Land Below a DBMS

Kaladhar Voruganti Jai Menon Sandeep Gopisetty
IBM Almaden Research
San Jose, USA

1 Introduction

The focus of this paper is on the functionality of the storage infra-structure layer below a database manager system (DBMS). In most large enterprise environments, the mechanism of storing data on disks ,attached to a DBMS server, is being replaced by a complex networked storage infra-structure. This storage infra-structure industry is already a multi-billion dollar industry and it is quickly growing since more types of data are being digitized, and the volume of data being stored persistently is increasing. The goal of this paper is to give a snapshot of the key developments in this important research area because these developments have a direct impact on the overall performance, use, and management of a DBMS.

In this regard, this paper covers the following key topics:

- Section 2 of this paper presents a brief overview of the area below a DBMS. This background information is useful for understanding the contents of the later sections.
- Section 3 discusses key new developments in the file systems domain. It analyzes and compares the new emerging areas of storage area networks (SAN) and network file systems.
- Section 4 discusses the new developments in the area of block storage systems. Specifically, it discusses iSCSI, a new storage area network protocol, and compares this protocol with Fibre Channel protocol.
- A comparison between IDE and SCSI disks is presented in section 5. The cost, reliability and performance differences between these two types of disks is analyzed.
- Section 6 discusses some other non-disk forms of persistent storage media such as tapes, DVDs and non-volatile memories. It lists the competing technologies in each of these forms of persistent storage domains.
- Section 7 discusses the area of storage management. More specifically, it presents the latest developments in the area of storage management standards, storage virtualization and policy-based storage management.
- Finally, section 8 presents our conclusions and it lists some new interesting research areas.

2 Data Flow and Storage System Component Functionality

As shown in Figure 1, a DBMS can place its table data on top of raw block storage, or it can put its data into a file system. If a DBMS uses file system to manage its storage, then it can leverage the features of a file system. With respect to network storage systems, if a system uses raw storage, then it needs to utilize a block storage protocol to access remote networked storage. However, if a system is utilizing a file system interface then it can utilize either the block storage or networked file system protocols to access remote storage. For example, a DBMS can either be residing on top of a local file system that is using a block storage protocol to access remote storage, or it can reside on top of a network file system client that is interacting with a remote network file system server storage. Currently, a DBMS can use the following types of underlying storage infra-structures:

- **Direct Attached Storage (DAS):** In this model disks are typically attached to and managed by the DBMS server [Toi99]. The server contains a SCSI adapter card, and the disks are connected to the server via parallel SCSI bus. The server adapter directly accesses the storage present on disks. Other storage interconnect technologies such as Fibre Channel, or ATA can also be used if the corresponding host bus adapter is present at the host and the disks support the necessary interconnect protocol. The key point to note is that a server cannot directly access the storage that is attached to another server. In this approach, the DBMS can access block storage, and it can also access file storage if the DBMS server also contains file system manager functionality.
- **Storage Area Network (SAN):** This storage model [Toi99] allows multiple hosts (can be DBMS servers, file servers or other types of

servers) to share a pool of disks and storage controllers. The SAN network is usually Fibre Channel based (FCP), or IP based (iSCSI). Hosts can directly access disks via a host bus adapter, or they can indirectly access disks from their host bus adapters via the storage controllers. The hosts access data using a block storage protocol (SCSI block protocol such as SAM-2). Storage Controllers are usually separate computer boxes that provide RAID functionality, copy-services functionality (like data mirroring, snapshots, flash-copy etc), and they also contain large read and non-volatile write-back caches. A single logical storage controller can physically consist of a cluster of machines.

- **Network Attached Storage (NAS):** In this storage model [Toi99], the DBMS server places its data in a file system. The database server accesses its data via an intermediate network attached storage (NAS) server using a network file system protocol such as NFSv4, CIFS or DAFs. The NAS server can, in turn, access data using either the DAS or the SAN storage model. A single logical NAS server can physically be composed of a cluster of computer systems. NAS servers provide large file level caches, file level data mirroring, and integration with hierarchical storage management systems which automatically move unused files from disks to tapes.

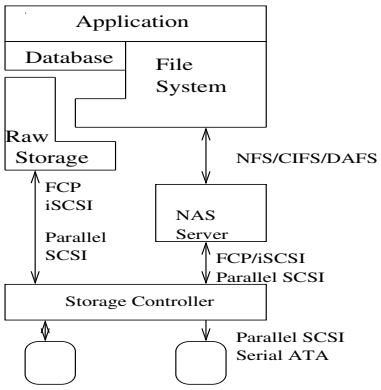

Figure 1: Network File System Infra-Structure

3 Network File Systems Analysis

DAS solutions are becoming less competitive than NAS and SAN solutions because, in DAS, storage devices are directly attached to (and owned by) a primary host server, and the server typically uses parallel SCSI transport protocol to access the storage devices. Since a parallel SCSI bus can support a maximum of 15 devices per channel, this limitation makes it necessary to add new servers in order to increase storage capacity. Furthermore, the free storage space available at a server cannot be shared by other servers. Centralized storage management becomes difficult in DAS environments because storage is partitioned and managed by multiple servers, whereas, NAS servers and SAN storage controllers allow centralized storage management as they allow multiple servers to share storage behind a single storage controller or a NAS server.

Until recently, NAS solutions have been considered less expensive than SAN solutions because NAS protocols run on commodity IP networks with relatively inexpensive network interface cards (NICs) and IP switches/routers, whereas, SAN solutions are comprised of expensive Fibre Channel networks. With the recent emergence of IP based SAN protocols, the cost of installing a SAN is decreasing. In the past NAS performance trailed SAN performance due to the following overheads:

- NAS protocols typically run on top of TCP/IP protocol. The use of software TCP/IP stack and commodity Ethernet NICs increases the CPU utilization at the hosts. The saturation of host CPUs throttles the maximum achievable throughput. Whereas, SANs utilize hardware protocol offload cards which help to reduce host CPU utilization. However, with the recent emergence of hardware TCP/IP protocol offload cards and fast host CPUs, this overhead becomes less of an issue for NAS.

- Older versions of the NAS protocols had inefficiencies such as the inability to aggregate multiple NAS commands into a single message, and the absence of flow control mechanisms. Newer NAS protocols have rectified most of these protocol problems.

- Since NAS systems provide file level data abstraction, client (host) file access requests lead to file meta data processing at the NAS server. Thus, there is a level of indirection (via the NAS server) between the host and the data residing in storage controller cache or disk drive. This level of indirection adds to the overall NAS performance overhead and this can also lead to bottlenecks due to queuing delays. In SANs, the hosts can access the desired data blocks by directly contacting the appropriate block storage controller, and thus, avoid potential bottlenecks, but in order to maintain single system image, shared-disk clustered systems utilize locking messages which degrade performance in workloads with high read/write data contention. In order to have a scalable NAS system with respect to storage, it is necessary for the NAS servers to be connected via a SAN in the back-end.

In order to leverage the strengths of both SAN and NAS systems, SAN file systems have been proposed [MPR+03]. SAN file systems allow applications to access data using the file system interface. However, underneath, a combination of network file system and block storage protocol are used to retrieve data from

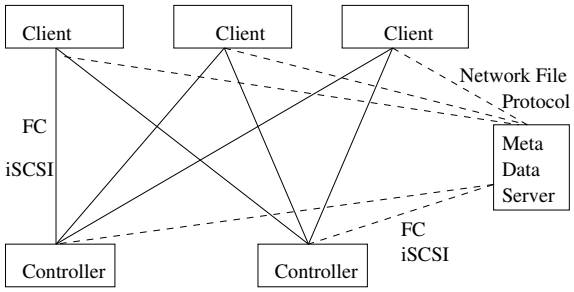

Figure 2: SAN File System

persistent store. As shown in Figure 2, hosts (contain file system clients) use a network file system protocol to contact a file system meta-data server. The meta-data server, in turn, informs the host about the location of the appropriate storage blocks. The host can then directly access the blocks from either the storage controllers or the disks. The host caches the relevant meta-data and, thus, minimizes the number of times it accesses the meta-data server. The key benefit of a SAN file system over conventional NAS systems is that the clients do not incur the latency present in a NAS environment due to an intermediate server (such as the NAS server). Moreover, the absence of an intermediate server from the data access path makes the SAN file system solution more scalable than a NAS solution with respect to the number of hosts that can be simultaneously supported. In conventional NAS file systems the NAS server performs authentication and ensures that only authorized users access the data. Therefore, in the SAN file system paradigm it is necessary for the storage controllers to provide the necessary security functionality. Currently efforts are under way to define a new object storage protocol [Aea03] that would be an alternative to the block storage access paradigm. The ability to provide security at a finer granularity than at the LUN level is being advocated as one of the key benefits of this new object storage protocol. However, it is too early to assess whether this approach will be successful.

4 SAN Protocol Analysis

Fibre Channel and Parallel SCSI are currently the most popular transport alternatives for carrying SCSI block commands. However, iSCSI is a new emerging transport protocol that has the potential to disrupt the SCSI transport protocol arena [VS01]. Parallel SCSI [Sch97] has distance (25 metres) and device connection limitations (15 devices per channel) that are not present in the other protocols. Fibre Channel [Ben96] partially overcame the distance limitation (it can support metropolitan area distances of a few kilometres) and it satisfactorily overcame the device connection limitation because it can operate in switched mode, and thus, support thousands of devices. However, the key drawback of FibreChannel is that it is not truly WAN (wide-area network) enabled. More importantly, an organization's investment in training their network administrators to acquire skills in IP based management tools such as SLP for discovery, IPSec for security, DiffServ/IntServ for QoS, SNMP for monitoring and configuration, and OSPF for routing, have not been transferred to the Fibre Channel domain. Also, Ethernet host adapters, and IP switches cannot be reused for Fibre-Channel networks. Thus, an organization's investment in regular networks cannot be carried over to Fibre Channel networks. Therefore, iSCSI protocol has been created in order to leverage the management advantages of IP networks, and the low cost advantages (due to high volume manufacturing) of Ethernet cards and IP routers. Storage devices supporting iSCSI protocol are just emerging and it will take a few years for iSCSI to compete with Fibre Channel.

iSCSI encapsulates SCSI commands and places them on TCP/IP. Since most TCP/IP networks run on Ethernet, until the arrival of Gigabit Ethernet, performance wise, iSCSI was not a feasible option. iSCSI protocol can be implemented completely in software (iSCSI and TCP/IP layers are executed by host CPU), or partially implemented in software (iSCSI layer is executed by host CPU, but the TCP/IP layer is offloaded to a network card), or completely offloaded to a network card (both iSCSI and TCP/IP layers execute on a host bus adapter). The offloading of the iSCSI stack along with TCP offload allows for the support of RDMA functionality in the HBA which is not available when one uses software iSCSI/TCP stack or software iSCSI layer in conjunction with TCP offload. Offloading TCP/IP stack alone does not allow for the support of RDMA because the TCP layer alone (needs iSCSI layer specific knowledge) cannot deduce the destination memory location of the incoming data packets. There have been some efforts in implementing RDMA without offloading the iSCSI or the TCP/IP layers. In these approaches, the incoming TCP segment header and data are scattered into separate locations in the kernel memory. Furthermore, the TCP data portion is remapped into the user space at the location desired by the user application. The drawbacks of this approach are that page remapping can be done only in 4K chunks, and therefore, if the size of the data being received is not a multiple of 4K then memory gets wasted. Furthermore, the user application buffers have to aligned at 4K multiples in order to utilize this page remapping technique and there is an overhead associated with the page remapping operation.

TCP/IP offload helps to minimize the number of host interrupts by the network card in comparison to the strictly software iSCSI stack because the network card can interrupt the host once for each TCP segment, whereas, when TCP layer is in software at the host, then a network card will interrupt the host once for each Ethernet frame. Current Gigabit Ethernet network cards have some form of interrupt coalescing but it is still not as effective as TCP segment level interrupt coalescing.

Moreover, since iSCSI is based on TCP/IP, it inherits some TCP properties which might be deemed

unsuitable in high speed LANs. TCP uses the slow start congestion control mechanism when the message sender discovers that packets have been lost in the network. Slow start drastically reduces the transmission window size, and it incrementally increases the size upon receiving successful acknowledgments from the receiver. Congestion is typically a transient phenomenon in over provisioned LANs and therefore, slow start congestion is not desirable in LANs. Secondly, TCP is a stream-based protocol. That is, each TCP segment, and thus, iSCSI header and data, could be spread across multiple Ethernet frames. Thus, if the Ethernet frame containing the iSCSI header information is dropped or it arrives out of sequence, then the TCP/IP offload cards need to buffer the trailing Ethernet frames until the frame containing the iSCSI header information is received. When the iSCSI protocol is operating at 10 Gbps then the network offload card would require a lot of memory in order to buffer the trailing Ethernet frames until the frame containing the iSCSI header arrives at the network card.

In order to leverage the benefits of IP networks, the advocates of Fibre Channel protocol have proposed FCIP and iFCP protocols [SVM+03]. The basic idea behind both these protocols is to put Fibre Channel on top of IP networks and thus, allow Fibre Channel legacy networks to overcome their limitations. In both iFCP and FCIP the end devices contain a fibre channel stack and islands of Fibre Channel devices communicate with other islands of Fibre Channel devices via iFCP or FCIP gateways. The key difference between iFCP and FCIP is the degree to which they utilize IP functionality. FCIP encapsulates Fibre Channel frames into TCP/IP segments, whereas iFCP maps Fibre Channel frames into TCP/IP segments. FCIP uses Fibre Channel fabric services (discovery and routing) whereas iFCP uses IP fabric services.

5 SCSI versus IDE protocols

SCSI and IDE/ATA [Sch97] are the two most popular protocols for accessing storage disks [ADR03, GH03]. Even though there are disks that support Fibre channel protocols, SCSI is the dominant disk access protocol in server environments and IDE/ATA protocol is the dominant disk access protocol in personal workstations and laptop computer environments. In the past IDE/ATA disks did not offer command queuing at the disks, multi-tasking between different user tasks and the lack of DMA option for placing the data directly without the host CPU involvement. Therefore, the absence of these features were viewed as key IDE disk drawbacks. However, currently these features are offered as options by IDE disk vendors and therefore, the differences between SCSI and IDE disks are diminishing. IDE disks currently support 2 devices per channel whereas, SCSI supports up to 15 devices per channel. IDE supports a maximum cable length of 18 inches whereas, SCSI typically supports maximum cable length of 12 metres. The ATA protocol allows for sector size of 512 bytes on the disk, whereas, SCSI allows for sector sizes greater than 512 bytes. Thus, in SCSI, one can choose to store the CRC for a 512 byte sector in the space following the data. Thus, after reading a sector of data, a controller can calculate the CRC of the data and check it against the stored CRC. Thus, support for larger than 512 byte block sizes allows SCSI disk based storage systems to have a higher degree of reliability than IDE disk based systems. The problem of off-track corrupt write operations is a problem for both SCSI and IDE disks. That is, a disk writes data at an incorrect location, and then it indicates that it has successfully written the data. Quorum-based data replication schemes help to overcome off-track write problems.

SCSI disks currently also have better performance characteristics than IDE disks. SCSI disks have higher rotational speeds and bus transfer rates than IDE disks. SCSI disks also typically have better disk servo control electronics to provide faster seek times. The higher rotational speeds in SCSI disks also lead to a reduction in the number of disk platters. Finally, the quality control process is much more stringent for SCSI disks (more elaborate testing, and use of higher quality electronic parts) than for IDE disks resulting in a mean time between failures of 1 million hours for SCSI disks in comparison to 500,000 hours or less for IDE disks. Thus, in reality, most of the performance and reliability benefits found in SCSI disks are protocol independent and they are not found in IDE disks simply because the disk vendors are positioning the IDE disks for the cost conscious personal workstation market.

Another new trend in the disk protocol realm is the emergence of serial ATA and serial SCSI protocols. The serial ATA protocol increases the bus length from 18 inches to 1 metre, and it also makes the ATA cable much thinner so that ATA can be used in blade server architectures. Serial ATA utilizes a point to point architecture and thus, the 2 devices to a bus ATA limitation is not present anymore. Serial SCSI has the same physical characteristics as serial ATA [ser02], and it thus extends the benefits such as thin wire and point to point connection to the SCSI domain.

6 Non-Disk Storage Media

Tapes, optical CDs/DVDs, WORM-tapes, and non-volatile memories are some of the other types of persistent storage media that are currently in use. Tapes have the lowest Gigabytes/dollar cost, and non-volatile memories have the highest Gigabytes/dollar cost. Tapes are the most preferred form of archival media. Tapes are also used as backup media, especially in situations where fast data restore times are not expected. DLT and LTO are the two competing tape standards and technically both have similar function and performance characteristics. It is too early to determine a clear winner between these two competing tape standards.

In the consumer market, optical CDs and DVDs are the most popular forms of WORM media for distributing both data and programs. Currently, DVD

forum and DVD alliance are the two competing DVD re-writing standards bodies and it is too early to pick a clear winner between the two standards. With respect to write-once DVD formats, DVD-R and DVD+R are the competing formats from DVD forum and DVD alliance respectively. At the enterprise level, WORM-tapes offer better storage densities than optical WORM media, but WORM-tapes cannot match the optical media random access times. Certain government regulations mandate that the user data be stored on WORM media to prevent data tampering. This type of data falls in a general class of user data that is called as *reference* data. The workload characteristics of reference data can be classified as write-once and read-seldom. Optical CDs/DVDs and WORM-tapes are ideal media for storing reference data. However, disks and non-WORM tapes are also being offered as WORM-like storage media for reference data. The content-addressable storage (CAS) interface allows disks/tapes to behave as a WORM device. In CAS, the input data object is hashed and the hash is also stored persistently in addition to the data. If the same data object is updated by the application program then the data object generates a new hash at the storage device, and it is thus stored as a new object.

Flash memory and battery-backed RAM are currently the two most prevalent types of non-volatile RAM technologies. Non-volatile memory is useful because it provides the notion of persistence to a write-back cache. Non-volatile write-back caches act as destaging buffers for write operations, and thus, they help to greatly improve the performance of write operations (like log writes to the disk). Flash memory cannot be used infinitely like DRAM (it has limited re-write capability) and it has slow access speeds in comparison to DRAM. One has to constantly check and replace the batteries in battery-backed non-volatile RAM solutions. Thus, there is a need for new non-volatile RAM solutions. New MRAM technology from Hewlett-Packard, Motorola and IBM is a promising non-volatile RAM technology [Tak02]. MRAM access speeds are faster than DRAM access speeds, and it can be written to an infinite number of times. Also its density (storage per area) is similar to DRAM. However, this technology is still in experimental stages, and it is yet to be seen whether it can be manufactured in large volumes. Ovonic NVRAM technology from Intel is another experimental technology. However, its access speeds are not as fast as MRAM, and it is more power hungry than MRAM. It is too early to determine a clear winner between MRAM and Ovonic memory technologies.

7 Storage Management

Storage management is currently a very active area of research in the storage infra-structure domain. The storage needs of organizations is increasing at a very fast pace, and also most organizations, for cost reasons, prefer to use storage hardware and software from many different vendors. This is leading to difficult storage management problems for organizations. End users want reliable, efficient and secure access to storage. In addition to ensuring that the storage needs of the end users are satisfied, storage administrators also want help in automatic monitoring, device configuration, resource provisioning (like automatically increasing the storage capacity for a user), resource usage prediction, resource discovery, performance (QoS) management, backup/restore, failure correlation, fault prediction, and capacity planning of the storage infra-structure. This section will discuss some of the key storage management building blocks that are required to implement the storage management functionality that will aid a system administrator. The storage management building blocks being discussed in this section are:

- Storage virtualization
- Storage model standards
- Policy-based storage management

7.1 Storage Virtualization

The goal of storage virtualization is to present block storage or file storage abstractions to the end users without requiring the users to worry about where the data resides (disks, tapes), how it is organized (logical and physical volumes, and RAID level) and how it is accessed (transport protocols used). Storage virtualization prevents the user/application from worrying about data reliability, data availability, and data access (performance) issues. Current virtualization functionality allows for the aggregation of storage from many devices and presenting it as contiguous storage to the higher level applications. However, in future, the objective is for the virtualization software to guarantee reliability and performance characteristics of the virtualized storage. For example, underneath the covers the virtualization layer replicates or backs up data to improve the reliability of the storage system. The virtualization layer can also expose QoS metrics to the users with respect to how quickly they can access their data. If slow user access times are acceptable to the end users, then the virtualization layer will store the data on slower physical media, and if the users want fast access times then the virtualization layer will cache the user's data in the server caches.

Virtualization functionality can reside either at request initiating hosts, or intermediate network switches or at target storage controller devices. Virtualization functionality can also reside at multiple levels like both at the hosts as well as at the network switches/storage controllers. The goal of each virtualization layer is to aggregate the storage underneath itself and prevent the layers above it from worrying about storage management issues. Thus, operating system, file system (logical volume managers) and storage management software vendors, network switch vendors and NAS server/storage controller vendors are all jockeying for a piece of this lucrative storage virtualization market.

Asymmetric and symmetric virtualization are two key virtualization techniques that are currently in use [vir00]. End user data, and meta-data required for virtualization purposes, are the two types of data that are accessed in a storage system. In the symmetric virtualization approach, the virtualization engine handles not only the virtualization meta-data, but it also handles the end user data. In the asymmetric approach, the virtualization engine only handles the virtualization related meta-data, and it does not handle the end user data. Thus, in the asymmetric approach, the clients first contact the virtualization engine and cache the necessary virtualization meta-data, and then they subsequently directly access the end user data. The disadvantage of the symmetric approach is that the virtualization engine can become a bottleneck and negatively affect the overall system performance. However, symmetric approach allows for better QoS and reliability support since the virtualization engine can monitor end-user data access traffic performance and dynamically adapt the storage system to satisfy the end-user needs.

SNIA storage standards organization is trying to standardize storage virtualization mechanisms so that a particular storage virtualization engine can interoperate with various software and hardware systems.

7.2 Storage Model Standards

Currently, it is difficult to build general purpose storage management software that can manage devices from multiple vendors because the vendors do not externalize the storage management APIs for their respective devices. That is, the vendors only provide the resource management APIs to their strategic partners. Thus, an effort has been underway at storage standards body (SNIA) to come up with a general storage model standard that models storage networking fabric (switches, hosts, HBAs, storage arrays), physical storage devices (disks, tapes), logical storage constructs (such as LUNs, Volumes, Extents etc), and storage operations (like mirroring, snapshots, zoning etc).

This storage model was previously known as DMTF Bluefin and it is currently known as SNIA SMI-S (Storage Management Initiative Services) [smi03]. Each storage hardware and software vendor will map their internal hardware and software resources into the canonical SMI-S model. The storage management software then simply deals with a single SMI-S API to manage the varying storage hardware and software. SMI-S uses the CIM modeling methodology to represent its storage model, and it converts the CIM model into XML and transports model data on top of HTTP/TCP combination. SMI-S chose CIM instead of SNMP as its data modeling methodology because of the better object-oriented modeling capabilities of CIM. SMI-S uses the client-server model where the storage management software or host servers typically act as CIM clients and the storage resource providers (storage arrays, switches etc) typically act as a CIM server. SMI-S also has the notion of agents and object managers (CIMOM). Agents typically reside at resource providers and they can be used to monitor storage resources. CIMOM can be thought of as an agent with richer functionality that can deal with multiple resources. Thus, a CIM client typically communicates with a CIMOM which, in turn, communicates with CIM service providers (servers) to satisfy the client requests. SMI-S uses the IETF SLP discovery protocol for discovering storage resources. The servers register their services at the SLP directory agent, and the client SLP agents either query the directory agent or directly query the SLP agents (server agents) present at the service provider. Currently, efforts are underway to combine the SMI-S transport mechanism (CIM/XML over HTTP) with the web-services (SOAP) communication approaches. SMI-S also contains a preliminary security model, that addresses authentication and encryption issues, an event handling model that handles asynchronous messages, and a locking/recovery model to co-ordinate resource access between competing clients. Finally, SMI-S standards group is also working on standards for SAN filesytems, NAS systems, storage virtualization boxes, and storage policy management infra-structure.

7.3 Policy-Based Storage Management

The storage needs of most organizations are increasing at a very fast rate. However, there is a limit on the amount of storage that can be managed by a single system administrator. Thus, an organization needs to hire more highly skilled and highly paid system administrators as its storage consumption increases. Therefore, the goal behind policy- based storage management is to increase the amount of storage that can be managed by a single system administrator. In policy-based storage management, the system administrator specifies high level goals with respect to the expected performance, reliability, security etc, and the storage management software maps these high level goals to lower level system management operations. This section briefly describes the necessary policy infra-structure concepts that allow for policy-based storage management. It is important to note that policy-based storage management is still in its infancy and that most current policy enabled systems provide only trivial policy-based storage management support.

A policy is specified as a 4-tuple construct. The four elements of the tuple are a) the "if" condition b) the "then" action clause c) the scope of the policy execution d) the priority for this policy. The policy scope determines whether a policy should even be considered for execution. The policy priority is used to resolve policy conflicts. Most of the policy definition standards (DMTF, IETF, Web Services etc) roughly follow the four tuple policy definition model. The policies specified by the system administrator are stored in the policy database. The policies are executed using the ECA (event-condition-action) model. That is, an event in the system triggers the checking of a par-

ticular policy condition, and based upon the result of the condition check, the policy action, specified in the policy 4-tuple definition, is taken by the storage management software. Currently, research effort is underway to automatically translate database table space management and backup-restore management policies into underlying storage management policies. Furthermore, converting high level policies into low level storage management actions is not a trivial task when dealing with a large number of policies, heterogeneous storage devices and a diverse range of possible storage management actions (e.g. replication, migration, buffer management, data prefetching size etc). Thus, research effort is underway to use case-base reasoning techniques to aid in this high level policy specification to low level storage actions mapping effort.

8 Conclusion

In conclusion, we would like to make the following observations with respect to the technologies presented in this paper:

- With the emergence of SAN-File systems, the areas of NAS and SANs can be viewed as complimentary in nature because a NAS protocol can be used for communication between SAN-File system clients and the SAN-File meta-data servers, and a SAN protocol can be used by the SAN-File system clients to directly access data from the storage controllers.

- With the emergence of IP SAN protocols such as iSCSI, an organization can leverage the same physical network infra-structure for its a) normal non-storage network traffic, b) for its NAS traffic and c) for its SAN traffic. Thus, the combination of Gigabit Ethernet/IP/TCP networks will eventually become the dominant storage transport protocol combination.

- Currently, many storage solution providers are devising storage solutions that use workstation class ATA disks to provide server class SCSI disk like solutions. The fundamental idea behind this approach is to leverage the lower cost of the ATA disks and use data replication technology to overcome the higher failure rates of the ATA disks.

- The SNIA SMI-S model is enabling storage management software vendors to provide solutions that span across devices from multiple vendors. SNIA SMI-S is helping storage management in the same manner SNMP/MIBs helped network management.

- Storage virtualization is quickly becoming a very active storage infra-structure area. Both network router companies and storage controller companies are building virtualization boxes which put them in direct competition with each other.

- In order for policy-based storage management to be successful, more research has to be done in the areas of policy conflict resolution, automatic conversion of business service level agreements into application, database and storage infra-structure policies, and automatic mapping of high level storage performance, availability and security policies into lower level device operations.

References

[ADR03] D. Anderson, J. Dykes, and E. Reidel. More Than an Interface: SCSI vs ATA. In *FAST*, 2003.

[Aea03] A. Azagury and et. al. Towards an Object Store. In *IEEE/NASA MSST*, 2003.

[Ben96] A. Benner. *Fibre Channel.* McGraw-Hill, 1996.

[GH03] E. Grochowski and R. Halem. Technological impact of magnetic hard disk drives on storage systems. In *IBM Systems Journal, Vol. 42, No. 2*, 2003.

[MPR+03] J. Menon, D. Pease, B. Rees, L. Duyanovich, and B. Hillsberg. IBM Storage Tank–A heterogeneous scalable SAN file system. In *IBM Systems Journal, Vol. 42. No. 2*, 2003.

[Sch97] F. Schmidt. *The SCSI Bus and IDE Interface.* Addison-Wesley, 1997.

[ser02] *Serial ATA II.* www.serialata.org, 2002.

[smi03] *Storage Management Initiative Specification.* www.snia.org, 2003.

[SVM+03] P. Sarkar, K. Voruganti, K. Meth, O. Biran, and J. Satran. Internet Protocol Storage Area Networks. In *IBM Systems Journal, Vol. 42, No. 2*, 2003.

[Tak02] D. Takahashi. The Search for Perfect Memory. In *Red Herring, Special Issue, No. 114*, 2002.

[Toi99] J. Toigo. *The Holy Grail of Data Storage Management.* Prentice-Hall, 1999.

[vir00] *Storage Networking Virtualization, What is it all about?* IBM Redbook, SG24-6210-00, 2000.

[VS01] K. Voruganti and P. Sarkar. An Analysis of three Gigabit Networking Protocols for Storage Area Networks. In *IPCCC*, 2001.