THE DIRECT ACCESS FILE SYSTEM

The Direct Access File System (DAFS)

The Direct Access File System (DAFS) is a newly developed network file system that is tailored to the use of RDMA. It is based upon NFS version 4, requires VI and can fully utilise its new possibilities. DAFS makes it possible for several DAFS servers together to provide the storage space for a large file system (Figure 4.8). It remains hidden from the application server – as the DAFS client – which of these DAFS servers the actual data is located in (Figure 4.9).

The communication between DAFS client and DAFS server generally takes place by means of RDMA. The use of RDMA means that access to data that lies upon a DAFS server is nearly as quick as access to local data. In addition, typical file system operations such as the address conversion of files to SCSI block addresses, which naturally also require the CPU, are offloaded from the application server to the DAFS server.

An important function of file sharing in general is the synchronisation of concurrent accesses to file entries – i.e. metadata such as file names, access rights, etc. – and file contents, in order to protect the consistency of the data and metadata. DAFS makes it possible to cache the locks at the client side so that a subsequent access to the same data requires no interaction with the file server. If a node requires the lock entry of a different node, then this transmits the entry without time-out. DAFS uses lease-based locking in order to avoid the permanent blocking of a file due to the failure of a client.

Furthermore, it possesses recovery mechanisms in case the connection between DAFS client and DAFS server is briefly interrupted or a different server from the cluster has to step in. Similarly, DAFS takes over the authentication of client and server and furthermore can also authenticate individual users in relation to a client-server session.

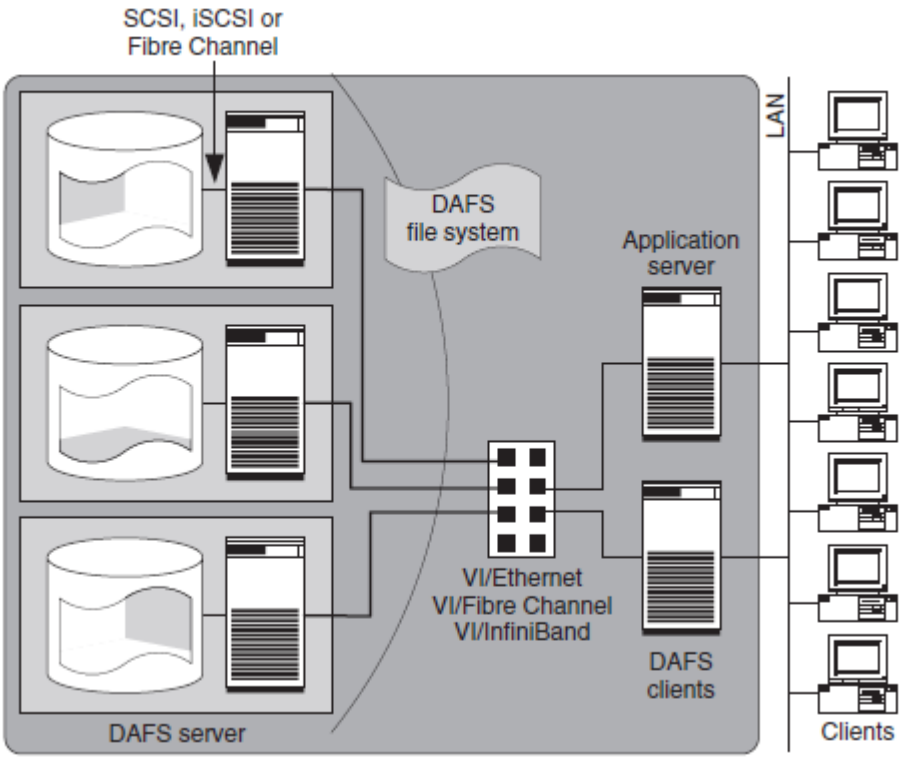

Figure 5.8 A DAFS file system can extend over several DAFS servers. All DAFS servers and DAFS clients are connected via a VI-capable network such as InfiniBand, Fibre Channel or Ethernet.

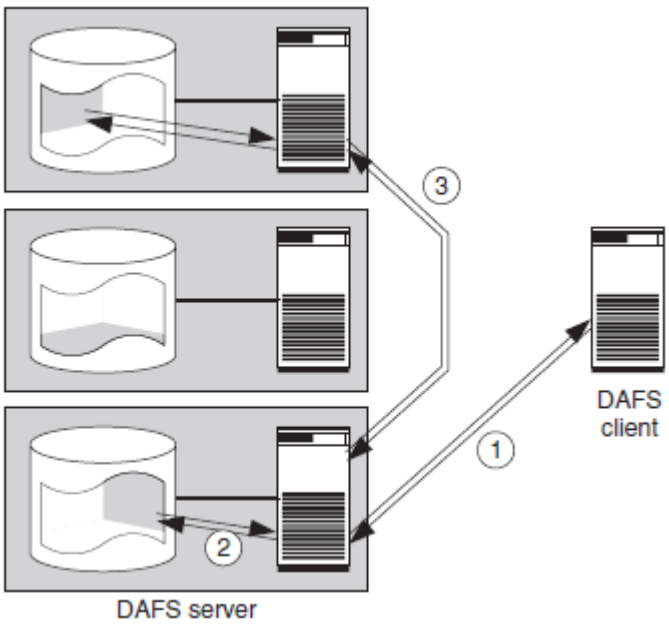

Figure 5.9 The DAFS client communicates with just one DAFS server (1). This processes file access, the blocks of which it manages itself (2). In the case of data that lies on a different DAFS server, the DAFS server forwards the storage access to the corresponding DAFS server, with this remaining hidden from the DAFS client (3).

Two approaches prevail in the discussion about the client implementation. It can either be implemented as a shared library (Unix) or Dynamic Link Library (DLL) (Windows) in the user space or as a kernel module (Figure 4.10). In the user space variant – known as uDAFS – the DAFS library instructs the kernel to set up an exclusive end-to-end connection with the DAFS server for each system call (or for each API call under Windows) by means of a VI provider layer (VIPL), which is also realised as a library in user space. The VI-capable NIC (VI-NIC) guarantees the necessary protection against accesses or faults caused by other processes. The user space implementation can utilise the full potential of DAFS to increase the I/O performance because it completely circumvents the kernel.

It offers the application explicit control over the access of the NIC to its private storage area. Although control communication takes place between the VIPL in the user space and the VI-NIC driver in the kernel, the CPU cost that this entails can be disregarded due to the low data quantities.

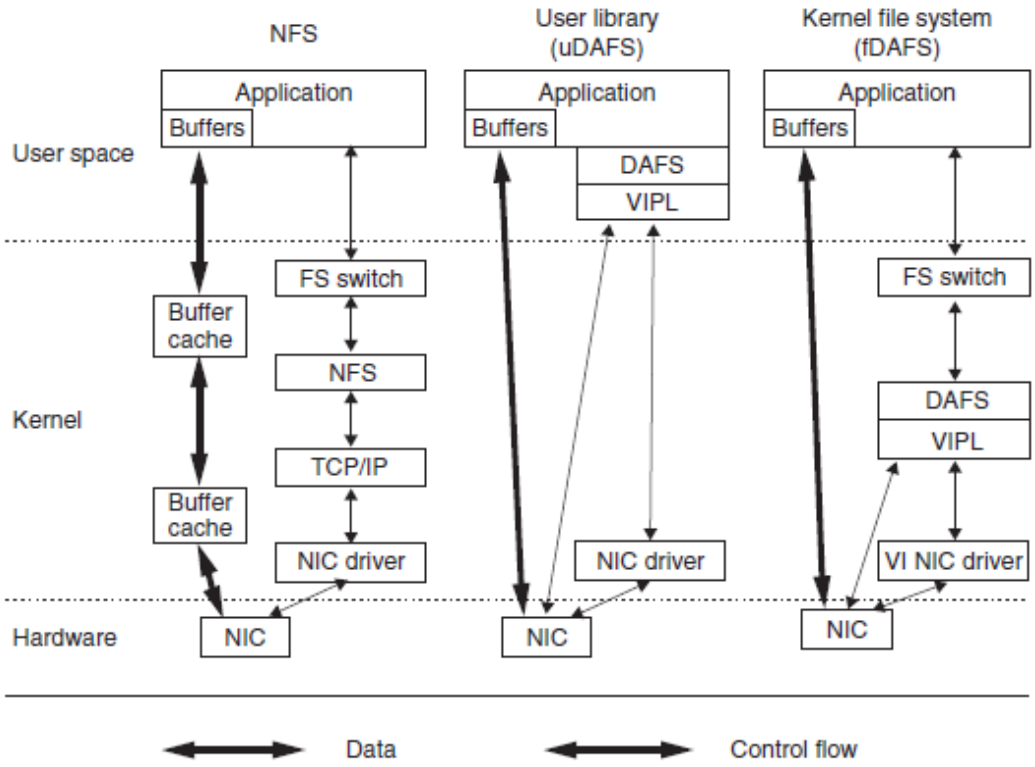

Figure 5.10 A comparison between NFS, uDAFS and fDAFS.

The disadvantage of the methods is the lack of compatibility with already existing applications, which without exception require an upgrade in order to use DAFS. Such a cost is only justified for applications for which a high I/O performance is critical. In the second, kernel-based variant the implementation is in the form of a loadable file system module (fDAFS) underneath the Virtual File System (VFS layer) for Unix or as an Installable File System (IFS) for Windows. Each application can address the file system driver as normal by means of the standard system calls and VFS or API calls and the I/O manager; DAFS then directs a query to the DAFS server. The I/O performance is slowed due to the fact that all file system accesses run via the VFS or the I/O manager because this requires additional process changes between user and kernel processes. On the other
hand, there is compatibility with all applications.

Some proponents of DAFS claim to have taken measurements in prototypes showing that data access over DAFS is quicker than data access on local hard disks. In our opinion this comparison is dubious. The DAFS server also has to store data to hard disks. We find it barely conceivable that disk access can be quicker on a DAFS server than on a conventional file server. Nevertheless, DAFS-capable NAS servers could potentially support I/O-intensive applications such as databases, batch processes or multi-media applications. Integration with RDMA makes it irrelevant whether the file accesses take place via a network. The separation between databases and DAFS servers even has the advantage that the address conversion of files to SCSI block addresses is offloaded from the database server to the DAFS server, thus reducing the load on the database server's CPU.
However, file servers and database servers will profit equally from InfiniBand, VI and RDMA. There is therefore the danger that a DAFS server will only be able to operate very few databases from the point of view of I/O, meaning that numerous DAFS servers may have to be installed. A corresponding number of DAFS-capable NAS servers could be installed comparatively quickly. However, the subsequent administrative effort could be considerably greater.

The development of DAFS was mainly driven by the company Network Appliance, a major NAS manufacturer. The standardisation of the DAFS protocol for communication between server and client and of the DAFS API for the use of file systems by applications took place under the umbrella of the DAFS Collaborative. Its website www.dafscollaborative can no longer be reached. Since the adoption of Version 1.0 in September 2001 (protocol) and November 2001 (API) the standardisation of DAFS has come to a standstill. Originally, DAFS was submitted as an Internet standard to the Internet Engineering Task Force (IETF) in September 2001; however, it has found very little support in the storage industry. Instead, widespread attention is being given to an alternative, namely an extension of NFS with RDMA DAFS is an interesting approach to the use of NAS servers as storage for I/O-intensive as a transport layer and the addition of DAFS-like lock semantics applications. Due to a lack of standardisation and widespread support across the industry, current DAFS offerings (2009) should be considered as a temporary solution for specific environments until alternatives like NFS over RDMA and CIFS over RDMA emerge. Furthermore, iSCSI is of interest to those who see DAFS as a way of avoiding an investment in

Fibre Channel, the more so because iSCSI – just like NFS and CIFS – can benefit from TOEs, the SDP and a direct mapping of iSCSI on RDMA (iSER) (Section 3.6.3). Shared-disk file systems (Section 4.3) also offer a solution for high-speed file sharing and in addition to that, storage virtualisation (Chapter 5) provides high-speed file sharing while it addresses the increasingly expensive management of storage and storage networks as well.

Source : http://elearningatria.files.wordpress.com/2013/10/cse-viii-storage-area-networks-06cs833-notes.pdf