    * [ncclGetErrorString](../api/comms.html#ncclgeterrorstring)
    * [ncclGetVersion](../api/comms.html#ncclgetversion)
    * [ncclGetUniqueId](../api/comms.html#ncclgetuniqueid)
    * [ncclCommInitRank](../api/comms.html#ncclcomminitrank)
    * [ncclCommInitAll](../api/comms.html#ncclcomminitall)
    * [ncclCommInitRankConfig](../api/comms.html#ncclcomminitrankconfig)
    * [ncclCommInitRankScalable](../api/comms.html#ncclcomminitrankscalable)
    * [ncclCommSplit](../api/comms.html#ncclcommsplit)
    * [ncclCommFinalize](../api/comms.html#ncclcommfinalize)
    * [ncclCommDestroy](../api/comms.html#ncclcommdestroy)
    * [ncclCommAbort](../api/comms.html#ncclcommabort)
    * [ncclCommGetAsyncError](../api/comms.html#ncclcommgetasyncerror)
    * [ncclCommCount](../api/comms.html#ncclcommcount)
    * [ncclCommCuDevice](../api/comms.html#ncclcommcudevice)
    * [ncclCommUserRank](../api/comms.html#ncclcommuserrank)
    * [ncclCommRegister](../api/comms.html#ncclcommregister)
    * [ncclCommDeregister](../api/comms.html#ncclcommderegister)
    * [ncclMemAlloc](../api/comms.html#ncclmemalloc)
    * [ncclMemFree](../api/comms.html#ncclmemfree)
  * [Collective Communication Functions](../api/colls.html)
    * [ncclAllReduce](../api/colls.html#ncclallreduce)
    * [ncclBroadcast](../api/colls.html#ncclbroadcast)
    * [ncclReduce](../api/colls.html#ncclreduce)
    * [ncclAllGather](../api/colls.html#ncclallgather)
    * [ncclReduceScatter](../api/colls.html#ncclreducescatter)
  * [Group Calls](../api/group.html)
    * [ncclGroupStart](../api/group.html#ncclgroupstart)

* [NCCL_IB_ROCE_VERSION_NUM](../env.html#nccl-ib-roce-version-num)
  * [Values accepted](../env.html#id14)
* [NCCL_IB_SL](../env.html#nccl-ib-sl)
  * [Values accepted](../env.html#id15)
* [NCCL_IB_TC](../env.html#nccl-ib-tc)
  * [Values accepted](../env.html#id16)
* [NCCL_IB_FIFO_TC](../env.html#nccl-ib-fifo-tc)
  * [Values accepted](../env.html#id17)
* [NCCL_IB_RETURN_ASYNC_EVENTS](../env.html#nccl-ib-return-async-events)
  * [Values accepted](../env.html#id18)
* [NCCL_OOB_NET_ENABLE](../env.html#nccl-oob-net-enable)
  * [Values accepted](../env.html#id19)
* [NCCL_OOB_NET_IFNAME](../env.html#nccl-oob-net-ifname)
  * [Values accepted](../env.html#id20)
* [NCCL_UID_STAGGER_THRESHOLD](../env.html#nccl-uid-stagger-threshold)
  * [Values accepted](../env.html#id21)
* [NCCL_UID_STAGGER_RATE](../env.html#nccl-uid-stagger-rate)
  * [Values accepted](../env.html#id22)
* [NCCL_NET](../env.html#nccl-net)
  * [Values accepted](../env.html#id23)
* [NCCL_NET_PLUGIN](../env.html#nccl-net-plugin)
  * [Values accepted](../env.html#id24)
* [NCCL_TUNER_PLUGIN](../env.html#nccl-tuner-plugin)
  * [Values accepted](../env.html#id25)
* [NCCL_PROFILER_PLUGIN](../env.html#nccl-profiler-plugin)
  * [Values accepted](../env.html#id26)
* [NCCL_IGNORE_CPU_AFFINITY](../env.html#nccl-ignore-cpu-affinity)

* [NCCL_SHM_DISABLE](../env.html#nccl-shm-disable)
  * [Values accepted](../env.html#id40)
* [NCCL_BUFFSIZE](../env.html#nccl-buffsize)
  * [Values accepted](../env.html#id41)
* [NCCL_NTHREADS](../env.html#nccl-nthreads)
  * [Values accepted](../env.html#id42)
* [NCCL_MAX_NCHANNELS](../env.html#nccl-max-nchannels)
  * [Values accepted](../env.html#id43)
* [NCCL_MIN_NCHANNELS](../env.html#nccl-min-nchannels)
  * [Values accepted](../env.html#id44)
* [NCCL_CHECKS_DISABLE](../env.html#nccl-checks-disable)
  * [Values accepted](../env.html#id45)
* [NCCL_CHECK_POINTERS](../env.html#nccl-check-pointers)
  * [Values accepted](../env.html#id46)
* [NCCL_LAUNCH_MODE](../env.html#nccl-launch-mode)
  * [Values accepted](../env.html#id47)
* [NCCL_IB_DISABLE](../env.html#nccl-ib-disable)
  * [Values accepted](../env.html#id48)
* [NCCL_IB_AR_THRESHOLD](../env.html#nccl-ib-ar-threshold)
  * [Values accepted](../env.html#id49)
* [NCCL_IB_QPS_PER_CONNECTION](../env.html#nccl-ib-qps-per-connection)
  * [Values accepted](../env.html#id50)
* [NCCL_IB_SPLIT_DATA_ON_QPS](../env.html#nccl-ib-split-data-on-qps)
  * [Values accepted](../env.html#id51)
* [NCCL_IB_CUDA_SUPPORT](../env.html#nccl-ib-cuda-support)
  * [Values accepted](../env.html#id52)
* [NCCL_IB_PCI_RELAXED_ORDERING](../env.html#nccl-ib-pci-relaxed-ordering)

  * [Values accepted](../env.html#id53)
* [NCCL_IB_ADAPTIVE_ROUTING](../env.html#nccl-ib-adaptive-routing)
  * [Values accepted](../env.html#id54)
* [NCCL_IB_ECE_ENABLE](../env.html#nccl-ib-ece-enable)
  * [Values accepted](../env.html#id55)
* [NCCL_MEM_SYNC_DOMAIN](../env.html#nccl-mem-sync-domain)
  * [Values accepted](../env.html#id56)
* [NCCL_CUMEM_ENABLE](../env.html#nccl-cumem-enable)
  * [Values accepted](../env.html#id57)
* [NCCL_CUMEM_HOST_ENABLE](../env.html#nccl-cumem-host-enable)
  * [Values accepted](../env.html#id58)
* [NCCL_NET_GDR_LEVEL (formerly NCCL_IB_GDR_LEVEL)](../env.html#nccl-net-gdr-level-formerly-nccl-ib-gdr-level)
  * [Values accepted](../env.html#id59)
  * [Integer Values (Legacy)](../env.html#id60)
* [NCCL_NET_GDR_READ](../env.html#nccl-net-gdr-read)
  * [Values accepted](../env.html#id61)
* [NCCL_NET_SHARED_BUFFERS](../env.html#nccl-net-shared-buffers)
  * [Value accepted](../env.html#id62)
* [NCCL_NET_SHARED_COMMS](../env.html#nccl-net-shared-comms)
  * [Value accepted](../env.html#id63)
* [NCCL_SINGLE_RING_THRESHOLD](../env.html#nccl-single-ring-threshold)
  * [Values accepted](../env.html#id64)
* [NCCL_LL_THRESHOLD](../env.html#nccl-ll-threshold)
  * [Values accepted](../env.html#id65)
* [NCCL_TREE_THRESHOLD](../env.html#nccl-tree-threshold)
  * [Values accepted](../env.html#id66)

* [NCCL_ALGO](../env.html#nccl-algo)
  * [Values accepted](../env.html#id67)
* [NCCL_PROTO](../env.html#nccl-proto)
  * [Values accepted](../env.html#id68)
* [NCCL_NVB_DISABLE](../env.html#nccl-nvb-disable)
  * [Value accepted](../env.html#id69)
* [NCCL_PXN_DISABLE](../env.html#nccl-pxn-disable)
  * [Value accepted](../env.html#id70)
* [NCCL_P2P_PXN_LEVEL](../env.html#nccl-p2p-pxn-level)
  * [Value accepted](../env.html#id71)
* [NCCL_RUNTIME_CONNECT](../env.html#nccl-runtime-connect)
  * [Value accepted](../env.html#id72)
* [NCCL_GRAPH_REGISTER](../env.html#nccl-graph-register)
  * [Value accepted](../env.html#id74)
* [NCCL_LOCAL_REGISTER](../env.html#nccl-local-register)
  * [Value accepted](../env.html#id75)
* [NCCL_LEGACY_CUDA_REGISTER](../env.html#nccl-legacy-cuda-register)
  * [Value accepted](../env.html#id76)
* [NCCL_SET_STACK_SIZE](../env.html#nccl-set-stack-size)
  * [Value accepted](../env.html#id77)
* [NCCL_GRAPH_MIXING_SUPPORT](../env.html#nccl-graph-mixing-support)
  * [Value accepted](../env.html#id79)
* [NCCL_DMABUF_ENABLE](../env.html#nccl-dmabuf-enable)
  * [Value accepted](../env.html#id80)
* [NCCL_P2P_NET_CHUNKSIZE](../env.html#nccl-p2p-net-chunksize)
  * [Values accepted](../env.html#id81)
* [NCCL_P2P_LL_THRESHOLD](../env.html#nccl-p2p-ll-threshold)

__[NCCL](../index.html)

* * *

Point-to-point communicationÂ¶

(Since NCCL 2.7) Point-to-point communication can be used to express any communication pattern between ranks. Any point-to-point communication needs two NCCL calls : a call to [`ncclSend()`](../api/p2p.html#c.ncclSend "ncclSend") on one rank and a corresponding [`ncclRecv()`](../api/p2p.html#c.ncclRecv "ncclRecv") on the other rank, with the same count and data type.

Multiple calls to [`ncclSend()`](../api/p2p.html#c.ncclSend "ncclSend") and [`ncclRecv()`](../api/p2p.html#c.ncclRecv "ncclRecv") targeting different peers can be fused together with [`ncclGroupStart()`](../api/group.html#c.ncclGroupStart "ncclGroupStart") and [`ncclGroupEnd()`](../api/group.html#c.ncclGroupEnd "ncclGroupEnd") to form more complex communication patterns such as one-to-all (scatter), all-to-one (gather), all-to-all or communication with neighbors in an N-dimensional space.

Point-to-point calls within a group will be blocking until that group of calls completes, but calls within a group can be seen as progressing independently, hence should never block each other. It is therefore important to merge calls that need to progress concurrently to avoid deadlocks.

Below are a few examples of classic point-to-point communication patterns used by parallel applications. NCCL semantics allow for all variants with different sizes, datatypes, and buffers, per rank.

SendrecvÂ¶

In MPI terms, a sendrecv operation is when two ranks exchange data, both sending and receiving at the same time. This can be done by merging both ncclSend and ncclRecv calls into one :

```
    ncclGroupStart();
    ncclSend(sendbuff, sendcount, sendtype, peer, comm, stream);
    ncclRecv(recvbuff, recvcount, recvtype, peer, comm, stream);
    ncclGroupEnd();
```

One-to-all (scatter)Â¶

A one-to-all operation from a `root` rank can be expressed by merging all send and receive operations in a group :

```
    ncclGroupStart();
    if (rank == root) {
      for (int r=0; r<nranks; r++)
        ncclSend(sendbuff[r], size, type, r, comm, stream);
    }
    ncclRecv(recvbuff, size, type, root, comm, stream);
```

```
    ncclGroupEnd();
```

All-to-one (gather)Â¶

Similarly, an all-to-one operations to a `root` rank would be implemented this way :

```
    ncclGroupStart();
    if (rank == root) {
      for (int r=0; r<nranks; r++)
        ncclRecv(recvbuff[r], size, type, r, comm, stream);
    }
    ncclSend(sendbuff, size, type, root, comm, stream);
    ncclGroupEnd();
```

All-to-allÂ¶

An all-to-all operation would be a merged loop of send/recv operations to/from all peers :

```
    ncclGroupStart();
```

```
    for (int r=0; r<nranks; r++) {
      ncclSend(sendbuff[r], sendcount, sendtype, r, comm, stream);
      ncclRecv(recvbuff[r], recvcount, recvtype, r, comm, stream);
    }
    ncclGroupEnd();
```

Neighbor exchangeÂ¶

Finally, exchanging data with neighbors in an N-dimensions space could be done with :

```
    ncclGroupStart();
    for (int d=0; d<ndims; d++) {
      ncclSend(sendbuff[d], sendcount, sendtype, next[d], comm, stream);
      ncclRecv(recvbuff[d], recvcount, recvtype, prev[d], comm, stream);
    }
    ncclGroupEnd();
```

[Next](threadsafety.html "Thread Safety") [Previous](groups.html "Group Calls")

* * *