RoCEv2-based Collective Communication Offloading (In-network Computing)

draft-liu-nfsv4-rocev2-00

https://datatracker.ietf.org/doc/draft-liu-nfsv4-rocev2/

Rubing LIU (H3C)

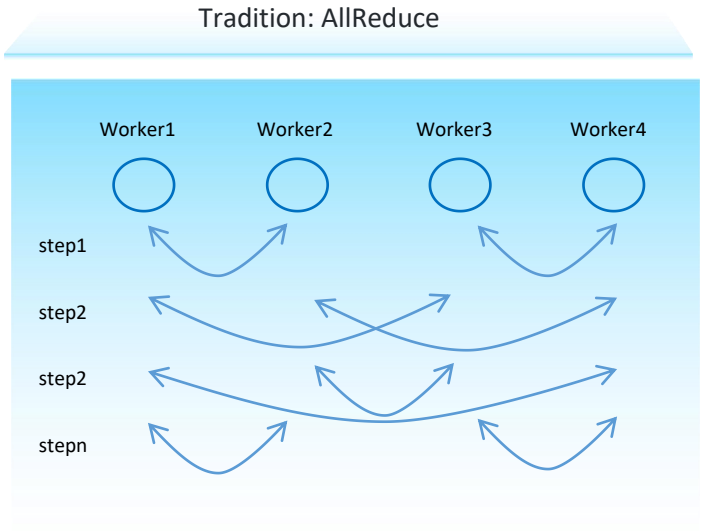
liurubing@h3c.com

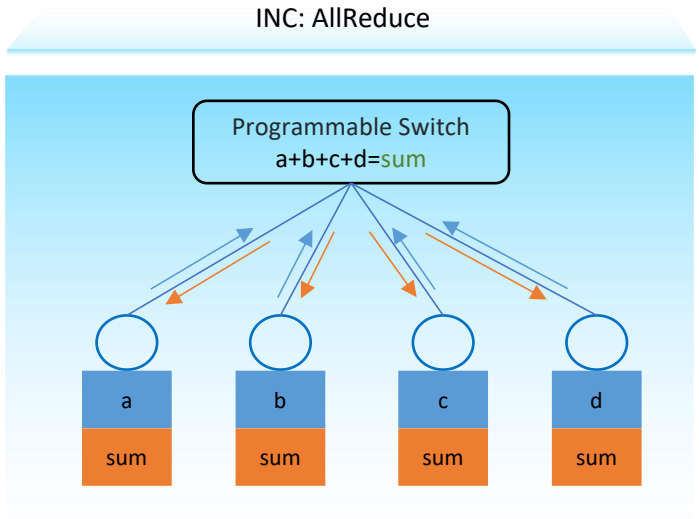
Background

The total time for HPC and AI is the sum of the computing time on CPUs/GPUs and collective communication time.

The focus is on reducing collective communication time.

The design proposes RoCEv2-based collective communication offloading. (In-Network Computing)





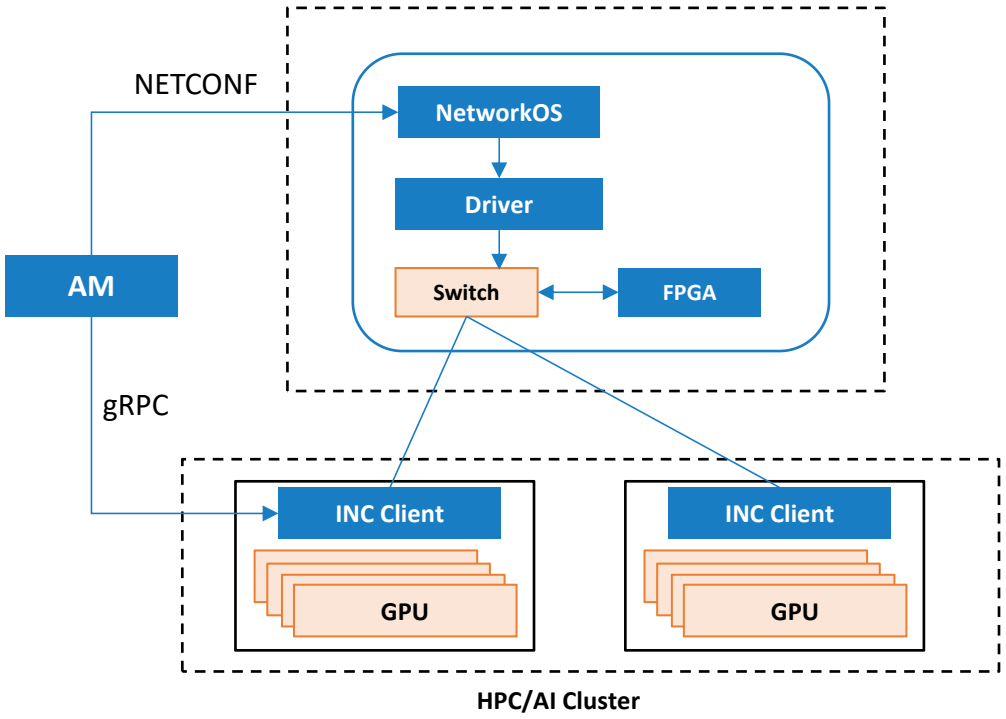
Issues:

1. Multiple interactions between worker nodes, affecting efficiency.
2. Large communication between worker nodes, increasing network workload.

Advantages:

1. Reducing interactions between worker nodes to improve the efficiency of cluster computing.
2. Decreasing the total communication among worker nodes to lighten the network workload.

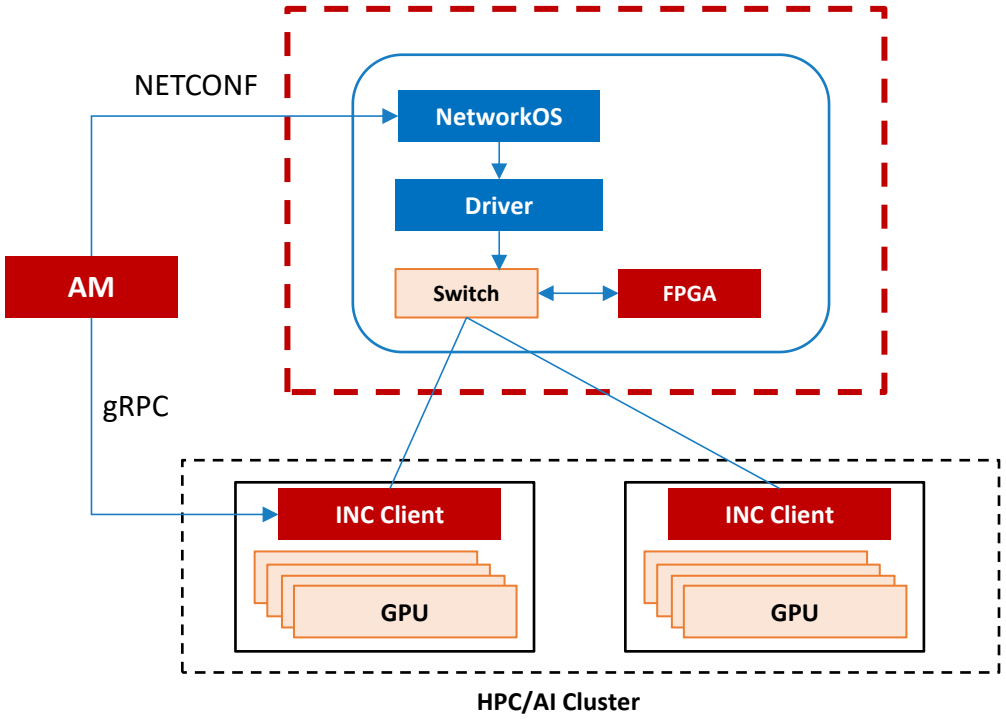
INC Architecture



Key components of in-network computing:

- **INC switch:** Responsible for implementing in-network computing based on the aggregation tree generated by INC Aggregation Manager. FPGA for initial in-network computing, with a future transition to higher-performance switch chips.
- **INC client (Host):** Deployed on computing nodes, integrates with MPI and NCCL libraries to send collective communication messages to INC switch.
- **INC Aggregation Manager (AM):** Generates and manages the aggregation tree, deploys in-network computing-related flow rules to the switches, and monitors the status of in-network computing tasks.

Minimize change, Maximize profit



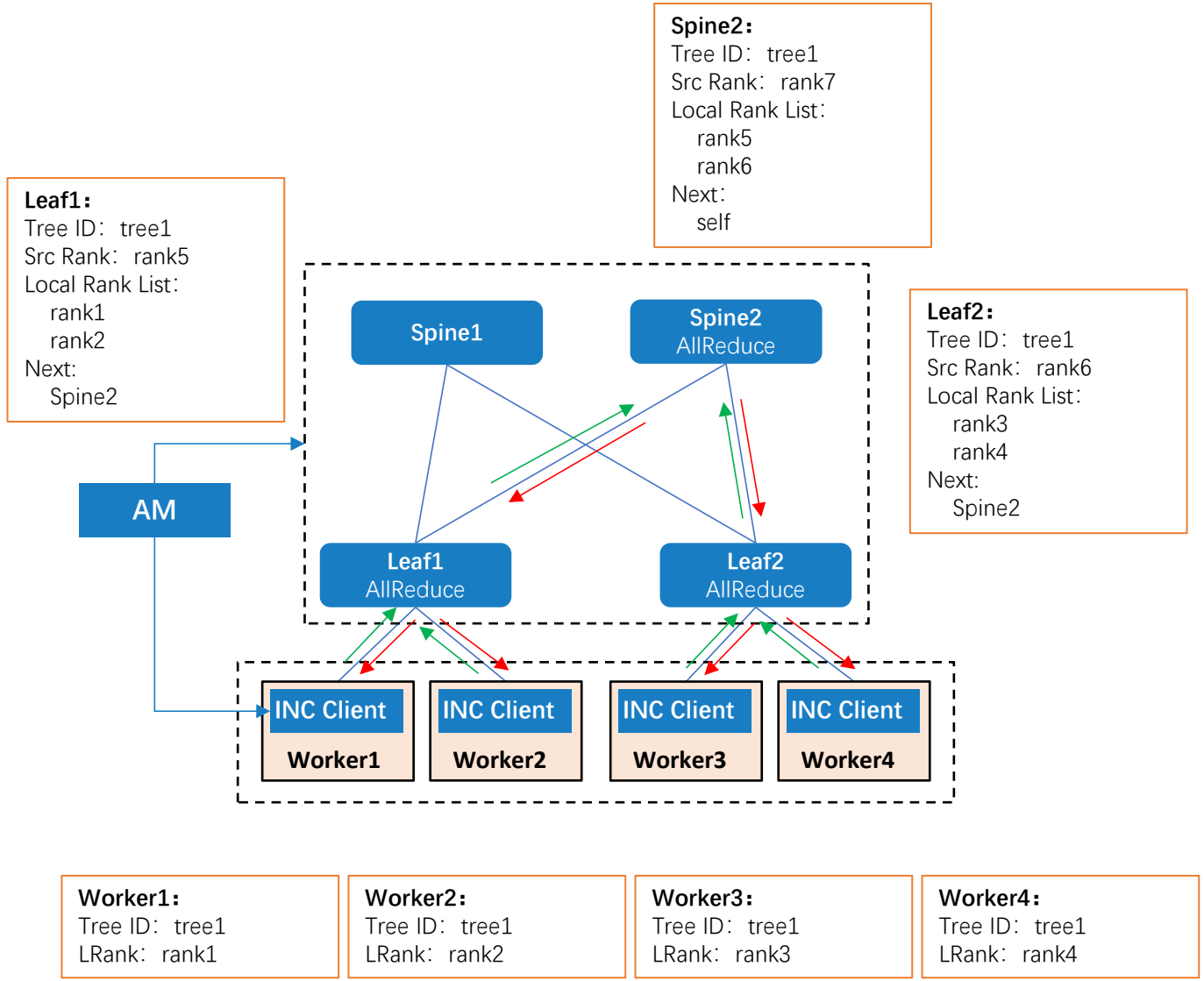
The design goals

1. Reduce the interaction between computing nodes to improve collective communication efficiency by over 50%.
2. Reduce the overall communication of computing nodes by over 30% and reduce network workload.
3. Linearly increase the computing capacity of the cluster with the scale of the cluster.

AM	New controller
FPGA	New INC Engine
INC Client	New plugin for PMI or NCCL
GPU	No Change
NIC	No Change

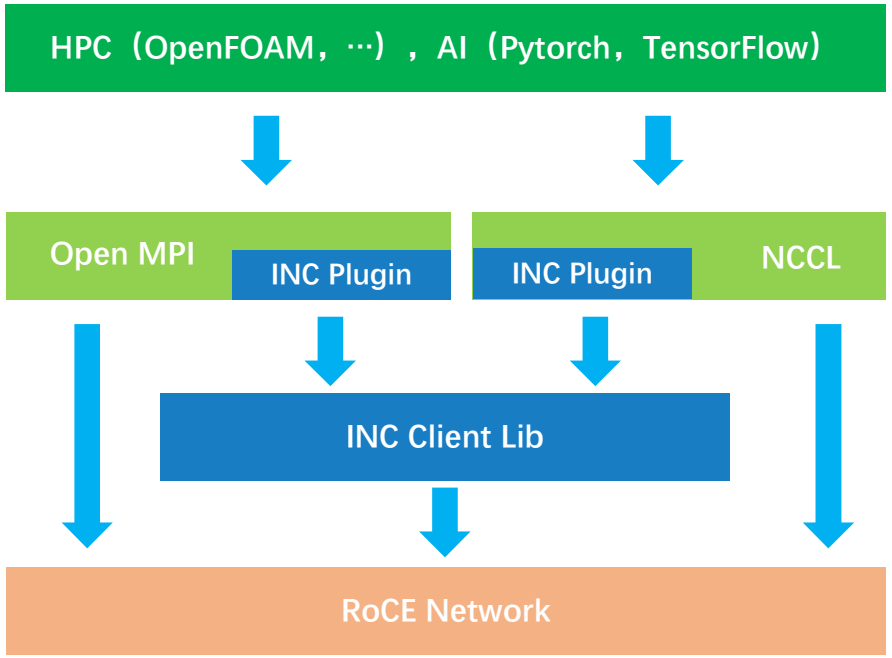
INC Switch

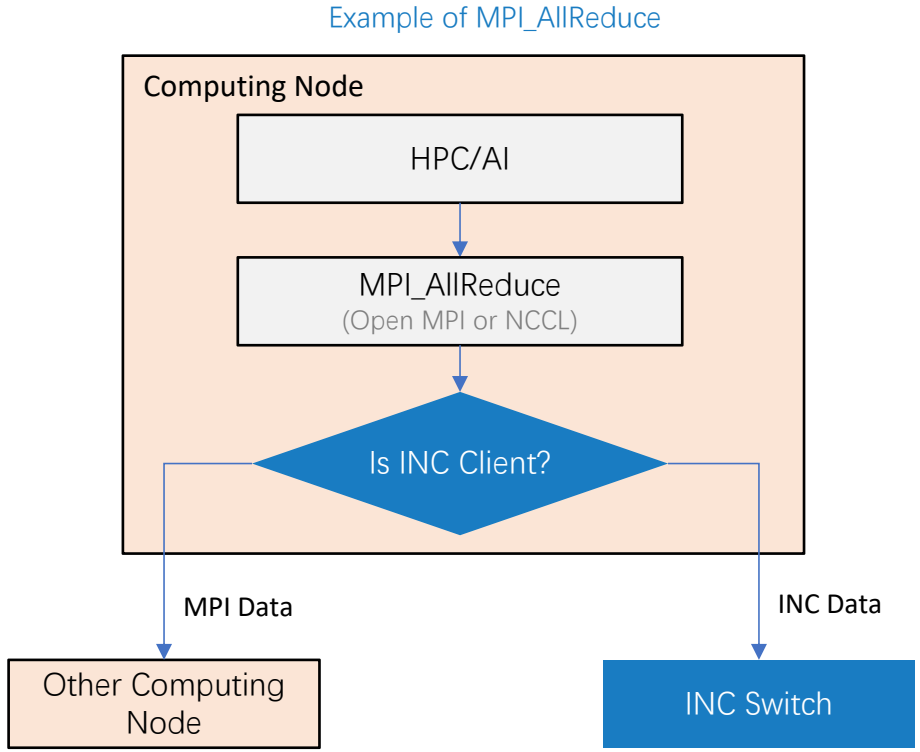
- Only data packets from the same computation task can be aggregated together.
- AM generates aggregation trees and assigns Tree IDs for different computation tasks. Then, it sends the aggregation tree information, including the Local Rank List, to the switch.
- When the INC switch receives data packets from the lower level, it performs local aggregation based on the Local Rank List. The aggregation can only be completed when data from all ranks in the list are received.
- If the INC switch is the root, it indicates that the aggregation is completed. The aggregated result is then broadcasted to all members in the Local Rank List.
- If the INC switch is not the root, it indicates that multi-level aggregation is required. The local aggregation result is sent to the upper-level INC switch for further aggregation.
- The lower-level INC switch, upon receiving the aggregated result from the upper-level INC switch, continues to broadcast the aggregation result to all members in the Local Rank List.



INC Client (Host)

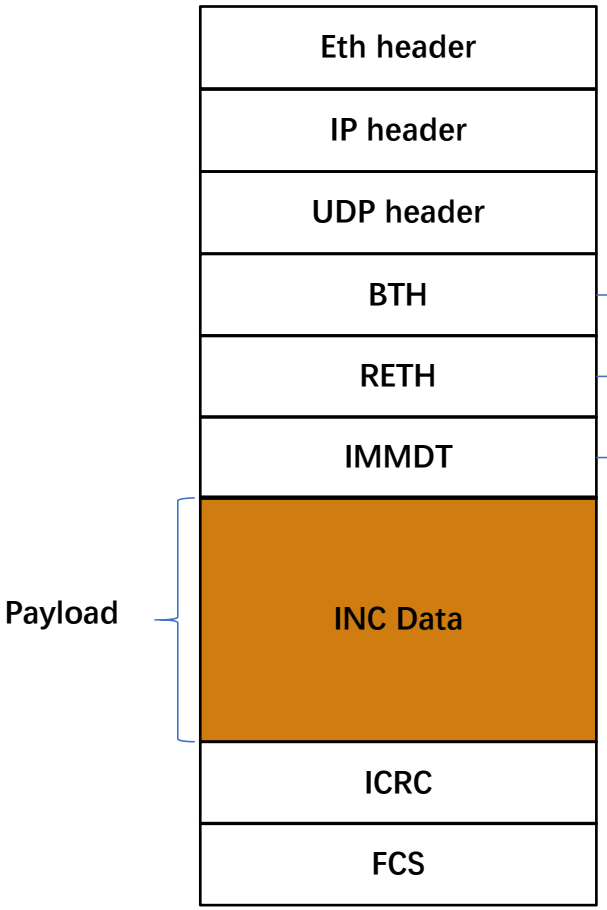
OpenMPI and NCCL define standard MPI collective communication interfaces, but they allow third-party to have their own implementation. INC Client essentially takes over the MPI collective communication interface of OpenMPI and NCCL and implements its own set of MPI collective communication algorithms.
When the application calls the MPI_AllReduce interface of OpenMPI or NCCL, the code executed is actually in INC Client. The INC Client sends the data of MPI collective communication to the INC switch in the encapsulated format of in-network computing. The INC Client is also responsible for receiving in-network computing data packets from the INC switch and returning them to the application.





RoCEv2 encapsulation

RoCEv2-based



The data packets between INC client (host) and INC switch are encapsulated using RoCEv2 and transmitted over UDP with a destination port number of 4791.

IMMDT included: Tree ID, Collective communication type, Data type, Operation type.

bits / bytes	31-24			23-16				15-8	7-0
0-3	OpCode			SE	M	Pad	TVer	Partition Key	
4-7	F/Res1 [a]	B/Res1 [a]	Reserved 6 [a]	Destination QP					
8-11	A	Reserved 7		PSN - Packet Sequence Number					

bits / bytes	31-24	23-16	15-8	7-0
0-3	Virtual Address (63-32)			
4-7	Virtual Address (31-0)			
8-11	R_Key			
12-15	DMA Length			

bits / bytes	31-24	23-16	15-8	7-0
0-3	Immediate Data			

Conclusions

- Collective communication offloading to Switch is a best way to deliver performance and efficiency gains, including low latency, improved resource utilization, flexible resource allocation, scalability.
- RoCEv2-based collective communication offloading.
- No change of Computing Node hardware. Compatible with multi-vendor GPU and NIC.

Next Step

- **Feedback / collaboration highly welcome!**
- **H3C built the original prototype in laboratory. The disclosure of test is coming soon.**

Thank you!