



登临 HammingTM V2

NCCL 使用说明

DL-DG/SW-063A-01

2024-10-30

Copyright©苏州登临科技有限公司，2019 - 2025，版权所有。

未经苏州登临科技有限公司事先书面同意，不得以任何形式或方式复制或传播本文件的任何部分。

商标和许可



和其它苏州登临科技有限公司的其它登临科技的图标为苏州登临科技有限公司的商标。本手册中提及的所有其他商标均为其各自所有者的财产。

通知

所购买的产品、服务和特性由苏州登临科技有限公司与客户签订的合同规定。本文件中描述的所有或部分产品、服务和特性可能不在采购范围或使用范围内。除非合同中另有规定，本文件中的所有声明、信息和建议均按“原样”提供，无任何明示或暗示的保证或陈述。

本手册中的信息如有更改，恕不另行通知。本文件在编制过程中已尽一切努力确保内容的准确性，本文件中的所有声明、信息和建议不构成任何明示或暗示的保证。

苏州登临科技有限公司

苏州工业园区扬富路11号南岸新地一期商务楼5号1101室，江苏，中国

<http://www.denglin.ai>

Email : support@denglin.ai

更新历史

版本	更新描述
01	第一次发布

目录

目录

1 简介

2 功能说明

2.1 intra-node

2.2 inter-node

3 使用说明

3.1 intra-node

3.2 inter-node

1 简介

登临Hamming™ V2的nccl基于开源社区[nccl v2.12.12-1](#)进行开发并且基于pci-e 进行p2p通信，对应nccl版本的[官方文档](#)。

以下简称登临NCCL为DCCL，NVIDIA原生NCCL为NCCL。

2 功能说明

DCCL和NCCL功能基本一致，可运行基本的collection op和p2p op以及大部分nccl api，以下分为intra node和inter node说明DCCL尚未支持的功能。

2.1 intra-node

1. DCCL的bfloat16/double相关操作会将bfloat16/double转为float实现，因为如果使用bfloat16/double可能会有精度问题；
2. Group call Aggregated Operations的功能不支持，如果使用Group call，DCCL会将group内的多个operation拆分成多个operation分别launch，并不能一次性将所有operation打包然后launch到一个kernel里；
3. DCCL不支持CUDA Graphs；
4. 自定义的reduction type在NCCL上未经过测试；
5. DCCL不支持Simple proto，因此DCCL的Point-to-point communication primitives(send/recv)是利用LL proto协议实现的（NCCL是采用Simple proto实现的），针对send/recv组成的neighbor exchange算法DCCL未能pass，NCCL也未能pass；
6. DCCL 支持LL和LL128协议，但LL128协议有更改：NCCL的LL128 proto 表示为120B Data + 8B flag，DCCL的LL128 proto 表现为2 * (56B data + 8B flag)；
7. DCCL针对LL proto launch bounds 512 threads，针对LL128 launch bounds 480 threads；
8. DCCL在intra-node对Ring是充分测试过的，对Tree是没有测试过的，当然intra node使用Tree意义也不是很大。

2.2 inter-node

1. inter-node使用需要设置**RNIC的MaxPayload和MaxReadReq为128B**才可使用；
2. 对于send/recv op，DCCL在inter-node仅支持LL proto；
3. 对于collection op，DCCL在inter-node支持ring + LL proto、ring + LL128的组合，同时也支持tree + LL 的组合（此组合并未进行充分测试，理论上是支持的）；
4. 不支持collnet插件；
5. PXN功能不支持。

3 使用说明

可以参考《登临Hamming V2 GPU P2P使用说明》主要对机器或板卡做一些配置。

如果拿到的是2024.10.16之后的SDK版本，不需设置ko的加载参数；否则需要按文档《登临Hamming V2 GPU P2P使用说明》设置参数。

3.1 intra-node

1. 加载denglin.ko，enable sdk环境

```
sudo insmod sdk/kernel/denglin.ko
source sdk/env.sh
```

2. 配置NCCL，为了尽量使用理想的状态运行，我们设置以下环境变量

```
export NCCL_P2P_LEVEL=SYS
export NCCL_PROTO=LL
export NCCL_ALGO=Ring
```

3.2 inter-node

1. 加载denlin.ko, denglin-peermem.ko，enable sdk环境

```
sudo insmod sdk/kernel/denglin.ko
sudo insmod sdk/kernel/denglin-peermem.ko
source sdk/env.sh
```

2. 配置NCCL环境变量

```
export NCCL_P2P_LEVEL=SYS
export NCCL_PROTO=LL
export NCCL_ALGO=Ring
```

另外配置下多机网络相关的宏，在DCCL中这些宏和NCCL保持一致。