Design of High Performance Disk Storage Protocol

SHEN Jinxing

School of Computer Science,

Guangdong Polytechnic Normal University, Guangzhou, 510665, China

E-mail:jx_shen@sohu.com

Key words：Storage System；Transport Protocol；Zero-copy；High Performance

Abstract. This paper introduces a light weight transport protocol which is referred to High performance Disk Storage Protocol（HDSP）. Like the zero-copy technology，HDSP improves the throughput of networked storage system and reduces the system workload by reducing protocol layers and data copy times，and employing efficient flow control. The test result shows that HDSP has better performance than TCP/IP in Ethernet storage environments and is able to overcome the flaw of TCP/IP due to multiple copies and context switching.

Introduction

The explosive growth of information makes the data storage requirements increase rapidly. The traditional direct attached-storage DAS has been gradually replaced by networked storage system NAS/SAN [1]. In traditional TCP/IP communications, a data has to go through several memory copies and switch frequently between operating system kernel mode and user mode during transmission, which makes the overhead to process TCP/IP protocol too large and adds on excessive workload to networked storage system. Processing TCP/IP protocol is thus becoming a bottleneck for networked storage system.

HyperSCSI[2] and AOE[3] protocols are two replacements of TCP/IP protocol for data transmission in LAN (Local Area Network) storage environments. The former can be applied to SCSI hard disk storage, while the later supports ATA disks. However, during data transmission, these two protocols still have to copy memory for several times. That is, the overhead of process has not been fundamentally improved. Therefore, it is practically significant to design a lightweight transport protocol to replace traditional TCP/IP protocol.

High-speed disk storage protocols (High-performance Disk Storage Protocol, HSDP), which uses a variety of techniques to improve the throughput of the networked storage system and reduce server load, is a lightweight data transmission protocol for LAN storage system.

HDSP Protocol Design Principles

When transferring data by TCP/IP protocol, the system has to do CPU copy three times and context switch twice in order to transfer the data received on network card into disk. Generally the data will go through the system bus twice during single CPU copy. As a result, in a process of receiving or sending data, the data will go through the memory bus in system as many as 6 times. That is, the memory bus of system has to provide at least 6 times the bandwidth of the network data stream [4]. Thus, the overhead on frequent memory copies and context switching is the bottleneck of overall system performance.

In fact, in networked storage systems, especially on the server side, the data usually does not have to interact on user layer. Taking design of zero-copy technology [5] as reference, HDSP protocol establishes a direct link between network card and disk in kernel mode. Therefore, the network card is able to directly access HDSP buffer zone through DMA (direct memory access). HDSP protocol will then read the data from buffer zone and transfer it into disk buffer. In the end, the system manages to complete the transmission from network card to disk through only one CPU copy and no more context switch.

The letter D in HDSP refers to Disk, which can be ATA, SATA or SC-SI disk. HDSP protocol is designed to provide remote access to disk in LAN. In this transmission system, data in disk can be accessed by transferring disk instructions between the host and Ethernet hard disk drives. So the traditional disk data cable can be replaced by Ethernet cable. HDSP by Pass file system, in the form of a block device, is able to access data directly through the address in disk sector, which provides a storage solution based on the 'block transmission'. HDSP is a kind of protocol located in the link layer, so that it provides reliable connection-oriented data transmission services, which transfers data efficiently in blocks by using flow rate control mechanism in combination of window and rate. With the help of HDSP protocol, the disks in LAN can also be self-organized to be a reliable networked storage system with more capacity.

HDSP Protocol Design

HDSP protocol is designed for Ethernet. As shown in Figure 1, DA and SA are destination address and source address respectively, carried by standard Ethernet frame. Ether-type is the type of data package carried by Ethernet frame. HDSP protocol packet (HDSP Packet) is encapsulated in an Ethernet frame for transmission. According to the provisions, if value of Ether-type is 0X6666, the Ethernet frame is carrying a HDSP protocol packet.

There are 2 kinds of HDSP protocol packets. One is data packet called PDU (Packet Data Unit), and the other one is control packet called PCU (Packet Control Unit). The PDU contains the specified data to be transmitted, while the PCU carries all the controls for connection management and confirmation. PDU and PCU are sharing the same HDSP Baotou (HDSP Header).

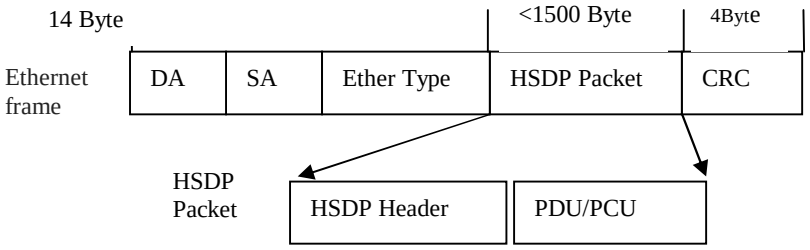

Figure 1 HSDP Packet and Ethernet frame

HDSP protocol common header

HDSP protocol common Header is shown in Figure 2. Ver. Refers to the version of HDSP protocol. Flag is a 4-digit field, and the last 2 digits (Z bits) are reserved by system. When the client is sending a request to server, R bit is cleared as 0. When the packet is sent by the server, the R bit is set as 1. Therefore, the server will only answer the packet with R bit as 0 while the client will only answer the packet with R bit as 1. The rule ensures that the server and client are able to process the correct packets in LAN accordingly. When E bit is set as 1, the server is not available to the client's request for some reason and Error field will show the specific error code. The value in PType field indicates whether the HDSP packet is PDU or PCU. DType field indicates the type of server disk. Disk Address field contains the address of server disk. HDSP protocol provides connection-oriented service, and Server Port along with Client Port in the header are used to identify the connection. Thus, with specific MAC address, disk address and port number, a disk in HDSP server can be accessed by any node in the network.

HDSP protocol packet header

HDSP protocol is able to encapsulate different disk instructions including ATA, SATA, and ISCSI. Figure 3 shows an example of SATA Command packet. When the W bit in DFlag field is set as 1, the system is supposed to write on SATA disk, which means this packet is delivering information as 'write'. Feature field and CMD field store the information to be written into flag-register and command-register in SATA Controller. In the same way, it means 'read' when W bit is 0. Status field and Err field carry the information taken from status-register and error-register in SATA Controller. E

(Extend) bit indicates whether the packet supports addressing mode of extended disk sector LBA (Logical Block Addressing). When the E bit is cleared, LBA0, LBA2 and LBA4, these three fields carry a 24-bit LBA address. While when E bit is 1, LBA0 to LBA5, these six fields carry a 48-bit LBA address. Sector Count Field indicates the number of sectors to be accessed. From the 28th bit, Data field fulfills the data in disk sector during actual transmission.

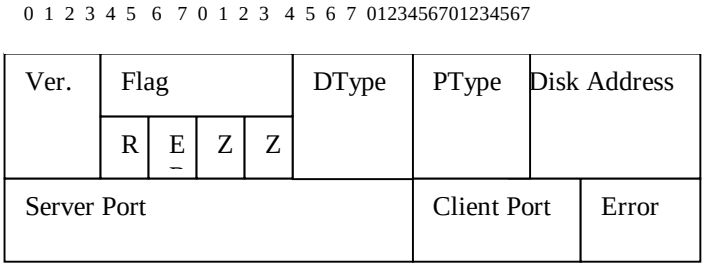

Figure 2 HSDP Packet

Before transmission, HDSP segments data into a series of groups. Group ID field represents the group number and Serial Number field contains the packet's sequence number within a group. Receiver will sort packets according to their group number and serial number, and restructure the data by making sure all packets in correct order and without repetition. The S bit in DFlag field represents synchronous. When multiple clients are trying to access the same disk in a server simultaneously, the disk has low throughput and has to waste a lot of time on tedious head seek, due to the random address of accessed data. In HDSP protocol, when client write data into sector on a server disk in sequence, S bit is supposed to be cleared, indicating there is other packet to be sent to this sector. Server usually returns to next packet immediately after it puts a packet into memory and it will transfer all the data in memory to disk, once it finds next packet with S bit as 1 or there has been as many as M packets in the memory. Thus, the server is able to transfer as much data as possible into multiple sectors in sequence with only one head seek, which significantly improves the time cost of head seeking.

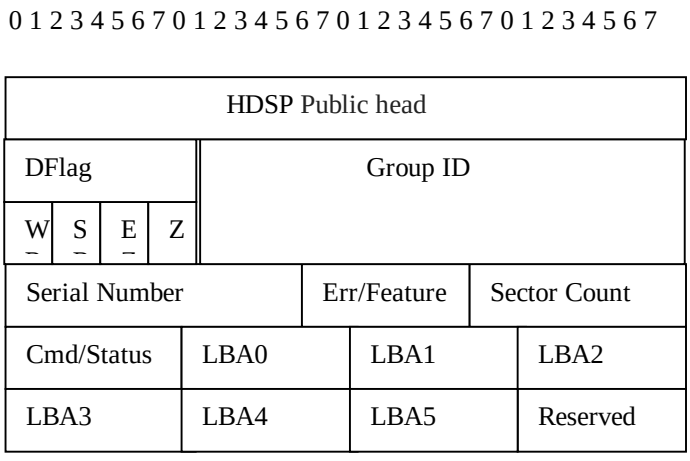

Figure 3 HSDP Protocol SATA Command packets

HDSP protocol control packet and connection control

HDSP's control packet encapsulates all the information for control, query, diagnose and other operations of connection. The format is shown in Figure 4. The first 8 bytes are the common header of protocol and all operation instructions are saved in Operation Class field. To be noted, the length of data carried by instructions can vary from each other and length is saved in Operation Code Length field. The specific instruction is saved in Operation Code field, which starts from 16th bit.

HDSP builds up connections by using three-way handshake mechanism. In that process, both sides have to complete the identity authentication and negotiation of transmission parameter, such as the transmission window size W, data packet size N, etc. The connection can be released by either side.

0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7

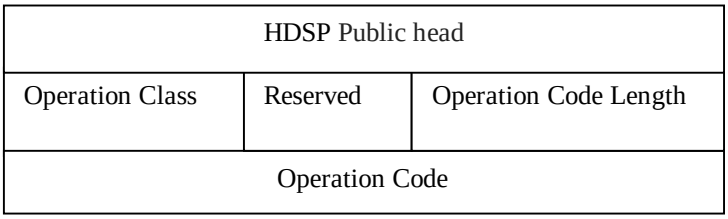

Figure 4 HSDP Protocol control packet

data transfer and flow control

HDSP improves bandwidth utilization and reduces congestion by controlling both window and rate. There is few message error in LAN and the errors are nearly negligible because of LAN's access control policies and verification policies. So it is not necessary to use great amount of ACK signals.

HDSP divides data into series of groups during transmission and each group includes as many as N packets. The receiver has two kinds of verification, which are ACK and NACK. ACK verifies data by a cumulative method to inform the sender that it has correctly received the data with largest group number (all groups before that have been received correctly). NACK, which works in another way, will inform the sender which packet is not received correctly. And the sender uses a strategy of selective retransmission, only resending the missing packets.

The sender maintains a send window with size of W, which can send W packets continuously before receipt of ACK from any group. Every time a packet is sent out, W bit will be decremented by 1. When W bit is 0, the sender will stop sending data groups until an ACK is received and W bit becomes non-zero. Parameter N and W, which are set during connection build-up, can be adjusted dynamically according to network condition, receiver overload and other circumstances.

A Copy of the Data Path Mechanism Design

Assuming the size of network card's receiving ring RX_Ring is N, the driver will look for N free blocks in idle queue FreeQ and write the address of these blocks into EX_Ring. In the meantime, the driver will move the descriptors of these blocks from FreeQ to ReceiveQ. Besides, block_addr points to the physical address of a block and pkt_len stores the length of an Ethernet frame.

When a complete Ethernet frame arrives at the storage unit inside network card, DMA control module will firstly write the Ethernet frame buffer into a buffer block pointed by RX_Ring (the descriptor of this block is already in ReceiveQ), and then execute a hardware interrupt.

The program, which receives interrupt signal, will fulfill pkt_len field in block_des1 with the length of the Ethernet frame. After that, it takes another descriptor block_des2 from FreeQ and puts this descriptor at the end of ReceiveQ. In addition, the physical address of this block will be written into RX_Ring in network card. This address will be the loop address for next DMA cycle, which is the destination address for the N+1 DMA transmission. Then, the program has to wake up the stuck process HDSP_Listen, which is waiting for generation of new Ethernal frame. HDSP_listen process will distribute descriptor block_desc of HDSP protocol frame (EtherType = 0X6666) in ReceiveQ to HDSP protocol process with corresponding port number for further processing. If the frame contains a packet with request (PDU), HDSP protocol process writes data into disk buffer according to given sector address (copied by CPU). For Ethernet frames with different protocols, Listen process sends a copy to the protocol stack in upper layer for processing, so that joint of HDSP does not affect the

normal operation of other protocols. Descriptor block_desc in ReceiveQ will be sent to FreeQ for recycle after process.

Thus, CPU only has to complete one copy from ReceiveQ to disk buffer in the course of HDSP writing a received Ethernet frame into disk. Conversely, from the disk to the card, CPU only has to copy once from disk buffer to SendQ. Furthermore, the data path for copy by HDSP is implemented in the system kernel, so there is no more overhead on context switching between user mode and kernel mode.

HDSP Protocol Performance Analyses

The test platform consists of two identical PCs with AMD Athlon4000+CPU, 1 GB of memory and Realtek 8169 1000 MB / s Ethernet card. The two machines are connected directly via a crossover cable. HDSP is realized in Fedora 10 operating system (kernel as Linux 2.6.27.5) after modifying the built-in driver of Reaktak 8169 to connect DMA module to HDSP buffer directly. HDSP module has to be installed on both server and client. A test partition in server disk can be virtualized to be a local partition for client. Bonnie++ is supposed to test the partition's read/write performance and SAR will be used to sample load of server CPU. Also, another process by NFC protocol is in need to compare the performance between TCP and HDSP.

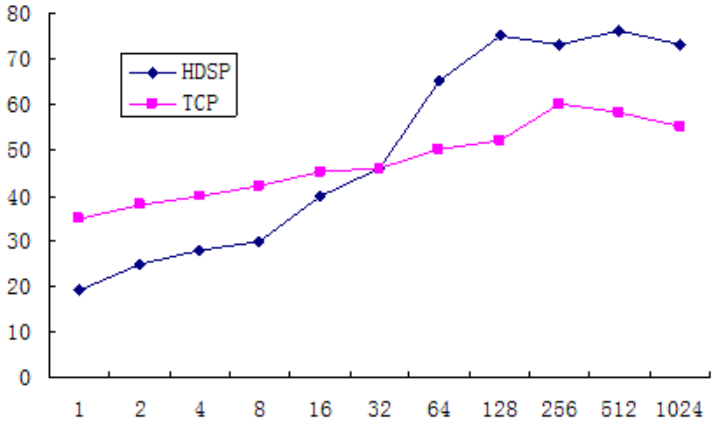

Figure 5 HDSP and TCP Throughput comparison chart

Figure 5 shows a sequence throughput by server of random data blocks with size of 1 KB, 2 KB, 4 KB, 8 KB ... 1 MB. Figure 6 implies CPU load on the server side. As a comparison, both figures include the result of TCP protocol. As can be seen in Figure 5, HDSP is not as good as TCP when processing smaller blocks. As the size of data block increases, HDSP's performance improves rapidly. When the data size is larger than 32 KB, the throughput of HDSP is 21% higher than TCP's on average.

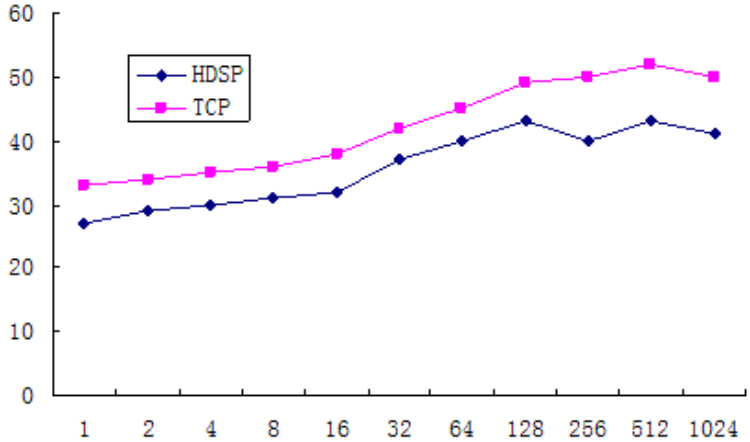

Figure 6 Server CPU Load comparison chart

Server CPU load trend in Figure 6 is similar to the throughput trend in Figure 5. As the size of data block and throughout by server increases CPU utilization of both HDSP and TCP increase as well. CPU utilization rate of HDSP is 16% lower than TCP's on average.

Conclusions

A high-speed disk storage protocol HDSP is designed to improve the poor performance of networked storage system, which is caused by huge overhead when processing TCP/IP protocol. HDSP protocol is fundamentally optimized by reducing the number of copies, streamlining the protocol layers and applying efficient flow control mechanism. The test result shows that, compared to TCP/IP protocol, HDSP not only improves throughput but also reduces system load, which makes it more suitable for networked storage system with large data transmission in LAN network environment.

References

[1] Chase J S Gallatin A J Yocum K G.End system optimizations for high-speed TCP [J].IEEE Communications Magazine，2001:68-74.

[2] Yong W，Wang Hong，Yeo H N，et al. Design and development of Ethernet-based storage area network protocol [J].Computer Communications，2006，29：1271-1283.

[3] Coraid Inc. The ATA over Ethernet protocol [EB/OL]. 2009, http://www.coraid.com.

[4] Wang Sheng, Su Jinshu. The reviews of acceleration technology research on TCP. [J] Journal of Software, 2004, 15 (11): 1689-1699.

[5] Wang Boling, Bin-Xing Fang, Yun Xiaochun. Study and implementation of zero-copy capture platform. [J] Journal of Computers, 2005, 28 (1): 46-52.