RDMA技术调研

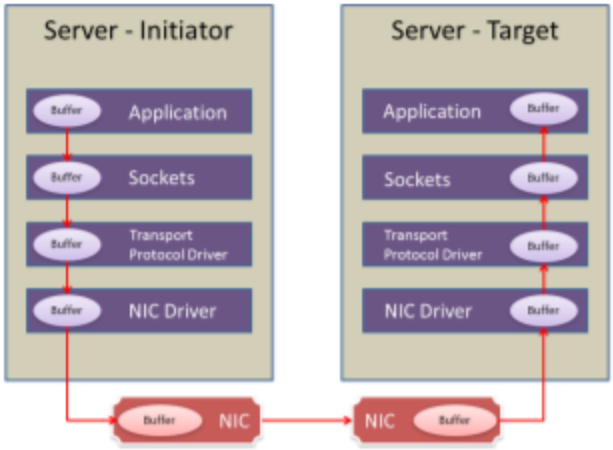
一.RDMA介绍

1.1 RDMA介绍

直接内存访问 (DMA) 是设备无需 CPU 干预即可直接访问主机内存的能力。

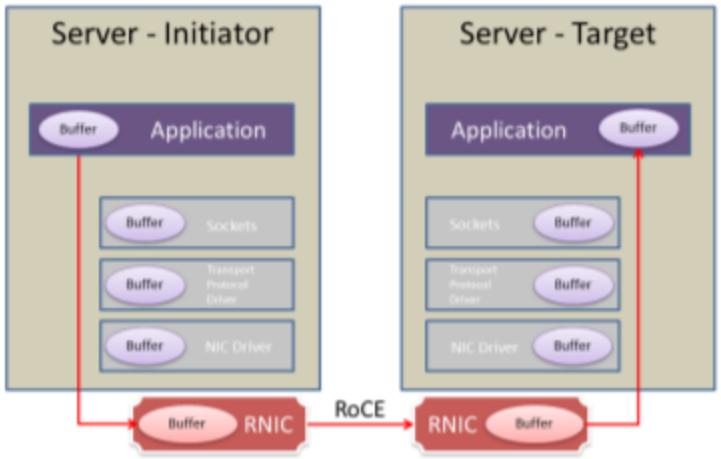
远程直接内存访问（RDMA）是访问（即读取或写入）远程机器上的内存而不中断该系统上 CPU 处理的能力。

- 传统网络中的转发过程：



传统网络协议栈的过程下，无论是发送端还是接收端，都需要CPU的指挥和控制，包括网卡的控制，中断的处理，报文的封装和解析等等。

- 使用了RDMA技术之后网络的转发过程：



在使用了RDMA技术时，两端的CPU几乎不用参与数据传输过程（只参与控制面）。本端的网卡直接从内存的用户空间DMA拷贝数据到内部存储空间，然后硬件进行各层报文的组装后，通过物理链路发送到对端网卡。对端的RDMA网卡收到数据后，剥离各层报文头和校验码，通过DMA将数据直接拷贝到用户空间内存中。

1.2 RDMA相较于传统网络的核心优势

RDMA是现代高速网络的具体实现方案，现主要用于数据中心内部的存储服务器之间的数据交互，相较于传统网络，主要有三大核心优势:

- **0拷贝:**

应用程序可以在没有网络软件堆栈参与的情况下执行数据传输，并且数据被直接发送到缓冲区，而无需在网络层之间进行复制。

- **内核旁路:**

应用程序可以直接从用户空间执行数据传输，而无需执行上下文切换。

- **CPU卸载:**

应用程序可以访问远程内存而不会消耗远程机器中的任何 CPU。远程内存机器将在没有任何远程进程（或处理器）干预的情况下被读取。远程 CPU 中的缓存不会被访问的内存内容填满。

1.3 什么样的业务场景需要用到RDMA

可以在至少需要以下一项的场景中找到 RDMA：

- 低延迟 - 例如：HPC、金融服务、web 3.0

- 高带宽 - 例如：HPC、医疗设备、存储和备份系统、云计算
- CPU 占用空间小 - 例如：HPC、云计算

1.4 三种RDMA协议介绍

RDMA本身指的是一种技术，具体协议层面，包含Infiniband（IB），RDMA over Converged Ethernet（RoCE）和internet Wide Area RDMA Protocol（iWARP）。三种协议都符合RDMA标准，使用相同的上层接口，在不同层次上有一些差别。

- Infiniband:

从一开始就原生支持 RDMA 的新一代网络协议。由于这是一项新的网络技术，因此需要支持该技术的网卡和交换机。

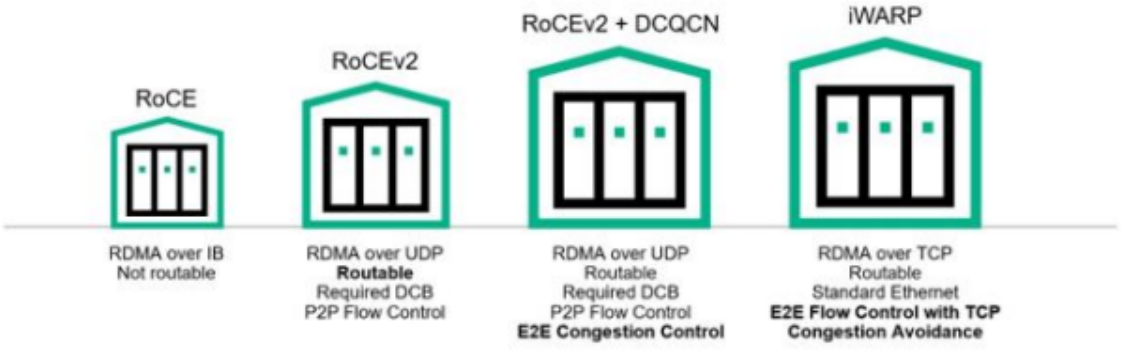
- RoCE:

一种允许在以太网网络上执行 RDMA 的网络协议。它的下部网络标头是以太网标头，其上部网络标头（包括数据）是 InfiniBand 标头。这允许在标准以太网基础设施（交换机）上使用 RDMA。只有 NIC 应该是特殊的并且支持 RoCE。

- iWARP:

一种允许通过 TCP 执行 RDMA 的网络协议。有些功能存在于 IB 和 RoCE 中，但在 iWARP 中不受支持。这允许在标准以太网基础设施（交换机）上使用 RDMA。只有 NIC 应该是特殊的并支持 iWARP（如果使用 CPU 卸载），否则所有 iWARP 堆栈都可以在 SW 中实现并失去大部分 RDMA 性能优势。

发展脉络如下图所示



1.5 如何使用RDMA

为了使用 RDMA，需要一个具有 RDMA 功能的网络适配器（例如 Mellanox 的 Connect-X 系列网卡）。

网络的链路层协议可以是以太网或 InfiniBand——两者都可以传输基于 RDMA 的应用程序。

二.RDMA术语及基本流程介绍

2.1 RDMA术语

在学习RDMA过程时总是会涉及到各种专有名词及术语，常用的缩略语及其全称如下所示。

缩略语	全称
WQ	Work Queue
WQE	Work Queue Entry/Element
QP	Queue Pair
SQ	Send Queue
RQ	Receive Queue
SRQ	Shared Receive Queue
CQ	Completion Queue
CQE	Completion Queue Entry/Element
WR	Work Request
WC	Work Completion

- WQ

WQ即Work Queue，即存放了工作请求的队列。

- QP

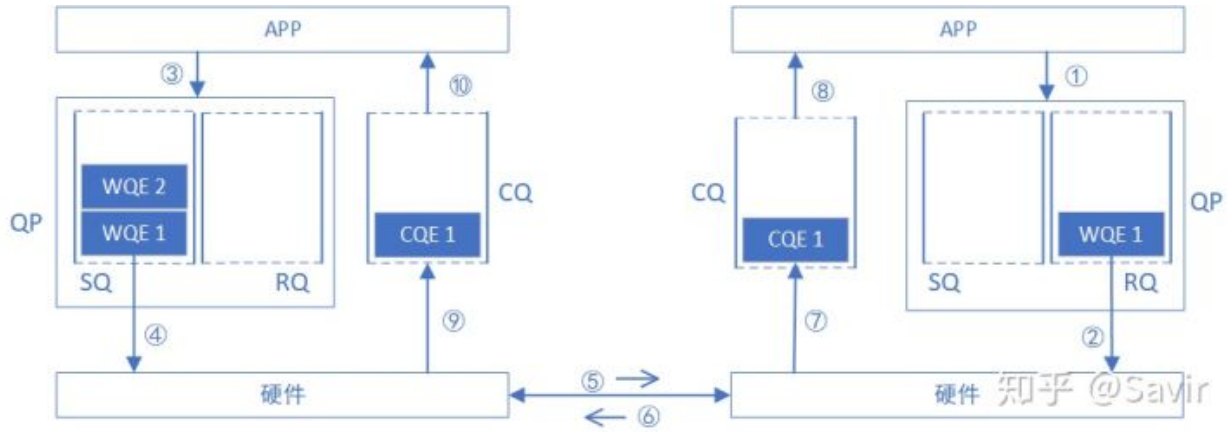
当需要向外发送工作请求的时候，就有了SQ的概念；当需要从外接受工作请求的时候，就有了RQ的概念。SQ和RQ统称为QP。

- CQ

当任务完成时需要一个CQ，即Completion Queue，用于宣告工作请求的顺利完成。

2.2 RDMA通信的基本流程

下面我们把CQ和WQ（QP）放在一起，看一下一次SEND-RECV操作中，软硬件的互动（图中序号顺序不表示实际时序）：



简而言之。

在通信前，接收端APP需要提前在RQ中准备好接收任务工作区。

发送起始，发送端APP需要向SQ下达SEND任务。

接收端收到数据后会第一时间向对方回复ACK，之后生成CQ，并通过CQ告知APP收到了消息。

发送端也会通过接收对方刚刚生成的ACK，生成CQ，进而通知APP任务完成。

三.RDMA编程方法介绍

3.1 RDMA编程基本概念介绍

通信操作

RDMA支持以下4种通信操作

- SEND/RECV

SEND操作允许将数据发送到远程QP的接收队列。接收器之前必须发布了一个接收缓冲区才能接收数据。发送方无法控制数据在远程主机中的所在位置。

- WRITE/READ

系统将从远程主机中读取一段内存。调用者指定要复制到的远程虚拟地址以及本地内存地址。

在执行RDMA操作之前，远程主机必须提供适当的权限来访问其内存。一旦设置了这些权限，就会执行RDMA读取操作，而无需发出任何通知到远程主机。对于RDMA的读写操作，远程端都是不知道正在执行此操作的（除了准备权限和资源之外）。

- ATOMIC

对RDMA操作的原子扩展。

- SRQ_RECV

通过共享RQ的方式，将原先的一个QP中一个SQ对应一个RQ的模式，变成了多个SQ共用一个RQ的模式，减少了内存占用。

传输模式

- RC

可靠连接，类似于TCP

- UC

不可靠连接，做了连接，但是没有做重传

- UD

不可靠数据报，类似于UDP

几种传输模式和支持的操作如下表所示：

Operation	UD	UC	RC	RD
Send (with immediate)	X	X	X	X
Receive	X	X	X	X
RDMA Write (with immediate)		X	X	X
RDMA Read			X	X
Atomic: Fetch and Add/ Cmp and Swap			X	X
Max message size	MTU	1GB	1GB	1GB

- DC

使用 UD 的主要优点是单个 QP 可以用来与任何其他 QP 对话；

而使用 RC 时，需要创建与通信对等点数量一样多的 QP。

Mellanox 的最新优化引入了 DCT（动态连接传输），可以很好地解决 QP 可扩展性问题。

关键概念

- **Send Request**

SR定义了将发送多少数据，从哪里、如何、发送到哪里。

- **Receive Request**

RR定义了要为非RDMA操作接收数据的缓冲区。如果没有定义缓冲区，并且发送器尝试发送操作或RDMA立即写入，则将发送接收未准备好(RNR)错误。

- **Completion Queue**

完成队列是一种通知应用程序关于已结束的工作请求（状态、操作码、大小、来源）信息的机制。

- **Memory Registration**

内存注册是一种机制，它允许应用程序使用虚拟地址来描述一组虚拟连续的内存位置或一组物理上连续的内存位置到网络适配器，作为一个虚拟连续的缓冲器。

- **Protection Domain**

保护域用于将队列对与内存区域和内存窗口关联起来，作为启用和控制网络适配器对主机系统内存的访问的一种手段。

- **Scatter Gather**

包含

Address：数据将从其中收集或分散到的本地数据缓冲区的地址。

Size：将从该地址读取/写入的数据的大小。

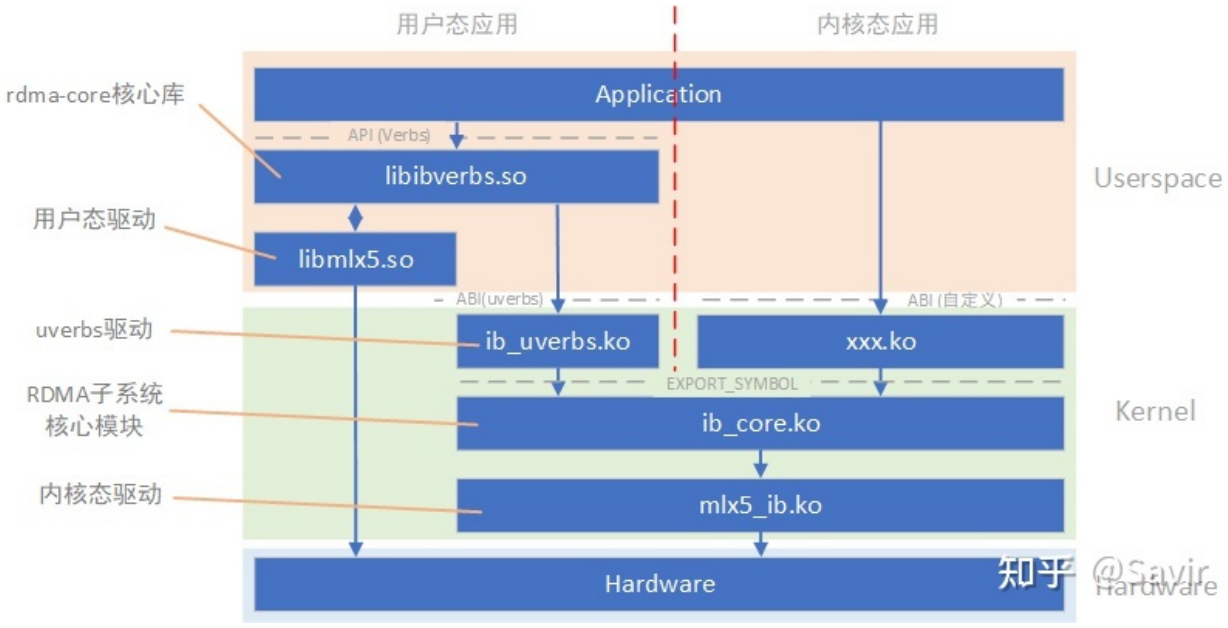
L_key：已注册到此缓冲区的先生的本地密钥。

- **Polling**

轮询CQ是获取已发布的WR（发送或接收）的详细信息。

3.2 rdma-core介绍

rdma-core:RDMA Core Userspace Libraries and Daemons，是RDMA给用户态程序提供的库，也是进行RDMA编程的基础，在系统中具体所在的位置如下图所示。



具体到rdma-core内部，主要是两个库在起作用，即**Libibverbs library**和**Librdmacm library**

- libibverbs:

Libibverbs库使用户空间进程使用远程直接内存访问（RDMA）动词

- librdmacm:

librdmacm库在verbs的基础上，提供通信管理器（CM）功能和一套通用的远程直接内存访问（RDMA）的CM接口

更详细的编程方法介绍，可以参考mellanox官方给出的216页完整手册。

> https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ff157e8e-9bbf-418f-a877-5d3694ff0c41/RDMA_Aware_Programming_user_manual.pdf

---

小结

本文主要从RDMA是什么、RDMA通信逻辑以及如何进行RDMA编程三个方面初步介绍RDMA技术。

具体到更实际的代码编写、通信过程、协议优化等，以后会新开一篇介绍，敬请期待。