The Case for Reexamining Integrated File System Design

Prashant Shenoy
Department of Computer Science,
University of Massachusetts at Amherst
shenoy@cs.umass.edu

Abstract— **Research in integrated file systems—file systems that support different types of data and applications with diverse performance requirements—has stagnated since the commercial failure of video-on-demand systems in the 1990s. However, the past two years have seen an acceleration in the development and deployment of several new technologies aimed at web-based multimedia applications. These technologies have opened up a fertile research ground for new file system optimizations and rekindled interest in integrated file systems. In this paper, we describe some of these technology trends, examine their implications on file system design, and present a sampling of the new research problems that arise due to these technological changes. Based on these observations, we argue for a renewed effort in designing next-generation integrated file systems.**

I. Technology Trends

An integrated file system is a general-purpose file system that supports applications with diverse performance requirements (e.g., interactive, throughput-intensive, soft real-time) accessing data with different characteristics (text, images, streaming media). While integrated file systems have evolved from continuous media servers, they are inherently different and more complex than a video-on-demand server. Specifically, a continuous media server is a special-purpose file system optimized for a single data type, namely streaming media, while an integrated file system supports a diverse set of applications and data types found in general-purpose computing environments. An integrated file system is also different from a conventional file system—the former is aware of the semantics of the stored data and employs mechanisms that exploit these semantics to cater to different types of applications, while the latter is typically oblivious of semantics of files and provides a simple best-effort service to all applications.

Whereas several integrated file systems were designed in the mid 1990s [4], [5], [6], [11], research in integrated file systems has stagnated in the recent past. However, the development and deployment of several new technologies for emerging web-based multimedia applications has necessitated a rethinking of the design of such file systems. Consider the following technology trends that are fueling the growth of web-based multimedia applications.

This research was supported in part by an NSF CAREER award CCR-9984030, NSF grants ANI-9977635, CDA-9502639, Intel, Sprint, and the University of Massachusetts.

TABLE I
PROJECTED GROWTH IN NUMBER OF USERS WITH HIGH-SPEED INTERNET ACCESS

Year	Cable Modem	DSL	Wireless broadband
2000	1.6 million	0.78 million	52,000
2001	3.1 million	1.7 million	160,000
2002	5.3 million	3.1 million	337,000
2003	8.3 million	5.0 million	445,000

- *Proliferation of streaming media content:* Although envisaged in the early 1990s in the context of video-on-demand servers, it is only in the past two years that streaming media has gained widespread use. A recent survey has found that the number of Internet users accessing streaming media content has exceeded 50 million and is growing. To cater to this clientele, an increasing fraction of web sites today provide content in Real Audio/Video, Quicktime, or Windows Media formats. Audio files in MP3 format are immensely popular (although illegal) on most university campuses. This trend is only likely to accelerate with the advent of inexpensive multimedia PCs and widely available, free software decoders.
- *Broadband to the home:* Concurrent to the growing use of streaming media, technologies such as cable modems and digital subscriber lines (DSL) have emerged, which promise high-speed Internet access to home users. In contrast to the slow 56 Kbps analog modem speeds, these technologies provide either a 10 Mbps shared connection (in case of cable modems) or a 384 Kbps dedicated connection (in case of DSL modems). Table I depicts the projected growth of users with high-speed Internet access [7]. The wide availability of higher bandwidth connections is resulting in increasing use of bandwidth-starved applications such as 3D multi-player games, online virtual worlds, and high-quality continuous media.
- *Proliferation of networked devices:* The past few years have seen a substantial increase in the use of networked

personal digital assistants (PDAs). Popular PDAs, such as Palm VII, come equipped with a low bandwidth wireless connection that enable them to instantly access information on the Internet. Moreover, many cellular phone providers now offer wireless data services that enable a user to access the Internet via a mobile computer or a custom browser built into the phone. A key characteristic of these networked devices is that they are *resource-scarce*—they have significantly smaller processing and display capacities as compared to a typical desktop as well as a lower bandwidth network connection (see Table II)

- *Emergence of storage area networks:* Traditionally, storage servers have employed the server-attached disk architecture, in which disks are locally attached to servers and clients access data on disks via the server (see Figure 1(a)). Recently, a new storage architecture has evolved that envisions a separation of storage devices from servers [1], [2], [9], [12]. This architecture consists of a *storage area network (SAN)* to which storage devices such as disks are attached; servers access these devices via the storage area network (see Figure 1(b)). Such a network-attached disk architecture is markedly different from the traditional server-attached disk architecture: (i) it allows clients to directly access data from disks without the server in the data path; (ii) it is more resilient to failures, since a server failure can be handled by employing another server to manage the set of disks.

We believe that these technology trends will profoundly impact how file systems will be designed and used in the next decade. In what follows, we examine the impact of these trends on file system design.

II. Technology Implications

The first two technology trends—the growing use of continuous media files and the availability of high-speed Internet access—argue for file systems that can efficiently handle streaming media data. However, special-purpose file systems that support only streaming media data but not other data types (e.g., video-on-demand servers) are unsuitable for this purpose. This is because emerging applications access a diverse set of data types and not just streaming media. To illustrate, a typical distance learning application accesses class materials consisting of textual and image data and class lectures consisting of streaming media; even commonly used presentation packages such as PowerPoint use a combination of text, images and streaming media data for multimedia presentations. This argues for the design of a single file system that integrates the storage of different types of data and efficiently services different types of applications (e.g., streaming applications, traditional interactive and throughput-intensive applications). We refer to such a file system as an *integrated file system*. Integrated file systems simplify system administration tasks since they reduce the number of disparate systems that need to be maintained; it has been argued that system administration is often the dominant cost in computing environments [3]. Integrated file systems also increase utility to uses by eliminating the need to store and manage different types of files on different servers (e.g., store textual files on a conventional file system, audio and video files on a streaming media server). Furthermore, performance studies have shown that sharing of resources in integrated file systems yields significant performance benefits over a partitioned architecture employing separate servers [10]. While these benefits come at the expense of increased file system complexity, a number of conventional file systems are nevertheless moving towards an integrated architecture by providing native support for data types such as audio and video [4], [5].

Most existing file systems, whether integrated or partitioned, are based on the traditional server-attached disk architecture. The emergence of storage area networks will require us to rethink the policies and mechanisms employed by existing file systems. As explained in Section I, in a storage area network, storage devices such as disks and disk arrays are attached to a network; servers access these devices via the storage area network resulting in an inherently distributed architecture [1], [2], [9], [12]. This architectural change has several implications. First, since disks are attached to a network, the architecture allows clients to directly communicate with disks without the server in the critical path; the file server is involved only in infrequent operations such as authentication and validation. This not only improves client response time and but also allows the server throughput to scale with the number of disks [8], [14]. Second, since disks are no longer attached to servers, this architecture enables high availability of data; a server failure can be easily handled by having another server take over the responsibility of managing disks. Third, with falling costs of microprocessors, network-attached disks are likely to consist of controllers with powerful embedded processors. A file system can harness these computational resources by offloading simple file system tasks to individual disks. To illustrate, a recent study has shown that a file system can offload the placement policy to the on-disk microprocessor and improve performance—the on-disk microprocessor can determine the current disk head location and place data close to this location so as to reduce head movement and reduce the latency of write requests [13]. Partitioning file system functionality between servers and disks has fundamental implications on the policies for placement, retrieval and caching. For instance,

TABLE II

CHARACTERISTICS OF PDAS AND TYPICAL DESKTOPS.

Characteristic	PDA (Palm Pilot)	Desktop
Processor	16.6 MHz Motorola 68328	700MHz Intel Pentium III
Memory	8MB	128MB
Display	150x150, 1 bit color	1280x1024, 16 bit color
Network	9.6 Kbps wireless	100Mb/s Ethernet

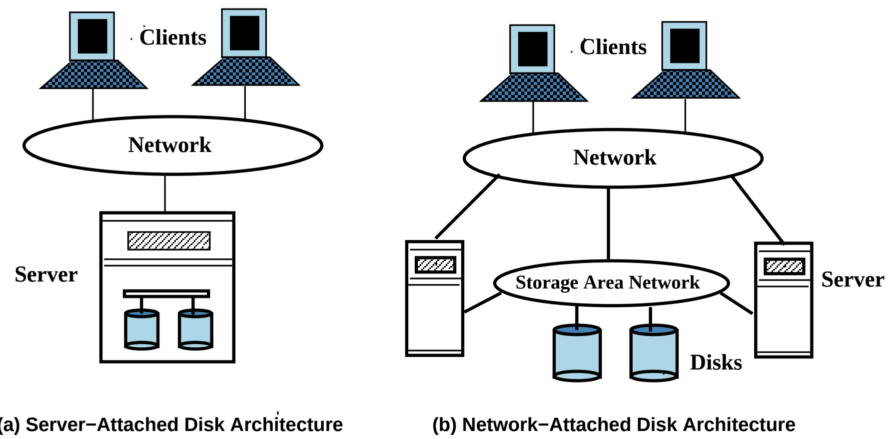

Fig. 1. The server-attached disk architecture and network-attached disk architecture for file system design.

network striping of data on disks in a storage area network will involve different tradeoffs as compared to striping files in a centralized file system. Similarly, the efficacy of a buffer cache at a server will need to be reexamined if users bypass the server and communicate directly with storage devices. Further, file system interfaces will need to be appropriately extended to enable users to communicate with storage devices (in addition to communicating with the server). These examples illustrate the need for redesigning integrated file systems so that they can take advantage of the features offered by a network-attached disk architecture.

Clearly, an integrated file system that employs the network-attached disk architecture can efficiently meet the needs of a large number of users and demanding multimedia applications. However, with the proliferation of networked devices (e.g., PDAs), these file systems will not only need to scale up to demanding applications, but also *scale down* to accommodate resource-scare end-hosts. Moreover, the file system protocols employed by such file systems will need to be reexamined to address this heterogeneity in end-host capabilities. For instance, file system protocols will need to be lightweight to handle resource-scare end hosts (protocols such as NFS that require the entire state information to be sent with every request may impose excessive communication overheads on such hosts). Employing a proxy to act as an intermediary between a server and an end-host is one technique for handling this mismatch. However, native file system support will be required in the long term to handle this heterogeneity. This is because the number of resource-scarce networked devices is likely to far exceed the number of high-end workstations accessing a server.

As file systems become more complex and distributed, new techniques will need to be developed to improve the *manageability* of such systems. It has been argued in [14] that, a manageable file system should: (i) be *self-healing* i.e, not require people to intervene for correcting faults and failures; (ii) be *self-managing* i.e., take as input declarative specifications of what the system is trying to achieve, not how it should be done; and (iii) be *self-configuring* i.e., adapt to changing workloads and isolate performance bottlenecks. Design of such manageable file systems remains a nascent research topic.

III. CONCLUDING REMARKS

In conclusion, several technology trends are likely to impact file system design in the next decade. In particular, (i) the growing use of streaming media, (ii) easy availability of high-speed Internet access, (iii) proliferation of networked devices, and (iv) emergence of storage area networks are four trends that will dictate how file systems of the future are built and used. Based on these observations, we argued that next generation file systems should: (i) employ an integrated architecture, (ii) be optimized for the network-attached disk architecture, (iii) should efficiently support the heterogeneity in end-host requirements, (iv) be manageable. We provided a sampling of research problems that need to be addressed to meet these requirements. We believe that these issues provide a fertile ground for future research on integrated file system design and are currently designing a file system to address these challenges.

REFERENCES

[1] G A. Gibson et. al. A Case for Network-Attached Secure Disks. Technical Report CMU-CS-96-142, School of Computer Science, Carnegie Mellon University, 1996.

[2] G A. Gibson et. al. A Cost-Effective, High-Bandwidth Storage Architecture. In *Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-VIII), San Jose, CA*, pages 92–103, Oct 1998.

[3] R. Golding, E. Shriver, T. Sullivan, and J. Wilkes. Attribute-managed Storage. In *Proceedings of the Workshop on Modelling and Specification of I/O (MSIO), San Antonio, TX*, October 1995.

[4] R. Haskin. Tiger Shark–A Scalable File System for Multimedia. *IBM Journal of Research and Development*, 42(2):185–197, March 1998.

[5] M. Holton and R. Das. XFS: A Next Generation Journalled 64-bit File System with Guaranteed Rate I/O. Technical report, Silicon Graphics, Inc, Available online as http://www.sgi.com/Technology/xfs-whitepaper.html, 1996.

[6] C. Martin, P. S. Narayan, B. Ozden, R. Rastogi, and A. Silberschatz. The Fellini Multimedia Storage System. *Journal of Digital Libraries*, 1997.

[7] PC Computing Report, April 2000.

[8] E. Riedel, G A. Gibson, and C. Faloutsos. Active Storage For Large-Scale Data Mining and Multimedia. In *Proceedings of the 24th international Conference on Very Large Databases (VLDB '98), New York, NY*, August 1998.

[9] Storage Networking: The Evolution of Information Management. Technical report, Seagate Technology, Inc., Available online as http://www.seagate.com/corp/vpr/literature/papers/sn.shtml, 1998.

[10] P. Shenoy, P. Goyal, and H M. Vin. Architectural Considerations for Next Generation File Systems. In *Proceedings of the Seventh ACM Multimedia Conference, Orlando, FL*, November 1999.

[11] P J. Shenoy, P. Goyal, S S. Rao, and H M. Vin. Symphony: An Integrated Multimedia File System. In *Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking (MMCN'98), San Jose, CA*, pages 124–138, January 1998.

[12] The Intelligent Storage Network. Sun Microsystems, Inc., http://www.sun.com/storage/vision, 1998.

[13] R Y. Wang, T E. Anderson, and D A. Patterson. Virtual Log Based File Systems for a Programmable Disk. In *Proceedings of third symposium on Operating systems design and implementation (OSDI), New Orleans, LA*, pages 29–43, Feb 1999.

[14] J. Wilkes, R. Golding, T. Sullivan, and C. Staelin. Introduction to the Storage Systems Program, Hewlett-Packard Laboratories. Available online from http://www.nsic.org/nasd/technol.html, August 1995.