip certain days.

Intelligent scheduling allows systems to fit more disks on each tape and to perform backups in less time. It is hard to evaluate from the literature available how well each system performs. In general, advanced scheduling requires less work of system administrators as the system performs the load balancing on the fly.

User Interfaces

One of the more striking differences between the systems examined is the sophistication of the user interfaces. The commercial systems all have what seem to be nice graphical front ends, some for the system administrators and others that the end users can use to request restores. None of the free systems have any graphical front ends, but some have programs to generate graphical performance information.

The command interfaces for the free systems vary from rudimentary to full description languages. Without playing with the interfaces it is difficult to assess which ones are appropriately matched to the system features.

All the systems offer some reporting, ranging from reporting only errors to full status reports. It is hard to compare the systems as most do not document what exactly is reported and in what form. It seems that the commercial systems have superior reporting facilities. The important thing to look for is whether the reports include enough information, highlight all discrepancies, and give some hints to novice operators what the problem may be.

Backup Programs

In table 1 we list the underlying backup programs each system supports:

Error recovery

There are number of things that can go wrong each time a backup is to be performed. One of the most common errors is that the right tape is not in the tape drive. Jukeboxes are less likely to suffer from this problem. All the systems have some mechanism to check if there is a tape in the drive and it is the right one. The systems that support carousels have an advantage, as they can automatically change the tape to the correct one.

In a large installation it is not uncommon that some hosts fail each night for various reasons. Most systems handle this to some extent, but the static schedule systems may have some difficulty overcoming this problem as this can delay the next night's backup significantly, or cause full backups to be skipped.

Restores

The reason people do backups is of course to be able to perform restores. The speed of restores is important to many. It is limited by a number of factors: where the data is on the tape, how fast it can be accessed, and how many tapes need to be scanned to search for the data. All the commercial systems have full file catalogues that allow them to identify quickly which tapes to restore from. DeeJay and CUCCS Network Backup support this feature, Backup-2.6 and Amanda both plan to support this in the future.

Epoch and Budtool offer graphical tools that end users can use to select files to be restored, and the requests can even be handled without operator assistance, if the tapes are available in a carousel. All others seem to require the operator to do most of the work when restoring, and use textual tools for this operation.

On the other hand, when full restores of a disk have to be done it seems that most of the systems will take similar time, depending on how incremental backups are performed and how many levels of backups have been done. All the systems seem to allow restores to remote hosts.

Per-System Highlights

In *Amanda* all scheduling and configuration is done on the tape server host. This means that no new files are created on the other machines: only `.rhosts` and `inetd.conf` have to be changed. Amanda is invoked the same way each time. Generally, all the system administrators need to do once the system is operational is to add or delete disks. Load balancing is performed by the system. Operator intervention is required for restores and after tape failures to `amflush` data from the holding disk to tape.

Ohio State University *Backup-2.6* has the ability to backup each host multiple times each night to different tapes to prevent data loss from bad tapes. It also has an explicit support for off site storage of tapes. Great care has been put into this system to allow it to overcome all kinds of problems with data loss and site errors, but it has not been tuned as much for performance as some of the other ones. Due to its inflexible scheduling, system administrators must perform operations to load balance the system including delaying adding new disks.

Carleton University *Network Backup* is designed more from the mainframe point of view. It supports index files, tar and dump, and knows about administrative domains. The system is designed to allow a central facility to backup many administrative domains. It and its tools are only supposed to be used by a hierarchy of system administrators, and there are controls on what each level can do. It has multiple configurations and supports PCs to some extent, but at the same time it is not geared at the large populations that Amanda and OSUB handle so well.

DeeJay was designed around a carousel and incorporates advanced tape management for backup of many machines. The system manages the tapes as one infinite tape. Because the carousel has multiple tape drives, it can perform backups to each one at

	Dump	GNU TAR	CPIO	Special	Index
Amanda-2.2	x	x	x		
Backup-2.6	x	x			
DeeJay	x		x		x
CUCCSNB	x	x	x		x
Budtool	x	x	x		
EpochBackup				x	x
Networker				x	x

Table 1: Comparison of Backup Programs

the same time. DeeJay has a fixed schedule of full and incremental backups for each disk: the options are weekly, monthly, or never.

Delta Microsystem's Budtool performs backups in parallel by controlling multiple tape drives on multiple hosts at the same time. It provides a simple setup procedure where users can specify the exact commands to be executed on each host to backup the system. It supports tar, dump and cpio, among others.

EpochBackup is in many aspects similar to Amanda: it provides a total hands off operation when use with EpochMigration. Unlike Amanda, EpochBackup does not run backups in parallel. Epoch claims that their special backup program is much faster than dump or tar. This system will detect changes in the configuration and notify system administrators if new disks are not being backed up. One of the advanced features claimed by this product is that restored directories will not contain deleted files, as tar based backup schemes will.

Legato *Networker*'s main distinction is that it uses nonstandard backup programs and tape formats. It performs parallel backups by multiplexing to the tape. This mechanism allows it to eliminate the holding disk, but at the cost of complex data format on the tape. Legato supplies clients for many Unix variants as well as for PC-DOS.

Future Directions

Amanda is still under active development. Some improvements not described in this paper are running in the lab (with varying degrees of solidity) and should be available about the time you read this, including:

- generalized backup program support, including `tar`, `cpio`, and eventually VMS, Macintosh, and PC-DOS clients.
- Kerberos Authentication, including sending encrypted data over the network.
- Generic carousel/stacker support. Supporting subsystems for particular hardware will need to be written.

In the longer term we are investigating the addition of a browseable file index, automatic tape verification, an X-based graphical user interface, writing to two tape drives at once, and interleaving backups on tape to allow good performance without a holding disk.

Availability

Amanda is copyrighted by the University of Maryland, but is freely distributable under terms similar to those of the MIT X11 or Berkeley BSD copyrights. The sources are available for anonymous ftp from ftp.cs.umd.edu in the pub/amanda directory.

There is also an active Internet mailing list for the discussion of Amanda, send mail to amanda-users-request@cs.umd.edu to join the list.

Author Information

James da Silva was a high-school hacker in the early '80s. He received a National Merit Scholarship from Georgia Tech in 1983, but soon left school to work in the Real World. He was a Systems Engineer for Electronic Data Systems, writing applications programs first on the PC, then under Unix. In 1987 he escaped the Real World and headed for the ivory towers of the University of Maryland. He finally got his degree after avoiding it for years by working on the Computer Science Department's computers rather than their classes. He now works there full time as a Faculty Research Assistant for the Systems Design and Analysis Group. Jaime can be reached at jds@cs.umd.edu.

Ólafur Guðmundsson was born in Reykjavík, Iceland. He graduated from the University of Iceland in 1983 with a B.S. in Computer Science. He worked as a systems programmer on VMS machines at the University of Iceland from 1983 to 1984. In 1984 he joined the graduate program of the Department of Computer Science at the University of Maryland where he had to learn some new operating systems, most of them unpleasant mainframe systems, until discovering Unix. Olafur obtained a Masters degree in 1987. Since then he has worked as a Faculty Research Assistant in the Department of Computer Science at University of Maryland. In this position he has been primarily involved in research, development and implementation of a distributed, hard-real-time operating system *Maruti*, but he has also worked on practical problems in computer networks and operating systems. Ólafur can be reached at ogud@cs.umd.edu or ogud@rhi.hi.is.

References

1. Steve M. Romig, ''Backup at Ohio State, Take 2,'' *Proceedings of the Fourth Large Installation Systems Administration Conference*, pp. 137-141, The Usenix Association, Oct 1990.
2. Paul Anderson, ''Effective Use of Local Workstation Disks in an NFS Network,'' *Proceedings of the Sixth Large Installation Systems Administration Conference*, pp. 1-7, The Usenix Association, Oct 1992.
3. Rob Kolstad, ''A Next Step in Backup and Restore Technology,'' *Proceedings of the Fifth Large Installation Systems Administration Conference*, pp. 73-79, The Usenix Association, Sep 1991.
4. *EXB-8200 8mm Cartridge Tape Subsystem Product Specification*, Exabyte Corporation, Jan 1990.
5. James da Silva and Ólafur Guðmundsson, ''Performance of a Parallel Network Backup

Manager,'' *Proceedings of the Summer 1992 Technical Conference*, pp. 217-225, The Usenix Association, Jun 1992.

6. Elizabeth D. Zwicky, ''Torture-testing Backup and Archive Programs: Things You Ought to Know But Probably Would Rather Not,'' *Proceedings of the Fifth Large Installation Systems Administration Conference*, pp. 181-190, The Usenix Association, Sep 1991.
7. Steve Shumway, ''Issues in On-line Backup,'' *Proceedings of the Fifth Large Installation Systems Administration Conference*, pp. 81-87, The Usenix Association, Sep 1991.
8. Steve Romig, ''The OSU-CIS Backup and Restore System,'' *Manual available from* `archive.cis.ohio-state.edu`, Ohio State University, Jan 1993.
9. R. Mallet, ''Carleton University Computing and Communications Services Network Backup Services,'' *Manual available from* `alfred.ccs.carleton.ca`, Carleton University, Apr 1992.
10. Melissa Metz and Howie Kaye, ''The Dump Jockey: A Heterogeneous Network Backup System,'' *Proceedings of the Sixth Large Installation Systems Administration Conference*, pp. 115-125, The Usenix Association, Oct 1992.
11. *BudTool Sales Literature,* Delta Microsystems, Inc., 1993.
12. *EpochBackup Technical Summary,* Epoch Systems, Inc., Oct 1992.
13. *Legato Networker Sales Literature*, Legato Systems, Inc., 1993.