IMPLEMENTATION CONSIDERATIONS ON VIRTUALISATION

In the following we want to draw up general requirements and considerations for the implementation of the virtualisation entity and illustrate how the difficulties described in Section 5.2 can be solved with the aid of storage virtualisation. For example, storage virtualisation also facilitates the integration of higher storage functions that previously had to be realised by means of other software products.

6.4.1 Realisation of the virtualisation entity

First of all, it is important that a storage virtualisation entity can be administered from a central console regardless of whether it is implemented as hardware or software and where it is positioned in the storage network. It is desirable for all tools that are required for the administration of the storage device to run via this console. All operations performed by the virtualisation entity should take place in a rule-based manner and orientate themselves to the applicable data profiles. Policy-based operation allows the storage administrator to configure and control the operations of the virtualisationentity. Profile-orientation makes it possible for the data to be automated according to its specific properties and requirements.

Because virtualisation always intervenes in the data stream, correct implementation is indispensable if data corruption is to be avoided. The virtualisation entity itself should therefore also be backed up so that access to the virtualised storage resources is still possible in the event of a failure. The concepts for server-clustering introduced in Section
6.3.2 are suitable here.

In order to achieve the greatest possible degree of compatibility to servers and applications and also to win acceptance amongst users it is necessary for a virtualization entity to remain hidden from its users. Servers, applications and users must always have the impression that they are working with physical storage media and must not notice the existence of a virtualisation entity. Furthermore, for reasons of compatibility, access on both file and block level, including the required protocols, must be supported (Section 5.5).

In addition to virtualised storage, the classical non-virtualised storage access options should continue to exist. This facilitates first of all an incremental introduction of the virtualisation technique into the storage network and second, allows applications, servers and storage devices that are incompatible with virtualisation to continue to be operated in the same storage network. For example, in our practical work during the testing of virtualisation software we found that the connection of a Windows server functioned perfectly, whilst the connection of a Solaris server failed. This was because of minor deviations from the Fibre Channel standard in the realisation of the Fibre Channel protocol in the virtualisation software used.

6.4.2 Replacement of storage devices

When using storage virtualisation the replacement of storage devices is relatively easy to perform, since the servers no longer access the physical devices directly, instead only working with virtual storage media. The replacement of a storage device in this case involves the following steps:

1. Connection of the new storage device to the storage network.
2. Configuration and connection of the new storage device to the virtualisation entity.

3. Migration of the data from the old to the new device by the virtualisation entity whilst the applications are running.
4. Removal of the old storage device from the configuration of the virtualisation entity.
5. Removal of the old storage device from the storage network.
The process requires no configuration changes to the applications. These continue to work on their virtual hard disks throughout the entire process.

6.4.3 Efficient use of resources by dynamic storage allocation

Certain mechanisms, such as the insertion of a volume manager within the virtualization entity, permit the implementation of various approaches for the efficient use of resources.

First, all storage resources can be shared. Furthermore, the virtualisation entity can react dynamically to the capacity requirements of virtual storage by making more physical capacity available to a growing data set on virtual storage and, in the converse case, freeing up the storage once again if the data set shrinks. Such concepts can be more easily developed on the file level than on the block level since on the file level a file system holds the information on unoccupied blocks, whereas on the block level this information is lacking. Even if such concepts have not previously been realised, they can be realised with storage virtualisation.

In this manner it is possible to practically imitate a significantly larger storage, of which only part is actually physically present. By the dynamic allocation of the physical storage,

additional physical storage can be assigned to the virtual storage when needed. Finally by dynamic, data-oriented storage allocation it is possible to achieve a more efficient utilisation of resources.

6.4.4 Efficient use of resources by data migration

If a virtualisation entity is oriented towards the profiles of the data that it administers, it can determine which data is required and how often. In this manner it is possible to control the distribution of the data on fast and slow storage devices in order to achieve a high data throughput for frequently required data. Such data migration is also useful if it is based upon the data type. In the case of video data, for example, it can be worthwhile to store only the start of the file on fast storage in order to provide users with a short insight into the video file. If the user then accesses further parts of the video file that are not on the fast storage, this must first be played back from the slower to the fast storage.

6.4.5 Performance increase

Performance can be increased in several ways with the aid of storage virtualisation. First of all, caching within the virtualisation entity always presents a good opportunity for reducing the number of slow physical accesses (Section 2.6). Techniques such as striping or mirroring within the virtualisation entity for distributing the data over several resources can also be used to increase performance (Section 2.5). Further options for increasing performance are presented by the distribution of the I/O load amongst several virtualisation entities working together and amongst several datapaths between server and virtual storage or virtual storage and physical storage devices (Section 5.1).

6.4.6 Availability due to the introduction of redundancy

The virtualisation entity can ensure the redundancy of the data by itself since it has complete control over the resources. The appropriate RAID techniques are suitable here (Section 2.5). For example, in the event of the failure of a storage device, operation can nevertheless be continued. The virtualisation entity can then immediately start to mirror the data once again in order to restore the redundancy of the data. As a result, a device failure is completely hidden from the servers – apart from possible temporary reductions

in performance. It is even more important that information about a device failure is reported to a central console so that the device is replaced immediately. The message arriving at the console can also be forwarded by e-mail or pager to the responsible person (Section 10.3). Multiple access paths, both between servers and virtual storage and also between virtual storage and physical storage devices can also contribute to the improvement of fault-tolerance in storage virtualisation (Section 6.3.1).

6.4.7 Backup and archiving

A virtualisation entity is also a suitable data protection tool. By the use of appropriate rules the administrator can, for example, define different backup intervals for different data. Since the virtualisation entity is responsible for the full administration of the physical storage it can perform the backup processes in question independently. All network backup methods (Chapter 7) can be integrated into storage virtualisation.

6.4.8 Data sharing

Data sharing can be achieved if the virtualisation entity permits access to the virtual storage on file level. In this case, the virtualisation entity manages the file system centrally. By means of appropriate protocols, the servers can access the files in this file system in parallel. Currently this is permitted primarily by classical network file systems such as NFS and CIFS (Section 4.2) and fast shared disk file systems (Section 4.3). Some manufacturers are working on cross-platform shared disk file systems with the corresponding protocol mechanisms that permit the fast file sharing even in heterogeneous environments.

6.4.9 Privacy protection

The allocation of user rights and access configurations can also be integrated into a virtualisation entity, since it forms the interface between virtual and physical storage and thus prevents direct access to the storage by the user. In this manner, the access rights of the data can be managed from a central point.

Source : http://elearningatria.files.wordpress.com/2013/10/cse-viii-storage-area-networks-06cs833-notes.pdf