First results of using IDE file servers for PC farms

V.Kolosov
ITEP, Moscow
N.Kruglov, A.Kryukov
SINP MSU, Moscow

Abstract

Monte Carlo production for LHC experiments require a lot of hard disk space. For examples during CMS MC production in this Spring there was about 1 TByte disk space was occupied. The traditional approach that based on SCSI disk arrays is too expensive. So, the IDE files servers can be considered as cheap alternative to SCSI ones.

In the paper we consider the using of IDE file servers for PC farms. The file servers are based on standard technology that present on the market. The solution include dual Pentium-III CPU, 512 MByte RAM and RAID0. Such approach permit build IDE file servers with capacities until 1.8 TByte per box.

The authors present first results of using such file servers and discuss the problems with them and give some recommendation.

1 Introduction

The experiments at the CERN Large Hadron Collider (LHC) will begin taking data in 2006. The Petabytes of data are expected to be collected. Even Monte Carlo production for LHC experiments requires a lot of hard disk space. The traditional approach that based on SCSI disk arrays is too expensive to provide the acceptable solution for these goals. In last few years with the development cheap IDE RAID controllers the disk arrays on ATA disks have become good alternative to SCSI ones.

2 Setup

The experimental setup had the following configuration: Dual PIII-650MHz on Intel(R) L440GX+ motherboard with 512MB of RAM. 8 IBM DTLA-307075 disks were grouped into hardware RAID0 on 3ware 6800 IDE RAID controller. Network connection were provided by Intel(R) PRO/1000 Ethernet card and Intel(R) 530T switch. Ethernet controller was plugged into 66 MHz PCI slot while 3ware controller used 33 MHz PCI slot on different bus. IBM DTLA-307045 disk on onboard controller was used as system disk.

The described setup was tested under Linux. We used kernel 2.2.18 with Intel's driver for Intel(R) PRO/1000 Ethernet card. Ext2 filesystem was created on 600 GB RAID0.

3 Filesystem tests

First of all we tested the disk array without network. For initial tests we used Bonnie (http://www.textuality.com/bonnie/) with options “-s 2000”. The results were the following:

Input:		
	Per char:	10230 KByte/sec
	Per block:	64040 KByte/sec
	Rewrite	22107 KByte/sec
Output:		
	Per char:	11001 KByte/sec
	Per block:	79553 KByte/sec
Random seeks:		400.2/sec

The second method used the measurements with standard dd utility. The six flows were run in parallel:

For input: `dd bs=1024 count=2000000 if=/dev/zero of=file#`

For output: `dd bs=1024 count=2000000 if=file# of=/dev/null`

The results in these tests were almost the same as in the previous one.

For six processes:

Input: 65-66 MByte/sec

Output: 78-79 MByte/sec

4 Network tests

The first group of tests with network connections was done on Linux NFSv2 filesystem from six clients on Fast Ethernet links. The filesystem was mounted in asynchronous mode with block size of 8192 bytes. The method of the testing was the same as in tests for local filesystem by means of `dd`.

For input: `dd bs=1024 count=2000000 if=/dev/zero of=file#`

For output: `dd bs=1024 count=2000000 if=file# of=/dev/null`

The results was the following:

For six clients:

Input: 220-230 MBit/sec

Output: 260 MBit/sec

After that we rechecked the transfer rate in the different way to measure the influence of the NFS implementation:

For input:

```
rsh hostname dd bs=1024 count=2000000 if=/dev/zero | \
                                           dd bs=1024 of=file#
```

For output:

```
dd bs=1024 count=2000000 if=file# | \
                  rsh hostname dd bs=1024 of=/dev/null
```

The results were the same as for NFS:

For six clients:

Input:	220-230 MBit/sec
Output:	260 MBit/sec

At last we measured the pure network speed turning off the disk operations:

For input:

```
rsh hostname dd bs=1024 count=2000000 if=/dev/zero | \
                               dd bs=1024 of=/dev/null
```

For output

```
dd bs=1024 count=2000000 if=/dev/zero | \
                  rsh hostname dd bs=1024 of=/dev/null
```

In this configuration we obtain the significant increase in speed:

For six clients:

Input:	450 MBit/sec
Output:	450 MBit/sec
Bidirectional test:	350 MBit/sec

The result for bidirectional test, when input and output flow were equal, looks similar to the result from tests which were done by "Network Computing" magazine[1].

5 Conclusions

The main result from our tests is the possibility to build the storage system which is based on cheap IDE RAID solution. For this configuration we found the limitation on the network access rate to the disk array at the level of 200-250 Mbit/sec. This limitations are common for IDE and SCSI systems on PC architecture for the same configuration. We suppose that the locking common system bus by the device to provide the direct access to the memory leads to impossibility to operate for other devices at the same moment and decrease in performance for the system as a whole.

This rates mean that it is possible to use more cheaper EtherChannel on Fast Ethernet cards to provide the same performance, though GigaEthernet allow us to build more scalable system.

In any case more detailed study of other options with new Linux kernel, faster system bus speed, other RAID configuration etc is needed.

6 Acknowledgments

This project was carried out with financial support from INTAS (grant 00-00440 INTAS CERN).

References

[1] Joel Conover, Gigabit Ethernet Switches Set To Take On The Enterprise, Network Computing, http://www.nwc.com/916/916r1.html