WHITE PAPER

USS 3.0

USDesign

Introduction

USS is platform independent software that provides access to storage devices through use of Internet and host-based access mechanisms. USS provides a consistent management and access methodology across all host platforms. USS is implemented using standards based software development tools such as JAVA™, HTML, and RPC. It provides management interfaces through standard HTML and RPC interfaces. Its file systems are based on industry standard based specifications, which provides for media and data investment protection. USS is extensible, allowing for easy addition for readability of additional file system formats, additional host systems, and additional storage devices.

Goals

Many storage management products currently on the market do a good job of addressing a specific need. But as the computer industry evolves rapidly each year, more flexible storage management software becomes highly desirable. USS intends to become the desired storage management software product by addressing the following needs.

- Platform independent

  The storage industry, because of the continued growth and change over the next five to ten years, will present the customer with unique problems. USS addresses these problems with industry unique solutions.

  Today the computer industry is comprised mostly with storage control software that is written at the driver level, thereby tying the software to a specific host platform. For example, if a driver is written for the Windows NT operating system, then it will only operate under Windows NT. Another problem with driver based approaches is that if an operating system is upgraded, then the driver software must also be upgraded.

  Objective Data Storage's goal is to produce a storage management software product that is not dependent on a particular host computer or operating system, allowing customers to use the storage software on a variety of computer platforms.

- Media interchange between different computer systems

  Storage management software typically uses host based file systems to provide data storage. Since these host based file systems are rarely standard, the customer is either needlessly tied to that host system or must endure expensive media conversion if they change systems. A standard based media interchange can be used to provide data accessibility from a variety of host systems. This gives potential customers flexibility in choosing the computer system that best meets their needs. It also frees the storage management software from being required to use only the host provided file system format. The result is storage management software that supports a "many to one" relationship regarding the underlying file systems instead of a "one to one" relationship.

  Objective Data Storage's goal is to produce storage management software that provides a standard media interchange format between different computer systems.

- Consistent management interface

  Typically, the interface for storage software is based on the host system's abilities. Some systems support more "friendly" interfaces than others support. Other software provides only graphical interfaces, which make automated management difficult. The result is a plethora of interfaces, each of which has its own quirks and "look-and-feel". As each new storage software product is deployed, training for that product must be provided. This leads to greater training costs and less "cross-training" benefits.

  Objective Data Storage's goal is to produce a storage management product that provides a consistent interface, no matter the host system. Addition-ally, Objective Data Storage recognizes that not all customers use software in the same way. As a result, Objective Data Storage's goal is to produce software that has many interfaces, yet provides consistency not only across platforms, but also across interfaces.

- Standardize external interfaces

  Storage management software, due to its dependence on an underlying host system, generally does not provide standard external interfaces. The interfaces that are provided are those that extend the host system's interface, which typically are not standard. Through the use of standard interfaces, customers can leverage existing tools to manage storage and to provide better "cross-training" of personnel.

  Objective Data Storage's goal is to produce storage management software that provides standard external interfaces.

- Extensible and Flexible

  The storage market is constantly changing. Storage management software struggles to keep up with new advances in media and drives. As a result, customers experience frustration and costly maintenance until software "catches up" to the current advances.

  Objective Data Storage's goal is to produce storage management software that is able to quickly adapt to changes in the storage market.

- Proactive device management

  Storage software does not usually perform proactive management of its devices. Devices tend to produce "out-of-bounds" events before errors are produced. These errors are typically diagnosed "after the fact". The result is costly down time. Storage management software must be able to identify "out-of-bounds" events and report those conditions to the responsible personnel through a variety of mechanisms.

  Objective Data Storage's goal is to produce storage management software that performs proactive device management and reports potentially serious conditions "before the fact".

- Customer hardware and data investment protection

  Optical media shelf life is rated between 10 and 100 years. To provide access to the data stored on these media, the customer must maintain obsolete and costly equipment or endure expensive media conversion. Further, data investment is lost if the customer cannot access that data in a timely fashion. Data recorded on new media may be needed on older systems, leaving the customer with bewildering options and interoperability issues.

  Objective Data Storage's goal is to produce storage management software that allows a customer to preserve their investment in new and existing hardware and media.

How does USS measure up?

- Platform independent

  USS is platform independent. It is written in the Java™ programming language. This language allows for one common code base that executes across a number of computer platforms. A common code base allows for more robust software, reduced maintenance costs, and greater investment protection.

- Media interchange between different computer systems

  USS allows media interchange between different computer systems by providing the ability to read several proprietary file systems from a number of media types. This allows customers using these media types and file systems to preserve a large part of their investment in hardware and media. Future investments are protected through use of the standard based file system format – Universal Disk Format (UDF). The UDF file system is a standard created by the Optical Storage Technology Association (OSTA). Many vendors provide implementations of UDF. OSTA ensures that each implementation is compliant.

  Another positive by-product of the design of USS is media can be physically interchanged with remote sites, even if the remote site uses a different host computer system.

  USS truly broadens the number of valid combinations between media type, on-media file format, and host computer system in a manner that is valuable and useful.

- Consistent management interface

  USS provides a consistent management interface across all platforms. USS management includes both HTML and RPC interfaces. Using graphical web browsers, USS may be managed from remote machines. The RPC based API allows programmatic management from local or remote machines. Each function available through the HTML interface is also available through the RPC interface.

- Standardize external interfaces

  The external interfaces used by USS are standards based. The management interfaces are based on HTML and RPC, both of which are standards that are platform independent. USS implements a file system based on the Universal Disk Format (UDF) specification standardized by the Optical Storage Technology Association (OSTA). USS currently supports versions 1.02 and 1.50 of the UDF specification.

- Allow for change

  The module construction of USS allows for easy modification, be it user interface modules, file system formats, or the addition of new functionality. USS is extensible.

- Proactive device maintenance

  USS allows for the identification of "out-of-bounds" events to be monitored. USS will alert the system manager to potential device problems should one of the events be detected. This functionality, combined with the management interface, provides a powerful tool in minimizing down time of the storage system.

- Protect customer hardware and data investment

  USS allows customers to use many existing hardware storage solutions and media types, which preserves much of their existing investment. The combined features of USS allow for the customer to change one or more components of their storage solution while continuing to use remaining components. Newer storage technologies can be incrementally added to their environment in the way customers find most beneficial, without having to make wholesale computer system changes. In addition, USS supports multiple legacy file systems through use of a single software package, thereby eliminating costly media conversion processes.

Implementation

To implement the goals of storage management software, USS defines four modules. These modules work together to provide both data access and management interfaces, while maintaining extensibility. These four modules are:

- File System Module
- Network Interface Module
- Medium Changer Module
- Web Maintenance Module

The modules are service-based in that they offer services to each other and to the customer. This service-based approach lends itself to extensibility and consistency, not only within USS, but also towards the customer. Figure 1 shows the basic interaction among these modules. The following sections present an overview of each module.

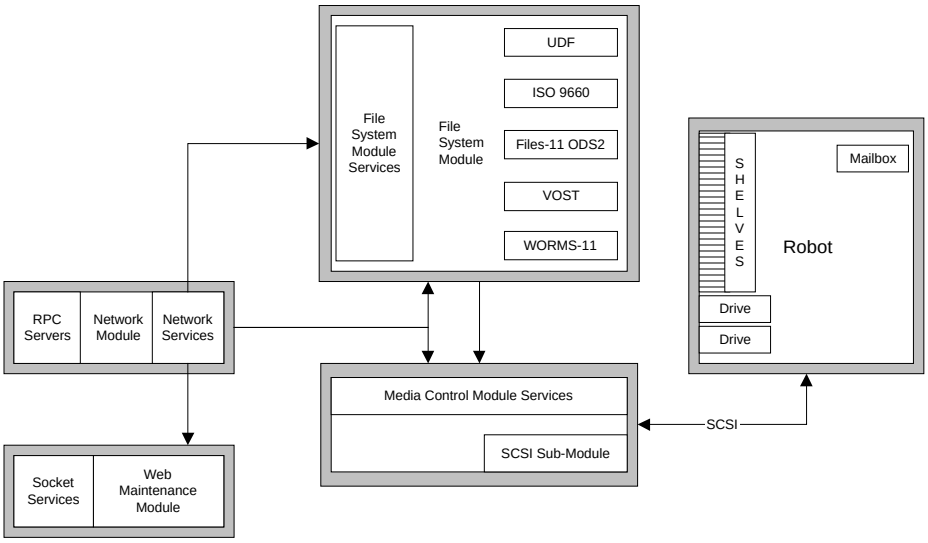

Figure 1 – USS Architecture

File System Module

The File System Module (FSM) provides mechanisms to implement file management of the storage data sets managed by USS. The FSM presents a common and transparent layer to all the file system formats that USS recognizes. This transparent layer is modeled after the NFS interface. The service presented by the FSM is state-less. It is the client's responsibility for keeping state information about file operations. For example, the client must maintain the list of files that are currently "open" for access. The state-less nature of the FSM provides better recovery in the event of USS failure.

The FSM is itself divided into sub-module file systems. Each sub-module provides accessibility to data recorded by different file systems. The FSM provides a "one-to-many" relationship with its sub-modules, resulting in one product that can provide accessibility to many different file system formats (see Figure 2).

Additional file systems are easily added to the FSM due to the sub-module architecture. Each sub-module is self-contained and is solely responsible for providing access to data stored on its associated media.

The FSM has been extended in USS version 3.0 to enable the caching of directories and file headers (metadata) for volumes managed by USS in an internal database. The metadata will consist of directory or file name, directory path, creation date, access date, modification date, size and any other information relative to the directory or file. This new capability, called "Meta-Cache," provides [metadata] read access of optical media in a jukebox library at the performance of hard disk.

Meta-Cache provides users the option of initializing the media surface and adding its metadata to the internal database; thereby improving read requests for directory hierarchy transversals on the media surface. This greatly improves the performance when accessing jukebox volumes via NT Explorer or API function calls for read requests to the metadata contained on the volume(s) selected for Meta-Cache journalling. Performance is best achieved for "write-active" optical media, or media that has read-intensive access. Meta-Cache will not cache the actual file data for the volume.

File systems supported in a write and read mode are UDF Version 1.02 for re-writeable media and UDF Version 1.5 (Sequential Mode) for write once media. File systems supported in read only mode are ISO 9660 and in the future the readability of Files-11 ODS-2, VOST and WORMS-11 file systems.

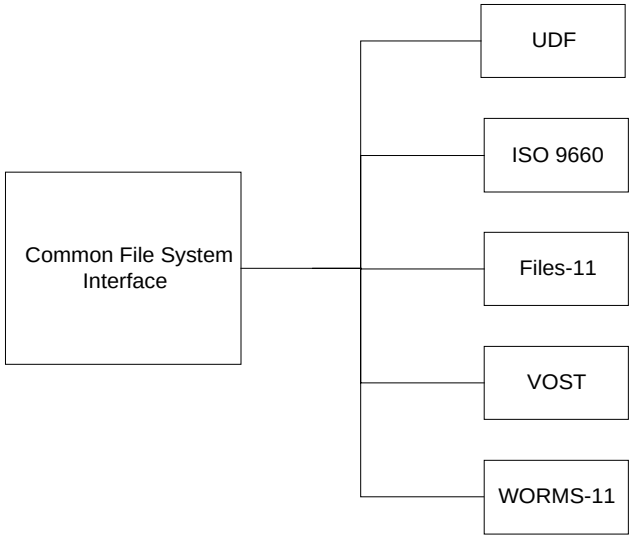

Figure 2 – File System Module

Network Interface Module

The Network Interface Module (NIM) provides network mechanisms for access to both the File System Module and the management of media and USS. The NIM provides protocols that enable access to the File System Module through NFS and its related protocols. In addition, the NIM provides network transport of USS management commands.

The NIM implements several standards based protocols that allow access to USS (see Figure 3). These standards are based on the TCP/UDP socket level interface and include the following:

- Remote Procedure Call (RPC) version 2
- External Data Representation (XDR)
- Network File System (NFS) version 2 and 3
- Mount version 1 and 3
- PC-NFS version 1 and 2

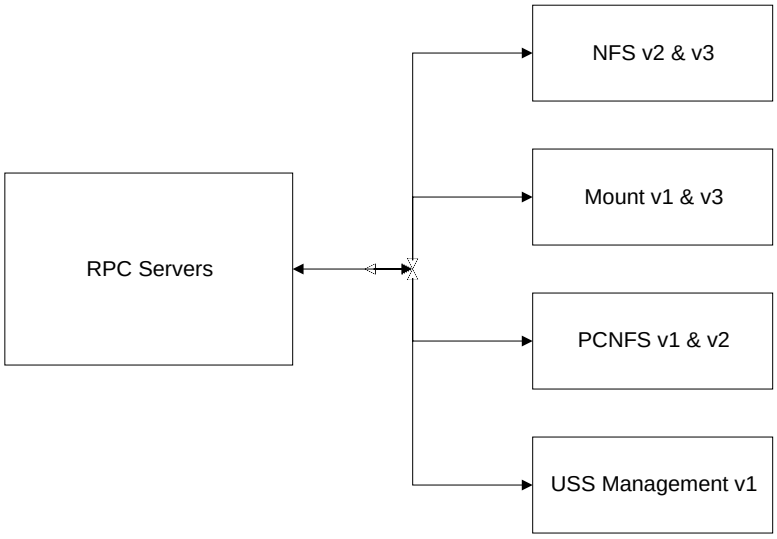

Figure 3 – Network Interface Module

Medium Changer Module

The Medium Changer Module (MCM) provides and enforces an association of a storage data set, such as a disk volume, with the addressable data space of physical media. This physical media may be optical disk or magnetic tape. This physical media may be removable from the hardware I/O device. The enforcement action of the MCM translates to physical media movement when the physical media is contained within a jukebox. In this environment, the MCM ensures that the physical media in any given hardware I/O device is associated with the storage data set being processed by the File System Module. The MCM physically moves media, as required, to the hardware I/O devices from media storage locations within the jukebox, and vice versa (see Figure 4).

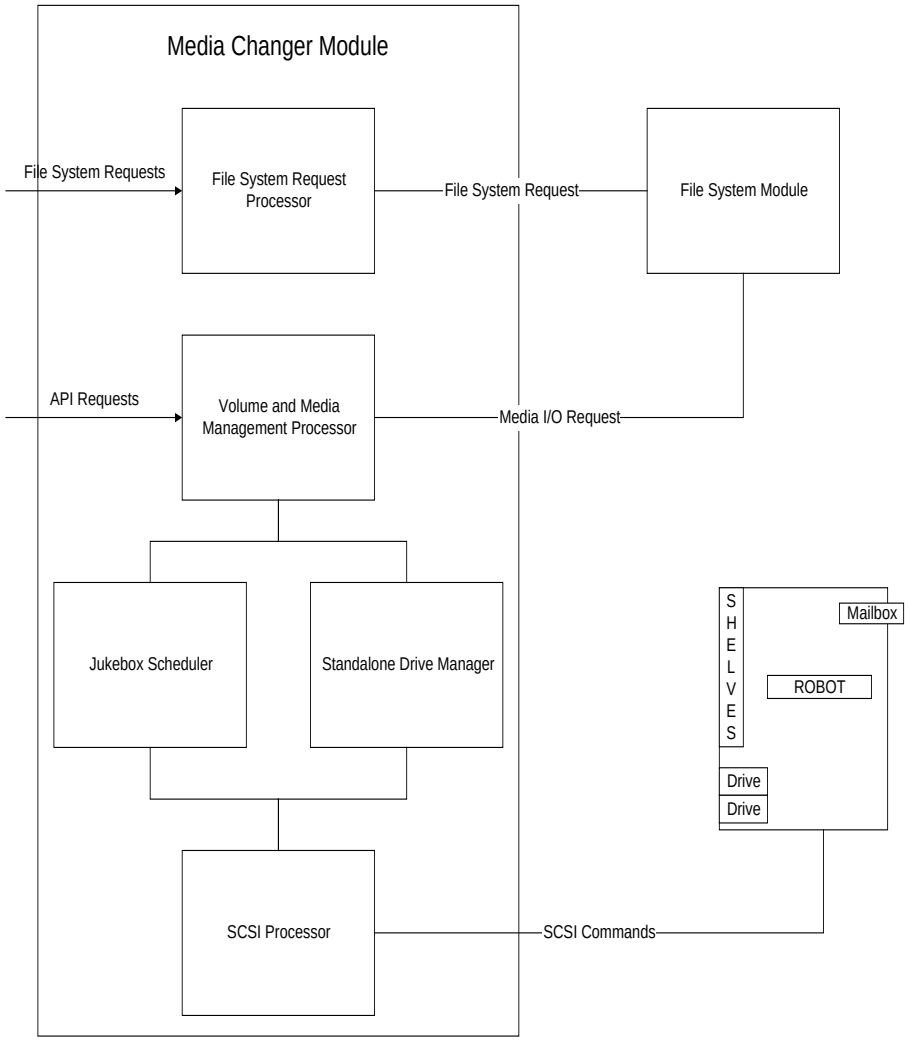

Figure 4 – Medium Changer Module

Web Maintenance Module

The Web Maintenance Module (WMM) provides HTML and RPC API access to USS management features and proactive device management. Because HTML and RPC are both standards, USS can be managed from a wide variety of client systems.

USS provides a built-in web server, saving the customer configuration and maintenance costs. The HTML interface is defined by the HTML 4.0 standard, providing the interoperability of a wide range of both text and graphical based web browsers. The customer is not forced to use a specific browser.

USS provides a built-in RPC server that handles requests for the management interface. The server is based on the standard RPC version 2 protocol. The specification for the management API is available for those customers who require programmatic control of USS. Because the HTML interface uses the RPC API to present information to the customer, functional consistency is maintained across both the HTML and RPC management interfaces (see Figure 5).

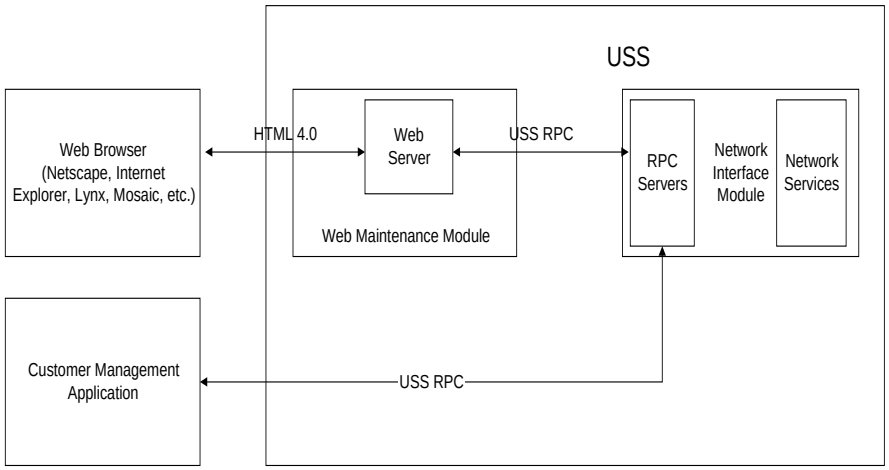

Figure 5 - Web Maintenance Module

The WMM also provides proactive device management. As events occur within the USS management domain, WMM will notify system managers based on action rules and event filters. USS ships with a default set of action rules and event filters, which generally are sufficient. However, interfaces are provided which allow the system manager to customize the proactive device management and notification mechanisms (see Figure 6).

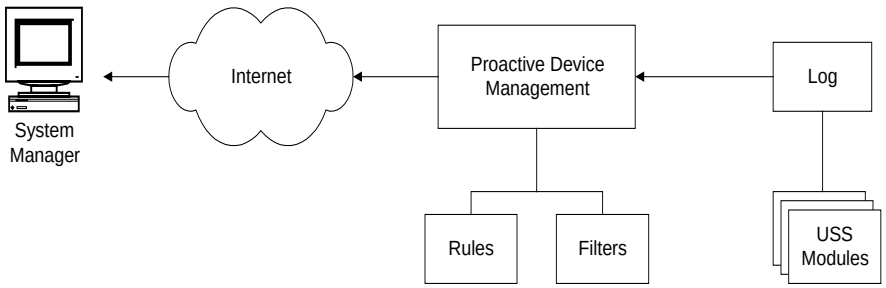

Figure 6 – Proactive Device Management

Figure 7 shows the design architecture and basic interaction among the USS modules.

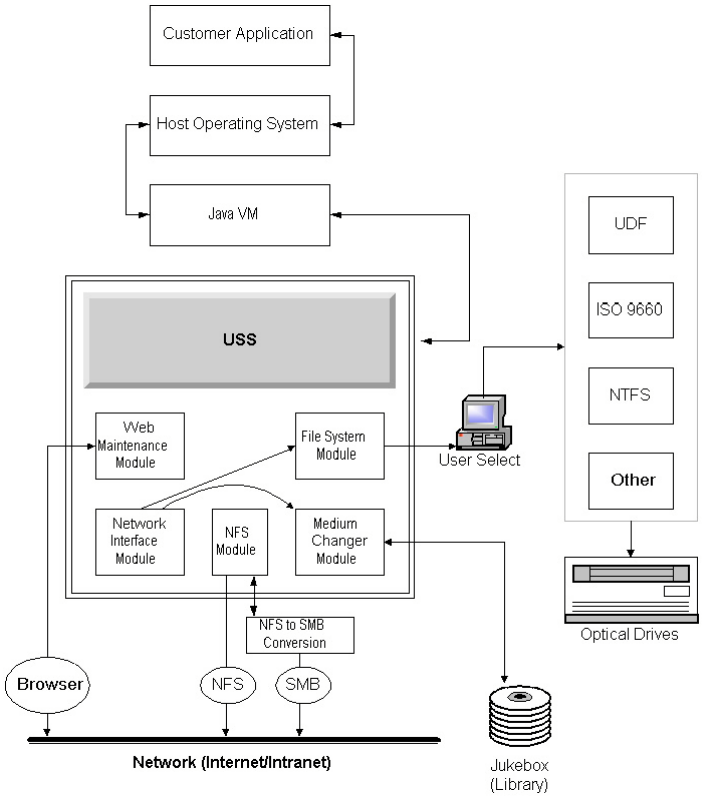

Figure 7 – USS Architecture