使用IB进行设计

使用IB进行设计

As with any network architecture, the InfniBand Architecture provides a service; in the case of InfniBand it provides a messaging service. That message service can be used to transport IPC (Inter Process Communication) messages, or to transport control and data messages for storage, or to transport messages associated with any other of a range of usages. This makes InfniBand different from a traditional TCP/IP/Ethernet network which provides a “byte stream” transport service, or Fibre Channel which provides a transport service specifc to the Fibre Channel wire protocol. By “application,” we mean, for example, a user application using the sockets interface to conduct IPC, or a kernel level fle system application executing SCSI block level commands to a remote block storage device.

和其它任何的网络体系结构相似，IB架构提供了一个Messaging Service服务。Messaging Service可以被用于传输进程间通讯信息、控制信息、存储数据的信息等。这使得IB区别于其它网络架构（其它网络架构一般采用字节流的形式传递数据）。

The key feature that sets InfniBand apart from other network technologies is that InfniBand is designed to provide message transport services directly to the application layer, whether it is a user application or a kernel application.This is very different from a traditional network which requires the application to solicit assistance from the operating system to transfer a stream of bytes. At first blush, one might think that such application level access requires the application to fully understand the networking and I/O protocols provided by the underlying network. Nothing could be further from the truth.

IB区别于其它网络技术的关键技术特征是：它在不同的应用程序层直接提供了消息传递服务，无论该应用是用户态应用程序还是内核态应用程序。这和传统的网络技术需要调用操作系统实现数据传输不同。

The top layer of the InfniBand Architecture defnes a software transport interface; this section of the specifcation defines the methods that an application uses to access the complete set of services provided by InfniBand.
IB体系结构的顶层定义了software transport interface， 这部分定义了应用程序通过IB访问完整IB服务的方法。

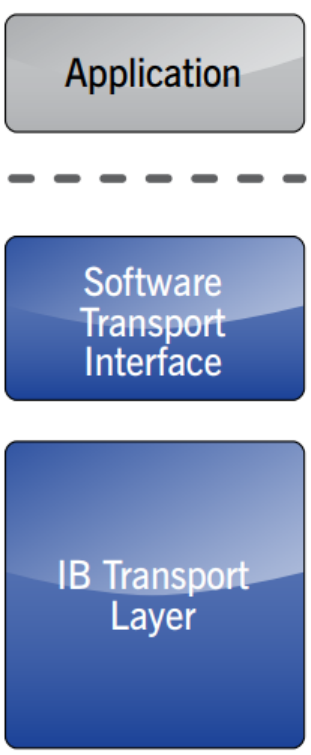
An application accesses InfniBand's message transport service by posting a Work Request (WR) to a work queue. As described earlier, the work queue is part of a structure, called a Queue Pair (QP) representing the endpoint of a channel connecting the application with another application. A QP contains two work queues; a Send Queue and a Receive Queue, hence the expression QP. A WR, once placed on the work queue, is in effect（实际上） an instruction to the InfniBand RDMA message transport to transmit a message on the channel, or perform some sort of control or housekeeping function on the channel.

应用程序通过提交一个Work Request （WR）请求到Work Queue来访问IB的Messaging Service服务。按照第二章的描述，Work Queue是一个数据结构，它被叫做Queue Pairs （QPs），它是channel的端点，在这里channel用来连接两个应用程序（该应用程序可以是用户态应用程序，也可以是内核态应用程序）。一个QP包含两个Work Queue，分别是一个Send Queue和一个Receive Queue。一个被放置在Work Queue中的Work Request实际上是一条指令，该条指令指示IB RDMA在channel上传输数据，或者是指示IB RDMA执行一系列的控制序列等。

The methods used by an application to interact with the InfniBand transport are called verbs and are defined in the software transport interface section of the specifcation. An application uses a POST SEND verb, for example, to request that the InfniBand transport send a message on the given channel. This is what is meant when we say that the InfniBand network stack "goes all the way up to the application layer." It provides methods that are used directly by an application to request service from the InfniBand transport. This is very
different from a traditional network, which requires the application to solicit assistance from the operating system in order to access the server's network services.

应用程序和IB交互使用的方法被叫做verbs，这些verbs被定义在software transport interface部分（在该文档的第二章已经说明过这个问题）。应用程序使用POST SEND verb请求IB在给定的channel上传输一条信息。IB提供的方法直接被应用层使用，应用层发出请求IB传输数据的请求。这使得IB不同于传统的网络技术，传统的网

络技术需要太多的操作系统参与用于传输字节流数据，而IB不需要操作系统的参与就能传输Message。



Application ("Consumer of InfiniBand message services"):
– posts 'work requests' to a queue
– each work request represents a message…a unit of work

Channel interface (verbs):
– allows the consumer to request services

InfiniBand Channel interface provider:
– Maintains work queues
– Manages address translation
– Provides completion and event mechanisms

IB Transport:
– Packetizes messages
– Provides transport service
  - reliable/unreliable, connected/unconnected, datagram
– Implements RDMA protocol
  - send/receive, RDMA r/w, Atomics
– Implements end-to-end reliability
– Assures reliable delivery

为什么在IB的体系架构中不定义APIs而是定义了一个充足的Verbs呢？我的理解这主要是由于如果要东一APIs，那么对于不同的操作系统需要定义不同的APIs，但是Verbs不用，Verbs指定了一系列需要的行为，它规定了一个具体的互操作，(Verbs)这同时保证了很好的跨平台特性。

说明

本文的内容大部分翻译自：Introduction to InfniBand™ for End Users 文档第四章。

参考文献

1. https://www.mellanox.com/pdf/whitepapers/Intro_to_IB_for_End_Users.pdf