PERFORMANCE BOTTLENECKS IN FILE SERVERS

Performance bottlenecks in file servers

Current NAS servers and NAS gateways, as well as classical file servers, provide their storage capacity via conventional network file systems such as NFS and CIFS or Internet

protocols such as FTP and HTTP. Although these may be suitable for classical file sharing, such protocols are not powerful enough for I/O-intensive applications such as databases or video processing. Nowadays, therefore, I/O-intensive databases draw their storage from disk subsystems rather than file servers.

Let us assume for a moment that a user wishes to read a file on an NFS client, which is stored on a NAS server with internal SCSI disks. The NAS server's operating system first of all loads the file into the main memory from the hard disk via the SCSI bus, the PCI bus and the system bus, only to forward it from there to the network card via the system bus and the PCI bus. The data is thus shovelled through the system bus and the PCI bus on the file server twice (Figure 4.7). If the load on a file server is high enough, its buses can thus become a performance bottleneck.

When using classical network file systems the data to be transported is additionally copied from the private storage area of the application into the buffer cache of the kernel

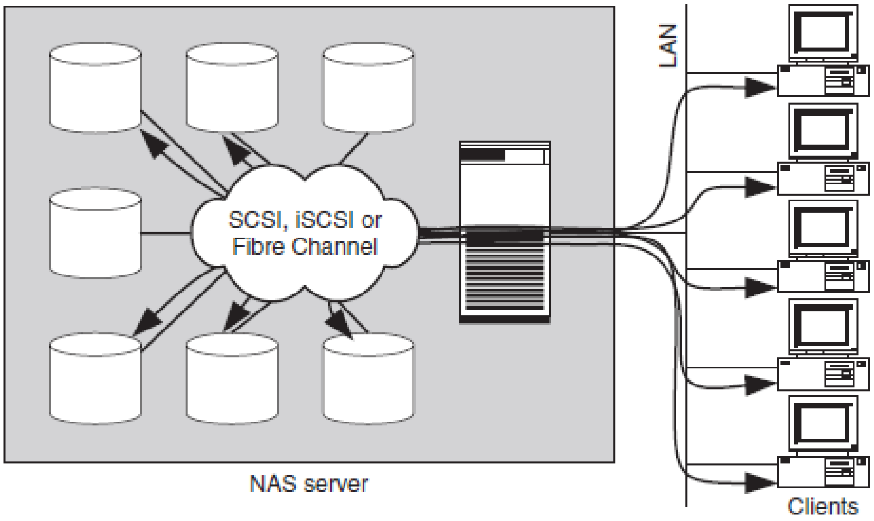

Figure 5.7 The file server becomes like the eye of the needle: en route between hard disk and client all data passes through the internal buses of the file server twice. on the transmitting computer before this copies the data via the PCI bus into the packet buffer of the network card. Every single copying operation increases the latency of the communication, the load on the CPU due to costly process changes between application processes and kernel processes, and the load on the system bus between CPU and main memory.

The file is then transferred from the network card to the NFS client via IP and Gigabit Ethernet. At the current state of technology most Ethernet cards can only handle a small part of the TCP/IP protocol independently, which means that the CPU itself has to handle the rest of the protocol. The communication from the Ethernet card to the CPU is initiated by means of interrupts. Taken together, this can cost a great deal of CPU time (Section 3.5.2, ‘TCP/IP and Ethernet as an I/O technology’).

Acceleration of network file systems

If we look at the I/O path from the application to the hard disks connected to a NAS server (Figure 4.15 on page 157), there are two places to start from to accelerate file sharing: (1) the underlying communication protocol (TCP/IP); and (2) the network file system (NFS, CIFS) itself. TCP/IP was originally developed to achieve reliable data exchange via unreliable transport routes. The TCP/IP protocol stack is correspondingly complex and CPU-intensive.

This can be improved first of all by so-called TCP/IP offload engines (TOEs), which in contrast to conventional network cards process a large part of the TCP/IP protocol stack on their own processor and thus significantly reduce the load on the server CPU (Section 3.5.2). It would be even better to get rid of TCP/IP all together. This is where communication techniques such as VIs and RDMA come into play (Section 3.6.2). Today there are various approaches for accelerating network file systems with VI and RDMA. The Socket Direct Protocol (SDP) represents an approach which combines the benefits of TOEs and RDMA-enabled transport (Section 3.6.3). Hence, protocols based on TCP/IP such as NFS and CIFS can – without modification – benefit via SDP from RDMA-enabled transport. Other approaches map existing network file systems directly onto RDMA. For example, a subgroup of the Storage Networking Industry Association (SNIA) is working on the protocol mapping of NFS on RDMA. Likewise, it would also be feasible for Microsoft to develop a CIFS implementation that uses RDMA instead of TCP/IP as the communication protocol. The advantage of this approach is that the network file systems NFS or CIFS that have matured over the years merely have a new communication mechanism put underneath them. This makes it possible to shorten the development and testing cycle so that the quality requirements of production environments can be fulfilled comparatively quickly.

A greater step is represented by newly developed network file systems, which from the start require a reliable network connection, such as the DAFS (Section 4.2.5), the family of the so-called shared disk file systems (Section 4.3) and the virtualisation on the file-level (Section 5.5).

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