Cluster File System Topologies

Paul Taysom and Robert Wipfel
paul_taysom@novell.com, rawipfel@novell.com
July 19, 2005 : December 9, 2004

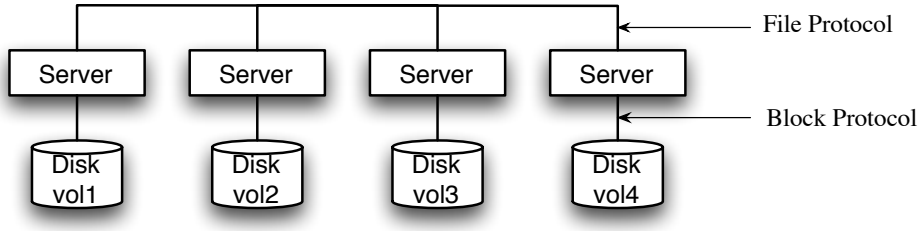
. . . the tendency underlying all the evil of our technology, the tendency to do what is "reasonable" even when is isn't any good. Pirsig

The topology of a cluster file system describes both the physical (disks, servers, network, *etc.*) and logical (volumes, file systems, *etc.*) layout.

Simplex Cluster File System

Each server is directly connected to a disk with its own set volumes. Because you can make a single name space using NFS or junctions this is also called a directory cluster. This architecture can be placed on Multiplex Cluster (see below) and you get the type of cluster Novell sells today.

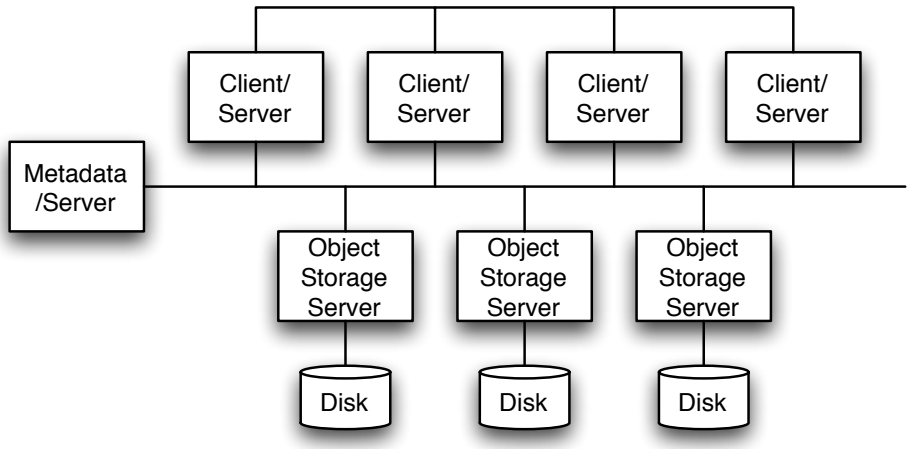
Examples: NFS and NetWare.



Split Meta/Data CFS

The user data is spread across multiple servers but the meta-data is controlled by a single server. The client/server merges the two separate views into a single file system.

Examples: Lustre, NASD, Ibirx, and Microsoft Tiger.



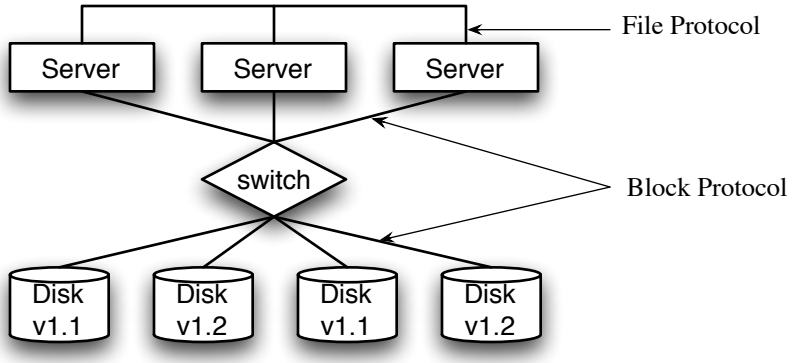
Multiplex Cluster File System

Each server is connected to every disk by a switch. The disk volumes are partitioned across the disk. This partitioning can be done by the disk raid controllers, the switch, or the volume manager in the server. Servers share access to the same volume and can directly write to each volume. This control can be either asymmetrical or symmetrical. In asymmetrical control, one server controls the volume with the others being slave. In symmetrical control, all servers control the volume with a distributed lock manager.

The file system can assume the underlying storage is reliable (durable).

Asymmetrical Examples: C-VxFs, Solaris CFS, and OpenSSI-CFS

Symmetrical Examples: OCFS-2, GPFS, GFS, and PolyServe Matrix.

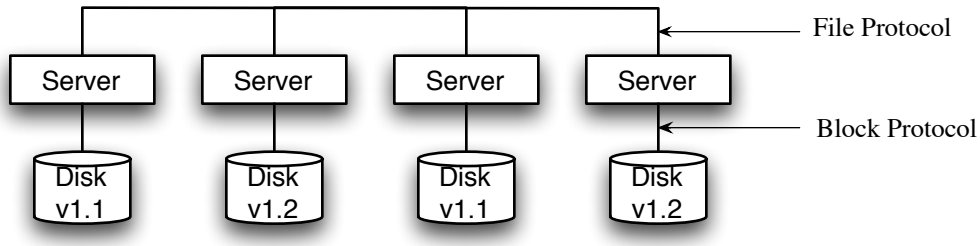


Omniplex CFS

The file system spreads and replicates the volume across multiple servers. The file system detects when a server fails and then uses copies on other servers to reconstruct the failed servers data.

The file system assumes the underlying storage will fail.

Examples: Google FS, Inktomi, and Berkeley FSX.



Cached CFS

The file system is stored on one reliable system and performance is achieved by caching the file system locally on each cache server.