

Backups Using Amanda

The Advanced Maryland Automatic Network Disk Archiver

Matt Hyclak

February 13, 2003

1 Traditional Backups vs. Amanda

Traditional thinking on backups has been to do a full backup of your systems over a weekend, and then do incremental backups during the week. This is not a bad plan, and makes keeping track of which tapes you need to restore a file fairly easy to keep track of. The down side of this is that you must make sure you have enough tape capacity to hold all of your full backups. When you are talking about small capacity tape drives, single drives, or a combination of both, that makes unattended backups rather difficult. This is the fundamental problem that Amanda addresses.

Amanda does things a little differently. Instead of backing up a large amount of data on the weekend, and then small incrementals throughout the week, amanda spreads the level 0s out over the course of your dumpcycle (be that a week, a month, etc.). Amanda tries its best to schedule backups so that the amount of data is evenly spread over each tape in the run. Fortunately, amanda also keeps track of what is on each tape, so you don't need to keep track of tapes at all.

Amanda uses the client/server model to do backups. The general flow of events goes something like this:

1. Sometime before `amdump` runs, `amcheck` runs to make sure the appropriate tape is in the drive, and that all the clients are reachable. If there is a problem, it sends mail to the specified address(es) to notify the administrator(s) of the problems so they can be fixed before the run.
2. `amdump` contacts all of the clients and requests estimates for each Disk List Entry (DLE) that needs to be done, each client does the estimates, and returns the results to the server.
3. Amanda makes sure it has room on the tape/holding disk for all of the data that is scheduled to be backed up. If not, it will make adjustments. Assuming everything is acceptable, amanda will begin dumping the DLEs.
4. If the wrong tape, or no tape, is in the tape drive, amanda will change to the appropriate tape (if it is an autochanger) or otherwise retain the

backups on the holding disk (depending on the settings for the holding disk, it may degrade some backups to level 1s so that they will fit).

5. After amanda has dumped everything, it will send a report to the configured mail address(es) that summarizes any problems that occurred during the run, the amount of data that was written, and what levels were done for each DLE (among other things).

2 Configuring Amanda

2.1 amanda.conf

Amanda has three basic configuration options that allows it to do its job.

- `dumpcycle` - The maximum length of time between full dumps for a particular DLE
- `runspercycle` - The number of times amdump will run during `dumpcycle`
- `tapecycle` - The total number of tapes you have

Amanda uses this information to schedule backups (when to increment, when to return to a level 0, etc.). Amanda guarantees that a level 0 will be done for each DLE **at least** once every `dumpcycle`, but it may promote a DLE early if it will save enough space. Generally speaking, you want `tapecycle` equal to at least $2 * \text{runspercycle}$, so that you always have one good set of full backups for each DLE. `runspercycle` is useful if you don't have an autoloading tape device, so you would only want to run amanda on weekdays. Another use for this is to save tapes and only run amdump once a week with a `dumpcycle` of 4 weeks.

Some other useful or interesting configuration options are as follows:

- `runtapes` - the number of tapes that can be used in a single run. If you need more than one tape, you will have to configure some sort of changer, whether that is an actual autoloader, or `chg-manual` with which amanda will prompt you for the next tape.
- `chunksize` - if you are using a holding disk that uses a filesystem that has a file size limit, `chunksize` will get you around it. For a long time `ext2` had a 2GB file size limit. Setting chunk size to 2000MB or so will split the dump on the holding disk into "chunks" that can be stored on disk, and then when it writes to tape, it puts them all back together.
- `autoflush` - if dumps have been saved to the holding disk, the next time amdump runs with the correct tape in the drive it will flush them while dumping the current DLEs to disk

2.2 disklist

The disklist contains all of the entries that you would like to back up. The basic format is

```
hostname mount-point/disk dumptype
```

The dumptypes are defined in `amanda.conf`. Some common ones include `root-tar`, `comp-root-tar`, `comp-user`. Reading through `amanda.conf` will give you a good idea as to what options are available.

When it comes to choosing what dumptype to use, you have two basic options. The `dump` program (in one of its many incarnations) or `gnutar.dump` will only do entire partitions, which can get rather large. It has the added convenience of the `restore` program that gives you a nice interactive interface to select and restore files from. Recently on the `amanda-users` mailing list, however, someone made a reference to something that Linus had said about `dump` being reliable only 99% of the time, due to how closely it is tied to the specific filesystem, and how `ext2/3` is implemented. It is because of this that I would recommend using `tar`. No, you don't get the nice interface, but it allows you much more flexibility. For example, you can split filesystems up (especially useful since `amanda` can't span tapes yet) and use `include/exclude` lists.

The other option to keep in mind is compression. You can use the hardware compression on your tape drive, but that makes it much harder for `amanda` to know how much data it can fit on the tape. Software compression is also generally speaking more effective, but has the disadvantage of taking more resources on the backup machine. To combat this, `amanda` lets you do compression on the client. If the machine is extremely slow, you may want to consider having the server do the compressing, but otherwise let the client do the work.

3 Restoring Your Data

We all know it's going to eventually happen. A file will accidentally be deleted, a hard drive will die, you will overwrite your code with a copy of Larry's recipe for chili. Afterall, that's exactly **why** you're setting `amanda` up in the first place. `Amada` gives you 3 ways to do this.

3.1 Low-tech

`Amada` uses only common unix utilities to do its backups. Essentially `dd`, `gnutar`, and `dump/restore`. You can use a utility such as `mt` to control your tape device, and `dd` (in addition to `dump` or `tar`) to restore your data. This is obviously not the best way to go about things, since there will be multiple files on the tape, etc., but at least it can be done. This is especially useful if your backup system has crashed and you need to recover the `amanda` data to restore it.

3.2 amrestore

`amrestore` gives you a higher-level interface to the standard commands. It allows you to do something along the lines of

```
amrestore /dev/nst0 hostname diskname YYYYMMDD
```

The `diskname` and date are optional, but it will at least let you get back the dump from a specific date for a specific drive. You do need to know ahead of time the appropriate tape. `amadmin` can find this for you.

3.3 amrecover

If you turn indexing on in your DLEs, you can use `amrecover` to restore specific files. This gives a similar interface to `restore`, however I have not yet been successful in getting it to work.

4 Resources

- <http://www.amanda.org/>
- <http://www.backupcentral.com/amanda.html>
- `#amanda` on `irc.freenode.net`