File Systems: State of the Art

Florin Isaila

Department of Computer Science
University of Karlsruhe
florin@ipd.uni-karlsruhe.de

Abstract. File systems are parts of the operating systems that manage the storage devices. The storage devices may be either computer-attached or network-attached. The devices are organized in linear files. A logical file name space and a protection mechanism are offered to the applications.
In this paper we provide an overview of the state of the art in file systems. We present issues related to both local and distributed file systems. We describe distributed file system architectures for different kinds of network connectivity: tightly connected networks (clusters), loosely connected networks (computational grids) or disconnected networks(occurring in mobile computing). File systems architectures for both network-attached and computer-attached storage are reviewed. We show how the parallel file systems address the parallel application I/O requirements. We explore different file sharing semantics in distributed and parallel file systems. We present how efficient metadata management can be realized in journalled file systems.

1 Introduction

The ever increasing gap between processor and memory speeds on one side and disk systems on the other side have exposed the I/O subsystems as a bottleneck for the applications with intensive I/O requirements. Consequently, file systems, as low-level managers of storage resources, have to offer flexible and efficient services in order to allow a high utilization of disks.

Local file systems manage the storage devices attached to a particular computer. They organize the disk in linear, non-overlapping files. The file system provides also a logical name space, typically represented as a tree. As they are often the building blocks of distributed and parallel file systems, this chapter will give a short introduction to local file systems.

Distributed file systems store files on one or several machines. They provide users with a coherent, shared access to the same set of files. Two basic architectures have emerged: client-server [SG85,BR98] and server-less[AD95,FO95,TM97]. A file server is typically a process that offers a file service to a collection of clients. In a server-less file system, all the machines are equal peers that cooperate for providing applications a file service. The paper will compare and contrast this architectures and show the impact each of them has on performance and scalability.

Efficient and simultaneous access to network-attached storage devices is usually the target of*shared file systems*[OK98]. For instance, Fibre Channel offers an open standard in network-attached storage interfaces. We will show how shared network storage influences the architecture of file systems.

High-performance distributed computing extended lately from tightly-connected supercomputers or clusters to computational grids composed of heterogenous systems, spread over large geographical areas. *Grid file systems*[OK01] manage resources spread over several administrative domains. They have to deal with unpredictable performance variations and with changeable system architecture.

Another important trend in high-performance computing is the increasing usage of mobile devices. Mobile devices have to address changing network characteristics ranging from high connectivity to no connectivity. For instance, the file system has to be able to adapt to variable bandwidths[BR98]. Data from disconnected computers has to be consistently reintegrated in the system after reconnection.

For scientific parallel computing, the distributed file systems have to specialize in order to expose the parallelism of the I/O subsystem to the applications. Several studies of parallel I/O access pattern of scientific applications [NK+96,SR97,SR98] have shown that the lack of parallelism inside the I/O subsystem drastically affects the performance. Therefore, *parallel file systems*[DC92,CF96,CL00,HE95,IT01] allow the parallel access to files. Files are striped over several independent devices and therefore, parallel accesses to files might translate to requests to parallel operating disks.

For an efficient sharing of files, distributed and parallel file systems must allow to user nodes the local caching of file blocks. Consequently, cache consistency has to be addressed. UNIX semantics of file sharing, that guarantees that each file modification is instantly visible to all processor, may be very costly in a distributed environment. Therefore, we will show how distributed and parallel file systems use relaxed semantics in order to reduce the overhead of file sharing.

Another important issue is how the file systems manage file metadata. In this respect, there is an increasing number of file system that use the journaling technique [BE00,PB00,SG00]. *Journaling* is typically implemented by using a log that records all metadata operations. The log and the data have to be updated on the disks such that recoverability is guaranteed in case of failure.

In the remainder of the paper we will present an overview of different local, distributed and parallel file systems, as outlined in this section. We will use mainly the following basic criteria for characterization:

1. **File distribution:** Local or Distributed File System.
2. **Network connectivity:** Tightly-connected, loosely-connected or disconnected networks.
3. **Storage attachment:** Network-attached or computer-attached storage.
4. **Resource heterogeneity:** Does the file system assume that resources (disks, networks) are heterogenous or homogeneous?
5. **Parallel file access:** This attribute tells if parallel file access may translate into parallel disk accesses and if this behavior is controllable by applications or is system dependent.
6. **Semantics of file sharing:** Unix semantics or other relaxed semantics.
7. **Physical File Placement Control:** Is the file placement controllable by applications or is fixed by the system?
8. **Caching/Prefetching:** The kind of caching and prefetching used by the system.
9. **Scalability:** We will analyze if the system is scalable in terms of both data and metadata.
10. **Metadata management:** Journaled or non-journaled file systems.

References

[AD95] Tom Anderson, Michael Dahlin, Jeanna Neefe, David Patterson, Drew Roselli, Randy Wang. Serverless Network File Systems. 15th Symposium on Operating Systems Principles, ACM Transactions on Computer Systems , 1995.

[BR98] Braam, P. J. Linux Journal, Number 50 June 1998
[BE00] Steve Best. JFS overview. January 2000.
[CL00] Philip H. Carns, Walter. B. Ligon III, Robert B. Ross, and Rajeev Thakur. PVFS: A Parallel File System For Linux Clusters. In Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, GA, October 2000.
[CF96] Peter F. Corbett, Dror G. Feitelson. The Vesta Parallel File System. ACM Transactions on Computer Systems, 14(3):225-264, August 1996
[DC92] Erik DeBenedictis, Juan Miguel De Rosario. nCUBE Parallel I/O Software. In Proceedinfs of 11th International Phoenix Conference on Computers and Communication, April 1992.
[FC+95] Dror G. Feitelson, Peter F. Corbett, Yarsun Hsu, Jean-Pierre Prost. Parallel I/O Systems and Interfaces for Parallel Computers. In Multiprocessor Systems - Design and Integration C.L.Wu(Ed.), World Scientific Publication Co., 1995.
[FO95] John H. Hartman and John K. Ousterhout. "The Zebra Striped Network File System" , ACM Transactions on Computer Systems 13, 3, August 1995, 279-310.
[HE95] James V. Huber, Cristopher L. Elford, Daniel A. Reed, Andrew A. Chien, David S. Blumenthal, PPFS: A High Performance Portable File System. In Proceedings of of the 9th ACM International Conference on Supercomputing, Barcelona 1995.
[IT01] Florin Isaila, Walter F. Tichy, Clusterfile: A Flexible Physical Layout Parallel File System. In Proceedings of of the Third IEEE International Conference on Cluster Computing, Newport Beach, Oct. 2001.
[MPI97] Message Passing Interface Forum. MPI2: Extensions to the Message Passing Interface. July, 1997.
[NK+96] Nils Nieuwejaar, David Kotz, Apratim Purakayastha, Carla Schlatter Ellis, Michael L. Best. File Access Characteristics of Parallel Scientific Workloads. IEEE Transactions on Parallel and Distributed Systems, 7(10), October 1996.
[NK97] Nils Nieuwejaar and David Kotz. The Galley Parallel File System. Parallel Computing, 23(4), June 1997.
[OK98] Matthew T. O'Keefe. Shared File Systems and Fibre Channel. In the Proceedings of the Sixth NASA Goddard Space Flight Center Conference on Mass Storage Systems and Technologies, March 23-26, 1998, College Park, MD.
[OK01] Ron Oldfield and David Kotz. Armada: A parallel file system for computational grids. In Proceedings of the IEEE/ACM International Symposium on Cluster Computing and the Grid, pages 194-201, Brisbane, Australia, May 2001.
[PB00] Kenneth W. Preslan, Andrew P. Barry, Jonathan Brassow, Russell Cattelan, Adam Manthei, Erling Nygaard, Seth Van Oort, David Teigland, Mike Tilstra, Matthew O'Keefe, Grant Erickson, Manish Agarwal: Implementing Journaling in a Linux Shared Disk File System. IEEE Symposium on Mass Storage Systems 2000: 351-378
[SG85] R. Sandberg, D. Goldberg, S. Kleinman, D. Walsh, and B. Lyon. Design and Implementation of the Sun Network Filesystem. In Proc. of the Summer 1985 USENIX, pages 119-130, June 1985.
[SG00] Margo I. Seltzer, Gregory R. Ganger, M. Kirk McKusick, Keith A. Smith, Craig A.N. Soules, and Christopher A. Stein. Journaling versus Soft Updates: Asynchronous Meta-data Protection in File Systems. In the Proceedings of the 2000 USENIX Annual Technical Conference, June 2000, pages 71-84.
[SR97] Evgenia Smirni and Daniel A. Reed. Workload Characterization of I/O Intensive Parallel Applications. Proceedings of the Conference on Modelling Techniques and Tools for Computer Performance Evaluation,Springer-Verlag Lecture Notes in Computer Science, June 1997.

[SR98] Huseyin Simitici and Daniel A. Reed A Comparison of Logical and Physical Parallel I/O Patterns. International Journal of High Performance Computing Applications, special issue (I/O in Parallel Applications), 12(3), 1998.

[TM97] C. A. Thekkath, T. Mann, and E. K. Lee. Frangipani: A scalable distributed file system. In Proceedings of the 16th ACM Symposium on Operating System Principles, 1997.