Ethernet RDMA Technologies

technology brief

Abstract

As a leader in the development of innovative technologies and a trusted advisor on future data center directions that provide lasting value for information technology (IT) customers, HP is closely involved with development of future fabric interconnect technologies such as RDMA over TCP. This technology brief will discuss RDMA over TCP in detail.

Introduction

Advances in computing and storage technologies are placing a considerable burden on the data center's network infrastructure. For example, as Ethernet speeds increase and greater amounts of data are moved, it takes more central processing unit (CPU) processing power to process this communication.

Compounding this problem, incoming Transmission Control Protocol/Internet Protocol (TCP/IP) data consumes significant memory bus bandwidth because this data typically crosses the memory bus three times: once when it arrives and twice more when it is processed and stored in its final destination in the application. This overhead keeps the CPU from doing other useful work, adds latency, and can limit the scalability of the data center network.

In addition, a typical data center today uses a variety of disparate interconnects to link servers to servers and servers to storage. The use of multiple system and peripheral bus interconnects decreases compatibility, interoperability, and management efficiency and drives up the cost of equipment, software, training, and the personnel needed to operate and maintain it. To increase efficiency and lower costs, data center network infrastructure must be transformed into a unified, flexible, high-speed fabric.

The concept of a unified high-speed data center infrastructure is relatively new. Such a system requires a high-bandwidth, low-latency fabric that can move data efficiently and securely between servers, storage, and applications. Evolving fabric interconnects and associated technologies promise more efficient and scalable computing and data transport within the data center by reducing the overhead burden on processors and memory. Use of more efficient communication protocols, some of which run over existing infrastructures, frees processors for more useful work and improves infrastructure utilization. In addition, the ability of fabric interconnects to converge functions in the data center over fewer, or possibly even one, industry-standard interconnect presents significant benefits.

Remote direct memory access (RDMA) technology is an emerging fabric technology that promises to accomplish these goals. RDMA is a data exchange technology used by systems, applications, and storage to communicate directly over a fabric infrastructure, such as Ethernet.

HP is a leader in developing networking and communication technologies, including cluster interconnects such as ServerNet, the Virtual Interface (VI) Architecture, and InfiniBand™ (IB), which represent the origins and evolution of RDMA technology. Today, HP is at the forefront of the RDMA technology initiative. HP is a founding member of the RDMA Consortium, an independent group formed to develop the architectural specifications necessary to implement products that provide RDMA capabilities over existing TCP/IP networks.

This technology brief will discuss RDMA over TCP in detail. For information on other emerging fabric technologies, read the HP technology brief titled *emerging fabric technologies* at ftp://ftp.compaq.com/pub/supportinformation/papers/tc030410tb_rev2_us.pdf.

The problem

TCP and IP represent the suite of protocols that drive the Internet. Every computer connected to the Internet uses these protocols to send and receive information. Information is transmitted in fixed data formats (packets), so that heterogeneous systems can communicate. The TCP/IP stack of protocols was developed to be an internetworking language for all types of computers to transfer data across different physical media. The TCP and IP protocol suite involves over 70,000 software instructions that provide the necessary reliability mechanisms, error detection and correction, sequencing, recovery, and other communications features.

Computers implement the TCP/IP protocol stack to process outgoing and incoming data packets. Today, TCP/IP stacks are usually implemented in operating system software and, therefore, packets are processed by the main system CPU. As a result, protocol processing of incoming and outgoing network traffic consumes CPU cycles—cycles that could be used for business and other productivity applications. The processing work and associated time delays may also reduce the ability of applications to scale across multiple servers. As network speeds move beyond 1 gigabit per second (Gb/s) and larger amounts of data are transmitted, CPUs become burdened by TCP/IP protocol processing and data movement.

The burden of protocol stack processing is compounded by a finite amount of memory bus bandwidth. Incoming network data consumes the memory bus bandwidth because each data packet must cross the memory bus at least three times. The receiving device writes data to the device driver buffer, copies it to an operating system buffer, and then copies it into application memory space. These copy operations add latency. Considering that the original data often must be split into smaller chunks, data moving at 1 Gb/s could consume a considerable percentage of the memory bus bandwidth, forcing all CPUs to stall for memory. In fact, the TCP/IP protocol overhead associated with 1 Gb of Ethernet traffic can increase CPU utilization by 20 to 30 percent. With 1 Gb of TCP/IP networking traffic requiring up to 1 gigahertz (GHz) of processing power, the ensuing software overhead as 10-Gb Ethernet arrives has the potential to overwhelm system CPUs.

RDMA overview

TCP/IP protocol overhead and constrained memory bandwidth are obstacles to deployment of faster Ethernet networks in the data center. The use of TCP/IP offload engines (TOE) and RDMA over TCP technology can diminish these obstacles. RDMA is often confused with TOE, but they perform different functions. TOE offloads TCP/IP protocol processing overhead from the host CPU to a network adapter that supports TOE. TOE and RDMA can work together: TOE can be used to connect to any TCP/IP device while RDMA can be used to connect to other RDMA devices with a more efficient protocol. For more information on TOE, read the technology brief titled *emerging fabric technologies* at ftp://ftp.compaq.com/pub/supportinformation/papers/tc030410tb_rev2_us.pdf.

RDMA technology was developed to move data from the memory of one computer directly into the memory of another computer with minimal involvement from their CPUs. Additional information included in the RDMA protocol allows a system to place the communicated data directly into its final memory destination without any additional or interim data copies. This “zero copy” or “direct data placement” (DDP) capability provides the most efficient network communication possible between systems.

RDMA provides a faster path for applications to transmit messages between servers. Similar RDMA functionality is provided for both IB and IP. Both these interconnects can support all of the emerging and existing network standards such as Sockets Direct Protocol (SDP), iSCSI Extensions for RDMA (iSER), Network File System (NFS), and Direct Access File System (DAFS).

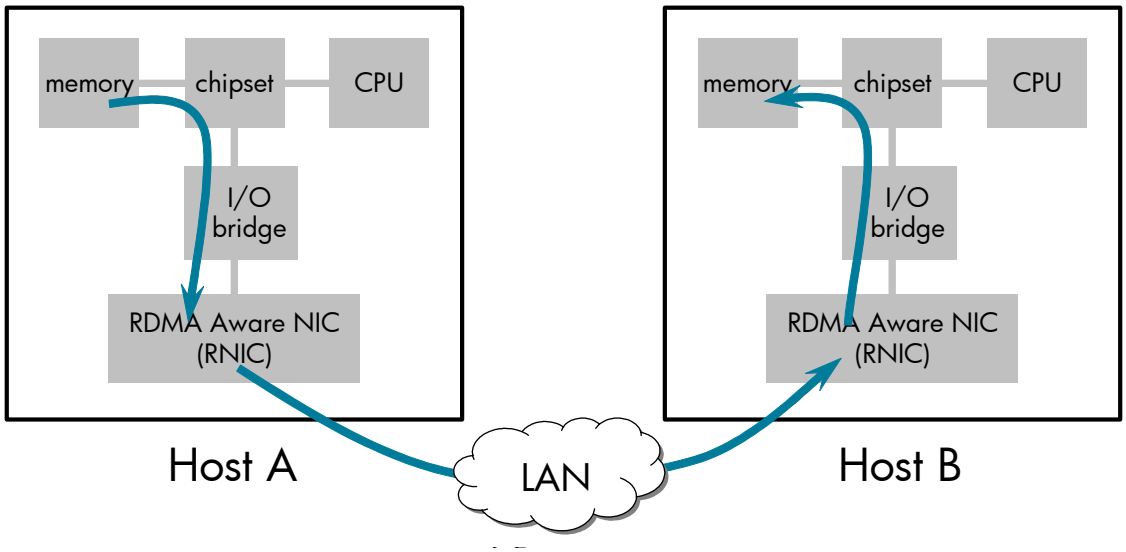
RDMA over TCP

Ethernet is the most prevalent network interconnect in use today. Customers have invested heavily in Ethernet technology and most are unwilling to tear out their networks and replace them. Their reliance on Ethernet is justified by its low cost, backward compatibility, and consistent bandwidth upgrades over time. Current Ethernet speeds in the data center are 100 megabits per second (Mb/s) and 1 Gb/s. Next-generation speed will increase to 10 Gb/s. Customer migration to 10-Gb Ethernet will be tempered by the input/output (I/O) processing burden it places on servers. The addition of RDMA capability to Ethernet will reduce CPU utilization and increase the benefits realized by migrating to 10-Gb Ethernet.

The addition of RDMA capability to Ethernet interconnects will allow data centers to converge their infrastructure over fewer types of interconnects. This simplifies server deployment and improves infrastructure flexibility. As a result, the data center infrastructure will become less complex and easier to manage. The data center will also have an infrastructure that is more adaptive. For example, bandwidth can be diverted from networking to storage when provisioning a database server and the bandwidth can be diverted back to networking if the server is reprovisioned as an application server.

RDMA over TCP is a communication protocol that moves data directly between the memory of applications on two systems (or nodes), with minimal work by the operating system kernel, and without interim data copying into system buffers (Figure 1). This capability enables RDMA over TCP to work over standard TCP/IP-based networks that are commonly used in data centers today.

Figure 1. RDMA over TCP places the data from the memory of one host directly into the memory of another host with no need for multiple buffer copies or CPU intervention.



RDMA over TCP allows many classes of traffic (networking, I/O, file system and block storage, and interprocess messaging) to share the same physical interconnect, enabling it to become the single unifying data center fabric. RDMA over TCP provides more efficient network communications, which can increase the scalability of CPU-bound applications. RDMA over TCP also leverages existing Ethernet infrastructures and the expertise of IT networking personnel.

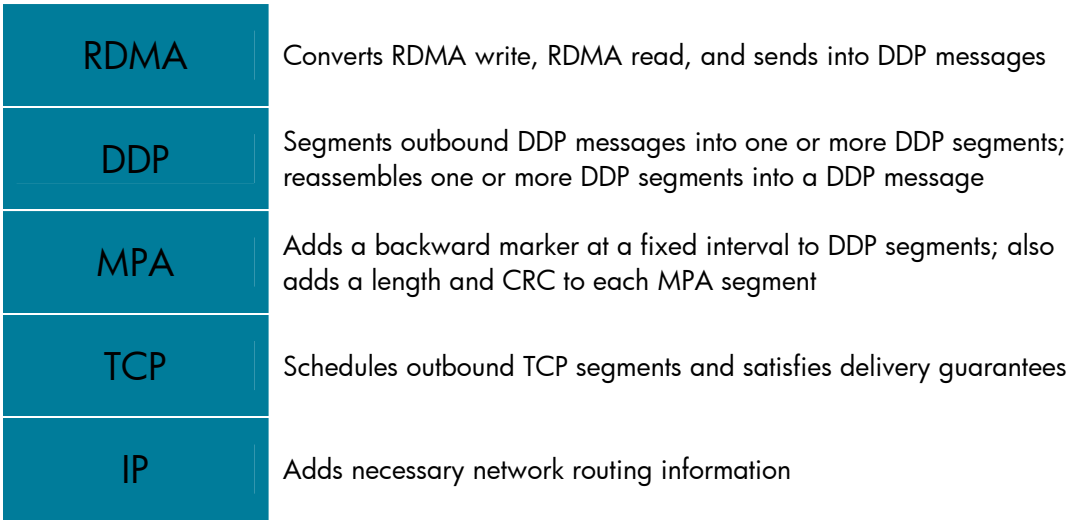
RDMA protocol details

The RDMA protocol suite eliminates data copy operations and therefore reduces latencies by allowing a local application to read and write data to memory on a remote computer. RDMA does this with minimal demands on memory bus bandwidth and CPU processing overhead, all while preserving memory protection semantics.

Layering overview

A series of protocols are layered to perform RDMA over TCP (Figure 2). The RDMA protocol converts RDMA write, RDMA read, and sends into Direct Data Placement (DDP) messages. The DDP protocol segments outbound DDP messages into one or more DDP segments, and reassembles one or more DDP segments into a DDP message. The marker-based protocol-data-unit-aligned (MPA) protocol adds a backward marker at a fixed interval to DDP segments, and it adds a length and cyclical redundancy check (CRC) to each MPA segment. TCP schedules outbound TCP segments and satisfies delivery guarantees. IP adds necessary network routing information.

Figure 2. A series of protocols are layered to form RDMA over TCP.



MPA

TCP uses a stream of octets to communicate data, while DDP uses fixed protocol data units (PDUs). To enable RDMA, DDP needs a framing mechanism for the TCP transport protocol. MPA is a marker-based PDU-aligned framing mechanism for the TCP protocol.

MPA provides a generalized framing mechanism that enables a network adapter that uses DDP to locate the DDP header. The adapter can then place the data directly in the application buffer based on the control information carried in the header. MPA enables this capability even when the packets arrive out of order. By enabling DDP, MPA avoids the memory copy overhead and reduces the memory requirement for handling out of order packets and dropped packets.

MPA creates a framed PDU (FPDU) by pre-pending a header, inserting markers, and appending a CRC after the DDP segment. MPA delivers the FPDU to TCP. The MPA-aware TCP sender puts the FPDUs into the TCP stream and segments the TCP stream so that each TCP segment contains a single FPDU. The MPA receiver locates and assembles complete FPDUs within the stream, verifies their

integrity, and removes information that is no longer necessary. MPA then provides the complete DDP segment to DDP.

Direct data placement

DDP allows upper layer protocol (ULP) data, such as application messages or disk I/O, contained within DDP segments to be placed directly into its final destination in memory without processing by the ULP. This may occur even when the DDP segments arrive out of order.

A DDP segment is the smallest unit of data transfer for the DDP protocol. It includes a DDP header and ULP payload. The DDP header contains control and placement fields that define the final destination for the payload, which is the actual data being transferred.

A DDP message is a ULP-defined unit of data interchange that is subdivided into one or more DDP segments. This segmentation may occur for a variety of reasons, including segmentation to respect the maximum segment size of TCP. A sequence of DDP messages is called a DDP stream.

DDP uses two data transfer models:

- Tagged buffer model – A tagged buffer model is used to transfer tagged buffers between the two members of the transfer (the local peer and the remote peer). A tagged buffer is explicitly advertised to the remote peer through exchange of a steering tag (STag), tagged offset, and length. An STag is simply an identifier of a tagged buffer on a node, and the tagged offset identifies the base address of the buffer. Tagged buffers are typically used for large data transfers, such as large data structures and disk I/O.
- Untagged buffer model – An untagged buffer model is used to transfer untagged buffers from the local peer to the remote peer. Untagged buffers are not explicitly advertised to the remote peer. Untagged buffers are typically used for small control messages, such as operation and I/O status messages.

RDMA data transfer operations

The RDMA protocol provides seven data transfer operations. Except for the RDMA read operation, each operation generates exactly one RDMA message. The RDMA information is included inside of fields within the DDP header.

With an RDMA aware network interface controller (RNIC), the data target and data source CPUs are not involved in the data transfer operations, so they can continue to do useful work. The RNIC is responsible for generating outgoing and processing incoming RDMA packets. In addition, the data is placed directly where the application advertised that it wanted it and pulled from where the application indicated it was located. This eliminates the copies of data that occur in the traditional operating system protocol stack on both the send and receive sides.

Send operations

RDMA uses four variations of the send operation:

- **Send operation** – A send operation transfers data from the data source (the peer sending the data payload) into a buffer that has not been explicitly advertised by the data target (the peer receiving the data payload). The send message uses the DDP untagged buffer model to transfer the ULP message into the data target's untagged buffer. Send operations are typically used to transfer small amounts of control data where the overhead of creating an STag for DDP does not justify the small amount of memory bandwidth consumed by the data copy.
- **Send with invalidate operation** – The send with invalidate message includes all functionality of the send message, plus the capability to invalidate a previously advertised STag. After the message has been placed and delivered at the data target, the data target's buffer identified by the STag included in the message can no longer be accessed remotely until the data target's ULP re-enables access and advertises the buffer again.

- **Send with solicited event** – The send with solicited event message is similar to the send message except that when the send with solicited event message has been placed and delivered, an event (for example, an interrupt) may be generated at the recipient, if the recipient is configured to generate such an event. This allows the recipient to control the amount of interrupt overhead it will encounter.
- **Send with solicited event and invalidate** – The send with solicited event and invalidate message combines the functionality of the send with invalidate message and the send with solicited event message.

RDMA write

An RDMA write operation uses an RDMA write message to transfer data from the data source to a previously advertised buffer at the data target. The ULP at the data target enables the data target tagged buffer for access and advertises the buffer's size (length), location (tagged offset), and STag to the data source through a ULP-specific mechanism such as a prior send message. The ULP at the data source initiates the RDMA write operation. The RDMA write message uses the DDP tagged buffer model to transfer the ULP message into the data target's tagged buffer. The STag associated with the tagged buffer remains valid until the ULP at the data target invalidates it or until the ULP at the data source invalidates it through a send with invalidate operation or a send with solicited event with invalidate operation.

RDMA read

The RDMA read operation transfers data to a tagged buffer at the data target from a tagged buffer at the data source. The ULP at the data source enables the data source tagged buffer for access and advertises the buffer's size (length), location (tagged offset), and STag to the data target through a ULP-specific mechanism such as a prior send message. The ULP at the data target enables the data target tagged buffer for access and initiates the RDMA read operation. The RDMA read operation consists of a single RDMA read request message and a single RDMA read response message, which may be segmented into multiple DDP segments.

The RDMA read request message uses the DDP untagged buffer model to deliver to the data source's RDMA read request queue the STag, starting tagged offset, and length for both the data source and the data target tagged buffers. When the data source receives this message, it then processes it and generates a read response message, which will transfer the data.

The RDMA read response message uses the DDP tagged buffer model to deliver the data source's tagged buffer to the data target, without any involvement from the ULP at the data source.

The data source STag associated with the tagged buffer remains valid until the ULP at the data source invalidates it or until the ULP at the data target invalidates it through a send with invalidate or send with solicited event and invalidate. The data target STag associated with the tagged buffer remains valid until the ULP at the data target invalidates it.

Terminate

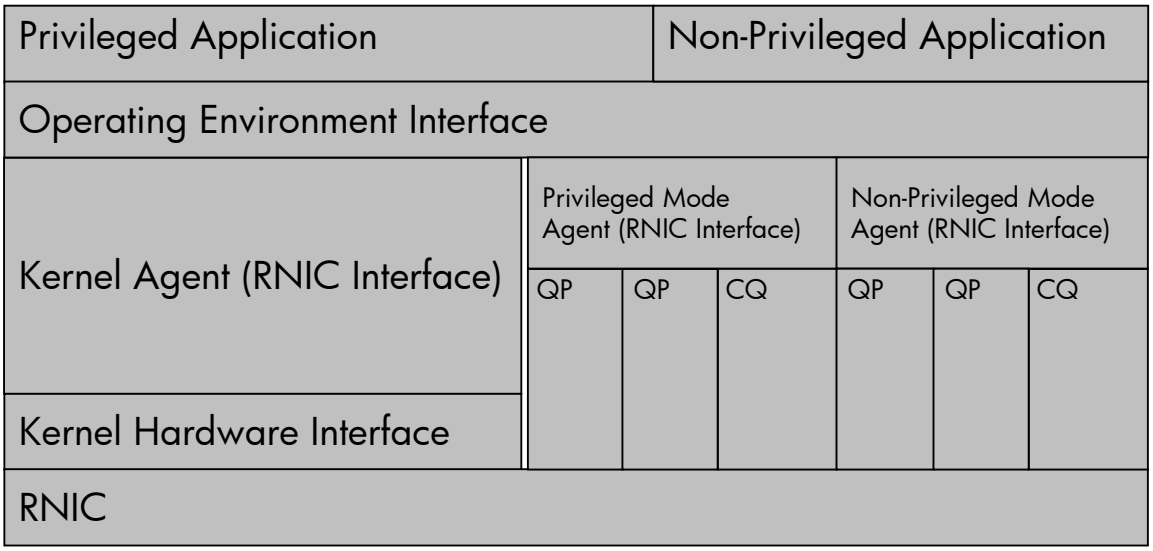
A terminate operation is included to perform an abortive tear down of the connection when an error is encountered. The terminate operation transfers to the data target information associated with an error that occurred at the data source. The terminate message uses the DDP untagged buffer model to transfer the message into the data target's untagged buffer.

Verbs

The RDMA Verbs Specification describes an abstract interface to an RNIC. The RNIC implements the RDMA protocol above a reliable transport, such as MPA over TCP. The verbs provide a semantic definition of the RNIC interface.

A verbs consumer uses the capabilities of the RNIC to accomplish some objective. In some instances, the verbs consumer may be an operating system kernel thread, in others a non-privileged application, and in still others, some special, privileged process. RDMA provides verbs consumers the capability to control data placement, eliminate data copy operations, and significantly reduce communications overhead and latencies by allowing one verbs consumer to place information directly in the memory of another verbs consumer, while preserving operating system and memory protection semantics. Figure 3 is a conceptual diagram that describes an architectural model that includes privileged mode consumers, non-privileged mode consumers, RNIC components, and the RNIC interface.

Figure 3. The architectural model includes privileged mode consumers, non-privileged mode consumers, RNIC components, and the RNIC interface.



RNIC interface

The RNIC interface provides coordination between the consumer of RNIC services and the RNIC. Verbs specify behavior of the RNIC and enable creation and management of queue pairs, management of the RNIC, management of work requests, and transferring error indications from the RNIC interface.

A fundamental function of the RNIC interface is management of the RNIC. This includes arranging access to them, accessing and modifying their attributes, and shutting them down.

Queue pairs are a key component required for the operation of the RNIC interface. They are the RNIC resource used by consumers to submit work requests to the RNIC interface. A queue pair is used to interact with an RDMA stream on an RNIC. There may be thousands of queue pairs per RNIC. Each queue pair provides the consumer with a single point of access to an individual RDMA stream.

Queue pairs consist of a send queue and a receive queue. Sends, RDMA reads, RDMA writes, and other operations are posted to send queues through work requests. Receive queues contain the buffer information for arriving send messages. Work requests provide the mechanism for consumers to place work queue elements onto the send and receive queues of a queue pair.

Completion queues allow the consumer to retrieve work request completion status. Notification mechanisms are provided to help a consumer identify when work requests have completed processing

in the RNIC interface. There may be thousands of completion queues per RNIC, and they can be associated with the send queues and receive queues in any combination the consumer requires.

Event handlers provide the mechanism for notifying consumers of asynchronous events that occur within the RNIC interface but that cannot be reported through the completion queues because they are asynchronous or they are not easily associated with a work completion.

Conclusion

The breakthrough economics and broad adoption of Ethernet, along with RDMA capabilities, open the possibility of truly pervasive fabrics in the future. New network connections, delivered through RNICs, will allow Ethernet infrastructures to efficiently handle most kinds of data transfers that today require specialized networks. Many of the concepts and technologies leveraged for RDMA over TCP come from high-end servers, such as HP NonStop and SuperDome servers, and from established networking used in storage area networks (SANs) and local area networks (LANs). As a leader in the development of innovative technologies and a trusted advisor on future data center directions that provide lasting value for IT customers, HP is closely involved with development of future fabric interconnect technologies. HP has led the development of RDMA over TCP and is a founding member of the RDMA Consortium. By leveraging HP's experience, emerging fabric interconnects will be more cost effective and able to compete in an industry-standard marketplace.

For more information

For more information on fabric technologies, read the HP white paper titled *emerging fabric technologies* at ftp://ftp.compaq.com/pub/supportinformation/papaer/tc030410tb_rev2_us.pdf.

For more information on RDMA, go to the RDMA Consortium web site at www.rdmaconsortium.org.

Call to action

To help us better understand and meet your needs for ISS technology information, please evaluate this paper by completing the short survey at www.zoomerang.com/survey.zgi?4LTTXWWEAY3F1Q01VYDGDAG5.

Note: This URL will be active through December 31, 2003. Please send questions and further comments about this paper to: TechCom@HP.com.

TC031003TB, 10/2003

invent