Abstract

The iSCSI contributors are currently considering the framing/delimiting of iSCSI messages in a TCP stream, especially with an eye towards being able to process segments in a TCP stream out-of-order.

Another concern in iSCSI is reducing host overhead, especially for data phase transfers. An RDMA abstraction on top of TCP has been proposed as a solution.

VI/TCP addresses both those needs. It provides a generic, reliable
message and RDMA interface. In addition, VI has other compelling applications, including clustering and filesystem access. As such, it is a good candidate for a generic mechanism to place on a network interface.

1. Introduction

In situations where there is totally in-order delivery of IP packets from the network, framing issues are largely a matter of implementation complexity.

In the presence of reordering, however, it is best to minimize the dependencies between packets, so as to be able to do as much processing up-front as possible. Such processing ensures that data can be copied to correct buffers the first time, decreasing the need for dedicated reassembly buffers as well as the latency and bandwidth related to extra copies.

Decreasing the need for reassembly buffers is crucial for high-speed WAN links. On a 10 gigabit connection with a 200 millisecond round-trip time, hundreds of megabytes of re-order buffering may be necessary.

The VI/TCP protocol, when used optimally with TCP, adds a self-describing header to each segment. For data transfers, this header is sufficient to place the data in the correct receiving buffer. For messages, the header is sufficient to place the data in the correct places in a message queue.

2. Changes necessary to iSCSI to support VI

* The iSCSI command and ready-to-transmit PDUs must be changed to pass a 64-bit virtual address and 32-bit memory handle.

* VI has a maximum message size, which may be less than the iSCSI maximum. iSCSI PDUs over the size of VI’s maximum message size will need to be divided across VI messages.

Problems with MTU should only come into play on unsolicited writes. However, an alternate scheme for unsolicited writes is possible with VI. The initiator could be issued a memory region (or multiple memory regions) for immediate writes.

* Data transfers will no longer happen at the iSCSI layer in iSCSI.
PDUs, but instead be initiated as RDMA.

3. Q & A

Does using VI/TCP imply that I will have to use the VI software interface?

No, the VI/TCP protocol is entirely self-contained and does not mandate any software interface.

Will adopting the VI/TCP protocol place any constraints on iSCSI?

Recovery from failed TCP connections: Since RDMA is at a lower layer than iSCSI, recovery in the middle of an iSCSI data phase would be harder with VI/TCP. However, iSCSI is not currently proposing to recover starting in the middle of a data phase.

Will adding VI hurt performance of software-only implementations?

No. It may help performance by using common RDMA code for many applications.
4. Author’s Address (send comments to:)

Constantine Sapuntzakis  
Cisco Systems, Inc.  
170 W. Tasman Drive  
San Jose, CA 95134  
USA  

Phone: +1 408 525 5497  
Email: csapuntz@cisco.com

5. References


[VI/TCP] VI/TCP (Internet VI), draft-dicecco-vitcp-00.txt

Expires January 2001