HEWLETT PACKARD

A Fibre Channel Based Architecture for Internet Multimedia Server Clusters

Shenze Chen, Manu Thapar
Broadband Information Systems Laboratory
HPL-96-125
August, 1996

Fibre Channel, Internet, World Wide Web, cluster, multimedia servers

Recently, the Internet has experienced an explosive growth by the World Wide Web technology. New users are increasing at an exponential rate, and many Web servers have been overloaded by client requests. System administrators are realizing the limitations of single processor systems to handle all of these requests. In this paper, we present a Fibre Channel (FC) based architecture for a cluster of Internet multimedia servers. A significant advantage of the FC based cluster is that it allows "direct storage attachment to the interconnect". Because of this feature, FC based clusters will change the fundamental data sharing model of existing and proposed clusters by eliminating remote data accesses. This in turn will improve the cluster cost and performance. It also achieves better load balancing across the storage devices in the cluster. Many of these aspects are critical to real-time multimedia applications, such as audio and video services. We address the cluster control mechanism, scalability, and fault tolerance issues of the FC based architecture.

Internal Accession Date Only

© Copyright Hewlett-Packard Company 1996

1. Introduction

Recently, the use of the Internet has been growing at a phenomenal rate. It is reported that in 1985, the Internet had only 50 sites and 1000 hosts, but now the numbers are well over 40,000 and 6,000,000, respectively (Figure 1), and they continue to grow at astonishing rates. The number of hosts increased by 81% in 1994 alone [1]. Each month, it is estimated that there are 150,000 new users joining the existing 20,000,000 Internet users [2]. By far, the WWW (World Wide Web) traffic has been observed as the largest and fastest growing Internet segment. Hundreds of gigabytes of data has been made available via Web servers by various commercial, educational, and government organizations [3]. Since multimedia can convey more information and is easier for people to understand, more and more multimedia data will be stored in Web servers and delivered across the Internet to users.

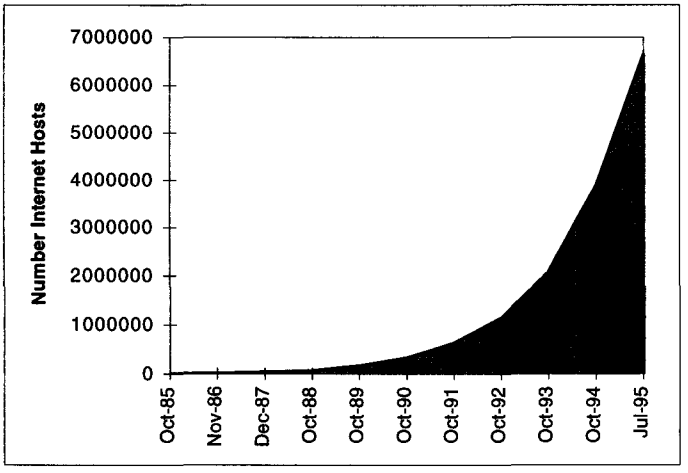

Figure 1: Internet Growth (Source: http://www.nw.com/zone/host-count-history).

To support the continued growth and provide better services to Internet users, during the past several years, considerable work has been done to develop protocols, tools, applications, algorithms, and systems. Today, there are various browsers available running on workstations, PC's, and notebooks that enable users to surf the Internet. Some venders even supply dedicated Internet browsing stations by using downsized PC's. These cheap appliances will enable more people to access the Internet. To support real-time multimedia applications across the Internet, network researchers have proposed to change the current Internet "best-effort" service model to a new service model that allows users to explicitly reserve the Internet resources so that their quality of service (QoS) can be guaranteed [4,5,6]. Given the exponential increase rate, it is predicted that the available IP addresses based on the current IPv4 protocol will soon be run out. Thus the next generation IP protocol IPv6 has been proposed to solve the problem, which will also provide real-time support, network security support, multicast support, and some other features [7].

At the other end of the service chain are the Internet Web servers. These Web servers need to manage hundreds gigabytes of data, which may easily grow to multiple terabytes given the increasing demand for the multimedia information from the servers. These multimedia Web serv-

ers differentiate themselves from traditional file servers by requiring real-time delivery of media objects, such as audio and video. This continuous media has to be retrieved from storage and sent to the network under time constraints in order to provide the QoS to the users. In a recent study, Kwan et al [3,8] reported that the NCSA (National Center for Supercomputing Applications) Web server experienced an explosive traffic growth of 11% per month from May 1994 to Jan. 1995. The number of requests received by the NCSA Web server amounts to about 690,000 per day, and they concluded no single processor system could serve all these requests. Therefore, for popular sites, a scalable solution is a necessity. Later in Section 2, we will discuss the NCSA's clustering solution, as well as some other existing and proposed cluster architectures.

In this paper, we present a scalable cluster architecture for Internet multimedia servers, which is based on the newly developed Fibre Channel technology. Fibre Channel (FC) is a new serial link defined by the ANSI X3T9 Technical Committee as an industry standard [9]. It provides a general transport vehicle for Upper Level Protocols (ULP), such as IPI, SCSI, HIPPI, IP, IEEE 802.2, etc. The bandwidth can be as high as one giga-bit per second. Multiple systems or storage devices can have point-to-point connections through FC switches. Fibre Channel also defines a loop topology that provides a shared media to multiple devices and hosts. In an FC loop, each node arbitrates the shared link following a fair arbitration algorithm, as discussed in Section 3.1. One of the main advantages of the FC based cluster is the "direct storage attachment to the cluster interconnect", which is unique to the Fibre Channel technology. This feature will change the fundamental file accesses paradigm of existing clusters. For non-FC clusters, storage devices have to be attached to a host and the host is attached to the interconnect. Therefore, storage devices attached to node A are not directly visible to node B. If node B wants to access files stored in node A, B has to request A's service to retrieve the files for him. These remote accesses are costly in terms of performance and resource utilization. In contrast, in an FC based cluster, because of the "direct storage attachment to the interconnect", all storage devices are immediately visible to all nodes. Thus, an FC based cluster can eliminate all of the remote accesses and therefore achieve both cost and performance benefits. We will discuss the FC based cluster architecture and its benefits in detail in the context of providing Internet multimedia services. We will also address FC cluster design issues such as data sharing, load balancing, cluster control mechanism, fault tolerance, and server scalability.

Fibre Channel is an emerging standard about which little can be found in the literature. Cummings [10] briefly described the general concept of Fibre Channel and the high level architecture of connecting peripherals to systems. Anujan Varma et al. studied the performance of FC switching systems [11,12]. Getchell and Rupert presented an FC-based local area network [13]. A comparison of FC and 802 MAC services was studied by Ocheltree, Tsai, Montalvo, and Leff [14]. Recently, Anzaloni et al. proposed a design solution for an FC-to-ATM internetworking gateway [15]. For FC loop topology, Chen and Thapar [16] studied the performance of a video server that uses FC loop as the storage interface. Early clustering concepts were represented by the DEC VAX clusters, which were based on the VMS operating system and provided file and resource sharing across nodes in the cluster. Currently, the most popular cluster architecture is the LAN based UNIX workstation cluster, which is available from most UNIX workstation vendors, such as IBM, HP, SUN, etc. Recently, server architectures based on ATM or other proprie-

tary switching technologies have been proposed. We will discuss these existing or proposed cluster and server architectures in detail in Section 2.

The remainder of the paper is organized as follows: Section 2 describes the current cluster and server architectures; Section 3 presents a Fibre Channel based architecture for a cluster of multimedia servers; Section 4 discusses some design issues of the FC based cluster; the performance issues related to the FC based cluster are addressed in Section 5; and Section 6 summarizes this paper.

2. Current Cluster and Server Architectures

In this section, we describe several existing and proposed cluster and server architectures. For the most popular UNIX workstation cluster, we take the HP-UX cluster [17] as an example for ease of discussion.

2.1. The HP-UX Cluster

HP-UX cluster is a collection of workstations that are interconnected by a typical FDDI / Ethernet LAN and is built on top of the NFS distributed file system [18]. One of the workstations is assigned as root node (r-node), and the rest are called c-node. Within the cluster, each c-node is booted from the r-node. After booting, each node perceives a single consistent file system view. To off-load the r-node, each c-node can have its local disks and create file systems on them. These file systems, once mounted, are immediately visible to all nodes in the cluster. The collection of workstations can work together to function as a powerful server complex, such as a Web server. When a cluster node receives an external request for a file access, if the file is in its local disk, then the node only performs a local disk operation, otherwise, it performs a remote file access and then return the results to the external client. In the later case, the requesting node acts as a NFS client, and the node carries the target file functions as a NFS server. Obviously, any remote file access incurs extra overhead (see Section 5 for more detailed discussions on the performance issues). As reported in [19], since NFS is a stateless system, which cannot properly manage the NFS-clients' file caches, the network traffic can be three times higher than that of stateful distributed file systems, such as AFS [19] and Sprite [20]. This may cause a serious performance problem when using the cluster as a Web server complex.

2.2. The UIUC / NCSA Web Server

The UIUC / NCSA Web server is based on the AFS distributed file system [3,8,21]. The server complex consists of multiple AFS file servers (SUN Sparc 10) and AFS clients (HP 9000/735). All of these nodes are connected via an FDDI ring. The AFS servers provide a consistent view of the file system to each AFS client, and the AFS clients collectively provide Web services to the external Web clients. One big advantage of the AFS based servers is good scalability. AFS allows heterogeneous server nodes. AFS client nodes can be easily added or removed to the Web server to handle varying Web traffic. In contrast to the stateless NFS, AFS is a stateful file system that enables an efficient management of the AFS clients' local file caches, which is critical

to the success of the NCSA Web server architecture [3]. Because of the AFS local caching, nearly 90% of the external Web requests are satisfied by the AFS clients without crossing the FDDI LAN to the AFS servers. The caching mechanism works well with current user access patterns, in which less than 1% of total requests accessing media objects such as audio and video. However, in the future, audio and video clips are expected to play a larger role in conveying multimedia information. Since these media objects are usually very large in size, this changing access pattern will have a significant negative impact on the efficiency of the caching strategy, and therefore on the overall performance of the server cluster.

2.3. ATM Based Server Architecture

There have been several proposals and prototypes on the ATM based media server architecture appeared in the literature. In [22] , Buddhikot et al proposed a Cluster Based Storage (CBS) architecture, in which multiple independent storage clusters are interconnected through an ATM switch. A central manager attached to the switch manages the ATM switch, performs the network signaling, and co-operates with the cluster-level storage managers to perform admission control. The cluster-level storage manager manages multiple storage nodes which are connected via a daisy-chained custom APIC (ATM Port Interconnect Controller) chips. The bi-directional APIC chain (each direction with 625 Mbit/sec bandwidth) functions as an I/O backplane in the cluster [23]. Since each cluster-level storage manager manages its data independently, it is not clear how to achieve data sharing across clusters in this configuration.

In another study, Ito and Tanaka [24] presented a video server architecture, which consists of a system manager, a File Distributor (FD), multiple Video segment File Servers (VFS), and multiple Sequence Control (SC) brokers. All of these nodes are interconnected through an ATM switch.

Based on the observation that future multimedia systems will perform not only data storage and retrieval but also data manipulations, Rooholamini and Cherkassky [25] proposed using ATM as the subsystem interconnect in a shared-memory multiprocessor multimedia server. In this architecture, processor module, memory module, network interface module, storage controller module, and other subsystem modules are all connected through one or more ATM switches. This architecture is quite different those "loosely-coupled" clusters discussed above.

2.4. Other Cluster Architectures

In [26], Haskin and Williams presented a cluster architecture for IBM's Web/video servers, which is based on the IBM SP-2 proprietary switch technology. At a high level, this architecture is similar to the one by Ito and Tanaka [24], with "File System Node" corresponds to the "SC brokers" and "Storage Node" corresponds to the "VFS servers". Moreover, each node is implemented by using a RS/6000 workstation.

An interesting on-going research work is the Berkeley NOW (Networks of Workstations) project [27], which tries to tackle the cluster computing in multiple dimensions, such as low overhead communications, cross-net memory sharing, a global layer UNIX, serverless network file sys-

tems, etc. FDDI was used as the cluster interconnect for the initial prototype, and faster interconnect such as Myrinet [28] will be used for the final demonstration.

3. A Fibre Channel Based Media Server Cluster Architecture

3.1. Fibre Channel Background

Fibre Channel, a newly defined industry standard, is a serial link used to connect various FC nodes and devices. These nodes or devices can be interconnected point-to-point by using a switch or share the same link by using the FC loop topology. Each link is bi-directional, with each direction having 1Gbit/sec of bandwidth. The physical media can be either fibre, which allows a distance of up to 10 kilometers, or copper, which allows a distance of 25 meters.

Like the ISO/OSI networking protocols, Fibre Channel also defines layered protocols for its data transfer (Figure 2). Because of its rich upper-level protocol support, Fibre Channel can be applied to a wide variety of areas, such as networking, storage interface, system interconnect, etc. Most of the FC chip implementations covers FC-0 to FC-2 in the protocol stack. Several FC chips also provide hardware assistance to some FC-4 level protocols, such as SCSI and IP, for performance purpose.

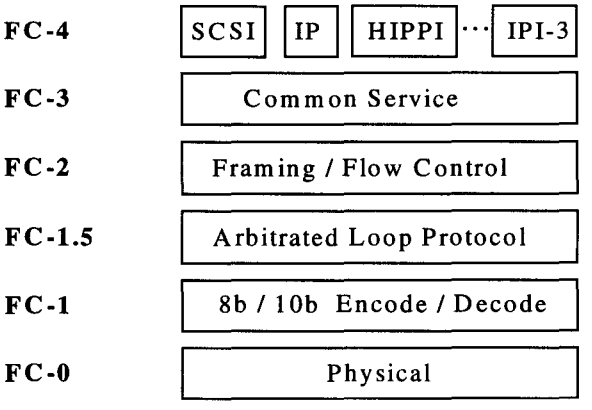

Figure 2: Fibre Channel Protocol Stack.

The current Fibre Channel standard defines three classes of service, which are distinguished primarily by the methodology used to allocate and retain the communication circuit between nodes.

- Class 1 -- Dedicated Connection:

  Dedicated connections must be established before transferring data frames. Once a connection is established, the bandwidth is guaranteed until one party releases the connection.

- Class 2 -- Multiplex:

  This is a connectionless service on a frame-by-frame basis. The Fabric, if present, routes the frame to the destination node. An end-to-end acknowledgment is required in this class of service.

- Class 3 -- Datagram:

This is also a connectionless service and is distinguished from the Class 2 service by not requiring the end-to-end acknowledgment.

In addition, Fibre Channel also defines an *"intermix"* of services that interleave Class 2 and Class 3 frames during an established Class 1 connection. In this case, Class 1 frames have a higher priority than Class 2 and Class 3 frames. These service classes will be further discussed in the next section.

Fibre Channel arbitrated loop provides a cost effective way of connecting multiple disks to a host or a switch port by sharing the same link media.

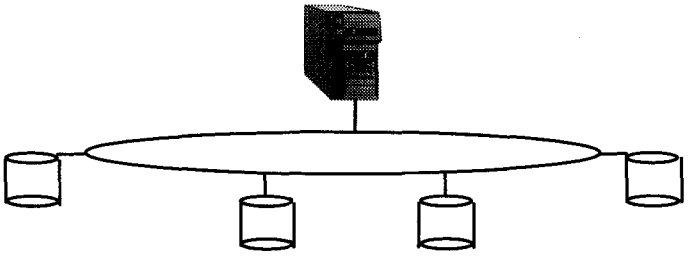

Figure 3: Fibre Channel Arbitrated Loop Topology.

Figure 3 shows a Fibre Channel Loop with five nodes, one host (or switch port) and four disks (multiple hosts are allowed). Each node has a port consisting of a transmitter and a receiver. The loop topology follows the same Fibre Channel point-to-point communication philosophy, i.e., only two nodes in the loop can communicate with each other at a time. Since all of the nodes share the same link, arbitration is required before a node is allowed to hold the link for transmission.

Unlike the SCSI priority arbitration rule, Fibre Channel defines a fair arbitration algorithm, which includes an access window. Whenever a node wants to arbitrate, it joins the window. If a node wins arbitration once, it must wait until all the other nodes in the current window have had a chance to win an arbitration. At that time, a new window is started, and each node has a chance to arbitrate again. This guarantees that each disk has an equal chance to transfer data. The only exception is that hosts can be assigned a high priority during the loop initialization phase. This allows hosts to send commands to multiple disks promptly.

3.2. An Architecture for Internet Multimedia Server Cluster

In this section, we present a Fibre Channel based architecture for Internet media server cluster, as illustrated in Figure 4.

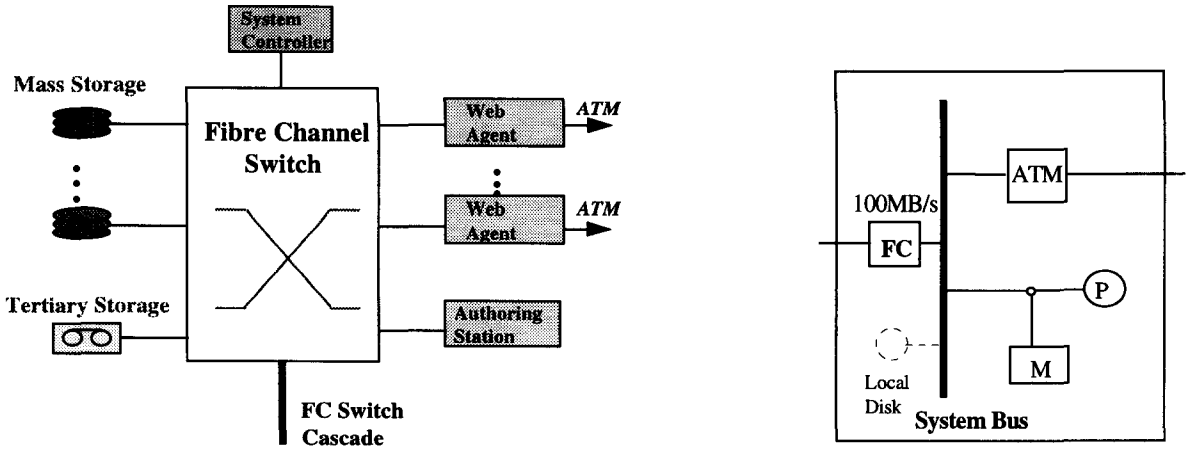

Figure 4: A Fibre Channel Based Media Server Cluster.

As shown in Figure 4(a), multiple server nodes and storage devices are attached to a Fibre Channel switch. The mass storage in the figure can be an individual FC disk, an FC disk array, or multiple FC disks / RAIDs attached to an FC loop. In the case of FC loop, although a maximum of 126 disks /RAIDs are allowed for each loop, in reality, less than 30 disks / RAIDs are recommended per loop to fully utilize the effective disk bandwidth. The simplified diagram for a cluster node, which can be a system controller, a Web agent, or an authoring station, is shown in Figure 4 (b). Here one FC interface replaces multiple SCSI interfaces and possibly the traditional Ethernet / FDDI LAN interface typically found in existing cluster nodes. Each cluster node can have an (optional) local disk for booting and local swap. The cluster nodes need not to be symmetrical. For example, the authoring station may have more CPU power for the image, video, and audio processings; the system controller may have higher fault tolerance requirement; and the Web agent can be optimized towards data transfers, including real-time continuos media deliveries such as audio and video. Since the server cluster is expected to store and manage a large database for the media objects, a tertiary storage subsystem is required. The data stored in tertiary storage subsystem can be managed to stage into disks or deliver directly to server nodes (Web agents). The communication between server nodes is done by using the Fibre Channel IP protocol and the data transfer between server nodes and storage devices is done by using the Fibre Channel SCSI protocol. The server cluster is connected to the Internet through ATM or other network interfaces.

3.3. Advantages of the Fibre Channel Based Architecture

While the architecture presented in Figure 4 looks similar to many of those discussed in Section 2, there is a main difference between an FC based cluster and existing clusters. In fact, many of those cluster architectures discussed in Section 2, such as NCSA's Web server, IBM's Web/Video server, HP-UX cluster, some ATM based server, etc. [3,17,22,25,26,27,26], can be

abstracted to the one shown in Figure 5 (a), where the processor block on the left hand of the interconnect corresponds to the AFS server, Storage Node, VFS server, etc. For ease of discussion, in the remaining of the paper, we use a unified term "storage node" to refer to these nodes. The main purpose for a storage node in a cluster is to manage storage devices attached to it and provide data services to other nodes in the cluster. Some of these storage nodes use full-blown general purpose machines, such as SUN Sparc 10 and IBM RS/6000, and some others use custom designed special purpose storage control systems. The processor blocks on the right hand side of the interconnect correspond to "server node" that receive and service the requests from external Web clients.

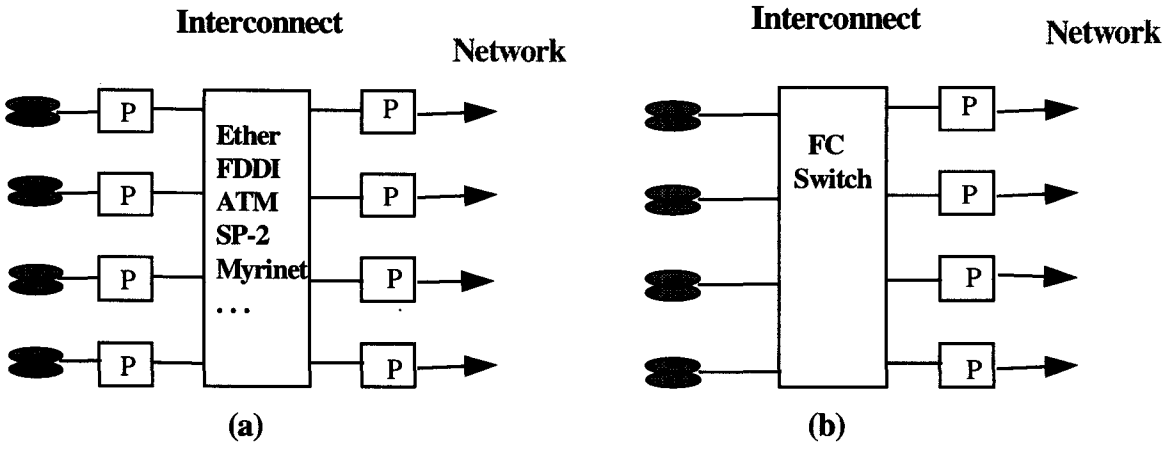

Figure 5: Difference between FC based Cluster and other Clusters.

Comparing Figure 5 (a) and (b), an obvious difference is that the FC based system, instead of connecting storage devices to the storage node, directly connects storage devices to the interconnect switch, and eliminates the storage nodes all together. This is because Fibre Channel devices, including FC disks, FC switches, and FC host interface card, can communicate directly using the SCSI protocol, which enables the "true" direct attachment of storage devices to the interconnect. Notice that for non-FC systems, although the term "storage direct network attachment" often appeared in the literature, in practice storage devices are attached to storage nodes and it is the storage nodes that are attached to the network. In non-FC systems, storage devices, such as disks, cannot directly communicate with the interconnect or network.

Allowing FC disks to directly attach to an FC switch results in improved cost/performance. The eliminating of storage nodes immediately reduces the system cost. Even compared to the Ethernet /FDDI based clusters, which do not incur the switch cost, the FC switch expense can be greatly offset by the savings on the multiple of storage nodes. In terms of the cost of FC disks, manufacturers estimate a comparable price to that of fast/wide differential SCSI disks. The performance benefit of the FC based cluster will be discussed in detail later in Section 5.

4. Fibre Channel Based Cluster Design Issues

4.1. Data Sharing and I/O Load Balancing

Data sharing is one of the fundamental design goals for a cluster of media servers, i.e., the file system should provide a single consistent view of all files stored in the cluster to each server node. Fortunately, as a consequence of direct disk attachment to the switch, the data sharing can be easily achieved in an FC based cluster. One simple implementation can follow the HP-UX cluster paradigm, i.e., the system controller acts as the root node, and all other server nodes boot from the root node. One main difference is that in an HP-UX cluster, file sharing is achieved via the NFS mount mechanism, whereas in an FC cluster all disks attached to the switch are immediately visible to each server node when it boots. Therefore, all files created on these disks can be accessed directly by each cluster node.

In clusters that are based on distributed file systems (such as NFS, AFS, etc.), files are typically not allowed to be striped across nodes, i.e., each node with disks manages a set of files and serves other nodes' access requests on these files. This paradigm has a disadvantage for load balancing. If some files are hotter than others, then the node managing the hot files may become a bottleneck in the cluster. A current solution to solve this problem is to store replicate copies of these hot files on multiple nodes to share the workload. However, this may be costly for a media server cluster, since multimedia objects are typically very large in size. In addition, it also increases the system complexity, since load balancing algorithms are required to decide where, when, and for which files to create and remove the redundant copies. In an FC based cluster, all of the disks are cluster-wide resources which enables the file system to manage these disks in a single consistent way. Let's assume that multiple disks are connected through an FC loop and multiple of such loops are attached to FC switch ports. Then wide data striping can be achieved that stripes files not only across disks within a loop, but also across loops. In this way, file access load can be evenly distributed across all disks. One may argue that redundant copies are also necessary needed for fault tolerant reasons. However, whence the load balancing is no longer a problem, fault tolerance can be achieved by using some cheaper ways, which we will address in Section 4.3.

4.2. Distributed vs Centralized Control

There are two ways to control the server cluster to serve requests from external clients. Consider a Web media server cluster. In the distributed control mechanism, which is similar to the one in [3], the Web domain name http://www.hp.com is mapped to the IP addresses of multiple server nodes or Web agents in a round-robin manner by the Domain Name Resolver (DNR) using the Berkeley Internet Name Domain (BIND) code [29]. Each Web agent node serves Web requests routed to it independently (in this case, the system controller can also function as a Web agent). By using the round-robin mapping mechanism, hopefully each Web agent receives the same amount of requests to share the workload. This, however, may not be necessarily true because of the caching effect of some local DNRs [8]. In fact, multiple clients using the same local DNR will access the same server node, resulting in hot spots for content that has geographical relevance. One advantage of the distributed control is the ease of adding and removing server nodes,

since each node is functionally symmetrical and serves requests independent of each other. A disadvantage of the distributed control is that the round-robin server assignment mechanism failed to utilize the workload characteristics to optimize the server performance. For example, when a server node recently serviced a request, the data may still be in the node's main memory when the next request arrives. Therefore, assigning the next request for the same data to this node may achieve a better performance.

In contrast to the distributed control mechanism, the centralized control method assign one node (e.g., the system controller) as the interface to the external clients, i.e., all external requests are routed to this node. This central node may serve some small text files and redirect requests for large multimedia objects to other nodes by using hyperlinks. In this case, the central node can use high-end systems or multiprocessor systems to avoid a bottleneck. On the other hand, the remaining server nodes can be optimized to service multimedia objects, such as audio and video. These optimizations include data type dependent caching algorithm, quality of service (QoS) protocols, direct storage to network transfers, etc. The central server node can redirect requests to other nodes by using a simple round-robin mechanism, or using more intelligent mechanisms that keep track of access patterns.

Finally, a hybrid of the above two control mechanisms can be configured that maps the Web domain name to a subset of the server nodes which in turn redirect requests for media objects to those optimized media server nodes.

4.3. Fault Tolerance

As technologies progress and applications grow, fault tolerance is becoming more and more important to today's computer systems. For Fibre Channel based server clusters, fault tolerance can be provided at different levels, depending on the demands and budget.

- ***Disks and I/O Channels Failures***

Disks are the most unreliable components is the system. A well-known and simple way to protect from a disk failure is to use RAIDs. That is, instead of using individual disks, use RAIDs to attach to FC loops or switch ports. However, this only protects from disk failures, but not I/O channel failures. One way to protect channel failure is to use the "dual-loop" structure as shown in Figure 6. In this configuration, each FC disk/RAID has dual ports, one connected to an independent loop, and each loop is attached to a separate switch port. Therefore, by using the dual-loop structure, fault tolerance is achieved for not only the FC channel (the loop) but also the switch port. Furthermore, besides the fault tolerance, the performance is also improved because double channel bandwidth is available now, which allows to attach more disks/RAIDs to the loop.

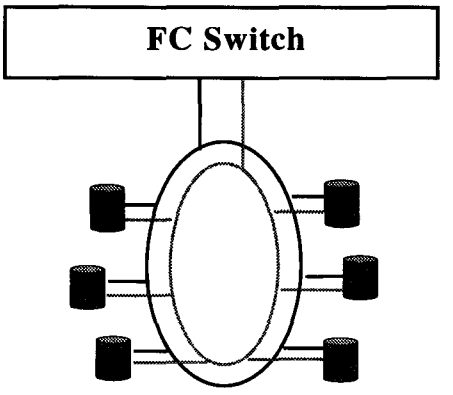

Figure 6: Dual-Loop Structure.

An alternative to provide fault tolerance on disk, loop, or switch port is shown in Figure 7. In this figure, a file *f* is striped across multiple loops and parity is used to protect a failure from either a disk, a loop, or a switch port. This configuration is cheaper than the previous one, but the cluster may suffer a performance degradation when any one of the three components fails.

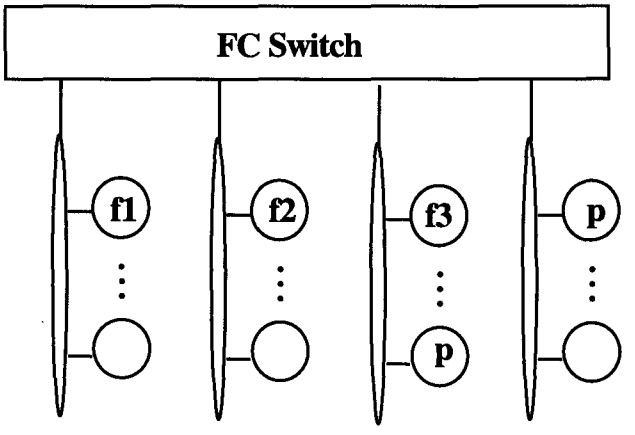

Figure 7: Striping across Loops with Parity Protection.

- ***Cluster Server Node and Switch Failures***

If clusters using the distributed control mechanism described in Section 4.2, when a server node fails, the DNR needs to be informed to temporarily remove the failed node from the round-robin distribution list. Since each node functions the same and operates independently, the cluster will be continuously operational but with less processing power. All of the current connections to the failed server node, however, are interrupted, and there is no way to recover these failed connections automatically. In this case, the Web clients connected to the failed node see an error message on their screens, and if they try to connect again, their requests are routed to surviving nodes that treat these requests as new requests and start to service them all over again.

In the centralized control situation, more sophisticated recovery mechanisms can be implemented by maintaining connection status of each server node at the central system control node(s). Various degree of recovery can be achieved depending on the complexity of the states and algorithms. The details of these recovery methods are beyond the scope of this paper and will not be discussed here. The control node is the key component of the cluster. If desired, the failure can be protected by providing a fully redundant fault tolerant control node. Or an alternative is to use the hybrid control mechanism, in which multiple control nodes are configured.

Last, for the interconnect Fibre Channel switch, we have discussed the protection of single port failure. However, the switch backplane is still a potential single point of failure, i.e., if the switch backplane fails, the whole cluster halts. So if a non-stop server without any single point of failure is required, a fully redundant server can be configured as shown in Figure 8.

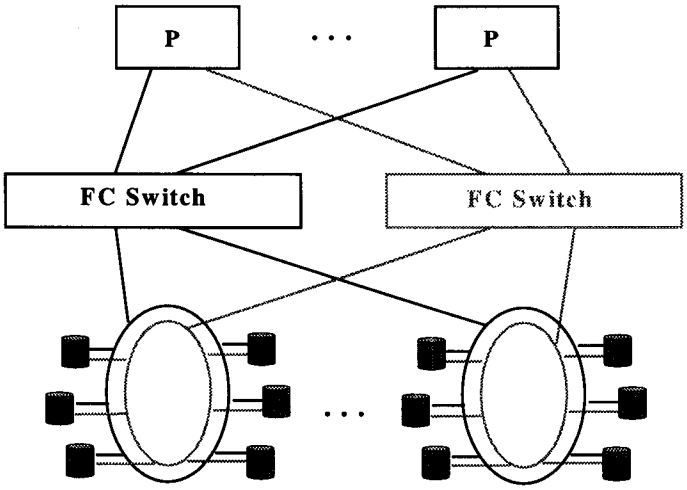

Figure 8: Fully Redundant Cluster Structure.

4.4. Scalability

Given the Internet growth rate, an Internet media server is required to be scalable to cope with the increasing service requests. The scalability of the Fibre Channel based server cluster can be achieved by adding/removing server nodes and/or storage devices to the interconnect switch. When more server processing power is needed, a server node is added to the switch. When more disk bandwidth or storage capacity is needed, one more loop is added to the switch. This provides flexibility to adjust the server size according to the demand. When the available switch ports run out, new switch can be cascaded to the existing switch.

5. Performance Issues

In this section, we address some performance issues related to the Fibre Channel based media server cluster.

5.1. The Remote File Access Issue

As stated in previous sections, compared to other cluster architectures, one of the most significant advantages of using the Fibre Channel technology is the direct storage attachment to the interconnect. For non-FC clusters, when a server node receives a request for files that are not cached or stored in local disks, it must retrieve the files from remote (storage) node(s). A typical data path for a remote file access is illustrated in Figure 9 (a).

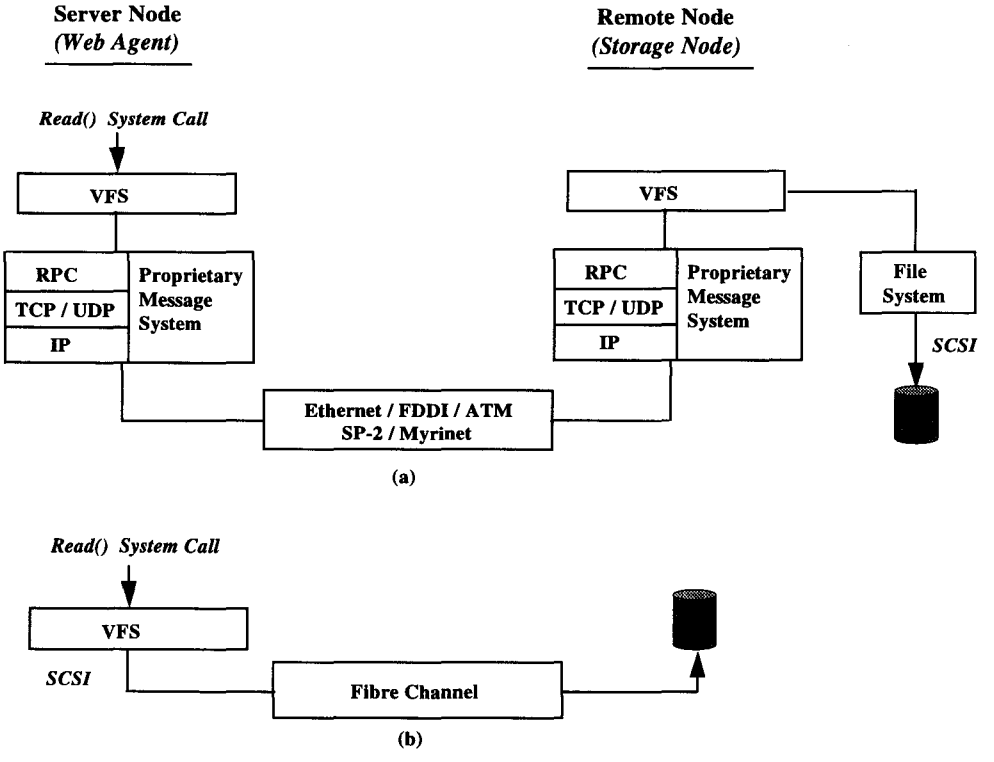

Figure 9: Data Path for File Accesses.

As we can see from the figure, remote accesses involve multiple components for non-FC based systems. To read a file, the system call is trapped into the kernel. The kernel virtual file system (VFS) layer identifies local files vs remote files. If the system call is for a remote access. The VFS then calls the network protocol stack if the interconnect is a LAN (Ethernet, FDDI, or ATM) or the proprietary message system layer if the interconnect is a proprietary switch. When the remote storage node receives the request, its VFS layer calls the local file system that translates the request into a SCSI command and then initiates the disk I/O. The requested file data is transferred into a memory buffer in the remote storage node which packages the data according to the packet or message format required by the interconnect and then ships the data back to the requesting Web agent node. There are many factors that may have negative impact on the performance of the remote accesses. First, the interconnect LAN can be slow. Today's typical Ethernet has 10 Mbit/sec of bandwidth, FDDI 100 Mbit/sec, and ATM/OC-3 link 155 Mbit/sec. Second, remote accesses have to suffer from the overhead of network protocol processing or

message processing and packaging, which could be very high if multiple data copies in the performance path are involved. Third, since there are many system resources/components are involved in the remote accesses, such as interconnect network, storage node processor, memory, bus, and disks, any contention on these resources will delay the data accesses, which is undesirable for real-time media object accesses, such as audio and video retrievals. Some studies have reported that remote file accesses are three times slower than local accesses [19].

5.2. The File Cache Issue

Although the current AFS/FDDI based NCSA Web server achieves satisfactory performance, its success so far greatly relies on the cache of frequently accessed files in the Web agent nodes (AFS clients). According to the current user access patterns, in which only 1% of the total requests access the audio or video files, it is reported that 90% of total requests can be serviced by the Web agents from their local caches [3,8]. However, since audio and video objects are expected to play a greater role in delivering multimedia information when the Internet infrastructure improves, the audio and video requests will increase rapidly. Since these audio and video objects are typically very large in size, caching them in the Web agents' local disk or memory will force the removal of many other (small) files from the local cache, which in turn will decrease the local cache hit ratio. Therefore, new data type dependent cache strategies need to be designed. In order to maintain a high cache hit ratio, probably only small portion of the audio and video data can be cached and most of them will be retrieved from remote storage nodes (AFS servers). As reported in the above study [3,8], the 1% of audio and video requests account for 28% of total bytes transferred. If the audio and video requests increase by 1%, simple mathematics reveals that the 2% of audio and video requests will account for 44% of the total bytes, and if 5% of the total requests are for audio and video then they will account for 67% of the total bytes transferred. This will significantly increase the traffic across the interconnect and the workload to the storage nodes. As a result, this may cause a severe performance degradation for the Web server.

In contrast, in a Fibre Channel based cluster, all storage devices attached to the interconnect switch are immediately visible to all server nodes. This allows a server node to access any file stored in the cluster by letting the file system directly issues SCSI commands to the storage devices, as shown in Figure 9 (b). This eliminates "remote" accesses and all file reads/writes are executed as "local" accesses. Because of the elimination of the protocol/ message processing and data/command format translation overhead for remote accesses, the performance for file accesses is expected to be higher for FC based clusters. In addition, using the Fibre Channel based cluster automatically solves the cache problem faced by an AFS-based server cluster, such as the NCSA Web server. All files effectively appear to be local to FC server nodes.

5.3. Fibre Channel Switch Overhead

Compared to truly local disk accesses (e.g., a disk directly attached to a server node's local PCI bus), the accesses of the disks attached to the Fibre Channel switch incur an extra delay to cross the switch. The minimum switch delay (i.e., there is no contention in the switch) for a Class 1 frame is 1 microsecond, and for Class 2 and 3 frames is about 8 microseconds. The Class 1 service, however, incurs an additional up front connection set up delay in the range of 150-250 mi-

croseconds. There are two types of traffic across the switch: (a) the IP traffic for the communications between server nodes; and (b) the SCSI traffic for data transfers between server nodes and storage devices. In an FC based multimedia server cluster, the inter-server communication traffic, as compared to server-disk data traffic, is relatively small, and can be delivered by either Class 2 or Class 3 services.

For the server-disk traffic, we identify two scenarios according to the traffic direction. The traffic from server node to disks is typically SCSI commands or control messages, which are usually very short and can be packed into a single frame (we assume that disk writes or database updates can be done off-line). Thus either Class 2 or Class 3 services can be used. On the other hand, the traffic from disks to server nodes is typically data requested by these nodes. To improve disk utilization while accessing media objects, typically media server file systems I/O requests access a large block of data (e.g., 128 to 256 Kbytes as compared to 4 to 8 Kbytes in traditional transaction processing systems).

When multiple disks are connected through an FC loop attached to a switch port, delivering these large data trunks using Class 2 or Class 3 services may suffer from the "*head of line blocking*" problem. For example, consider two loops attached to switch port 3 and 4 transferring data to the server node on switch port 1 by using Class 3 service. Another disk in the port 3 loop tries to send data to the server on port 2 which is idle. Since both port 3 and port 4 loops are contending for the same destination port 1, each port can only get half of the port 1 bandwidth, which means the port 3 loop can only transfer its data at half speed. Therefore, the next transfer in port 3 loop to port 2 is blocked by the slow transfer to port 1, even though the port 2 is currently idle. Some preliminary simulation results show that, with the head of line blocking, only 50% of the port bandwidth can be effectively utilized. This implies that, for a 1Gbit/sec port, it can only sustain 500 Mbit/sec of data rate. In a previous study [16], we observed that, if a 1 Gbit/sec FC loop is directly attached to a host PCI bus, it can sustain 700-800 Mbit/sec of data rate for real-time applications such as video-on-demand. Thus, if a switch between the host and the loop can only sustain 500 Mbit/sec data rate, the switch is apparently a bottleneck.

If Class 1 service is used, then in the above example, only one connection can be established to port 1 at a time. If the port 3 loop holds the connection to port 1, then the next request in port 3 loop only needs to wait for the first transfer to finish, which is conducted at a full speed of 1Gbit/sec. On the other hand, if the port 4 loop holds the connection to port 1, then port 3's connection request will be rejected, and the re-arbitration of the port 3 loop will make the next request a winner, which in turn can establish a connection to the idle port 2. Thus the head of line blocking can be avoided. The Class 1 connection setup overhead is a concern, but it can be justified by the large data transfer within a connection, which typically takes several milliseconds. Therefore, using Class 1 service for the data traffic from disks to server nodes might be appropriate for multimedia servers.

5. Summary

In this paper, we presented an architecture for a cluster of Internet multimedia servers, which is based on the Fibre Channel technology - a newly developed industry standard. Currently the most

popular clusters are the LAN based UNIX workstation clusters, which rely on distributed file systems, such as NFS, AFS, or Sprite, to achieve data sharing. There are other architectures proposed by researchers that use ATM or proprietary switching technologies as the cluster interconnect. Compared to these existing and proposed cluster architectures, the FC based cluster allows storage devices to be directly attached to the interconnect FC switch. This feature will change the fundamental data sharing model of existing clusters, in which remote file accesses are necessary if the required files are not stored or cached locally. These remote accesses are costly and could be three time slower than local accesses. In contrast, because of the "direct storage attachment to interconnect" feature, all storage devices in an FC based cluster are directly visible to all cluster nodes. This enables all file accesses to be performed as local I/O operations at any cluster node.

The benefits achievable by the FC based cluster architecture are manifold. First, it achieves the performance benefit by eliminating all of the remote data accesses. It also avoids the "caching large media objects" problem faced by AFS-based multimedia server clusters. Second, it achieves the cost benefit by eliminating the storage nodes or file servers found in many existing cluster architectures. Third, it enables a single uniform mechanism to manage all of the storage devices in the cluster. This allows a wide data striping across multiple disks and therefore achieves the load balancing across I/O channels and storage devices. This load balancing is extremely important for real-time retrieval of large continuous media objects. The load balancing amongst the server nodes can be achieved by intelligently routing external client requests to these server nodes. Fourth, in an FC based cluster, the server scalability can be achieved simply by adding or removing a server node or a loop of storage devices according to the demands. Last, the server fault tolerance can be achieved at various levels, ranging from RAIDs that protect only disks to fully redundant server clusters that have no single point of failure.

Acknowledgment:

The authors wish to thanks Lucy Cherkasova for her generous help on providing some of the simulation results on the Fibre Channel switch performance.

References:

[1] Shenker, S., "*Fundamental design Issues for the Future Internet*," IEEE J. Selected Areas in Communications, Sept. 1995.

[2] Pitkow, J. and M. Recker, "*Results from the First World-Wide Web User Survey*," Computer Networks and ISDN Systems, 27, pp.243-254, 1994.

[3] Kwan, T. and McGrath, R., "*NCSA's World Wide Web Server: Design and Performance*," IEEE Computer, Nov. pp.68-74, 1995.

[4] Zhang, L. et al, "*RSVP: A New Resource Reservation Protocol*, " IEEE Network, Vol. 5, pp.8-18, Sept. 1993.

[5] Braden, R. D Clark, and S.Shenker, "*Integrated Services in the Internet Architecture: an Overview*," RFC 1633, June 1994.

[6] Topolcic, C., "*Experimental Internet Stream Protocol: Version 2 (ST-II),*" RFC 1190, BBN, Oct. 1990.

[7] Bradner, S. and A.Mankin, "*Next Generation IP*," Internet RFC 1752, Jan. 1995.

[8] Kwan, T., R. McGrath, and D. Reed, *"User Access Patterns to NCSA's World Wide Web Server,"* Tech. Report UIUCDCS-R-95-1934, Dept. Computer Science, Univ. of Illinois, Urbana-Champain, Feb. 1995.

[9] ANSI Standard X3T9.3 *Fibre Channel Physical and Signaling Interface (FC-PH),* Rev 4.0, May 1993.

[10] Cummings, Roger, *"System Architectures Using Fibre Channel,"* Proc. 12th IEEE Symposium on Mass Storage Systems, pp.251-256, 1993.

[11] Varma, Anujan, Vikram Sahi, and Robert Bryant, *"Performance Evaluation of a High-Speed Switching System Based on the Fibre Channel Standard,"* Proc. of the 2nd IEEE Int'l Symposium on High-Performance Distributed Computing, Spokane, Washington, July 1993.

[12] Varma, Anujan, Shree Murthy, and Robert Bryant, *"Using Camp-on to Improve the Performance of a Fibre Channel Switch,"* Dept. of Computer Engineering, Univ. of California/Santa Cruz, CA 95064, 1993.

[13] Getchell, D. and P. Rupert, *"Fibre Channel in the Local Area Network,"* IEEE LTS, Vol.3, No. 2, pp.38-42, May 1992.

[14] Ocheltree, K.B., T.C. Tsai, R. Montaivo, and A. Leff,*"A Comparison of Fibre Channel and 802 MAC Services,"* Proc. of IEEE , pp.238-246, 1993.

[15] Anzaloni, A., M. De Sanctis, F. Avaltroni, G. Rulli, L. Proietti, G. Lombardi, *"Fibre Channel (FCS) / ATM Internetworking: A Design Solution,"* Proc. Global Telecommunications Conf., Vol.2, pp.1127-1133, 1993.

[16] Chen, Shenze and Manu Thapar, *"Fibre Channel Storage Interface for Video-on-Demand Servers,"* IS&T/SPIE Proc. Vol 2667 on Multimedia Computing and Networking, San Jose, CA, Jan. 1996.

[17] Hewlett-Packard Co. HP-UX Release 9.0, *Managing Clusters of HP 9000 Computers*, Aug. 1992.

[18] Sandberg, R., et al, "Design and Implementation of the SUN Network File System," Proc. Summer USENIX Conf. pp.119-130, Portland, OR, June, 1985.

[19] Howard, John H, et al, *"Scale and Performance in a distributed File System,"* ACM Trans. on Computer Systems, Vol. 6, No.1, pp.51-81, Feb. 1988.

[20] Nelson, M.N., B.B. Welch, and J.K. Ousterhout, *"Caching in the Sprite Network File System,"* ACM Trans. on Computer Systems, Vol.6, No.1, pp.134-154, Feb. 1988.

[21] Katz, E.D., M. Butler, and R. McGrath, *"A Scalable HTTP Server: The NCSA Prototype,"* Computer Networks and ISDN Systems, 27, pp.155-164, 1994.

[22] Buddhikot, M.M., G.M. Parulkar, and J.R. Cox, Jr, *"Design of a Large Scale Multimedia Storage Server,"* Computer Networks and ISDN Systems, 27, pp.503-517, 1994.

[23] Dittia, Z.D., J.R. Cox, Jr, and G.M. Parulkar, *"Using an ATM Interconnect as a High Performance I/O Backplane,"* Presentation at Hot Interconnects II, Stanford, CA, Aug. 1994.

[24] Ito, Yukiko and Tsutomu Tanaka, "A Video Server Using ATM Switching Technology," unpublished, 1994.

[25] Rooholamini, Reza and Vladimir Cherkassky, *"ATM-Based Multimedia Servers,"* IEEE Multimedia, Spring, 1995, pp.39-52.

[26] Williams, Robin and Roger Haskin, *"Tiger Shark Video Server,"* Presentation at Hot Interconnect III, Stanford, CA, Aug. 1995.

[27] Anderson, Thomas. E, et al, *"A Case for NOW (Networks of Workstations),"* IEEE Micro, Vol.15, No.1, pp.54-64, Feb. 1995.

[28] Seitz, C., "*Myrinet - A Gigabit per second Local-Area Network*," IEEE Micro, Vol.15, No.1, Feb. 1995.

[29] Albitz, P. and C. Liu, "DNS and BIND in a Nutshell," O'Reilly and Associates, Sebastopol, CA, 1992.