# Ubuntu 20.04 OS安装GPUDirect RDMA操作指南

## 1、准备

Ubuntu 20.04操作系统中，安装以下程序：

* GNU编译器(g++, gcc)
* CUDA 11.0,, 11.4, etc.
* GDRCopy源码
* Mellanox OFED驱动源码
* nv-peer-memory源码
* UCX 1.13
* OpenMPI 4.1.4

## 2、安装步骤

统一安装路径：export MY\_PATH=/home/lijian/GPUDirect-RDMA

2.1安装Mellanox网卡驱动

(1)下载源码

Ubuntu 20.04系统的OFED驱动镜像文件：

<http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-5.4-3.5.8.0&mname=MLNX_OFED_LINUX-5.4-3.5.8.0-ubuntu20.04-x86_64.iso>

（2）挂载镜像文件

mount -o loop mlnx-en-5.4-3.6.8.1-ubuntu20.04-x86\_64.iso /mnt/iso

**卸载：umount /mnt/iso/**

（3）安装OFED驱动

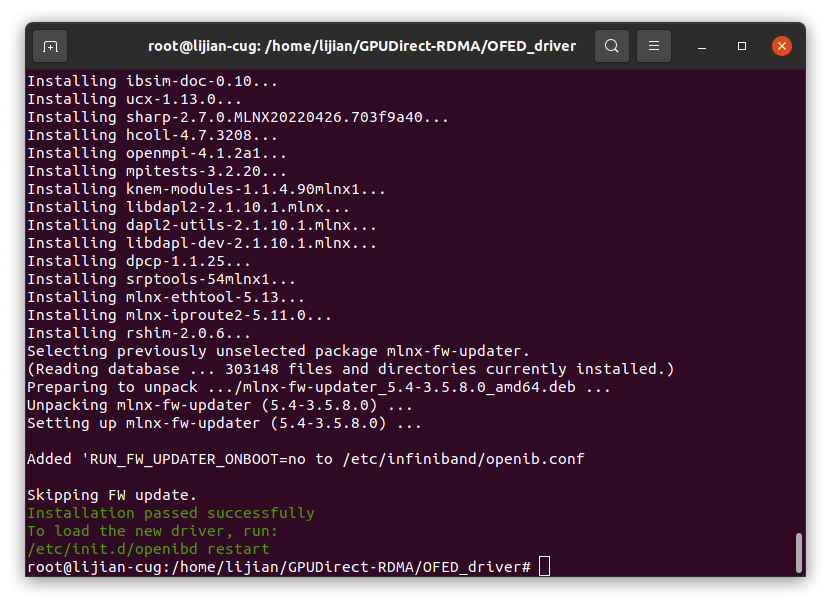
/mnt/mlnxofedinstall --without-dkms --add-kernel-support --kernel 5.4.0-131-generic --without-fw-update -force

注：查询Ubuntu 20.04系统当前的内核版本。

（4）启动Mellanox NIC驱动

/etc/init.d/openibd restart

OFED驱动安装并启动后屏幕显示如下：



2.2安装GDRCopy

(1)下载源码并解压

wget gdrcopy-2.3.tar.gz

tar xf gdrcopy-2.3.tar.gz

(2)安装

make prefix=$MY\_PATH/gdrcopy\_install CUDA=/usr/local/cuda-11.0 all install

./insmod.sh

问题：fatal error: check.h: No such file or directory

解决：sudo apt-get install check

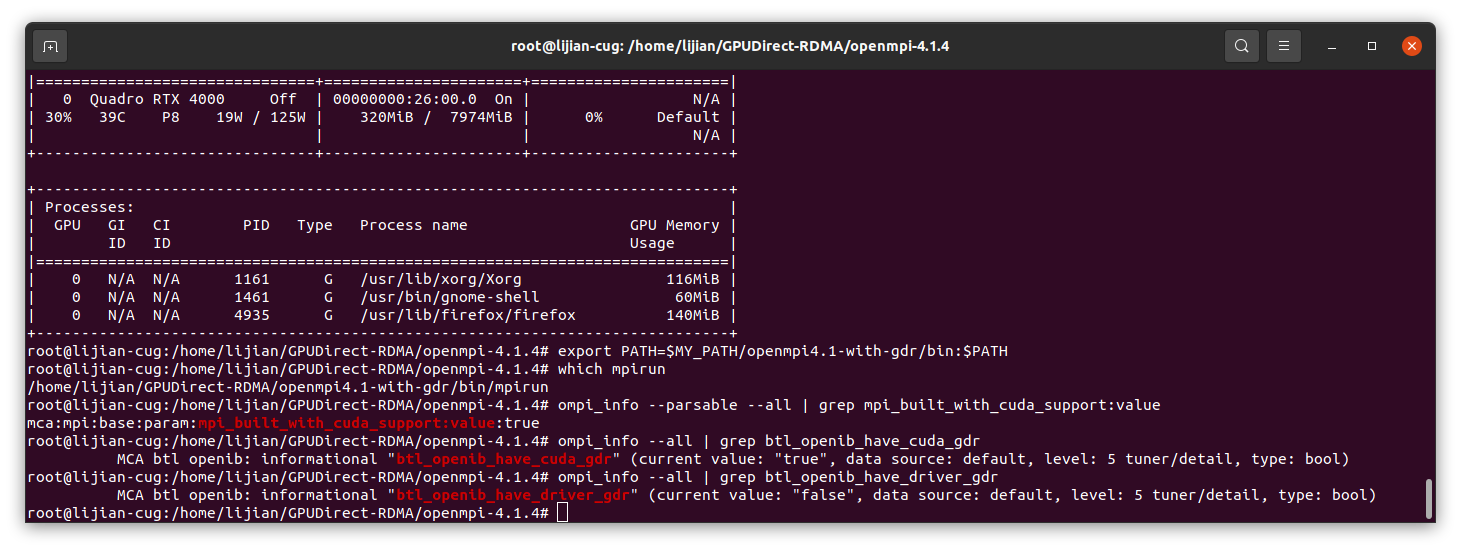
注：执行./insmod.sh，需要超级用户权限。

2.3安装nv\_peer\_memory

nv\_peer\_memory的功能是：实现GPU设备间的直连，基于OFED驱动。如果不安装nv\_peer\_memory，GPUDirect RDMA的驱动无法工作，即：

运行：ompi\_info --all | grep btl\_openib\_have\_driver\_gdr

显示：MCA btl: informational "btl\_openib\_have\_driver\_gdr" (current value: "false", …)



解决方法：需要安装nv\_peer\_memory

（1）下载nv\_peer\_memory源码并解压

wget https://github.com/Mellanox/nv\_peer\_memory/archive/refs/tags/1.3-0.tar.gz

tar xf nv\_peer\_memory-1.3.0.tar.gz

（2）Ubuntu 20.04系统下使用DEB安装方式

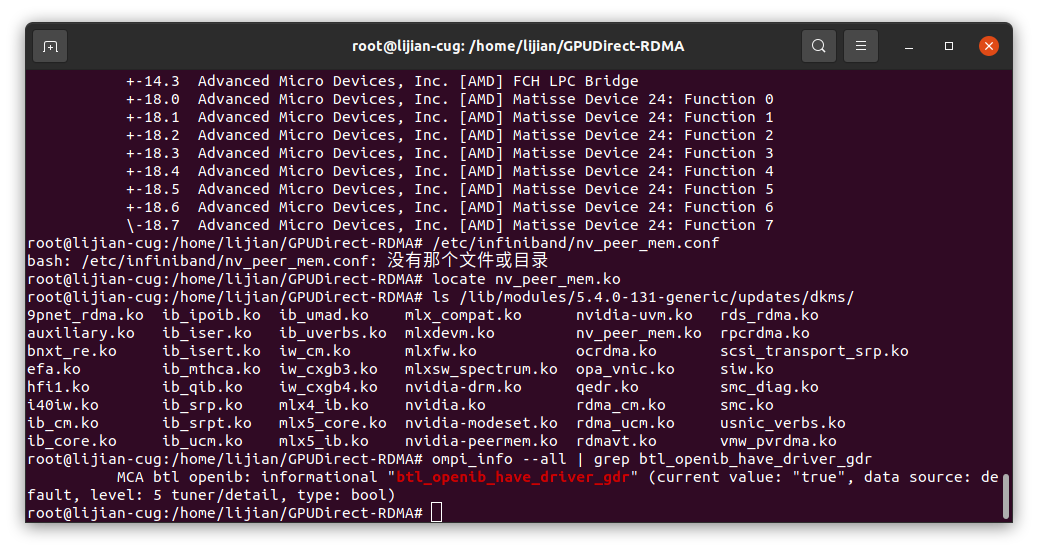
dpkg-buildpackage -us -uc

dpkg -i ../nvidia-peer-memory\_1.3-0\_all.deb

或者

dpkg -i ../nvidia-peer-memory-dkms\_1.3-0\_all.deb

（3）安装成功后，安装的程序包括：nv\_peer\_mem.ko等。



使用/etc/init.d/nv\_peer\_mem服务文件start/stop/status管理添加的核模块；

/etc/infiniband/nv\_peer\_mem.conf控制是否在启动时加载核模块（默认：YES）

（4）注意

为实现NIC与GPU的最佳性能，两者必须在物理上位于相同的I/O根环境，可以使用lspci -tv查看是否符合。

2.4安装带GDRCopy的UCX-1.13

为编译安装带GPUdirect RMDA功能的OpenMPI库做准备。

（1）下载UCX源码，并解压

wget https://github.com/openucx/ucx/archive/refs/tags/v1.13.1.tar.gz

tar xf v1.13.1.tar.gz

（2）配置并安装

./configure --prefix=$MY\_PATH/ucx-with-gdrcopy --with-cuda=/usr/local/cuda-11.0 --with-gdrcopy=$MY\_PATH/gdrcopy\_install

make -j$(nproc) install

2.5安装具备UCX CUDA支持的OpenMPI-4.1

（1）下载OpenMPI-4.0源码，并解压。

wget https://download.open-mpi.org/release/open-mpi/v4.1/openmpi-4.1.5.tar.gz

tar xf openmpi-4.1.5.tar.gz

（2）配置并安装。

./configure --prefix=$MY\_PATH/openmpi4.1-with-gdr --with-cuda=/usr/local/cuda-11.0 --with-ucx=$MY\_PATH/UCX-with-gdrcopy

make -j$(nproc) install

（3）设置临时环境变量。

export PATH=$MY\_PATH/openmpi4.1-with-gdr/bin:$PATH

## 3、检验CUDA-aware OpenMPI的安装及GPUDirect RDMA

（1）查看构建的MPI库是否具备了CUDA-aware支持?

shell$ ompi\_info --parsable --all | grep mpi\_built\_with\_cuda\_support:value

屏幕显示：

mca:mpi:base:param:mpi\_built\_with\_cuda\_support:value:true

表示：支持!

（2）查看MPI库是否具备支持GPUDirect RDMA功能？

shell$ ompi\_info --all | grep btl\_openib\_have\_cuda\_gdr

屏幕显示：

MCA btl: informational "btl\_openib\_have\_cuda\_gdr" (current value: "true", data source: default, level: 4 tuner/basic, type: bool)

表示：支持!

（3）查看OFED栈是否支持GPUDirect RDMA功能？

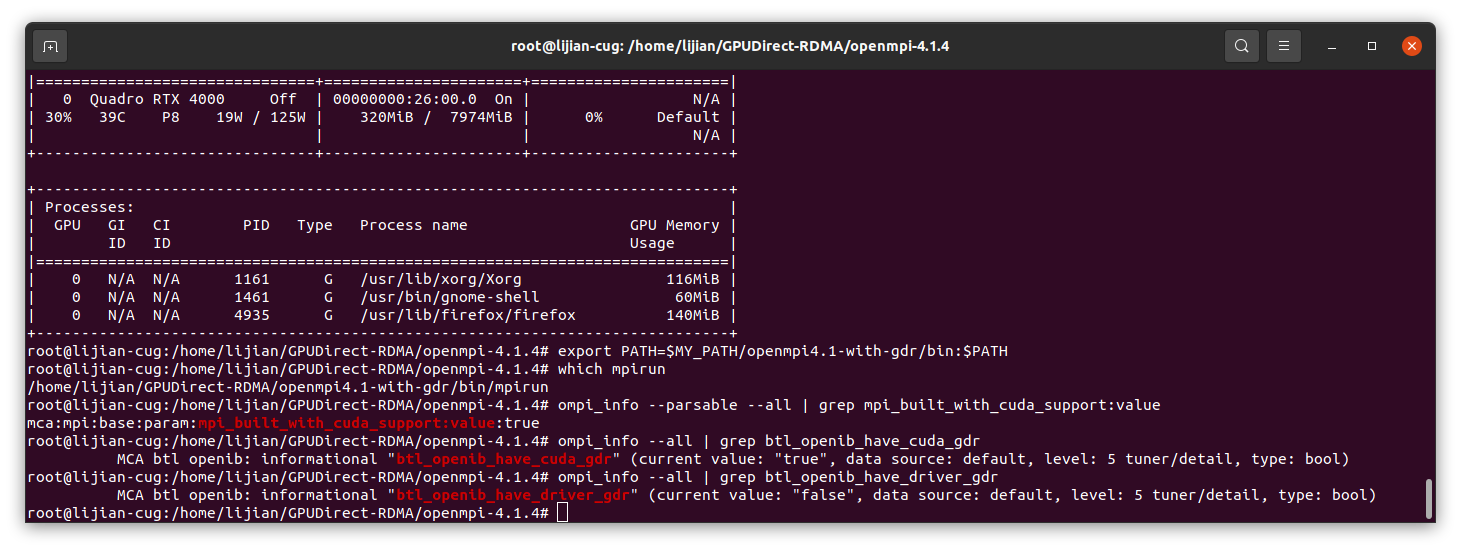
shell$ ompi\_info --all | grep btl\_openib\_have\_driver\_gdr

屏幕显示：

MCA btl: informational "btl\_openib\_have\_driver\_gdr" (current value: "true", data source: default, level: 4 tuner/basic, type: bool)

表示：支持!

注：上面的3个检测都必须是"true"状态才行。



## 4、运行支持GPUDirect RDMA的应用程序

mpirun --mca btl\_openib\_want\_cuda\_gdr 1 ...

例如mgcfd的多GPU运行脚本：

mpirun -hostfile $HOSTFILE -np 2 --mca plm\_rsh\_agent /opt/batch/agent/tools/dstart -x LD\_LIBRARY\_PATH -mca pml ucx -x UCX\_NET\_DEVICES=mlx5\_0:1 -mca btl ^vader,tcp,openib,uct -x UCX\_TLS=self,sm,rc --bind-to core --map-by socket --rank-by core mgcfd\_mpi\_cuda -i input.dat -m parmetis -r kway

注：（1）UCX\_TLS可用于调整MPI使用的通信协议。OpenMPI的UCX\_TLS可以添加：rc, cuda\_cpy，实现多GPU间的GPUDirect RDMA数据传输。

（2）可以使用：ucx\_info -d | grep cuda

查看是否UCX支持CUDA。