**Title:** TRANSPARENT INTELLECTUAL NETWORK STORAGE DEVICE

**Abstract:** The invention is an intellectual network storage device that uses a network to store and retrieve data, and that connects to an existing storage controller of a host computer. The device is transparent to the operating system of the host computer, and thus does not require additional software, such as a device driver, to operate. The device includes a device board, which is connected to an existing storage controller of a host computer via any suitable interface, a set of hardware or software acting as a remote storage server, and a connection that carries signals between the device board and the remote storage server.
Patent Application of:
Andriy Naydon, Sergiy Naydon, and Anton Kolomyeytsev

TITLE OF THE INVENTION:
Transparent Intellectual Network Storage Device.

CROSS-REFERENCES TO RELATED APPLICATIONS:
None.

BACKGROUND:
1. FIELD OF THE INVENTION:

The present invention relates to the way a "networked" storage device is plugged into the host system. In all other prior art storage networks, a dedicated storage controller maintains a plurality of "networked" storage devices, which requires a set of device drivers (typically, a driver for each networked storage device) running on the host central processor unit (CPU) under control of the operating system (OS).

In contrast, in the invention the intellectual network storage device (iNSD) is plugged directly into the existing storage controller already present in the host system, rather than to the system bus expansion slot. Thus in the invention, no additional software is required to make the iNSD operational, except the already present storage interface controller driver.

Said another way, from the viewpoint of the host machine OS, the iNSD of the invention represents a well-known storage device type (e.g., an HDD, CDROM, etc.) attached to an already recognized storage interface controller (e.g., an onboard PCI IDE controller) which is supported by the OS. This means that as far as the host machine OS is concerned, no new, unsupported storage controller with unknown architecture and an unknown set of storage devices is being attached to the host machine, and therefore no additional device driver or other software is needed.

2. DESCRIPTION OF THE RELATED ART:
In order to increase performance and capacity of an already installed system, new storage devices are added. However, it is not always possible to attach a new storage device directly to the host machine because of physical limitations (no more device expansion bays, power supply is limited, system case is under warranty and cannot be opened, etc.), and because of logical reasons (e.g., it is easier and cheaper to maintain a single large hard disk drive shared between multiple users than attach a small hard disk to each user's machine to get equivalent additional storage size). That's why network storage devices exist.

To overcome physical limitations or for the logical reasons discussed above, either existing or newly created network infrastructure is used to carry storage traffic, storage devices are added to one or multiple storage server nodes, and all user machines get equipped with some client software and/or hardware which is used to communicate with remote server nodes on behalf of requests arriving from the OS on client machines. Presently, two approaches are used:

1) A "fully software" solution which requires no additional hardware besides the already existing network interface card (NIC),

2) A "partially hardware" solution which involves some form of special storage controller card inserted into an expansion slot on the client computer and controlled by specialized driver software. A storage controller such as this has a direct network connection, and does not use any existing NICs.
Neither of these prior art approaches is satisfactory. A fully software solution is cheaper, but a partially hardware solution is faster and easier to use. In addition, in the case of the fully software solution, the following problems are apparent:

- All newly installed operating systems must have quite a complex set of software to make the network interface card (NIC) hardware work also as storage controllers.

- A fully software solution is very difficult to make bootable. Thus, at least one hard disk drive must be present in the machine to boot the operating system (OS), and only additional storage could be added as "network storage". Even if a fully software "boot enabled" solution is used, the client software is very complex and additional server software must be installed.

- The software set in a fully software solution is very central processor unit (CPU) hungry - i.e., it uses up a lot of CPU capacity. This degrades the performance of the entire system.

An example of a fully software solution is Microsoft's iSCSI initiator, which creates a set of storage devices in software (on a virtual SCSI controller) and propagates requests sent to these devices by OS to some remote server by means of Internet SCSI (iSCSI) protocol. But this solution has limited OS support, and also cannot be used as a boot device because virtual devices are not supported by BIOS.

With a partially hardware solution, most of the above-mentioned problems are solved. For example, such solutions are easier to boot as they may contain onboard BIOS, no additional
local storage is required, and they always have on-board processors to handle data management thus off-loading the host CPU.

An example of a hardware solution are iSCSI adapters manufactured by some vendors (e.g., Intel, etc.) which are inserted into PCI expansion slot on host computer. Nevertheless, there still must be software support (such as the installation of a driver set) from such iSCSI controller vendor or operating system vendor, and this is a major disadvantage.

SUMMARY OF THE INVENTION:

The invention is an intellectual network storage device that uses a network to store and retrieve data, and that connects to an existing storage controller of a host computer. The device is transparent to the operating system of the host computer, and thus does not require additional software, such as a device driver, to operate. The device includes a device board, which is connected to an existing storage controller of a host computer via any suitable interface, a set of hardware or software acting as a remote storage server, and a connection that carries signals between the device board and the remote storage server.

It is therefore an object of the present invention to provide a "fully hardware" network storage solution that needs no additional operating system support except the support already built-in for the existing storage controller. The invention is thus a "transparent" intellectual network storage device (iNSD).
It is a further object of the invention to provide an iNSD that can connect to different types of existing storage controllers, and connect to them via a variety of interfaces.

It is a further object of the invention to provide an iNSD that can communicate with the existing storage controller of a computer, and with a remote storage server, via either standard/known or non-standard communication protocols.

It is a further object of the invention to provide an iNSD that is easy to make bootable, does not require an additional hard disk to boot the OS locally, and does not consume excessive amounts of host computer CPU resources or degrade overall system performance.

Further objects and advantages of the invention will become apparent from a consideration of the ensuing description and drawings.

DESCRIPTION OF THE DRAWINGS:

Figure 1 is a diagram showing the major components of the transparent iNSD device, and how it interfaces with a host/client computer and a remote storage server.
Figure 2 is a diagram showing how a generic network storage controller (either software/virtual or hardware) is installed in prior art solutions, and what the created virtual storage device looks like in the operating system storage device stack.

Figure 3 is a diagram showing how the intellectual network storage device (iNSD) of the invention plugs into an existing storage controller expansion port (e.g., the primary channel of the onboard PCI IDE controller), and what the iNSD looks like in the operating system storage device stack.

DETAILED DESCRIPTION OF THE INVENTION:

The following provides a list of the reference characters used in the drawings:

10. Device board
11. Storage controller
12. Host computer
13. Remote storage server
14. Connection
15. Storage interface command packet
16. Response
17. Local disk
As shown in Figure 1, the transparent intellectual network storage device of the invention includes an actual device board 10, which is plugged into either the internal or external port of the existing storage controller 11 of the host computer 12. It should be noted that it does not matter what particular storage interface is used. As just example of two possible connection means between device board 10 and storage controller 11, device board 10 can connect directly to a PCI IDE controller via a 40-pin parallel ATA cable or a serial ATA cable, depending on the PCI IDE controller type and iNSD version.

The invention also includes a set of hardware or software acting as a remote storage server 13, and a connection 14 (either a physical wired connection or a wireless connection can be used) that carries signals between device board 10 and remote storage server 13.

Device board 10 is responsible for storage interface dependent device emulation - i.e., dealing with storage interface command packets 15 sent from storage controller 11, wrapping them into network interface dependent protocol, and sending them for execution to remote storage server 13. Device board 10 is also responsible for receiving responses 16 from remote storage server 13,
unwrapping the data and corresponding status codes contained therein, and providing storage controller 11 with the correct data and status codes which constitute the reply to the storage interface command packet 15 request which was previously issued.

Connection 14 (again, either a physical wired connection or a wireless connection) is responsible for providing all required network transport between device board 10 and remote storage server 13. The particular protocol used for communications may vary, depending on the storage interface type used and other factors.

Remote storage server 13 is responsible for receiving requests from device board 10 over connection 14, unwrapping storage interface command packets 15 that are wrapped in network protocol, and executing the storage interface command either in software or in hardware.

It should be noted that communication between device board 10 and storage controller can make use of absolutely standard means. Thus, the invention does not require any modification to the storage controller for communication - for example, it can use standard ATA/ATAPI protocol via 40-pin or serial cable, or use standard communication via USB wire in the same manner as any other external USB storage, e.g., memory sticks or USB flash cards.
Communication between device board 10 and remote storage server 13 can be performed using standard network packets - e.g., via standard TCP/IP protocol or low level Ethernet frames (if the server resides on the same network segment), depending on iNSD version and configurable options. No changes are required to existing network infrastructure.

As discussed above, to carry out these tasks the invention uses the already existing storage controller 11. Device board 10 is attached directly to storage controller 11, instead of plugging into the system bus expansion slot (for example PCI) as all other prior art storage controllers do. Thus instead of representing a new storage controller that would need additional software to operate, the invention represents a different class of hardware to the operating system of host computer 12 - that is, a storage device co-working with existing storage controller 11.

As a more specific example, the iNSD of the invention can be plugged directly into an onboard PCI IDE controller (configured either as master or slave), and can represent a type of hard disk drive (HDD). The PCI IDE controllers that are available today on practically any personal computer have built-in support in most operating systems, and thus no additional drivers are required to support the iNSD of the invention. Moreover, as the BIOS on most motherboards contains support for PCI IDE devices, it is even possible to boot from the iNSD of the invention without any additional hardware or software support, because the iNSD looks like a standard ATA/ATAPI device located on the IDE bus.
Such an architecture allows one to use the existing storage controller 11 and its drivers to make the iNSD of the invention accessible to the operating system. Thus the invention has all the advantages of a 'partially hardware" network storage controller (easy to make bootable, no need for additional hard disk to boot OS locally, and high system performance as dedicated CPU is used), without the disadvantage of needing an additional device driver.

The iNSD of the invention can also be configured to operate correctly regardless of whether host computer 12 uses a local hard disk to boot its operating system. If host computer 12 uses a local hard disk to boot its operating system, the iNSD (specifically, device board 10) can be plugged into storage controller 11, and because the OS already has a driver for storage controller 11 installed it would automatically recognize the iNSD device.

If host computer 12 does not use a local hard disk to boot its operating system, the iNSD of the invention can be plugged into an available storage controller device expansion slot. The operating system is then installed into the iNSD, in order to boot remotely from network storage. In this case, there would be still a need for a storage device driver; however, these are much more widely supported and there would be no need to provide a driver written specifically for the iNSD of the invention.

Figures 2 and 3 provide further perspective on the differences between prior art solutions and the iNSD of the invention:
Figure 2 shows how a generic network storage controller (either software/virtual or hardware) is installed in prior art solutions, and what the created storage device looks like in the operating system storage device stack. Local disk 17 represents a network storage controller - either a virtual device created by software on a host computer or a hardware device plugged into, for example, the PCI slot of the host computer. Whether software or hardware-based, local disk 17 is visible to the OS of the host computer. Every request sent by the OS to local disk 17 is propagated to server storage node 18 using network I/O 19. Server storage node 18 further communicates with a remote disk 20. It can be appreciated that in order for the OS to access server storage node 18 or remote disk 20, an additional vendor driver is required to drive the network storage controller (local disk 17) because the network storage controller usually has some special hardware/software implementation that has no built-in support in most OS.

Figure 3 is a diagram showing how the intellectual network storage device (iNSD) of the invention plugs into an existing storage controller expansion port (e.g., the primary channel of the onboard PCI IDE controller), and what the iNSD looks like in the operating system storage device stack. The intellectual network storage device (iNSD) of the invention represents a local hard disk visible to the OS of a host computer as a primary master 21 connected to a PCI IDE controller 22 onboard the host computer. Thus the iNSD requires absolutely no additional software driver support, because PCI IDE controller 22 already contains all the necessary support inside the OS and BIOS of the host computer.
Conclusions, Ramifications, and Scope:

Thus the reader will see that the network storage device of the invention is a "fully hardware" network storage solution that needs no additional operating system support except the support already built-in for the existing storage controller.

While the above descriptions contain many specificities, these shall not be construed as limitations on the scope of the invention, but rather as exemplifications of embodiments thereof. Many other variations are possible without departing from the spirit of the invention. Examples of just a few of the possible variations follow:

Although the description mentions a PCI IDE controller as the default storage controller used to communicate with our intellectual network storage device via PATA (parallel ATA) interface, the invention is not confined to using this interface alone. As just a few examples, the iNSD of the invention can be connected to a PCI IDE controller also via a SATA (Serial ATA) interface.

Even a completely different interface may be used, for example USB, if the iNSD is attached to an enhanced host controller implementing a USB 2.0 interface on the motherboard. In this case, the iNSD of the invention can be plugged directly into any available hub attached to the
enhanced host controller - internal or external, depending on the motherboard implementation.
The invention’s use with still other interfaces is envisioned, including those not yet developed.

The description and figures primarily discuss a PCI IDE controller as the existing storage controller of the host computer. However, the iNSD of the invention may be connected to another type of existing storage controller, including but not limited to an enhanced host controller as discussed above.

In sum, whatever type of existing storage controller the iNSD is connected to, or whatever type of interface is used to connect the iNSD to the storage controller, what is important is that the iNSD is considered by the OS of the host computer to be a storage device co-working with the existing storage controller - and therefore, the iNSD does not require any additional software to operate.

Although the figures show the device board communicating with just one remote storage server, in actuality there may be more than one remote storage server. That is, the device board can communicate with multiple remote storage servers simultaneously.

The description and figures show the device board communications with the storage controller, and the device board communications with the remote storage server, to be of a known or standard type - for example, standard ATA/ATAPI protocol or standard USB, and
standard network packets respectively. However, while the use of standard communication
protocols has its advantages, the invention is not confined to them. Non-standard
communications means may also be used.

The connection between the iNSD and the existing storage controller can be a physical wired
connection, or alternatively it may be a wireless connection.

Accordingly, the scope of the invention should be determined not by the embodiments
illustrated, but by the appended claims and their legal equivalents.
CLAIMS:

What is claimed is:

1. A storage device, comprising:
   (a) a device board that emulates a storage interface dependent device;
   (b) means for connecting said device board to a storage controller of a computer, such that said computer recognizes said device board as co-working with said storage controller; and
   (c) a remote storage server communicating with said device board;

   whereby said storage device does not require an associated device driver on said computer to operate.

2. The device of claim 1, wherein said storage controller is a PCI IDE controller.

3. The device of claim 1, wherein said storage controller is an enhanced host controller.

4. The device of claim 1, wherein said means for connecting comprises an ATA interface.
5. The device of claim 1, wherein said means for connecting comprises a USB interface.

6. The device of claim 1, wherein said communication between said device board and said remote storage server is via a wired connection.

7. The device of claim 1, wherein said communication between said device board and said remote storage server is via a wireless connection.

8. A remote storage system, comprising:

   (a) a device board that emulates a storage interface dependent device;

   (b) a computer having a storage controller;

   (c) a remote storage server;

   (d) a first communications connection between said device board and said storage controller, such that said computer recognizes said device board as co-working with said storage controller; and
(e) a second communications connection between said device board and said remote storage server.

9. The system of claim 8, wherein said storage controller is a PCI IDE controller.

10. The system of claim 8, wherein said storage controller is an enhanced host controller.

11. The system of claim 8, wherein said first communications connection comprises an ATA interface.

12. The system of claim 8, wherein said first communications connection comprises a USB interface.

13. The system of claim 8, wherein said second communications connection is a wired connection.

14. The system of claim 8, wherein said second communications connection is a wireless connection.

15. A method of communicating between a computer and a remote storage server, comprising the steps of:

(a) providing a device board that emulates a storage interface dependent device;
(b) providing means for connecting said device board to a storage controller of a computer, such that said computer recognizes said device board as co-working with said storage controller; and

(c) providing means for communicating between said device board and a remote storage server.

16. The method of claim 15, wherein said storage controller is a PCI IDE controller.

17. The method of claim 15, wherein said storage controller is an enhanced host controller.

18. The method of claim 15, wherein said means for connecting comprises an ATA interface.

19. The method of claim 15, wherein said means for connecting comprises a USB interface.

20. The method of claim 15, wherein said means for communicating comprises a wired connection.
Network Storage Controller (virtual, i.e. created by software on client computer, or hardware, i.e. some vendor device in PCI slot of client computer)

FIGURE 2
Network Storage Device (Primary Master, local disk)

Existing Storage Controller (for example onboard PCI IDE controller channel)

Server Storage Node

Remote Disk

FIGURE 3