Cray Connectivity to ESCON-attached Storage Devices at NCAR

John Merrill, National Center for Atmospheric Research, Boulder, CO

ABSTRACT: *With the advent of ESCON-attached storage devices, capable of much higher transfer rates than the older FIPS-60 devices, the Mass Storage Group at the National Center for Atmospheric Research (NCAR) was faced with the problem of how to incorporate these devices into the NCAR Mass Storage System (MSS) (1). The new devices were going to have to be accessible from all MSS host machines, which included several Crays. Since all MSS hosts were already HIPPI-attached, the solution was to bring in a special adapter from Network Systems Corporation (NSC) which would provide HIPPI to ESCON connectivity. This paper examines some of the problems encountered in the initial testing of the new data path, and looks at some of the results and data rates achieved using several different Cray machines.*

INTRODUCTION

The current Mass Storage System at NCAR, known as MSS-III, dates back to the mid 1980's. The design of this system (1) centered around the concept of a centralized Mass Storage Control Processor (MSCP) which would control all aspects of the system, including the Master File Directory (MFD), resource allocation, and internal data migration. All MSS devices, however, would be accessible directly from the host machines via high speed data paths. This design allowed for the fastest possible data rates when transferring MSS files to/from the host machines.

In order to implement this design, all MSS devices had to be multi-ported, so that the MSCP could access the devices, as well as the remote host machines. Not only that, but there had to be a way for many remote host machines to be able to access the same MSS devices, although not at the same time. The resulting implementation made use of a "Local Data Network" (LDN), with locally developed host software to control data movement to/from the MSS. The system was initially developed under the Cray Operating System (COS) running on a CRAY-1A, and also a CRAY X-MP/48. Modifications were made to the host software in 1989 to convert the system to UNICOS, running initially on a CRAY X-MP/18. The conversion to UNICOS is described in detail in a paper presented at the Tenth IEEE Symposium on Mass Storage Systems (2).

The system has gone through one major evolutionary change since the initial implementation, which was the conversion of the LDN from NSC HYPERchannel to a High-Performance Parallel Interface (HIPPI) datafabric, which is now known as the "High-Performance Data Fabric" (HPDF). The MSS devices, however, were still all FIPS-60 devices, with a maximum transfer rate of 3.0 MBytes/sec (slightly higher rates could be achieved using compression, coupled with a 4.5 MByte/sec control unit). What was needed was a faster device, with a larger capacity, which could be incorporated into the system with as few changes to existing software as possible. This paper will examine the results of evaluating ESCON-attached storage devices as one possible solution.

WHY ALL THIS LOCALLY DEVELOPED CODE?

The development of NCAR's MSS over the years has always been a process of getting the most out of existing hardware devices and data paths, using whatever tools were available at the time. This always seemed to lead to developing a fair amount of local code to get the job done. Off-the-shelf solutions never seemed to be available to solve our data storage needs, so we were forced into the "do it yourself" mode more often than not.

One advantage of the "do it yourself" approach, though, is that you end up with a great deal of on-site expertise in the resulting systems, and making mods and upgrades is somewhat easier and more flexible. When new hardware and software come along, they can always be integrated into the existing system without too much difficulty, but again there is a requirement to do more local mods.

SEPARATION OF CONTROL AND DATA

One of the guiding principles of the NCAR MSS development effort has always been the separation of control and data. This principle is one that has been advocated by many people over the years, and has always played a part in the development of the IEEE Reference Model for Mass Storage Systems (3).

For the purposes of this paper, I will not go into detail on the control path, except to help in setting up the framework for the overall system. The NCAR MSS is a centralized system, with all devices, data paths, and other resources controlled by the Mass Storage Control Processor (MSCP).The MSCP also maintains the Master File Directory (MFD), which is the repository for all information pertaining to user files stored in the system. Figure 1 shows a simplified view of the NCAR MSS.

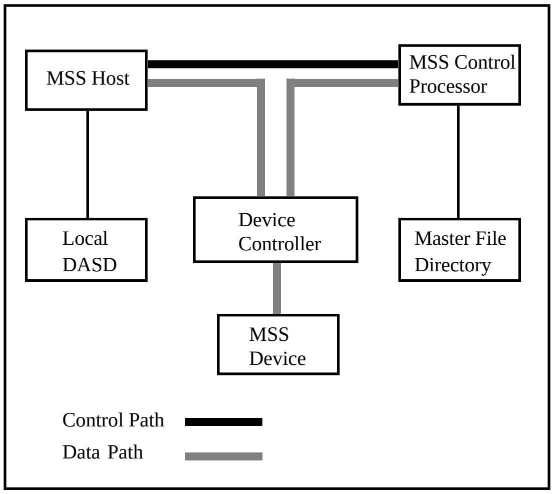

Fig. 1 - Simplified diagram of NCAR MSS

User access to MSS data is based on reading and writing entire MSS files, using special commands. MSS files are copied from local disk on the host machine to the MSS, and vice-versa, when a user invokes an mswrite or msread command. The device and data path involved in the transfer are allocated by the MSCP, and remain allocated until the host computer signals that it has completed the data transfer.

The MSCP needs to know enough about the host data path to be able to allocate resources, but it does not need to know the details of how the host computer actually transfers data to/from the various MSS devices. Also, the MSCP is not involved at all in the data transfer. This is one of the main advantages of the NCAR design -- it removes the MSCP as a potential bottleneck in the data path. This makes for a system which is much more scaleable, and allows more host machines to be added without having to upgrade the MSCP.

A CLOSER LOOK AT THE HOST DATA MOVER

Let's turn now to the real subject of this paper, which is the data movement portion of the system. Once the MSCP has done its job in setting up the appropriate resources for a transfer, all that is required is for the host machine to do the actual data transfer. All of the details required by the host machine to accomplish the data transfer are sent to the host via the control path. The message contains device-specific information needed by the host to set up the transfer.

In the current production NCAR MSS, all devices are accessed via a block multiplexer channel for IBM 370 (and compatible computer systems). The only way to read or write data to these devices via an appropriate control unit. This is where Network Systems Corporation (NSC) comes into the picture. They have been supplying us with adapters since the beginning that attach to the control units for the purpose of reading and writing data to the attached devices (4).These adapters are called RDME's (Remote Device Module - Extended Performance). On the device side, the RDME emulates an IBM host channel subsystem. It moves data to/from the attached devices by executing IBM channel programs. On the host side of the RDME, there is either a HYPERchannel interface (old NCAR system) or a HIPPI interface (new system).In the following discussion, I will only talk about the HIPPI version of the adapter, since we are using HIPPI exclusively at the present time.

The NSC adapter setup we are using means that there are two basic requirements for a host machine to be able to be attached directly to the MSS:

1. It must have a HIPPI interface capable of transmitting and receiving HIPPI packets to/from the NSC adapters using the HIPPI-FP protocol (5).
2. It must have the appropriate "mover" software running which is capable of building channel programs which are sent to the NSC adapters for reading and writing the MSS devices.

At the present time, there are 6 Crays, 7 Suns, 6 IBM RS6000's, and 2 SGI machines connected to the NCAR MSS via HIPPI channels. All of the devices currently in use are FIPS-60 attached devices -- IBM 3490E cartridge drives, IBM 3380 and 3390 disk drives, and a StorageTek Silo equipped with 4490 drives. We currently have six NSC RDME adapters, which are used for accessing these devices. Each RDME plugs into one channel interface on one of the control units for the devices (each device is accessible to two control units, each of which has four channel interfaces). Figure 2 shows a close-up of the data path from a Cray to a FIPS-60 device. The HIPPI switch in the data path is necessary to provide connectivity between multiple hosts and multiple RDME's.

MOVING THE BOTTLENECK AROUND

For any data transfer that involves multiple steps, utilizing different hardware and protocols along the way, there will always be a "bottleneck" which limits the overall data rate. Once you have identified the bottleneck in a particular data path, you can take steps to speed up that portion of the transfer. If you succeed in doing this, and speed the transfer up enough, the

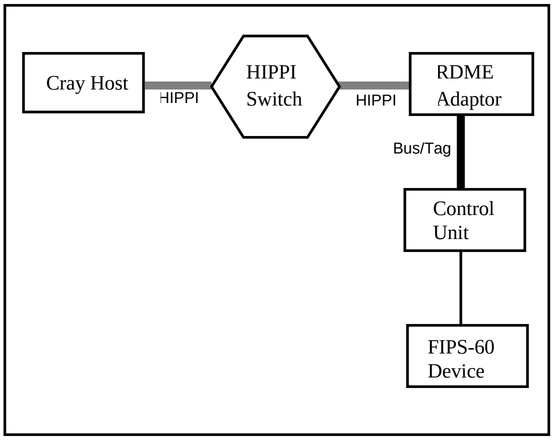

Fig. 2 - Cray to FIPS-60 Device

bottleneck moves to some other part of the datapath that had previously been running below its capacity.

In the case of the NCAR MSS, the data path we want to optimize is everything between the MSS device and the Cray's local disk subsystem. Table I shows the data rates for all of the different portions of the datapath for transfers between FIPS-60 MSS devices and several different Crays. These are sustained transfer rates.

TABLE I
Maximum Data Rates in MBytes/sec

	Local			FIPS-60	
Host Computer	disk	HIPPI	RDME	cntrl unit	device
CRAY Y-MP8/864	10 (DD-49)	80	25	4.5	3.0
CRAY Y-MP8I	20 (DD-60)	80	25	4.5	3.0
CRAY J916	12 (DD-5I)	78	25	4.5	3.0
CRAY J920	12 (DD-5I)	78	25	4.5	3.0
CRAY EL98	12 (DD-5I)	70	25	4.5	3.0

Looking at the FIPS-60 devices in our current system, it is clear that the devices themselves are the bottleneck, when it comes to overall datatransfer rates. If we had a fast enough device, with a fast enough control unit, we would be able to eliminate the device as the bottleneck. While we would effectively just be moving the bottleneck from one place to another, the end result would be an improvement in the overall transfer rate.

ESCON MAKES ITS DEBUT AT NCAR

Two factors played an important part in the decision to bring in ESCON-attached devices. One was the increase in the data rate to/from the devices. The other was the increase in storage capacity provided by new types of media, such as StorageTek RedWood cartridges. We would be able to get data into and out of the MSS faster, and we would also be able to reclaim precious floor space in the computer room, which is currently needed to store our large archive of lower capacity cartridges.

To set up our ESCON testbed, we brought in an NSC RDME equipped with an ESCON interface, in place of the bus/tag FIPS-60 interface. On the HIPPI side, it is identical to our existing RDME's. At the same time, we brought in a StorageTek RedWood drive for evaluation, which is configured with an ESCON interface. This gave us everything we needed to begin testing the data path from the Crays all the way down to the RedWood drive. All that was required on the Cray side was to modify the software that communicates with the RDME to include all of the necessary commands to set up the ESCON transfer. We were able to use the existing channel program software, since the RedWood device is 3490E compatible. Figure 3 shows a close-up of the data path from a Cray to a RedWood drive.

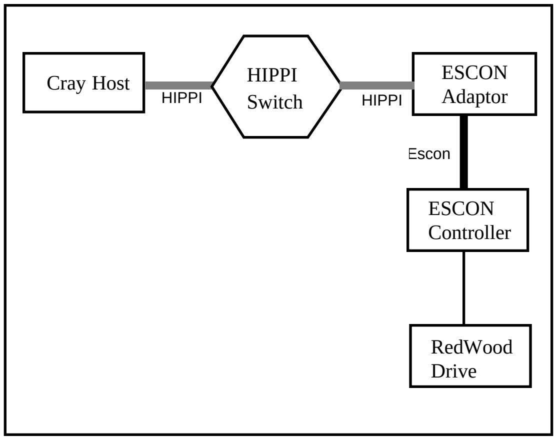

Fig. 3 - Cray to Redwood Drive

Most of the modifications necessary to support the ESCON data path wherein the portion of the mover software that communicates with the RDME. In the case of the FIPS-60 RDME, device selection was accomplished by means of the physical unit number. The select sequence broadcasts the device unit number to all attached control units, and the appropriate one responds to the selection request. All the host software has to do is supply the device unit number in the "Start I/O" message to the RDME.

The ESCON RDME requires a more complicated setup before a device can be accessed. Basically, what is required is to establish a logical path to the device, and then set up a connection. For more information on how this is done, refer to the NSC document "RDCOP-E Device Controller with Enterprise Software" (6). Once the connection has been set up, channel programs can be sent to the RDME to transfer data to/from the device.

MEASURING PERFORMANCE

The first tests that were done eliminated the Cray disk I/O portion of the transfer to determine the maximum transfer rates between the RedWood drive and Cray central memory. For these tests, a record size of 56,320 bytes was used. Each channel program transferred 8 records of this size in one I/O operation. The reason for picking this setup was to be compatible with existing record sizes already in use within the MSS.

For all of the Cray's tested (2 Y-MP's, 2 J-90's, and 1 EL98), the maximum data rate for writing/reading data to/from the

RedWood tape drive was between 9.5 and 10.0 MBytes/sec. This was the overall end-to-end transfer rate for moving a total of 30 MBytes of data. Hardware compression was active, and different data patterns were used. The data rate did not change measurably when a highly compressible data pattern was used, as opposed to an incompressible pattern. Turning off hardware compression altogether did not lower the data rate, and in fact seemed to increase the data rate 2 or 3 Mbits/sec in some cases. Some limited testing was done with larger record sizes. The highest rate measured was 12.0 MBytes/sec, using 225 KByte records.

The next test made use of the data mover code from the MSS host software to transfer files from Cray local disk to the RedWood device and read them back. Here again the record size being used was 56,320 bytes. The results varied considerably, due to differences in the Craydisk I/O rates, the types of disks used, and other factors. Table II shows the fastest rates achieved in the second set of tests. The rates shown are overall end-to-end transfer rates, and the size of the file being transferred was 30 MBytes. Two different test files were used -- one was not compressible, and the other compressed about 2 to 1 when written to the RedWood. The data rates were virtually identical for both files, as was the case in the previous set of tests.

TABLE II
Maximum Data Rates (MB/s) for Cray DASD to/from RedWood

Host Computer	Writing to RedWood	Reading from RedWood
CRAY Y-MP8/864	6.0	7.5
CRAY Y-MP8I	8.75	7.5
CRAY J916	8.0	9.0
CRAY J920	8.0	9.0
CRAY EL98	7.5	8.75

You can see that even with each portion of the data path being capable of at least 9.5 MB/s, the overall end-to-end transfer rate is somewhat slower. This is most likely due to timing considerations, and how double-buffering is used at various stages along the data path. We may still be able to do some further tuning to increase the overall transfer speed.

FUTURE PLANS

For all of the tests that were done, a single RDME was directly attached to a single RedWood ESCON controller. Before we will be able to go into production with the new system, we will need to bring in an ESCON director. This is a device which allows ESCON connections to be dynamically switched between multiple hosts and multiple devices. There will be a small amount of new code necessary to interact with the RDME for the purpose of setting up a data path through the ESCON director. The RDME has the capability to do this, but we have not yet been able to test it. The current plan calls for a total of five ESCON RDMEs to be installed. This would give us the capability to have an aggregate data rate between the Cray's and RedWood drives in the neighborhood of 40 MBytes/sec.

Another possibility we may look at is using a native ESCON interface on various Cray's and other MSS hosts to directly connect to the ESCON director, without going through HIPPI or the RDME. Whether or not we could do this would depend on the flexibility of the ESCON device driver, the performance of the interface, how errors are handled, etc. The advantage of using the RDME is that we already have all of our MSS host machines HIPPI attached, and equipped with software to talk to the RDME. With only a few minor changes to this software, these hosts can all be enabled to talk to an ESCON RDME, and any attached ESCON devices .The direct ESCON attach would require purchasing additional hardware for each machine (assuming it even exists) and installing device drivers, in addition to making changes to our local MSS software to make use of the new interface.

ACKNOWLEDGMENTS

I would like to thank Gene Harano, John Sloan, Erich Thanhardt, and George Williams for reviewing my paper and presentation. I would also like to thank Kristin Conrad for her help in preparing the final version of the paper and the viewgraphs.

TRADEMARKS

- ESCON is a trademark of the International Business Machines Corporation.
- CRAY, CRAY-1, CRAY X-MP, CRAY Y-MP, COS and UNICOS are trademarks of Cray Research, Inc.
- HYPERchannel is a trademark of Network Systems Corporation. StorageTek and RedWood are trademarks of Storage Technology Corporation.

REFERENCES

1. Nelson, Marc, David L. Kitts, John H. Merrill, and Gene Harano, "The NCAR Mass Storage System," Digest of Papers, Proc. Eighth IEEE Symposium on Mass Storage Systems, May 1987, pp. 12-20.

2. Merrill, John, and Erich Thanhardt, "Early Experience with Mass Storage on a Unix-based Supercomputer," Digest of Papers, Proc. Tenth IEEE Symposium on Mass Storage Systems, May 1990, pp. 117-121.

3. Miller, Stephen W., "IEEE Reference Model for Mass Storage Systems," Advances in Computers, Vol. 27, 1988, pp. 157-210.

4. Device Controller with Enterprise Software, 460465-03, Network Systems Corporation, Minneapolis, Minnesota, 1991.

5. High-Performance Parallel Interface -- Framing Protocol (HIPPI-FP), American National Standard X3.210-1992, April 1994.

6. RDCOP-E Device Controller with Enterprise Software, 461091-01, Network Systems Corporation, Minneapolis, Minnesota, 1995.