

The role of SCSI diagnostic tools in the iSCSI environment

As iSCSI begins moving from designs to real world products, diagnostic tools are needed for a variety of purposes. This paper will address a few real-world experiences gathered while working in this new environment.

What are the issues?

At the highest level, the question is "is this storage subsystem working?" Does the computer system recognize the disk on the other end of the wire? Is the capacity of the disk readable? Can the inquiry data be shown? Can the disk write and read data reliably?

Once communication is established and verified, lower-level functions need to be confirmed. Can the write cache on the drive be turned on/off? Can new firmware be downloaded into the drive? Does the entire storage subsystem respond in a reliable way when an error occurs? These errors could be drive related (a drive failure), or system related (an illegal command sent from a software application).

Device / Firmware development

In the real world, disk drives do not always operate strictly according to standards. Will your storage system crash or misbehave if a drive has a peculiarity? For instance, in experimenting with an iSCSI ->Fibre Channel bridge/router this week I discovered a particular brand of disk drive that did not support the SCSI command that the iSCSI bridge was relying on when it did fibre channel device discovery. The drive failed this command, and the bridge decided that that rack of drives was not there. Invisible drives!

By using a controlled environment SCSI design tool (PTI's SCSI toolbox32) we were able to quickly ascertain what the offending SCSI command was, duplicate that command and collect detailed information about how the drive was failing. We then took this information to the software engineers at the iSCSI bridge company, they made changes to their software, and voila – within 30 minutes our bridge could now use Hitachi fibre channel drives!

Functional and performance testing

The SCSI toolbox32 provides several "layers" of testing needed for iSCSI work. Its hot bus scanning allows discovering devices added to or removed from the iSCSI connection. Once a drive is discovered any SCSI command can be tested. In theory any legal or illegal SCSI command should be supported in the iSCSI environment. In today's reality we are dealing with bridges accomplishing the protocol conversion between iSCSI and SCSI/FC. Any time there is protocol conversion there is a possibility for errors, and the SCSI toolbox32 helps identify those errors. Since it generates known good (or known bad) SCSI commands, the bridge conversion process can be completely tested and understood. SCSI

compliance tests can be used to insure that all SCSI 2 and SCSI 3 commands are supported correctly.

Once command compliance is assured, testing can move into a performance phase. Writes and reads of varying blocks per transfer can be sent to one or more drives, from one or more source computers. Raw “best case” performance can be measured to one drive. “Real world” performance can be measured using multiple synchronized computers sending multiple data streams to one or more drives or volumes. Tests running 128 deep queued commands to multiple drives can easily generate enough data to completely swamp the iSCSI subsystem for “torture” type testing.

Surround your unknowns with knowns

In summary, testing an iSCSI HBA or an iSCSI->SCSI/FC bridge or router is easily accomplished with the following pieces:

1. A test tool that can generate known good SCSI traffic, and can eloquently deal with and report all data gathered during any error condition.
2. A known good SCSI or fibre channel disk drive.

In between these two “knowns” is placed the iSCSI HBA or iSCSI->SCSI/FC bridge or router. The theory is then “if something doesn’t work right, it’s the HBA or the bridge or the cables”. As I mentioned above about the “invisible drives”, this test setup can provide for very fast identification and correction of bugs.

One more example

In closing, another example came up when we used the SCSI toolbox32 to send an INQUIRY command that asked for 6 bytes of data to be returned (a perfectly legal thing to do). The iSCSI bridge received our iSCSI command, converted it to fibre channel, sent it to the drive, and got the data back. But then, instead of sending back the 6 bytes that we asked for, the bridge sent back 32 bytes of data. This made certain layers of the operating system device drivers very unhappy – trying to stuff 32 bytes of data into a 6 byte sack! The good news was that it was very easy to reproduce the error, the error information obtained was everything needed, and once again the firmware in the bridge was fixed in a very short time.

Our new Tee shirts are going to say “You can’t spell iSCSI without SCSI” – and you can’t develop iSCSI equipment without good SCSI tools!