Initial Tests of a USB/Firewire storage solution

Andreas Hirstius

Motivation

The current mass storage solution with tapes has a very long access time and a relatively limited transfer rate. In addition, the number of available tape units is a serious bottleneck, because the number of requests will be basically always larger than the number of tape units.
A mass storage solution using external harddisks, which are connected via USB 2.0 or Firewire could replace the tapes for certain applications, e.g. relatively short term storage.
This test was intended to investigate basic features like throughput and simple maintainability/manageability of external USB/Firwire harddisks connected to a Linux box.

1 Initial Setup and Tests

1.1 Test equipment

- 8x "old" 75GB IBM DTLA-307075 HDD
- external USB 2.0 / Firewire combo enclosure (Genesys Chipset)
- 1x Adaptec 4 (+1) Port USB 2.0 Hub
- 1x NoName 7 (+1) Port USB 2.0 Hub (Arp Datacon Art. Nb.: 244149 / Orig.Manu.Nb.: UH7U-2.0)
- 1x NoName USB 2.0 (4+1 Ports) / Firewire (3 Ports) Combo Hub (Arp Datacon Art. Nb.: 240002/ Orig.Manu.Nb.: UF-300; Rev. 1.2)
- 2x NoName Firewire Hub 6 Ports (Arp Datacon Art.Nb.: 224956; Orig.Manu.Nb.: FH-600)
- Adaptec USB 2.0 Controller (AUA-5100 5 ext. + 1 int. Ports; NEC chipset)
- Adaptec Firewire Controller (FireConnect 4300 3 ext. Ports)
- Intel chipset as on-board controller of IBM R40 (2722-GDG) notebook

All three controllers have only a 32-bit/33 MHz PCI interface. Any throughput measurement will have an upper limit set by this interface. There are reports on the web, that the maximum reachable transfer rate for the controller chip (NEC) used on the Adaptec USB 2.0 controller is below 30MB/s.

1.2 Test plan

Things that should be looked at:

- does Linux work with the controllers
- can Linux actually see (and work with) this number of external HDDs
- how does Linux behave with cascaded Hubs (and what's the transfer rate)
- Fault tolerance/maintainability; How does the system behave with:
  - exchange of disks with normal procedures (umount/unplug/plug/fdisk/mount)
  - dead disk (during access / w/o access) and exchange
  - dead cable(s) (permanent / temporary)
  - dead Hub (permanent / temporary / new [and different brand] Hub)

1.3 Test program

The benchmark program bonnie++ (http://www.coker.com.au/bonnie++/) was used the test the throughput of the disks. Later also the CERNgrown program io64 was used in a few tests.

2 Tests with USB 2.0

The Linux kernel supports USB 2.0 by default (EHCI driver). The usb-storage module talks to the SCSI-layer of the kernel, so an external USB disk appears as a SCSI device.

Figure 1: The laptop test setup.

2.1 Measurements on the Notebook

The host machine was an IBM R40 (2722-GDG) notebook with 1.5GHz Pentium-M processor and 256MB RAM. The OS installed was Fedore Core 1 running a 2.6.1 kernel (and a little patch to be able to read from the USB disks).

2.1.1 The First Look

The tests on the laptop were at first intended to be a simple test of the HDDs used, since they were in storage for some time. But it eventually developed into much more complex measurements.
Connecting a single disk to the laptop worked "out of the box" with the right kernel. Also a tree with six disks behind two of the hubs was correctly discovered and all devices were available. A disconnect and the discovery after reconnect of an unmounted device worked for a single disk and for the whole tree.
With a 2.4.24 kernel the disks could be partitioned, but attempts to format a disk always failed with strange I/O errors. This was not investigated further, since it was anyway foreseen to make the tests with a 2.6 kernel. But there is reason to believe, that this problem is connected to the reading problem, which is descriped later on (Sec. 2.3).

A much more serious problem was observed in the setup with six disks (Fig. 1). An attempt to copy data from one USB disk to another stopped after the transfer of 64k. At this point the first look with the laptop stopped. This problem was further investigated with the Dual Xeon box and it turned out to be connected to the chipset in the external disk enclosure and as a solution a kernel patch had to be applied. A more detailed description follows.

2.1.2 More Detailed Measurements

More detailed measurements could be done on the laptop. To test if the disks are usable bonnie++ was used (*bonnie++ -d <disk> -s 10g -n 5*). The result for a single disk was a surprise:

disk	Sequential Output			Sequential Input			Seq. Create	Random Create	
	Per Char	Block	Rewrite	Per Char	Block	Seeks	Create	Create	Delete
	kB/s	kB/s	kB/s	kB/s	kB/s	1/s	1/s	1/s	1/s
1	22080	23968	**11857**	**18377**	**25972**	84.4	2177	2223	11356

The transfer rate for block-read was higher than both rates for writing and higher than the rate on the Dual Xeon setup. And in addition, the rates for rewrite and per char-read were also higher than the rates on the Dual Xeon setup! Only both write-rates were smaller than the rates on the Dual Xeon setup.
The results of the measurement with both disks reveal even more differences! The aggregate per character read and write rates are slightly smaller than the rates in the Dual Xeon setup. But the block write rate is 34%, the rewrite rate is 49% and the block read rate is even 50% higher than the corresponding rate in the Dual Xeon setup!! And both block rates are well above 30MB/s!

disk	Sequential Output			Sequential Input			Seq. Create	Random Create	
	Per Char	Block	Rewrite	Per Char	Block	Seeks	Create	Create	Delete
	kB/s	kB/s	kB/s	kB/s	kB/s	1/s	1/s	1/s	1/s
1	11129	19064	9507	12423	20384	72.1	1796	2256	10764
2	11170	18867	9565	12375	20398	83	2075	1389	4942
Sum	22299	37931	19072	24798	40780	155.1			

The transfer rates with six disks showed very little variation between the disks. A summary of the measurements:

disk	Sequential Output			Sequential Input		
	Per Char	Block	Rewrite	Per Char	Block	Seeks
	kB/s	kB/s	kB/s	kB/s	kB/s	1/s
per disk	3690	7213	3501	3980	7305	87.6
Sum	22143	**43276**	21004	23878	**43829**	525.6

The block-read and -write rates are both 50% higher than the rates obtained with the Adaptec controller. This particular measurement lasted for about two days and during this time no serious problems or errors were observed. A few SCSI error were reported, but they didn't result in a disconnect of the device.
Judging by the transfer rates of block-I/O, the on-board USB controller of the laptop is much better than the NEC chipset (Adaptec controller).

2.2 The Measurements with the Dual Xeon host

The machine contained two 2.4GHz Xeons (Pentium-IV), 1GB RAM and a motherboard with Intel E7500 chipset. The external disks were connected to the PCI-controller cards. Fedora Core 1 with a 2.6.1, 2.6.2-rc2, 2.6.2-rc3 and 2.6.2 kernel was the OS.

2.2.1 First Experience with this Setup

As with the laptop, the discovery of the external disks worked without problems right from the beginning. Unfortunately a very high failure rate of the disks was observed. One reason was undoubtedly the unusually high temperature of the disks. To avoid further overheating a cooling solution was installed (Fig. 2).
The failure rate was still relatively high afterwards, but no disks were exchanged anymore and the first measurements started. Very frequent deadlocks of the system were observed and no error messages could be found. This turned out to be a problem with the SMP kernel. All further measurements were done using only one CPU (maxcpus=1 boot option).

- **The single disk test**
  The command line: *bonnie++ -d <disk>*
  And the result of the measurement:

disk	Sequential Output			Sequential Input			Seq. Create	Random Create	
	Per Char	Block	Rewrite	Per Char	Block	Seeks	Create	Create	Delete
	kB/s	kB/s	kB/s	kB/s	kB/s	1/s	1/s	1/s	1/s
1	25695	26691	9757	15148	22782	217.2	2362	2348	5475

Figure 2: The test setup in the Lab with the cooling solution.

The sequential writing rates (per character and block) were close to the quoted limit of the NEC controller, while the transfer rates for sequential reading were too low, especially the per character reads. This might be a consequence of the kernel-"patch" to enable the reading from those disks.

- **Two disks**
  Two instances of bonnie++ were started with two USB disks as target. The results were:

disk	Sequential Output			Sequential Input			Seq. Create	Random Create	
	Per Char	Block	Rewrite	Per Char	Block	Seeks	Create	Create	Delete
	kB/s	kB/s	kB/s	kB/s	kB/s	1/s	1/s	1/s	1/s
1	13560	13808	6470	11886	13535	148	2252	2294	5080
2	13709	14421	6341	11459	13564	149	2310	2155	5249
Sum	27269	28229	12811	23345	27099	297			

Looking at the aggregate transfer rates, the block-read rate is basically at the same level as the write-rates and the per char-read rate is much closer to the block-read rate.

- **Three disks**

  The same game with three disks:

disk	Sequential Output			Sequential Input			Seq. Create	Random Create	
	Per Char	Block	Rewrite	Per Char	Block	Seeks	Create	Create	Delete
	kB/s	kB/s	kB/s	kB/s	kB/s	1/s	1/s	1/s	1/s
1	10038	9604	4344	9060	9378	135.7	2022	1528	4095
2	9378	9483	4368	9058	9317	126.4	2172	1395	3304
3	9521	9397	4355	9048	9403	127.5	1875	1787	
Sum	28937	28484	13067	27166	28098	389.6			

  With three disks the per character and block read and write rates are basically equal!! The maximum rate is slightly below 29MB/s and marks the limit of the capabilities of the NEC controller chip.

- **Four or more disks**

  The aggregate transfer rates do not increase anymore with more than three disks, so the rate per disk decreases. All disks have basically the same transfer rate. Unfortunately it turned out, that all five ports of the Adaptec controller share only one controller chip!! Therefore the transfer rate of the controller card is limited solely by the chipset. This also means, that in any decent future setup one has to use another controller and it doesn't matter for the aggregate transfer rate if the disks are connected directly to the controller or if they are in a tree configuration behind one port.

2.3 The disk-to-disk copy problem

The USB disk-to-USB disk copy problem discovered during the initial tests with the notebook turned out to be "simply" a read problem from a USB disk. The reading doesn't actually stop! It's only very slow. The first 64k are read and then the device takes a break of 8.5 min, then another 128k are read and again waiting etc. The average transfer rate was around 256 bytes/sec!
On the mailing lists (mainly linux-usb-devel) there were (very) recent reports about strange behaviour/failures of external devices connected via the Genesys chipset in the external enclosure. Unfortunately the enclosures we use have this chipset inside. There is no real fix for this problem up to now. But as a workaround a degradation of the throughput is advised and

different ways of doing so are presented. The easiest and cleanest way to do so seems to be the following "patch":
in drivers/usb/storage/scsiglue.c change

```
/* limit the total size of a transfer to 120 KB */
        .max\_sectors =                  240,
```

to

```
/* limit the total size of a transfer to 64 KB */
        .max\_sectors =                  128,
```

and enable USB and *usb-storage* debugging.
After applying those changes, the reading problem disappeared.
For additional tests with external USB disks it is absolutely recommended to use external enclosures with a different chipset, although they are more expensive and even if a fix for the problem will be found, that doesn't degrade the throughput by default. A buggy chipset is a buggy chipset, even with a workaround and it's difficult, if not impossible, to find out if it injects some strange behaviour into a larger system.

2.4 Problems with the SMP-kernel

After installing the cooling several more deadlocks of the machine were seen, that could not be explained by dead or failing disks. And there were no error messages at all, that could give a hint of what's behind the problem. The machine freezes under the following conditions:

- 3 or more disks directly connected to the controller
- 2 or more disks connected to the hub(s)

A single disk always works fine.
When running the same (SMP-)kernel with the "maxcpus=1" option no deadlocks were observed any longer!
There were also no problems when connecting the disk setup to the laptop.
In order to check if the SMP problem is platform dependent, the Adaptec controller was installed in a 2-way Itanium2 box (HP rx2600). No deadlocks were observed when running on this box. Instead of having a deadlock, the transfer simply stopped with catastrophic errors and the filesystems had to be rebuild.

In order to check if this problem is also related to the Genesys chipset, the transfer size in scsiglue.c was reduced to 8kB (.max_sectors = 16). The transfer rates for a single disk was now ridiculously low (10MB/s) as expected. But a test with continuous data transfer to seven disks was runnning for four days without a deadlock of the machine. Even though two disks died during that period, the test went on after killing the jobs accessing the dead disks.
A test with a Maxtor OneTouch external disk (.max_sectors=240) also caused a deadlock of the machine! With .max_sectors at 128 or "maxcpus=1" no problems were observed.
And a call trace of the Oops revealed, that the cause is most likely a wrong spinlock/interrupt handling. The problem was sent to the linux-usb-devel mailing list.

```
Unable to handle kernel paging request at virtual address 00100104
 printing eip:
f8a34fb9
*pde = 00000000
Oops: 0002 [#1]
SMP
CPU:    0
EIP:    0060:[<f8a34fb9>]    Not tainted
EFLAGS: 00010246   (2.6.6-rc1)
EIP is at qh_completions+0x162/0x35d [ehci_hcd]
eax: 00100100   ebx: 00001c00   ecx: f6ea12d8   edx: 00200200
esi: f5ea12a0   edi: f75fd460   ebp: c0343eac   esp: c0343e68
ds: 007b   es: 007b   ss: 0068
Process swapper (pid: 0, threadinfo=f7f44000 task=c02d2bc0)
Stack: f7153900 f75fd460 00001000 00001c00 f6ea214c 01343eb0 00000000 00000004
       00000000 f6ea1338 f6ea1c60 00000000 c1810880 c0343ec0 f6ea214c f6ea2100
       f7153900 c0343ed8 f8a35ded f7153900 f6ea2100 c0343f8c c0343ef8 00000000
Call Trace:
[<f8a35ded>] scan_async+0x7e/0x125 [ehci_hcd]
[<f8a382c0>] ehci_work+0x35/0xbb [ehci_hcd]
[<f8a3842a>] ehci_irq+0xe4/0x135 [ehci_hcd]
[<f8ba0427>] usb_hcd_irq+0x35/0x64 [usbcore]
[<c010801d>] handle_IRQ_event+0x39/0x62
[<c010844d>] do_IRQ+0xdd/0x21a
[<c01128c9>] smp_apic_timer_interrupt+0xd9/0x141
[<c01067c8>] common_interrupt+0x18/0x20
```

```
[<c0103db9>] default_idle+0x0/0x31
[<c0103de5>] default_idle+0x2c/0x31
[<c0103e58>] cpu_idle+0x36/0x3f
[<c03448b6>] start_kernel+0x1c0/0x222
[<c0344454>] unknown_bootoption+0x0/0x108

Code: 89 50 04 89 02 c7 41 04 00 02 20 00 c7 46 38 00 01 10 00 89
 <0> Kernel panic: Fatal exception in interrupt
In interrupt handler - not syncing
```

2.5 Failure recovery

During the tests, the disks reported occasionally filesystem and/or SCSI errors. Basically always a filesystem error was followed by a SCSI error. At first the DTLA disks were suspected to cause this, because they're known to have a problem. But in the end the reports were too frequent to be blamed only on the DTLA disks.
These errors made it possible to investigate the behaviour of the system in case of real errors. This would be very difficult with perfectly well working disks, so the errors were very welcome in a sense.
The observations:

- If a filesystem error was seen, the transfer to this filesystem was slowed down or stopped, but the transfers to the other disks were not effected.
- After a SCSI error the transfer to this device was stopped and the other transfers were slowed down or even stalled for a while. After a while the SCSI device disappeared from the list of available devices and the transfers to the other disks reached full speed again (and went even faster due to the missing disk).
- Reconnected devices were discovered even when tests were running on other devices.
- Disconnect of a disk or the whole tree during operation (unplugging the cable) resulted in a lot of SCSI errors, but after reconnecting the devices, they could be used again without a reboot of the machine. Sometimes a filesystem had to be rebuild, but mostly recovery of the journal was sufficient. All processes that access the disk(s) have to be killed and the devices have to be unmounted, before they can be connected again.

After a disconnect of a disk, either through a SCSI error or through deliberate unplugging the external device had to be powercycled. Having done this, it appeared again in the list of available devices and was useable after a reconnect. Without powercycling the device was "invisible".
Since most of the disks didn't show the same error (filesystem or SCSI) at the same place, it couldn't be a filesystem or a disk problem. A disk that reported SCSI errors was therefore tested on the normal on-board IDE controller and it didn't show any problems. The maximum transfer rate there was 32MB/s, which is consistent with previous measurements (Bernd). The mailing-lists show similar reports about SCSI errors with external USB disks and in most cases those disks are connected via the Genesys chipset. Without testing another chipset final conclusions can not be drawn, but there is enough evidence to blame the Genesys chipset.

3 Tests with Firewire

The reports in mailing lists suggest, that a firewire solution is less problematic than a USB solution. Unfortunately the standard firewire configuration is a daisy chain of the devices. ALEPH uses daisy chained Maxtor disks and doesn't have any problems! Such a configuration is unfortunately not useful for our purpose (poor maintainability).
The first tries with firewire stopped right at the beginning. It was impossible to get even a single disk up and running!! This can be fully blamed on the Genesys chipset basically without any doubt.

4 Conclusions

- The transfer rates are at a reasonable level and are limited by the controller chipset.
- Discovery of new devices works even under load.
- Fault tolerance / maintainability:
  - exchange of disks with normal procedures (umount/unplug/plug/fdisk/mount) works fine
  - dead disk w/o access can be easily replaced (see previous point)
  - dead disk during access is removed from device list and the rest of the system is uneffected

  - dead cable/hub: devices report a lot of SCSI errors and are removed from device list. If a dead device (dead disk or behind a dead cable/hub) is mounted and used, the processes that access the device have to be killed and the device has to be unmounted.

- Deadlock of the machine when running a SMP-kernel (2.6.1, 2.6.2) on a x86 box

- Don't use the Adaptec controller because of the very limited available bandwidth.

- **Don't use external disk enclosures with the Genesys Chipset!!!**

The main reason for the problems seen during this test is definitely the Genesys chipset. Even the SMP deadlock might be caused by it. For further investigations other external disks are needed. A different controller would also be a good idea.
This first test was quite successful in checking the behaviour of an USB based solution in terms of failure management and failure recovery. With the current setup too many problems were observed to show that external USB storage might be an option, but this is most likely connected solely to the chipset in the external disk enclosures.
It is very likely that further tests with more reliable hardware would prove that an USB storage solution is a viable option.

Appendix A

Powerconsumtion

A measurement of the powerconsumption of the external disks under different access scenarios was carried out.
Both types of disks (IBM DTLA and Maxtor OneTouch) showed very similiar powerconsumptions.
The results:

mode	1 stream	2 streams	4+ streams
seq. read	14.7 W	14.8 W	14.9 W
seq. write	13.8 W	13.8 W	13.9 W
random read	14.8 W	15.2 W	15.8 W
random write	13.8 W	13.8 W	13.9 W
startup	<25 W		
mkfs	14.7 W		
idle	<11 W		

Table 1: The powerconsumption of on external HDD. Only the max. consumed power is quoted.

As expected, the highest powerconsumtion was observed during the startup phase of the disk.
During operation the disk never exceeded a powerconsumption of 16 W.
It is notable, that the disk consume in idle mode already $\sim$ 11 W.
Unfortunately, the power saving capabilities of the disks can not be exploited. This is because they are visible as SCSI devices and *hdparm* does not support spinning down SCSI disks.

Appendix B

Tests with a single Maxtor OneTouch 250GB disk.

In order to check the setup, tests with a Maxtor OneTouch 250GB FW/USB2.0 were carried out.
The disk was tested on the Laptop and on the Adaptec Controller in the original dual Xeon box and in a HP Proliant DL140 (also dual Xeon 2.4GHz). The second box was used to rule out hardware failures in the original dual Xeon box.
The measurement results on the Laptop:

	USB		FW	
	read	write	read	write
transfer rate	∼30MB/s	∼27MB/s	∼23MB/s	∼20MB/s
CPU usage module	∼2%	∼2%	n.a.	n.a.
CPU usage application	∼5%	∼10%	∼5%	∼6%

Table 2: Test results for the Maxtor OneTouch disk connected to the Laptop.

Running the tests on the dual Xeon boxes cause a deadlock of the machines. This happend on both dual machines. After reducing the .max_sectors value to 128 or running with maxcpus=1, the machines were stable again. During the FW tests another deadlock was observed and could be reproduced. Reading from the disk causes a deadlock (even with maxcpus=1). No problems were observed while writing (up to now). The call trace of this oops indicates, that it's again a spinlock/interrupt problem.
The measurements results for the Adaptec controller:

	USB		FW	
	read	write	read	write
transfer rate	∼32MB/s	∼22MB/s	∼23MB/s	∼20MB/s
CPU usage module	∼1.5%	∼1.5%	n.a.	n.a.
CPU usage application	∼4%	∼13%	∼4%	∼12%

Table 3: Test results for the Maxtor OneTouch disk connected to the Adaptec controller in a dual Xeon box.