Open source backup software Amanda

White paper

"As the founder of a new software development business, I have been keenly aware that we must diligently protect our intellectual property and effectively manage our costs as we scale our operations. Amanda network backup and recovery software has played an important role in providing us with the peace of mind that our valuable data has been protected. It has proven to be easy to implement and highly reliable. We are especially pleased to see Zmanda emerge as a provider of enterprise level support and services for Amanda."

Amit Narayan, PhD
Vice President of Engineering and Founder, Berkeley Design Automation

"I'm very pleased to see Zmanda's interest in bringing enterprise features, stability and support to Amanda. They have taken a leadership role in open source Amanda development and been very responsive to the Amanda development and user community. Their enterprise-level support offering should make it much easier for companies to take advantage of Amanda's technology for data protection."

James Da Silva, the original developer of Amanda

Abstract

This white paper provides brief technical overview of Amanda. It will help you to understand how Amanda works, how it is different from other backup software, and how it can help you solve your data protection requirements.

Table of Contents

Please send your comments about this white paper to ***feedback@zmanda.com***

Introduction to the most popular open source backup software Amanda

The purpose of this white paper[1] is to give brief technical overview of Amanda. We want you to understand how Amanda works, how it is different from other backup software, and how it can help you solve your data protection requirements. On the other hand, we don't want to overwhelm you with technical details that could be very specific to a particular setup or backup policy. Throughout this section we provide links to various places on www.zmanda.com where you can find up-to-date and easy to follow instructions and details about everything you need to know about deploying Amanda in production.

Amanda, the Advanced Maryland Automated Network Disk Archiver, is the most known open source backup software. Amanda was initially developed at University of Maryland in 1991 with the goal to protect files on a large numbers of client workstations with a single backup server. James da Silva was one of its original developers.

The Amanda project got registered on SourceForge.net on November 11, 1999. Jean-Louis Martineau of the University of Montreal has been the gatekeeper and leader of Amanda development in recent years. Over the years more than 250 developers contributed to the Amanda code and many thousands of users provided testing and feedback resulting in stable and robust Amanda. In April of 2006 Amanda was estimated to be deployed at more than 20,000 sites worldwide.

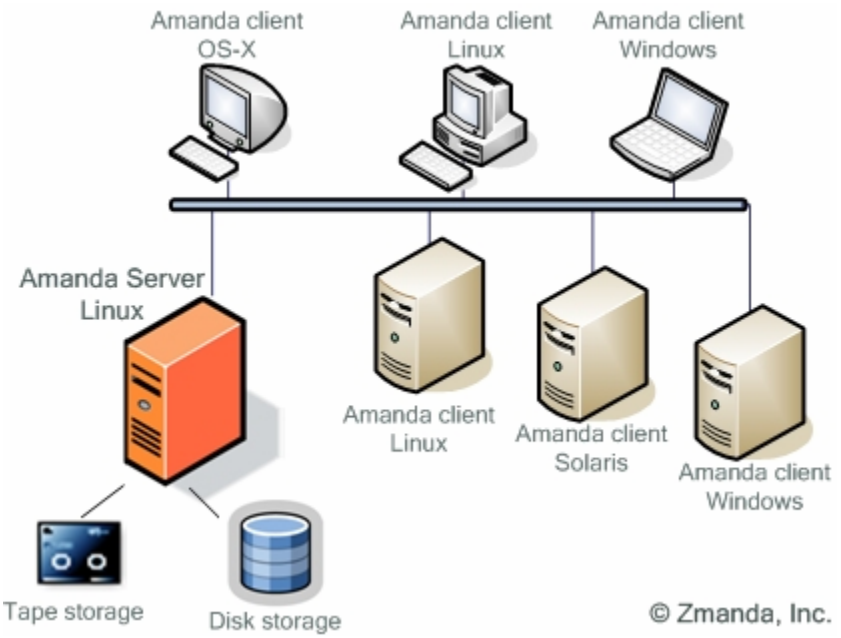
Originally Amanda was deployed in production mostly by universities, technical labs and research departments. Today with wide adoption of Linux in IT at large, Amanda is found in many other places, especially where focus is on applications deployed on a LAMP stack.

Over the years Amanda has received multiple awards from users. For example, in 2005 it received Linux Journal Readers' Choice Award for "Favorite Backup System".

---

[1] This white paper was written by Zmanda, Inc based on contributions by Dmitri Joukovski, John R. Jackson, Alexandre Oliva, Æleen Frisch, Paul Bijnens, Stefan G. Weichinger and many others who contributed to the wealth of published knowledge about Amanda.

Figure 1. Typical Amanda network.



Amanda allows you to set up a single master backup server to back up multiple Linux, UNIX, Mac OS-X and Windows hosts to a very large selection of tape, disk, and optical devices including tape libraries, autochangers, optical jukeboxes, RAID arrays, NAS devices and many others.

Here are a few real life examples of Amanda in production.

A language translation company uses 3 Amanda servers on CentOS in three countries to protect 30+ clients on Solaris 2.6 and 8, Linux and Windows 2000. Different versions of Amanda have been in production for 9 years. The total amount of protected data is more than 500 GB and data grows on average 8 GB per week. One of the sites performs backup to disk only, and the other two backup to both disk and LTO autoloaders. System Administrator recovers files at least once per week because users erase a file by accident. A few times over the years they lost servers because of failed hard drives. Amanda came to the rescue for bare metal recovery.

The University of Newcastle in UK has 2 Amanda servers on Fedora Core with 100+ Linux (Fedora Core, Red Hat Enterprise Linux), Mac OS-X and Solaris clients with more that 2 TB of data. One of the Amanda servers is dedicated to backup of SAP and Oracle on Solaris.

A cinematographic post-production company with credits for “Batman Begins” and “Harry Potter and the Goblet of Fire” has 3 Debian Amanda servers on 2 sites protecting 84 Linux and IRIX clients with total amount of data 26 TB. They recover files about twice per week due to user

error. In three years of production they had three instances of total volume loss despite using RAID arrays. Amanda was able to recover all three lost volumes.

Throughout this white paper we will use more examples of real life Amanda implementations. Based on feedback from many Amanda users with a great variety of configurations and different level of Amanda expertise, we believe that the key reasons for wide adoption of Amanda are:

- Amanda simplifies your life as a System Administrator because you can easily set up a single server to back up multiple networked clients to a tape, disk or optical storage system.
- Amanda is optimized for backup to disk and tape. Additionally, it provides the unique capability of writing backups to tape and disk simultaneously. The very same data could be available on-line for quick restores from disk and off-site for disaster recovery and long term retention.
- Since Amanda does not use proprietary device drivers, any device supported by an operating system works well with Amanda. The System Administrator does not have to worry about breaking support for a device when upgrading Amanda.
- Amanda uses standard operating system ***dump*** and ***GNUtar*** utilities. Since there are no proprietary formats, in case of emergency, data could be recovered with always available standard tools even without Amanda.
- Amanda's unique scheduler optimizes backup level for different clients in such a way that total backup time is about the same for every backup run. Amanda frees the System Administrators from having to guess the rate of data change in their environments.
- Amanda project has attracted a large and active community that grows every day.

Amanda software as a source code tarball and as RPMs for most common versions of Linux is available from www.zmanda.com. Additionally, source code is available from SourceForge.net at http://sourceforge.net/projects/amanda. Some older but stable versions of Amanda are packaged with all common Linux distributions, e.g. Fedora Core, Red Hat Enterprise Server, Debian, Ubantu, OpenSUSE, SUSE Linux Enterprise Server, etc. and not just for x86 architecture, but also for x64, Itanium, IBM p-Series and even IBM S/390 and z-Series mainframes.

Per a survey based on Google hits, approximately 16% of Linux System Administrators know Amanda and approximately 8% of UNIX System Administrators know Amanda. This wide availability of Amanda expertise along with a very friendly Amanda user list provides great resources for IT managers to install and configure Amanda in their environments.

Amanda is the only open source backup software with enterprise support – available from Zmanda, Inc. Support for Amanda is sold as a subscription service to the Zmanda Network (very much along the lines of subscriptions from Red Hat and MySQL). Zmanda also offers indemnification to select buyers of its Amanda Enterprise Edition subscription from any intellectual property infringement issues. In addition, professional services are available from Zmanda and several other organizations for installing and configuring Amanda. You can learn more about Zmanda Network here: http://zmanda.com/network_overview.html.

Amanda documentation written by users for users is available at the Amanda Wiki: http://wiki.zmanda.com. Ease of remote editing by multiple users, an on-going archival of changes and search capability are key features of this Wiki.

Amanda community uses various collaboration tools including Amanda forums: http://forums.zmanda.com.

Amanda users have a very friendly mailing list: amanda-users@amanda.org with archives available at: http://groups.yahoo.com/group/amanda-users

To wrap up this introduction to Amanda**,** we want to share just one of many success stories where Amanda saved the day and made a difference. The story has been told by long time Amanda user Jon LaBadie.

In 1999 I began consulting for a small service organization within one of the US Government Departments, basically their internal telephone company. They used about 40 Windows PC's and 3 Sun servers, the latter running Oracle. For backups they used two separate commercial products and were unhappy with each. A fourth Sun server was already purchased and the tasks were being shifted around, including the UNIX backups.

I was asked for suggestions for a replacement for their backup software before an additional copy was purchased and "non-"support contracts renewed. I did a bit of research and discovered Amanda. I installed it on my home systems, ran it for a week and suggested it to my management. But as was common in that time, free software would not be considered by management. Who would they get support from? What if something went wrong and it was discovered that free software was being used for such an important function as backup? How good can it be if it is free? Thanks, but no thanks. We'll make the safe choice; pay our thousands of dollars for software we are not happy with, just because it is sold by a large company.

So they migrated their backups to a different server with some difficulty. Meanwhile, without telling my management, I started a parallel backup system with Amanda using the oldest Sun server and a spare DAT drive. About a month later the crisis happened. A directory tree from several weeks earlier was needed. I was not involved in the recovery but I thought it was a good chance to compare recovery times from the two systems. About twenty minutes later I had used Amanda recovery to get what I thought they were seeking and copied it to a directory on their system under /var/tmp.

From the other camp I heard much cursing and hair pulling all morning. In the afternoon I ended their torture and, pointing to the /var/tmp directory, asked "Is this what you need?"

Later I learned the problem with the commercial backups was that the backup tapes were keyed to the backup server. Restores could only be made from the same server. The data they needed had been made on the previous backup server which now had neither installed software nor license. The backup tapes were basically worthless.

Management then decided to give Amanda a try as their primary backup system. Eventually they also backed up the PC's using Amanda. As of my last contact with them a year ago, Amanda was still in use in that department.

Summary of important features.

We will start with brief overview of Amanda architecture. This will help in understanding most important concepts in Amanda functionality.

Client server architecture focused on using non-proprietary tools

Amanda is designed to handle large numbers of clients and data, yet is reasonably simple to install and maintain. It scales well up and down, so small configurations, even a single client, are possible. There are many users who back up just a single client that is also the Amanda server. On the other hand, many Amanda users backup hundreds and even thousands of file systems (there could be multiple file systems per protected system) to a large tape library with multiple drives.

The Amanda code is written in C (with some Perl and Shell scripts) and the code is portable to any flavor of Linux and UNIX including Mac OS-X. Windows clients could be backed up today via Samba or via a Cygwin client, which is a Linux-like environment for Windows. Amanda community is actively working on providing a native client for Windows. New Windows client

will take advantage of Microsoft technologies such as Volume Shadow Copy Service (VSS) that provides snapshots of a system's volumes including snapshots of open files.

The biggest advantage of Amanda over any other backup software is that Amanda does not use any proprietary formats. For movement of data from the file system on a client to the tape, disk or optical disk Amanda uses standard operating system utilities such as ***dump*** and ***tar*** or open source utilities available in many operating systems such as ***GNUtar, smbtar*** and ***star***. Depending on which one is the best match for your file systems, directories and files, you can mix and match these utilities as you wish Since you use standard utilities, you can be confident that these utilities will always be available to you. Another advantage of using standard utilities is that in case of disaster recovery or any other emergency you can recover your data even without Amanda having being installed. We will explain how to recover data without Amanda when we discuss Amanda restores.

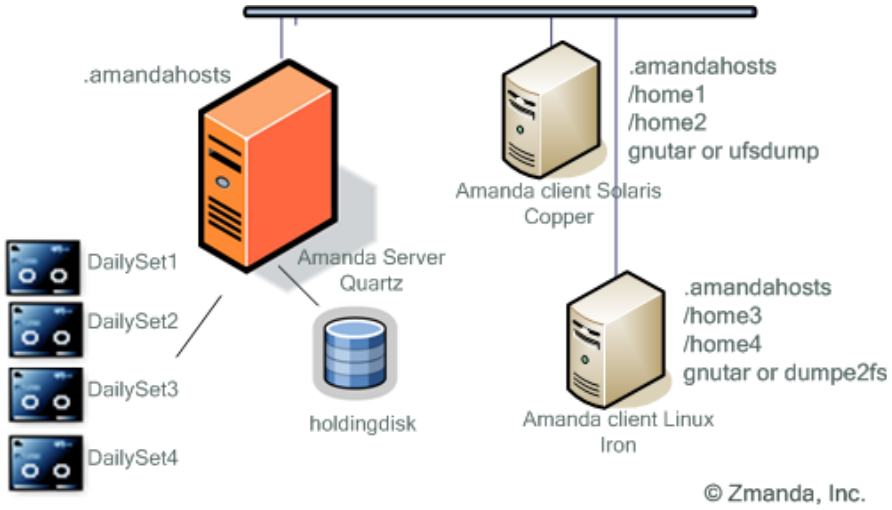
Since Amanda uses standard utilities it provides the following:

- Backup of sparse files
- Backup of hard links
- No changing of file timestamp during backup
- Exclusions of files and directories

From System Administrator perspective it is very important that Amanda does not use any proprietary device drivers and any device supported by an operating system works well with Amanda. In practical terms it means that Amanda supports a wide range of tape storage devices and new devices are usually not difficult to add. Many tape changers, stackers, jukeboxes and tape libraries are supported by using special tape changer scripts to provide truly hands-off and lights-out backup. Basically, if you can read and write to your tape drive and move tapes in your tape library with standard operating system commands such as ***mt,*** Amanda will work with your tape library. Since Amanda doesn't use proprietary device drivers, one more benefit is you don't have to worry about breaking support for a device when upgrading to the latest version of Amanda.

To understand Amanda architecture and inner workings, let's take a look at a simplified Amanda configuration and review an example of a backup cycle.

Figure 2. Amanda server with two backup clients.



To simplify our discussion let's assume that we have only two Amanda clients that run on two workstations: workstation "Copper" running Solaris and workstation "Iron" running Linux. Each workstation has two file systems with users' data that we want to protect. Amanda server Quartz is installed on a different Linux host and, for simplicity sake, we don't backup Amanda server itself. **In your production and even in your evaluation environment you should always backup the Amanda server itself**. Let's also assume that we want to run a full backup once in four days and incremental backups between full backups.

Amanda is designed as a traditional client - server architecture. The Amanda server, also historically known as the tape host, is connected either directly or over the Storage Area Network to a tape drive or tape changer. Each client backup program is instructed to write to standard output, which Amanda collects and transmits to the tape server. The client-server architecture provides these benefits:

- It ensures scalability of Amanda from environments with a single client and CD-ROM to environments with hundreds of clients and large tape libraries with multiple tape drives and hundreds of tapes.

- It allows all configurations to be done on the Amanda server. Once the initial configuration of Amanda is done, you can easily add additional clients without worrying about breaking your tested backup procedures.

- It allows some CPU-intensive operations such as compression or encryption to be done on a client before sending backup images to the Amanda server.

Considering the ever increasing importance of security for backup data from privacy and compliance perspective, let's go over a brief overview of Amanda security.

Amanda security

Amanda clients communicate with Amanda server via its own network protocols on top of TCP and UDP. The Amanda's client-server communications do not suffer from the security holes inherent in the traditional /etc/rmt approach used by dump, e.g. using an ***.rhosts*** file in root's home directory.

As in every other client-server setup, you should ensure that only your own and trusted Amanda server is able to communicate with Amanda clients. Amanda achieves that by using the file ***.amandahosts.*** You can see that on Figure 2 there are three ***.amandahosts*** files, one on the Amanda server "Quartz" and one per each Amanda client. On a client side you have to add the name of the Amanda server (or Amanda servers if you prefer the same host to be protected by multiple Amanda servers) and the Amanda user that are allowed to backup the client. For example, ***.amandahosts*** file for a Linux client "Iron" on Figure 2 should have the following entry:

quartz.zmanda.com amandabackup

That tells the Amanda client "Iron" to let Amanda server "Quartz" to communicate with user "amandabackup".

During restores you need an access to Amanda server. For configuration presented on Figure 2 ***.amandahosts*** file on a tape server "Quartz" should have the following entries:

iron.zmanda.com root
copper.zmanda.com root

These entries tell Amanda server to allow the "root" user on each client to run restores. For security reasons Amanda was designed to allow only the root user to restore data.

In addition to using authentication via ***.amandahosts*** for stronger data transport security and backup client authentication Amanda can use OpenSSH. This allows Amanda to protect the transfer of data between clients and backup server with strong authentication- and authorization-

mechanisms. The current 2.5 version of Amanda also features an abstracted secure communication API that enables developers to easily add different communication plugins between backup server and client.

To protect data on the backup media itself, Amanda 2.5 provides the ability to encrypt backup data with symmetric or asymmetric encryption algorithms (using either ***aespipe*** or ***gpg***). Encryption is very expensive in terms of CPU utilization and that is why depending on availability of CPU cycles, the Amanda encryption can be done either on the server or the client. In addition to unloading the Amanda server CPU, client site encryption also ensures security of data on a wire, which could be important for backing up remote clients. Because of CPU constrains you might choose to encrypt just some data. Amanda is flexible enough to configure data encryption for a single directory or even for a single file. If ***aespipe*** and ***gpg*** don't match your encryption requirements, Amanda will work with your custom encryption utilities.

Amanda works with Security-Enhanced Linux (SELinux) and it also works reasonably well with common types of firewalls between Amanda server and clients as long as you select UDP and TCP port ranges during initial setup. Please check installation and configuration details for firewall setup at http://wiki.zmanda.com.

To conclude the brief overview of Amanda security, we want to emphasize that the flexibility of the security configurations allows Amanda to fit well into security policies and processes of most IT environments including the organizations with strict security requirements.

You might recall that Amanda is actually an acronym, and 'D' in Amanda is for Disk. To explain how Amanda moves data from client to its final destination on tape or disk we will introduce a very important Amanda concept of holding disk.

Holding disk

On Figure 2 you can see that Amanda server "Quartz" has a "holding disk" attached. Holding disk is one or several directories on any file system that is accessible from the Amanda server. It could be as small as a single 10 GB directory on your Amanda server drive or as large as 5-10 TB on a fibre-attached RAID array. As the name suggests, holding disk is used as a cache to store backup data from all Amanda clients. Each set of backup data from a client file system or a client directory is just a bunch of files on the holding disk. Later, an independent process flushes individual backup images from holding disk to tape at maximum throughput possible to keep tape drive streaming. Using holding disk as a staging area for backups has several benefits:

- Modern tape drives are very fast, for example LTO-3 has throughput of 80 MB/s. Even Gigabit network can not feed backup data from a single client through the Amanda server to LTO-3 drive fast enough to avoid stop and wait known as ”shoe-shining” which reduces throughput and shortens life of media and the drive. Holding disk collects data from all clients and as soon as the first backup is complete, the holding disk starts feeding data to tape as fast as Amanda server can push it. However, many users prefer to complete backup of all clients before they start flushing data to tape.

- Holding disk can accept data streams from multiple clients in parallel to overcome sequential nature of a tape. Instead of writing one backup to tape after another, you can configure multiple backups running in parallel and make full use of your available network bandwidth and reducing total backup time. If network becomes your bottleneck for performance, you can reduce total backup time by adding another NIC to your backup server or dedicating a separate network for backups.

- Lastly, using holding disk provides additional safety in case you have a bad- or wrong tape, or no available tape at all. Your backup will be complete even if you forget to insert a new tape before taking the day off.

Amanda supports multiple holding disks so that backup images from different clients could be sent to different holding disks. That increases scalability of Amanda and provides better load balancing for I/O since holding disks could be on different controllers.

Often times new Amanda users ask how large the holding disk should be. Since for a typical “full and incrementals” backup cycle, most backups are just small incrementals, even a modest amount of holding disk space can provide better flow of backup images to a tape than without the disk. Good rule of thumb is there should be enough holding disk space for the two largest backup images at the same time, so one image can be coming into the holding disk while the other is being written to tape. So, for example if on Figure 2 full backup for “Copper” is 50 GB and full backup for “Iron” is 30 GB the optimal capacity of holding disk on “Quartz” should be at least 80 GB. If that is not practical, any amount that holds at least a few of the smaller incremental backups is better than no holding disk at all. With today’s low disk prices a good sized holding disk is well worth the investment.

On the other hand, some Amanda users have significantly larger capacities for holding disk. For example, a very large Japanese manufacturing company has 4 Amanda servers running on Solaris

and BSD protecting more than a hundred Amanda clients on BSD, Windows, Linux, HP-UX and Solaris running Oracle. One of their holding disks is on a RAID array with total capacity of 4 TB. Fast arrays and Amanda servers with high I/O allow streaming throughput from holding disk to tapes at approximately 120 MB/second.

Flexibility of Amanda allows configurations without holding disk with backups going directly to tape, but then backups can be written to tape only sequentially instead of parallel to the holding disk. Obviously, lack of holding disk will significantly reduce backup performance.

If the holding disk is for temporary keeping of backup files, than how does Amanda decide what to send to the holding disk in the first place? Let's take a look at Amanda unique way to schedule backups

Backup scheduling

Most backup products provide basically the same backup scheduling. The System Administrator configures software to perform full backup on Sunday, or every other Sunday, or the last day of the month, etc with different levels of incrementals between full backups. The biggest problem with this approach is that it does not provide any load balancing. You have to make sure that enough resources are available to manage peak demand for backup server CPU, network, and I/O during full backups. Since you perform full backups only once in a while, your resources are under-utilized most of the time. More often than anybody wants to admit, on Monday morning the System Administrator finds out that Sunday's full backup was not complete because there were not enough tapes available in a library. Other Mondays you might find that your full backups are still running and users are calling you to kill all backups. Of course, you can figure out yourself how to achieve load balancing by instructing your backup software to distribute full backups among your clients throughout the week, but than you have to make sure that no changes in your environment, for example, new clients break down your balancing schema.

Amanda provides a unique approach to scheduling that optimizes load balancing of backups and simplifies your life. Instead of giving Amanda the exact instruction "Do a full backup every Sunday for clients A, B, and C and full backups on Wednesday for clients D, E, and F and incrementals all other time" you just set up a few ground rules that control Amanda scheduling. For example, you might give Amanda the following rule "Do at least one full backup within 7 day period and do incrementals all other days with maximum time between full backups of 7 days". The maximum time between full backups is called "dump cycle".

For any dump cycle specified by you, Amanda finds an optimal combination of full and incremental backups from all clients to make the total amount of backup data per backup run as small as possible and consistent from one backup run to another. To find such a balance Amanda uses the following considerations:

- The total amount of data to be backed up as reported by each client based on amount of data changed since last backup.

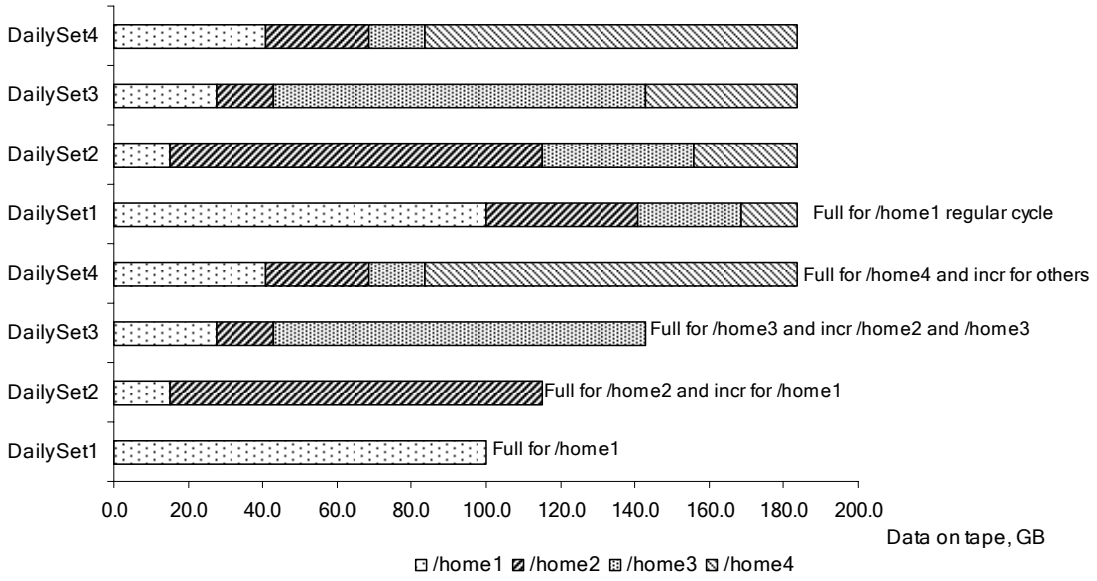
- The maximum time between full backups (dump cycle) specified by you.

- The size of backup media (tape or disk) available for each backup run.

To calculate the optimal backup level, Amanda starts every backup run with so called “estimate phase”. Every Amanda client runs special process to determine which files have changed and what is the total size of all changed files. The estimate phase can take some time especially if there are many clients and file systems. If some file systems are not very dynamic and files don’t change much, you can tell that Amanda, which will save time during estimate phase. After collecting data from all clients, Amanda goes into so-called “planning phase” and calculates the optimal combination of full and incremental backups for all clients.

Let’s take a look at how Amanda will schedule backups for clients on Figure 2 assuming that each home directory is 100 GB, data change rate is 15%, and dump cycle is 4 days. For simplicity let’s assume that Amanda writes each backup run to a new tape labeled DailySet1 to DailySet4 and that all incrementals are level 1 (level 0 is usually defined as a full backup) meaning everything that changed since the last full backup.

Figure 3. Illustration of Amanda scheduling



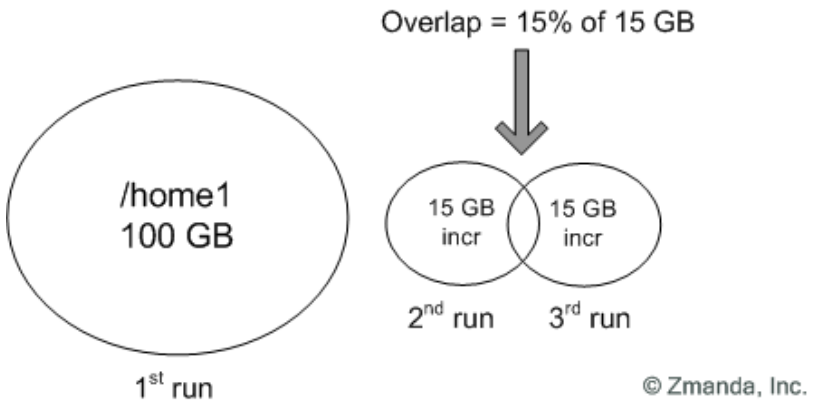
For each run Amanda schedules a full backup for exactly 1/(dump cycle) of the total amount of data. Since our dump cycle is 4 days, for DailySet1 Amanda will do the full backup for ¼ of the data, for example /home1. For DailySet2 Amanda will do a full backup for another ¼ of data, for example /home2, and an incremental backup for /home1 which is 15 GB (15% of 100 GB). For DailySet3 Amanda will do a full backup of /home3 and incrementals for /home1 and /home2. After initial startup period of 4 days, Amanda will run a full backup for one of the /home directories and incremental backups for all the others. Let's calculate what would be the total amount of data on each DailySet tape.

Table 1. Illustration of Amanda scheduling – total amount of data per DailySet tape.

	DailySet1	DailySet2	DailySet3	DailySet4	DailySet1	DailySet2	DailySet3	DailySet4
/home1	100.0	15.0	27.75	40.84	100.0	15.0	27.75	40.84
/home2		100.0	15.0	27.75	40.84	100.0	15.0	27.75
/home3			100.0	15.0	27.75	40.84	100.0	15.0
/home4				100.0	15.0	27.75	40.84	100.0
Total	100.0	115.0	142.75	183.59	183.59	183.59	183.59	183.59

It is trivial to calculate the total amount of data for DailySet1 and DailySet2. For the third run for backup of /home1 we have to consider that 15% of data that was backed up on DailySet2 which was 15% of 100 GB changed again, see Figure 4.

Figure 4. Clarification to total amount of data per tape discussion.



To avoid double counting we have to subtract that small overlap area from 30 GB. So for DailySet3 the size of the incremental for /home1 will be 30 GB - (15 GB x 15%) = 27.75 GB. Following the same logic, for DailySet4 the incremental for /home1 is not 45GB but 45GB - (27.75GB x 15%) = 40.84 GB.

This example is only an illustration for explaining Amanda's approach to scheduling. In reality Amanda uses all nine levels of incremental backups to minimize total amount of data on tape.

In addition to a traditional schema with full backups and incrementals in between, Amanda also supports:

- Periodic archival backup, such as taking full backups to an off-site.
- Incremental-only backups where full backups are done outside of Amanda, such as very active areas that must be taken offline, or no full backups at all for areas that can easily be recovered from vendor media, e.g. installation CD for an operating system.
- Always doing full backups of databases that change completely between each run or any other critical areas that are easier to deal with during an emergency if they are a single-restore operation.

It's easy to support multiple configurations on the same Amanda server, for example doing traditional full backups and incrementals in between on a weekly basis and also doing additional monthly full backups for off-site storage. Multiple configurations can run simultaneously on the same tape server if there are multiple tape drives.

When you decide on a length of your own dump cycle, you should take into consideration that shorter dump cycles, for example, 3-4 days make restores easier because there are fewer incrementals, but use more tape and require more time to backup. Longer dump cycles allow Amanda spread the load better over multiple tapes but may require more steps during a restore. More information about how to choose a reasonably balanced dump cycle depending on amount of data, tape drive capacity, etc is available at http://wiki.zmanda.com

Let's take a look at Amanda tape management.

Tape Management.

Each tape should be labeled before use by Amanda command ***amlabel***, e.g. DailySet1. There is a default template for labels, but you could define your own label templates. Labeling prevents overwriting of tapes with valid backup images and allows Amanda server to keep track on all tapes that were labeled.

At the present time Amanda starts a new tape for each backup run, for example, each nightly backup, and does not provide a mechanism to append a new run to the same tape as a previous run.

Based on your backup retention policy, Amanda keeps track on expiration date for each labeled tape and after an old backup image expires, Amanda will re-use that tape for new backups. However, you can configure Amanda not to re-use specific tapes. For some data you might choose to never expire your backup images and use Amanda for creating archives. Amanda' recent support for optical media becomes very useful for archiving.

For backup of large amount of data Amanda supports using multiple tapes in a single backup run, for example backups from clients A, B, and C could go be written on one tape and backups from clients E, F, and G could be written to another tape.

In the past Amanda could not span multiple tapes for a single backup image and System Administrators had to break large file systems into smaller chunks, e.g. into several directories.

Starting with version 2.5, Amanda can span multiple tapes. That alleviates a significant limitation and is a major step forward in terms of scalability and simplicity of its use. The size of the backed up images is no longer restricted to a single tape and there is no need for the System Administrator to artificially segment data into parts which can fit into a single tape.

Device management.

We already mentioned that Amanda does not use any proprietary drivers for tape or optical devices and if an operating system can write and read from your tape drive, Amanda will work with that device. You have to make sure your tape devices are configured as non-rewinding devices, e.g. /dev/nst0, /dev/nst1, etc. You also have to select so-called "tape definition" specific to your tape drive technology. There are many default tape definitions provided with Amanda. Here is an example of tape type definition for DLT8000:

```
define tapetype DLT8000 {
comment "Quantum DLT8000 created by tapetype"
length 38130 mbytes
filemark 29 kbytes
speed 5627 kps
}
```

Similarly, you will have to select a tape changer script for your tape changer. Examples of tape definition for most commonly used tape drives and details about configuring tape drives and tape changer scripts are available at http://wiki.zmanda.com.

For a long time Amanda provides the ability to use disk as the target media for backup. Dedicated directories are used as "virtual tapes" called ***vtapes***. You work with ***vtapes*** exactly the same way as you work with "real" tapes, for example you have to label ***vtapes*** before they could be used by Amanda. There are several usage scenarios for ***vtapes:***

- Backup to disk is a good way to test and evaluate Amanda before you decide to invest in an expensive tape library.

- Backup to disk has become a viable option for production from cost perspective. You get all the benefits of having backup of your data without challenges of managing finicky tape drives.

The most interesting scenario is using tapes and disk at the same time. Amanda provides an interesting functionality called RAIT, which stands for “Redundant Array of Inexpensive Tapes”. Initially RAIT was designed to increase redundancy. This is the same technology as RAID where data is striped over several disks. Amanda supports RAIT with 2, 3 and 5 drives.

A 3-drive RAIT will write 2 data streams and one parity stream, and give you twice the capacity, twice the throughput, and the square of the failure rate (for example, a 1/100 failure rate becomes 1/10,000, since you might loose data only if two tapes are faulty or not available). Similarly, a 5-drive RAIT set will give you 4 times the capacity and 4 times the throughput.

A 2-drive RAIT duplicates the output stream and each output stream can have either the same or different media targets. If you have the same media targets, for example, 2 tape drives you get the exact copies of your backup data called clones. You can keep one clone on-site for occasional restores and take another clone off-site for disaster recovery.

If you have different media targets, than you can keep your backup data on disk for 2-3 weeks for occasional restores, and for long term retention you have a copy on tape. Most restores happen within 10 days after a file has been lost and ability to restore data quickly from disk becomes very important.

The ability to write the very same data to disk and tape at the same time is another unique feature of Amanda. At the time of writing this text we don’t know any other backup product capable of doing that.

Since you already understand all important Amanda concepts, let’s take a look at how to configure Amanda backup cycle.

Configuring Amanda.

Detailed instructions how to install and configure Amanda client and server are available from http://wiki.zmanda.com. Here we want to provide the configuration roadmap.

The preferred way to install Amanda is from the RPMs found at www.zmanda.com. To compile Amanda from source:

- Create “amandabackup” user in the “disk” group
- Unpack, compile, and install from the source archive. Specify options for a client-only install.
- Add Amanda related entries to /etc/services and /etc/xinetd.d directory and restart ***xinetd***

  In any case, after you have installed Amanda, you must tell the client which servers are allowed to connect:

- Edit ***.amandahosts*** file to enable authentication between client and server

To install the Amanda server, you can also use RPMs. If you want to compile from source:

- Create “amandabackup” user in “disk” group.
- Unpack, compile, and install from the source archive. Specify options for a client-and-server install.
- Add Amanda related entries to /etc/services and /etc/xinetd.d directory and restart ***xinetd***

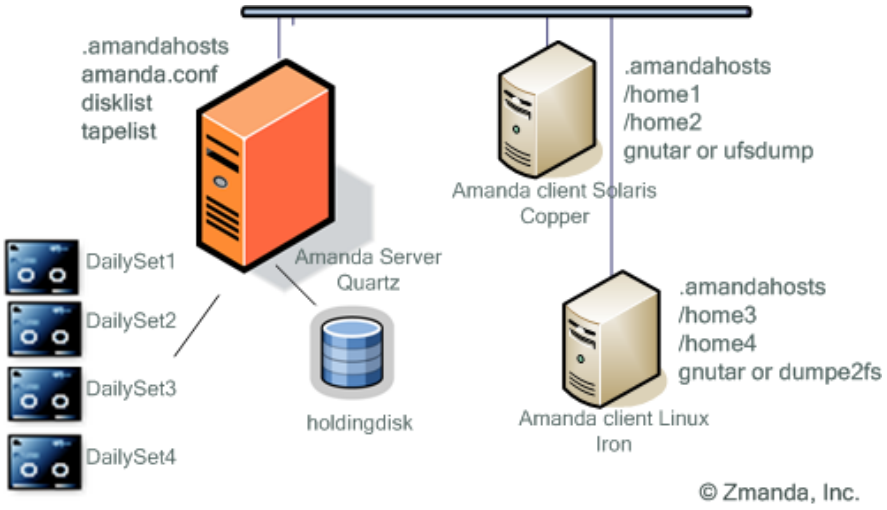
  Once installed, there are a few steps to configure the Amanda server.

- Create the Amanda configuration directory /etc/amanda/(config name) (in case you use Zmanda packages)
- Copy a sample of amanda.conf into /etc/amanda/(config name)
- Create ***disklist*** file in /etc/amanda/(config name)

- Edit ***.amandahosts*** file to enable authentication between client and server
- Edit configuration files ***amanda.conf*** and ***disklist***
- ***Set up the device (create device nodes or directories)***
- Label the media (tapes or vtapes) using ***amlabel***
- Configure ***cron*** job to schedule Amanda backup runs.
- Run ***amcheck*** program to verify there are no problems with configuration, client server communications, the holding disk and the tape.

  Note that while one machine can be both a client and a server, there is no need to perform both of the above procedures; installing the server normally includes the client.

Figure 5. Amanda configuration files.



The most important file for configuring your Amanda setup is ***amanda.conf*****.** The example file is quite large with more than 700 lines (and that is why we don't provide an example here, see details at http://wiki.zmanda.com), but self-explanatory with easy to follow comments and examples. That file defines HOW you do your backups by configuring the following parameters:

- Local settings such as name of your organization, e-mail where to send backup completion report and warnings, location of Amanda directories, etc

- Device and network settings such as names of your tape devices, tape type definitions, tape changer script, tape labeling template, network bandwidth for Amanda to use, etc

- Backup policy settings such as dump cycle, specifics about incremental backups, number of backup runs per dump cycle, number of tapes in rotation, etc.

- Holding disks locations with capacities available to Amanda

- Instructions for how to perform backups such as whether to use ***dump*** or ***gnutar,*** details about indexing data, encryption, compression, etc.

Instructions about WHAT to backup are provided in the ***disklist*** file. For example, for backing up /home directories for clients "Iron" and "Copper" on Figure 5 we will need the following Disk List Entries, often times referred as DLEs:

```
# hostname diskname dumptype
Copper /home1 stable
Copper /home2 stable
Iron   /home3 normal
Iron   /home4 normal
```

dumptype in the disk list entry refers to a "***dumptype***" that should be defined in the ***amanda.conf*** file. ***Dumptypes*** specify backup related parameters, such as whether to compress the backups, whether to record backup results in /etc/dumpdates, the disk's relative priority, exclude lists, etc. Here are the sample definitions for "stable" and "normal" dumpcycles that we use for "Copper" and "Iron" entries in ***disklist*** file:

```
define dumptype  normal {
comment "gnutar backup"
holdingdisk yes # (on by default)
index yes
program "GNUTAR"
priority medium
       }

define dumptype  stable {
comment "ufsdump backup"
holdingdisk yes # (on by default)
index yes
program "DUMP"
priority medium
       }
```

Many parameters in ***amanda.conf*** have default values that you don't have to edit, but because all parameters are available to your for editing you have full control over your backup environment.

So far we discussed the most typical situation with Amanda client configured on the system to be protected. However, there are various scenarios in which a System Administrator may decide to mount a file system via NFS or Samba on the Amanda server, and have the Amanda client running on the same system (the Amanda server) backup these networked file systems.

Backup up clients via NFS or Samba (SMB/CIFS)

Comparing to the traditional approach of using an Amanda client on the system to be protected, there are several advantages of backing up via NFS or Samba:

- You don't need to install and configure Amanda client software on the system to be protected. More systems can be added to the list of protected systems by modifying files on one system.

- A native Amanda client may not be available for a particular operating system. A NFS/SMB approach gives you the ability to backup such an operating system.

- You use significantly less CPU and memory resources of the protected system.

- Better integration with the company IT policy if the important data from the protected systems is already available via NFS or CIFS protocol

However, while considering this approach you should be aware of some trade-offs:

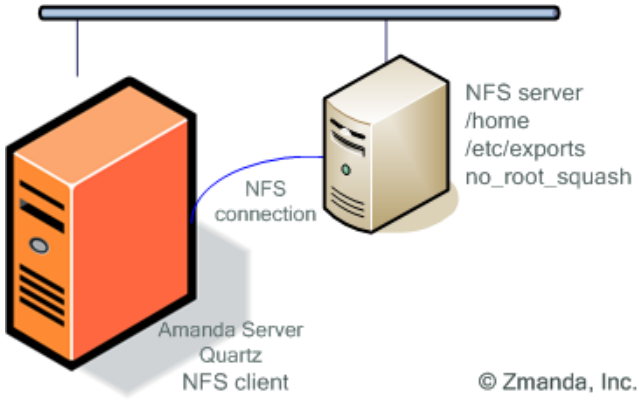
- You will need to consider the security issues of the mounting mechanism, i.e. SMB or NFS. You will not be able to use the mechanism provided by Amanda to encrypt the data while being transferred from the client to the server. You will also need to understand the security implications of running NFS or Samba on the system you want to protect in the first place. For example, if you do not want to run NFS server all of the time on the system you want to backup, you will need to craft a synchronization scheme, and whereby the NFS server on the system starts running just before backup is initiated on the Amanda server.

- Access privileges will need to be carefully worked out. Amanda server will need both read and write privileges on the NFS mount point. Read permission is necessary during the backup phase. Write permissions are necessary during a restore.

- You will not be able to use local file system based mechanisms for optimizing the backup process. For example, you will not be able to use dump or xfsdump on a file system being accessed over the network.

- You cannot backup any open files being accessed via Samba. You cannot backup extended file attributes via Samba.

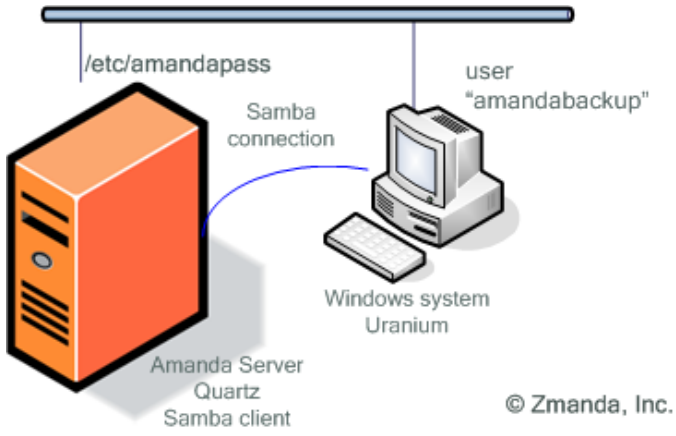
Figure 6. Configuration issues with NFS based backup



You need to install and configure NFS server on the target system, and a NFS client on the Amanda server.

At this point, export the file systems to be backed up (by listing them in /etc/exports file of the target system). You need to make sure that Amanda server can access all the files that are needed to be backed up. In many cases this means turning on the "***no_root_squash***" option on the NFS share that is being backed up - so the Amanda server can access all files. Note that the "hostname" in the corresponding Disk List Entry will be the system where NFS share is being mounted (not the target system), for example on Figure 6 it would be "Quartz"

Figure 7. Configuration issues while backing up a Windows based system using Samba.



You need to install Samba client on the Amanda server. You don't have to explicitly mount the remote file system. Amanda is well integrated with the ***smbclient*** utility (ftp-like client to access SMB/CIFS resources on servers). It uses the "-T" option of the ***smbclient*** utility to create tar compatible backups of all the files on a SMB/CIFS share. Amanda will clear the archive bit of the files (on the Windows based target) it backs up, hence enabling the incremental backup process.

A user must be created on the Windows system with full access rights (read/write) to the share, for an example on Figure 7 it would be user "amandabackup". Amanda will connect to the share via this user. If the user does not have full access, incremental backups will not work and the whole share will be backed up every time (because the archive bits are never reset). Note that if any other program on the Windows system goes and resets the archive bit of a file, Amanda may not backup that file.

Other than the standard Amanda configuration, you need to create a file ***/etc/amandapass*** on the system where ***smbclient*** utility is run. This file contains authentication information to access specific Windows shares. Also note that the "hostname" in the corresponding Disk List Entry is the system where ***smbclient*** is run, and not the Windows system being backed up, for example on Figure 7 it would be Amanda server "Quartz".

In conclusion of backup via Samba discussion we want to reiterate that many Amanda installations protect Windows servers and PCs in production. For example, the Radiology Department at large Mid West University is using Amanda since 1999. In the past they had their Amanda server running on IRIX, AIX, and Solaris, but the current Amanda server runs on Linux with indices replicated to another server. They backup more than 70 Linux, Solaris, IRIX, Mac

OS-X and Windows clients with total amount of backup data around 4 TB. The holding disk is 1.4 TB and the dump cycle is 90 days. All Windows clients are protected via Samba. Several times per months they recover files because of user error or hard drive failures and they never had data because Amanda was always able to recover lost files.

That brings us to a brief overview of Amanda recovery.

Amanda recovery

amrecover and ***amrestore*** are two programs to restore Amanda backups. ***amrecover*** restores files by using an interface that allows browsing of your backup file index to a certain date and choosing files you need to restore. Of course, in order to use ***amrecover*** you should enable indexing of backup files when you specify ***dumptype*** in ***amanda.config***. After you make your selection of files, Amanda finds the required tape, looks for backup image, decompresses the image if required, brings the image over the network to the client and pipes it into the appropriate restore program with the arguments needed to extract the requested files. In case you have to restore your files from incremental backups, Amanda will instruct you about correct order of tapes you need. For security ***amrecover*** must run as root on the client and you should list root as the remote user in ***.amandahosts*** on Amanda server.

Full file system recovery should be done with ***amrestore*** which retrieves the whole file system images from tape.

amrecover can be done on any client including Amanda server. ***amrestore*** can be done only on the Amanda server. You have to use ***amrestore*** when you don't have backup index.

If your backup policy specifies backup of everything including the operating system, you can do bare metal recoveries with Amanda:

- Replace the disk
- Boot with LiveCD, e.g. Knoppix
- Format and partition the disk
- Use ***amrestore*** on Amanda server to restore everything on a new disk including the operating system (you might need to restore incrementals as well)

- Create a boot loader on a new disk.

The Amanda tape format is deliberately simple so in case of emergency, restoring data could be done without any Amanda tools. The first tape file is a volume label with the tape Volume Serial Number and date it was written. It is not in ANSI format, but is plain text. Each file after that contains one image using 32 KB blocks. The first block is an Amanda header with client, area and options used to create the image. As with the volume label, the header is not in ANSI format, but is plain text. The image follows, starting at the next tape block, until end of file.

To retrieve an image with standard UNIX utilities if ***amrestore*** is not available, position the tape to the image, then use ***dd*** to read it:

```
# mt rewind
# mt fsf NN
# dd if=$TAPE bs=32k skip=1 of=dump_image
```

The skip=1 option tells dd to skip over the AMANDA file header. Without the of= option, ***dd*** writes the image to standard output, which can be piped to the decompression program, if needed, and then to the client restore program.

Since the image header is text, it may be viewed with:

```
# mt rewind
# mt fsf NN
# dd if=$TAPE bs=32k count=1
```

In addition to describing the image, it contains text showing the commands needed to do a restore. Here's a typical entry for /home2 file system on iron.zmanda.com. It is a level 1 dump done without compression using Solaris ***ufsdump*** program:

```
AMANDA: FILE 20060418 copper.zmanda.com /home2 lev 1
comp N program /usr/sbin/ufsdump
```

To restore, position the tape at start of file and run:

```
# dd if=$TAPE bs=32k skip=1 | /usr/sbin/ufsrestore -f... -
```

As with any backup system, you should test and retest your restore procedures so you know exactly what to do when disaster strikes.

Future plans

One of the main challenges in IT today is the overall security of systems. Since security is such a fundamental part of backup (especially when people loose un-encrypted tapes), Amanda team plans to continue hardening all aspects of security with Amanda.

There is fundamental shift in backup industry with disk becoming the primary media for backups. Even though Amanda has been designed for backup to disk from the very beginning, Amanda team plans many backup to disk improvements, for example providing multiple simultaneous backups and restores from disk.

Many of Amanda users have constant battle with overwhelming data growth. Amanda has to be up to the task and we are working on increasing scalability and performance.

Wide adoption of open source and especially Linux brings Amanda to production environments with Oracle, MySQL, SAP and many other applications. There are many users who successfully deploy Amanda in such demanding environments, and Amanda team is working on an application API that will simplify backup of those applications.

Amanda was always striving to simplify life of a System Administrator and we will continue to work on simplification of installation, administration and recovery while giving the System Administrator the full control of how you want to do your backups.

As you can see from this short list, the development of Amanda continues toward addressing of real-world requirements by people like you. The Amanda development team and the Amanda community will further maintain and enhance this powerful and well-known software suite.