StorTrends Whitepaper

"Advanced Snapshot Technology"

American Megatrends International GmbH
15.02.2008

INHALTSVERZEICHNIS

1. Advanced Snapshot Technology for AMI's StorTrends Storage Appliances

A snapshot is a point-in-time copy of a volume similar to taking a picture of the data at that instant. The Snapshot can be taken according to a schedule or under the direction of an administrator. Each snapshot is similar to an incremental backup of the data within the system; what distinguishes snapshots is the way that the data is backed up, improving both space utilization as well as performance. Snapshots look and behave like complete backups and they can be mounted as volumes. The ease and flexibility with which these operations may be performed has been instrumental in speeding up the adoption of snapshot technology in the IT industry.

SNAPSHOT PERFORMANCE

The storage industry has adopted snapshot technology because of its inherent benefits like ease of use, manageability and the utility it provides. There are drawbacks to traditional implementations, that is ongoing performance degradation penalty. This is often overlooked issue during the storage purchasing decision. In the traditional model, there is 60-90 percent performance degradation on the volume after a Snapshot is taken. The degradation becomes more acute after many Snapshots are created. AMI's Snapshot technology is unique which offers near ZERO performance degradation by reducing the number of IO operations by a factor of three. By avoiding unnecessary table and pointer updates, AMI's Snapshot is able to achieve the highest levels of performance while still offering up to 254 Snapshots per volume.

SNAPSHOT CREATION

Many companies set aside a partition that will be used for storing Snapshot data that reserve disk space and requires three IO operations for every write. With AMI's solution, there is no need to reserve disk space for creation of a Snapshot. This leads to more efficient disk usage as there is no need to pre-allocate space up front. With

AMI's StorTrends products, Snapshot creation takes less than half a second and up to 254 Snapshots for both block and file level can be taken per volume.

SNAPSHOT ROLLBACK

Rolling back to a previously created Snapshot is what makes the Snapshot technology so attractive to administrators. With very little training, one can rollback to any previously created Snapshot without the hassle and time delay that a restoration from tape would take. AMI's StorTrends Advanced Snapshot technology allows administrators to rollback to any Snapshot in less than one second. This drastically reduces the downtime window, which saves time and money.

StorTrends Advanced Snapshot Highlights

- Redirect on Write (ROW) snapshots
- Near-ZERO performance degradation
- Create a Snap in less than half a second
- Rollback in under a second to any Snap
- Up to 254 snapshots per volume
- Snapshot Scheduling
- Rollback to any Snapshot
- Writeable snapshot support
- Online defragmentation

THE NEED FOR SNAPSHOTS

In any storage system, the feature that is of paramount importance to the user is data protection. While the user takes for granted that data will not be lost due to a system crash or a design bug, the user also expects that a storage system will protect him or her from inadvertent deletions, unwanted modifications, malicious agents, etc.

Various solutions have been proposed by the storage industry, and have been adopted to various degrees. One very common method is by taking regular backups, either on the storage system itself, or onto tapes. This method is foolproof, but has the major disadvantage of requiring massive storage capacity, and additionally, of causing substantial disruption in service while a backup is being made. Additionally, because of these disadvantages, it is not possible to create backups that are more frequent than about once a day, causing a large window of loss if data must be restored.

Snapshots are rapidly being adopted as an industry-wide standard for protecting data at a very fine granularity. Either according to a schedule or under the direction of an administrator, the storage system takes point-in-time 'snapshots' of the contents of a volume in a storage system. Each snapshot is similar to an incremental backup of the data within the system; what distinguishes snapshots is the way that the data is backed up, improving both space utilization as well as performance.

Snapshots look and behave like complete backups – they can be mounted as volumes, read simultaneously without affecting the volume that they are snapshots of, rolled back onto the volume if necessary, and deleted to free space. Some snapshot implementations also allow writable snapshots, in which snapshots may be modified to create branches of volumes. The ease and flexibility with which these operations may be performed has been instrumental in speeding up the adoption of snapshot technology in the IT industry.

As snapshot technology has evolved into a mature and stable platform, various vendors have found ways of exploiting snapshots to make various other storage-related tasks faster and more convenient. Two typical examples are Information Lifecycle Management (ILM) and Remote Replication. These features are crucial to any storage system, but in a system that supports snapshots, their implementation becomes faster, simpler, and consequently, more reliable.

AMI's storage offerings beginning with the iTX 2.0 engine have industry-leading support for snapshots, offering an exhaustive array of innovative features, with an implementation that optimizes performance and storage capacity utilization.

IMPLEMENTING SNAPSHOTS IN A STORAGE SYSTEM

The idea that distinguishes snapshots and full-system backups is that in a system that has active snapshots, data at a particular address is not backed up until it is modified. In other words, snapshots take advantage of the fact that a backup is identical to the data everywhere except where the data is changed, and so the snapshot only needs to protect data where it is being changed.

Let us assume that a certain portion of the disk has been written to, and a snapshot is then taken. When a subsequent write request attempts to modify data that was written before the snapshot was taken, this new data must not be allowed to overwrite the old data.

There are two ways in which the old data is protected, and hence there are two kinds of snapshots. These two kinds of snapshots are called "Copy-On-Write" and "Redirect-on- Write", and are explained below.

COPY-ON-WRITE (COW) SNAPSHOTS

Some implementations of snapshots protect the old data by copying it to a separate snapshot space when there is a new write that needs to overwrite it. This is shown in the figure below.

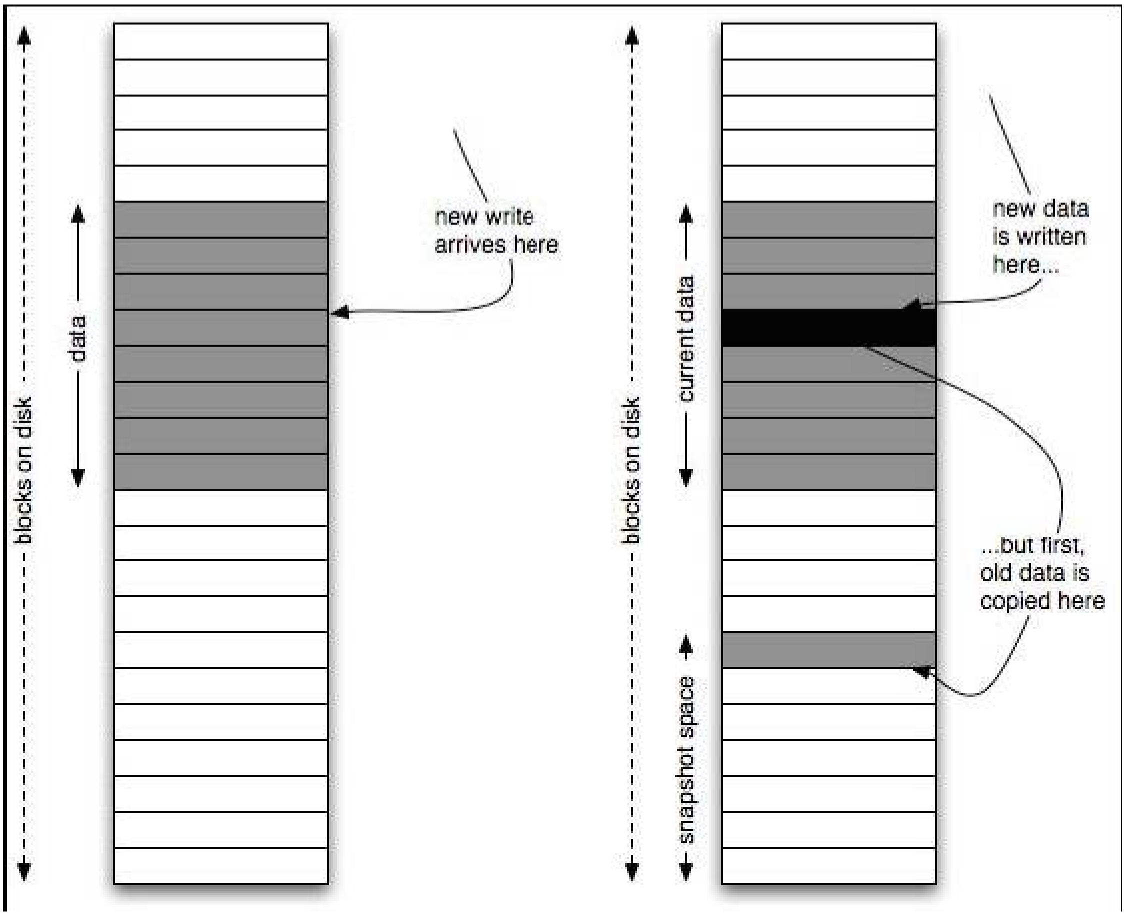

FIGURE 1: COPY-ON-WRITE SNAPSHOTS

After the old data has been copied to the new location, the data in the old location is overwritten with the new data. Until the next snapshot is taken, all subsequent writes that arrive for the same location overwrite the new data without touching the old data in the snapshot space.

In copy-on-write snapshots, therefore, every first write that tries to overwrite an old one, additionally spawns an additional read (of the old data) and a write (of the old data to the snapshot space).

The snapshot space may be organized in different ways depending on the implementation. Regardless of the implementation, however, it is necessary to maintain a table that keeps track of the logical addresses of data that have been copied to the snapshot space. Often, this table is just a hash (by logical address) of the location of old data versus the location of new data. This table needs to be consulted when a snapshot is mounted and read, and must preferably be stored in main memory to allow reads from the snapshot to be performed rapidly.

Since the quantity of old data that needs to be protected may be arbitrarily large, these tables may grow very large and unwieldy if they are maintained on a per-sector level.

Hence, most implementations choose not to track data on a per-sector level; rather, a larger granularity is chosen, such as 64kB. This means that data is always copied out in 64kB portions, regardless of the size of the new write request. AMI's implementation refers to this granularity as the chunk size. Other implementations variously refer to this as physical extent, snap granularity, copy size, etc.

REALLOCATE-ON-WRITE (ROW) SNAPSHOTS

Redirect-on-Write snapshots use a different method of protecting data that needs to be overwritten by new writes after a snapshot has been taken. They preserve the old data in its old location, and instead, redirect the new write to a new location. This is shown in the figure below.

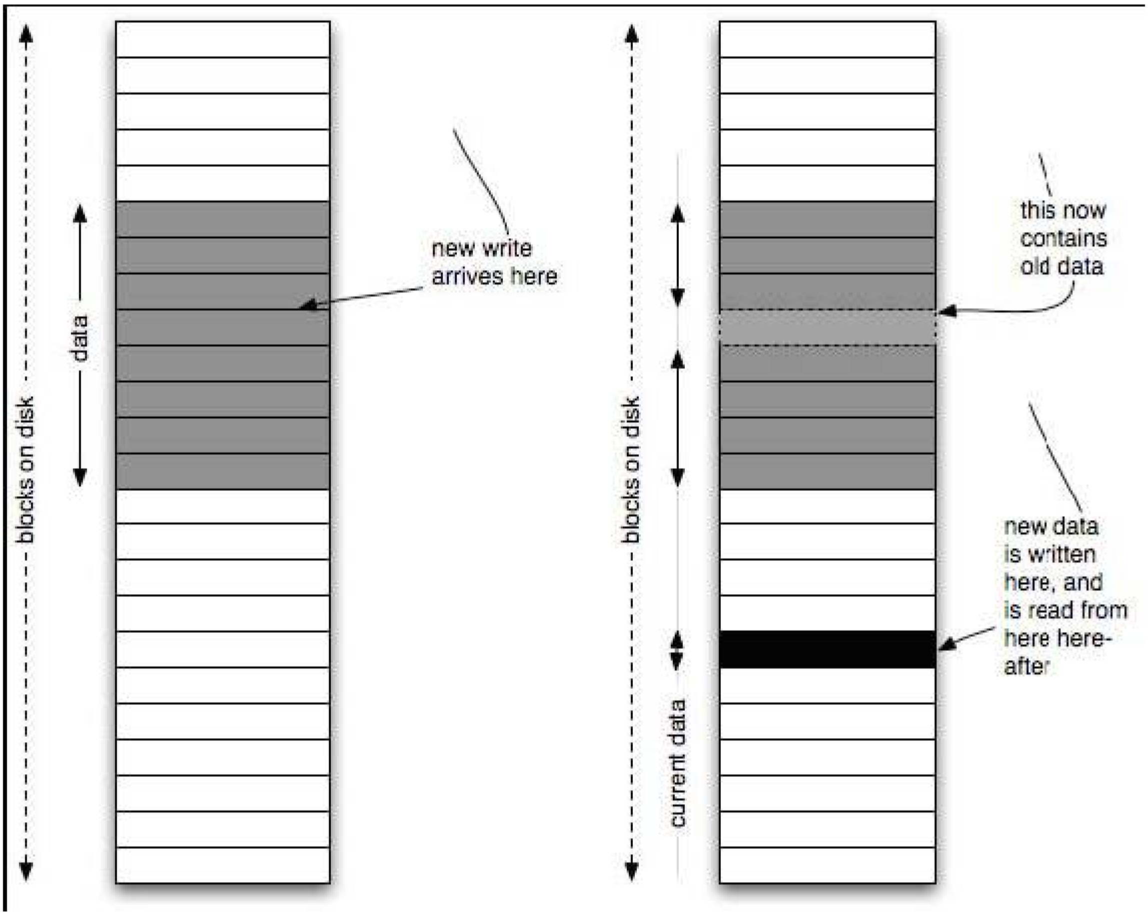

FIGURE 2: REDIRECT-ON-WRITE SNAPSHOTS

The physical address of the data has now changed for the current data. All subsequent reads and writes of data for the volume are performed at the new location. Snapshot reads continue to be performed from the old location.

As with Copy-on-Write snapshots, systems that implement Redirect-On-Write snapshots also need to maintain tables that contain a map of where old data is stored and where new data is stored. This mapping needs to be updated and referred to for volume reads in Redirect-on-Write, as opposed to merely for snapshot reads in Copy-on-Write. It is, therefore, very important to maintain these tables in main memory for Redirect-on-Write snapshot systems. In order to keep the amount of memory manageably small, the chunk size for these systems is usually more than one sector, for example 64kB.

Because write requests arriving to the disk may be smaller than the chunk size, and since a chunk's worth of data always needs to be valid when it is written to the new location, Redirect-on-Write snapshots sometimes require a read-modify-write cycle to keep data valid at the chunk granularity. If the chunk size is 64kB, and an 8kB new write arrives, then 64kB of old data is read, the 8kB change is applied, and the updated 64kB is written to the new location. Hence, the first write after a redirect-on-write snapshot may sometimes spawn an additional read.

A COMPARISON OF COW AND ROW SNAPSHOTS

While Copy-on-Write snapshots are most common in the market currently, Redirect-on-Write snapshots are slowly becoming more and more popular. The biggest benefit of redirect-on-write snapshots is, of course, greatly improved performance. The additional I/O load for first-time rewrites in ROW snapshots is, at most, one extra read. When a full chunk is written, this read is also avoided.

For COW snapshots, however, every new write necessarily requires an extra read and an extra write. Since the read takes place from the same location that the new write is to be written to, there is an additional translational latency associated with the I/O. This imposes a considerable performance penalty.

On the other hand, COW snapshots provide one degree of flexibility that ROW snapshots do not. Since the area that snapshots are stored in is clearly demarcated for COW snapshots, the administrator has the option of storing snapshot information, which may be considered 'stale', and therefore 'less important' in some kinds of storage systems, on lower cost storage. For example, it may be possible to store current data on a RAID-1 disk, and snapshot data on RAID-5 disks. Or the administrator may choose to store current data on fast SAS disks and snapshot data on slower (but cheaper) SATA disks.

This flexibility is not available with ROW snapshots, because the volume is scattered between the original location and the snapshot space.

The following table compares a few of the other relative benefits of ROW and COW snapshots.

Copy-on-Write snapshots	Redirect-on-Write snapshots
Flexibility to store volume and data separately	Better handling of depleted snapshot space - volume is not affected
Predictable amount of space recovered by deleting snapshots	Superior performance, constant regardless of the number of active snapshots
Superior space utilization, since uninitialized locations can be tracked and not redirected	Deletions and rollbacks are nearly instantaneous

Choosing between COW and ROW snapshots may be difficult. In complicated Information Systems, the best option available to an administrator may be to designate certain volumes as having COW snapshots, and certain others as having ROW snapshots.

AMI's implementation provides full support for both kinds of snapshots, thus providing that flexibility to the administrator. It is possible, therefore, to choose the right implementation for the right application, taking advantage of the benefits of both COW and ROW technologies.

In fact, AMI's implementation uses innovative methods of maintaining a common metadata structure for both COW and ROW snapshots, so that both may coexist seamlessly. In later versions of the implementation, it may even be possible to migrate from ROW to COW snapshots and vice versa.

In addition to supporting both kinds of snapshots, AMI's implementation also supports a potentially infinite number of snapshots. Current hardware limits the number to 254 snapshots, but this number scales with hardware resources available, such as processing power and memory.

2. OPERATIONS THAT CAN BE PERFORMED ON SNAPSHOTS

CREATION OF SNAPSHOTS

Snapshots may be created in one of two ways. An administrator may decide to create a snapshot schedule, which automatically takes snapshots at predefined intervals of time.

When there are a large number of volumes that need to be managed, this is often the only practicable option. Most management software packages available today offer tremendous flexibility in creating schedules.

In AMI's implementation, the time taken to create a snapshot, either ROW or COW, is less than 100 milliseconds. This provides a tremendous flexibility to the user, because the administrator is now free to create schedules with extremely fine granularities. For example, an administrator may create a snapshot every second, providing what is amounts to almost continuous data protection. In addition, because this time is so small, AMI's implementation of snapshots is fully compatible with initiator-side methodologies like Microsoft Virtual Shadow Service (VSS), which allow snapshots to be fully application-aware.

MOUNTING SNAPSHOT VOLUMES

In order to read from a snapshot volume, it is necessary to mount a snapshot as a readonly volume. For example, if a user has accidentally deleted a file, it may be required to mount a previous snapshot and recover the file. Sometimes, it may be necessary to mount multiple snapshots simultaneously, compare the contents of a particular file, and choose one of them to copy back to the original volume.

AMI's implementation allows all snapshots, as well as the data volume, to be mounted and accessed simultaneously. There is minimal performance degradation despite this. AMI's implementation aims to give administrators maximum flexibility when recovering data from snapshots.

SNAPSHOT DELETION

Administrators expect to routinely delete old snapshots to recover storage capacity. This is often part of a snapshot schedule, and at steady state, the oldest snapshot is deleted prior to the creation of a new snapshot.

Sometimes, though, administrators may choose a different strategy for snapshot deletion. For example, if a volume has not changed substantially between two snapshots, but has changed dramatically between two others, the administrator may prefer to delete one of the first pair rather than one of the second pair.

AMI's snapshot implementation allows administrators to delete any snapshot at random. Snapshot deletion is fully schedulable, and takes less than 100 milliseconds, providing the administrator with maximum flexibility in scheduling deletions. Additionally, freeing of space due to snapshot deletion is immediate, and so additional space becomes available at once when a snapshot is deleted.

SNAPSHOT ROLLBACK

When data has become irretrievably corrupt (for example, due to a virus), or has been destroyed entirely (for example, if a disk is accidentally formatted), the administrator has no choice but to perform a rollback from a snapshot onto the volume.

The most important criterion for rollback is that it must take as little time as possible. In highly available servers, the amount of time that a rollback takes is equal to the amount of time that service is interrupted.

In order to make the transition from a server that is down to a server that is up again as smooth as possible, AMI provides maximum flexibility for rollbacks. It is possible to roll back from any arbitrary snapshot, regardless of how far back in time it is, without having to roll back intermediate snapshots successively. Additionally, the time taken for a rollback is typically less than 100 milliseconds, and often substantially less. If the rollback has occurred from a ROW snapshot, the entire rollback is completed in that time, and there is no data migration. Since rollback from a COW snapshot involves data movement, the rollback is signaled as completed to the user, and the actual data migration takes place in the background. In either case, the volume is available for I/Os as soon as the rollback is signaled as completed, that is, in less than 100 milliseconds.

WRITABLE SNAPSHOTS

Some advanced snapshot implementations support allow snapshot volumes to be mounted as both read-only as well as read-write volumes. When a snapshot is mounted read-write, it is called a writable snapshot.

Administrators typically use writable snapshots to try experimental changes on a running system. For example, an administrator may wish to find out how much performance is improved if a defragmentation is done upon a particular volume, without actually performing the operation unless it is sufficiently beneficial. Or an administrator may want to find out if the application of an OS service pack causes any existing services to stop working. All of these experiments must, of course, be performed without interruption of service.

For applications such as these, writable snapshots would seem to be the best solution. The administrator would typically create a snapshot, on which to try out the experiment in mind, and continue to make the volume available for use. The snapshot would be mounted writable, and the experiment would be performed as needed. After the experiment has been completed, the administrator may decide to delete the writable snapshot, or the administrator may decide to discard the volume and convert the writable snapshot into the volume.

AMI provides the industry's most fully featured writable snapshot implementation. In AMI's implementation, any snapshot can have an associated writable snapshot, which can be mounted read-write. The writable snapshot may be created, deleted and rolled back with exactly the same feature set as read-only snapshots. Performance of writable snapshots is as good as the performance of read-only snapshots, and minimally degrades the performance of the main volume.

SNAPSHOT-ASSISTED FEATURES

Snapshots are an interesting and innovative extension to storage systems, and as is the case with such technologies, it enables various new features and simplifies several old ones. AMI's storage offering leverages on its snapshot implementation to provide the following advanced features.

3. REPLICATION

Synchronous replication and asynchronous replication are features that are becoming more and more important in storage systems that protect valuable data. One of the most important criteria that govern replication algorithms is the consistency of data at any replication point. Since applications that use a volume may write data in a certain order, failure to adhere to that order may render data unusable if the replication process is performed at an inconsistent point. This issue is known as write-order fidelity, and is a major consideration in replication systems.

When snapshots exist, write-order fidelity is easy to achieve. When a snapshot is taken, we are assured that all the data that has arrived before the snapshot was taken is part of the snapshot, and all the data that has arrived subsequently, is not. Additionally, if an agent such as Microsoft VSS is used on the client side to initiate the snapshot, the application using the volume is also aware of the snapshot, and may quiesce its data in a consistent state for the snapshot.

This snapshot can now be replicated to a remote location while maintaining write-order fidelity and application-level consistency. This replication does not interrupt or slow down I/O on the volume, because it reads from a snapshot volume and does not touch the data volume.

AMI supports various means of synchronous and asynchronous replication, and provides full support for snapshot-assisted replication. The fact that an administrator has a very fine control of snapshot creation and deletion means that it is easy for him or her to integrate snapshots and replication in a consistent and convenient manner.

4. INFORMATION LIFECYCLE MANAGEMENT

Information Lifecycle Management (ILM) is an increasingly important function of storage systems. Since information systems are usually used continuously over several years, it becomes necessary to differentiate old data from new, and give priority to new data over old data. An administrator may want to move old data to cheaper, lower performance media, and move frequently used new data to faster media.

Snapshots have the natural advantage of having time information inbuilt into them, thus making them an ideal choice around which to build an ILM implementation. AMI's implementation of ILM takes full advantage of its full-featured snapshots to timestamp data and migrate it accordingly.

5. PERFORMANCE AND UTILIZATION STATISTICS OF AMI SNAPSHOTS

(statistics pending beta testing and QA)

In AMI's implementation, further potential for acceleration of performance is provided by allowing the administrator partial flexibility to select a chunk size. It is well known that different applications provide different kinds of I/O loads. For instance, database applications typically provide random 8kB read and write loads to a server, whereas file servers often receive 64kB sequential read and write requests due to OS-side caching. To accommodate all these applications on the same platform, it is possible to tweak the chunk size to be either 8kB or 64kB. The benefit of synchronizing the chunk size and the most frequently received I/O request size is that if ROW snapshots are being used, a large number of additional I/Os (in the form of read-modify-write cycles) may be avoided. In the best case, it is even possible to perform penalty-less ROW snapshots. It is, of course, not possible for any other implementation to better such phenomenal snapshot performance.

6. CONCLUSION

In conclusion, snapshots appear to be the way forward to ensure data consistency and protection in modern storage systems. There are two methods of performing snapshots, viz. copy-on-write and redirect-on-write, each of which has its advantages and disadvantages. An administrator expects maximum flexibility in creating, deleting and rolling back snapshots, and expects that each operation take a minimum amount of time.

Some advanced snapshot implementations allow the administrator to mount snapshots as both read-only and writable volumes, for the purpose of experimentation. In addition to all the above features, an administrator expects stability and performance from any snapshot implementation.

AMI's snapshot implementation is arguably one of the best snapshot implementations in the industry, providing all of the above features, at a minimum performance premium, and with minimal wastage of space. In addition, AMI's snapshot implementation is lightweight and flexible, making it easy to add new features and fine-tune existing ones.

Finally, AMI uses its snapshot implementation to enable various other attractive features, primary among them being snapshot-assisted replication and information lifecycle management.

American Megatrends, Inc.
6145-F Northbelt Parkway
Norcross, GA 30071

American Megatrends International GmbH
D 81825 München , Wardeinstrasse 3 a
Deutschland

TRADEMARK AND COPYRIGHT ACKNOWLEDGMENTS

FOR ADDITIONAL INFORMATION

Call American Megatrends at 1-800-246-8600 for additional information. You can also visit us online at ami.com.

Call American Megatrends International GmbH at +49 89 96 999 510 for additional information. You can also visit us online at ami.de.

LIMITATIONS OF LIABILITY

In no event shall American Megatrends be held liable for any loss, expenses, or damages of any kind whatsoever, whether direct, indirect, incidental, or consequential, arising from the design or use of this product or the support materials provided with the product.

LIMITED WARRANTY

No warranties are made, either express or implied, with regard to the contents of this work, its merchantability, or fitness for a particular use. American Megatrends assumes no responsibility for errors and omissions or for the uses made of the material contained herein or reader decisions based on such use.

REVISION HISTORY

15.03.2007 Preliminary release

15.02.2008 Revised version

More information: europe@ami.com - www.ami.de - www.ami.com