

In-Network Data Consistency (INDAC)

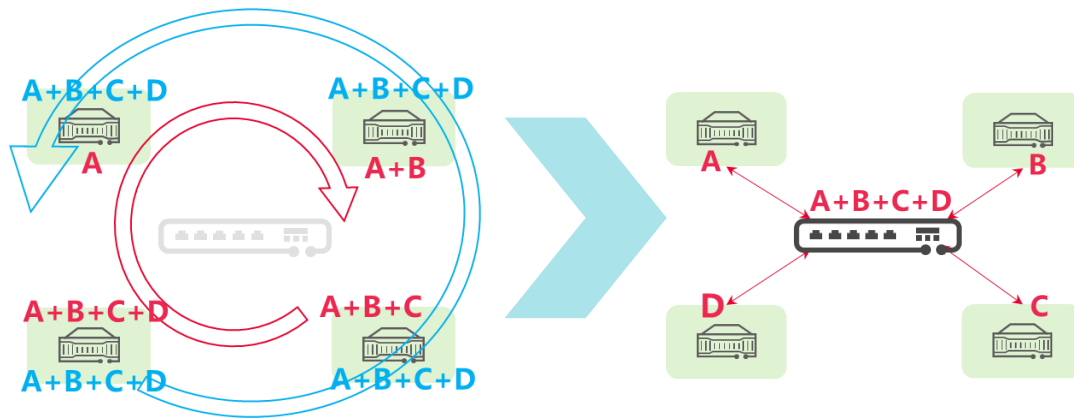
IETF 119 CCO Side Meeting

Yang Tian, Qiaoling Wang, Yizhou Li, Jian Yan, Xiaoyong Ming

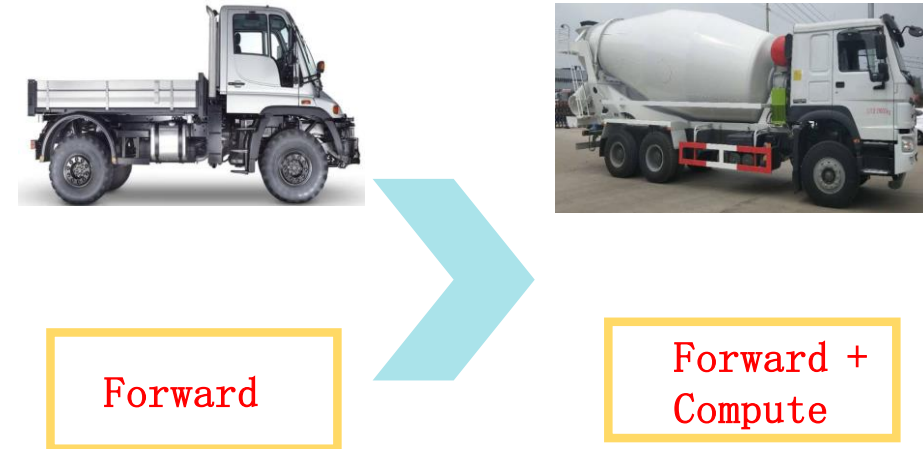
{tianyang21, wangqiaoling, liyizhou, jim.yanjian, liyizhou}@huawei.com, mingxiaoyong@hisilicon.com

In-Network Computing: Embed computing process in network

- Recently, In-Network Computing (INC) has been used in the AI training, BigData and HPC to accelerate the data aggregation process.
- INC leverages convergence position and high throughput capability of switches to aggregate data on-path, eliminating 50% data movement^[*].



INC: In-network aggregation
Data aggregation during communication



INC reduces node interaction times and data transmission amount by 50% ^[*].

In-Network Computing has more potentials than just aggregation

What are the strengths of INC switch?

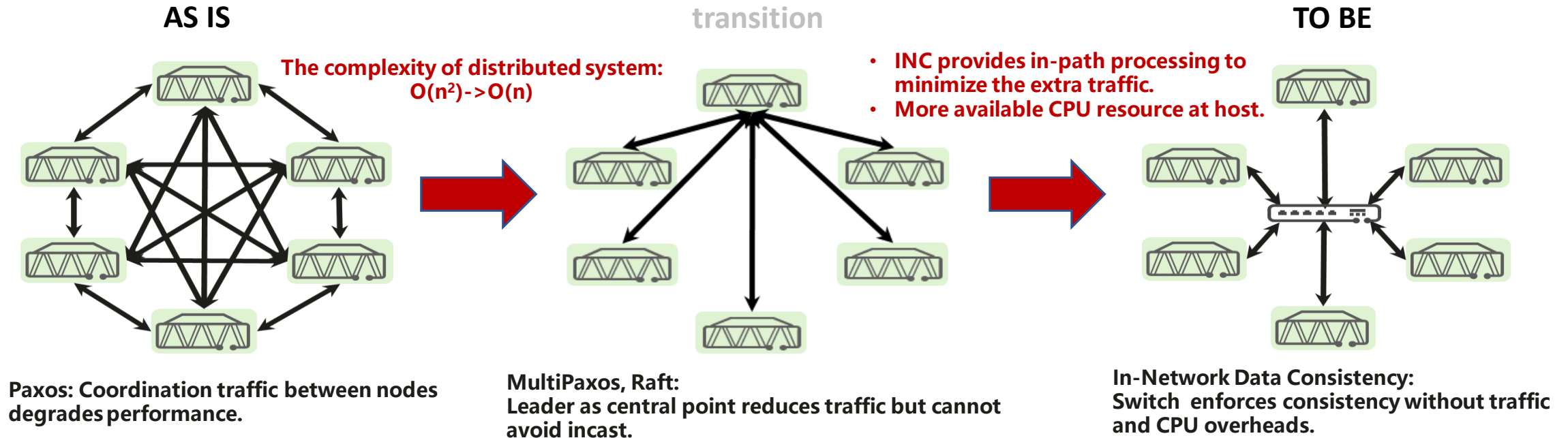
- At the convergence position of the network, switch can see all the traffic passing through, which can build a global view.
- Switch memory available for packet/flow level state information maintenance.
- Switch can provide various arithmetic operations in addition to sum(aggregation) when doing line rate forwarding.

INC can do more in

- Distributed storage
- Database
- Cloud

In-Network Data Consistency(INDAC)

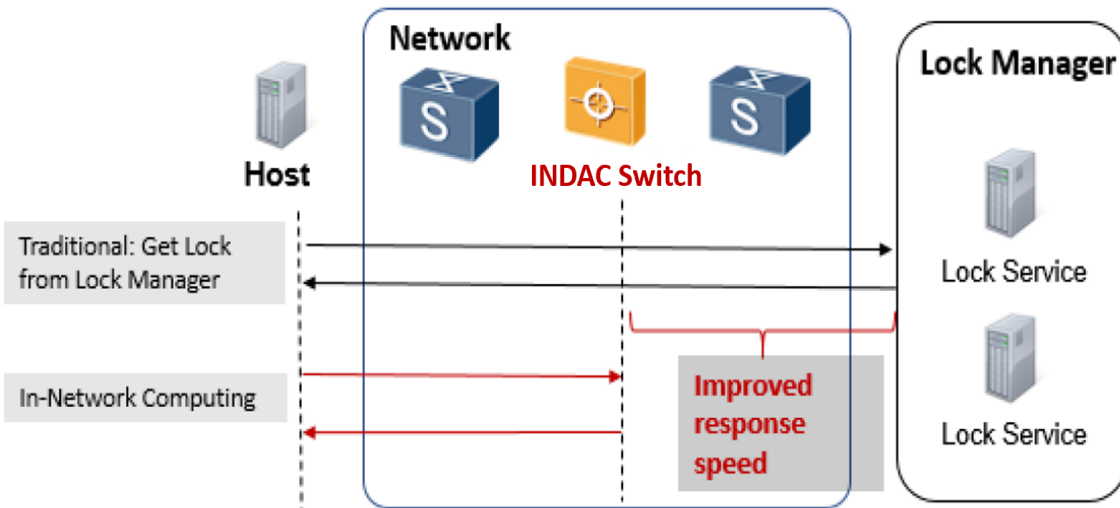
- In distributed systems, **data consistency** management involves complex coordination and synchronization mechanisms between nodes, such as locking and consensus protocols.
- Most mechanisms introduce **extra traffic and CPU consumption overhead** either between peer nodes or between a single server and other nodes. The overall system performance including latency, throughput and scalability is not optimal.
- INC provides new opportunities to achieve data consistency with lower latency and higher throughput.



Maintain the global consistency info in switch **to accelerate consistency protocols.**

An INDAC example for Concurrency Control: Lock mechanism enhancement

Switch grants **lock** to hosts with short latency and ~10TB throughput.



Server CPU → Switch Pipeline

Concurrency Control: For database and file system, lock is a common mechanism to achieve concurrent read/write data consistency.

Why INDAC accelerates consistency?

- Switch memory records consistency information. When packet passes through switch, check the record and modify the packet.
- Switch processes faster and more stable than CPU, ~ns for one packet.
- Switch has much higher throughput, i.e., >10TB, which can achieve billion PPS.

A general-purpose INDAC framework and the challenges

Starting from the lock function, we can build a general purpose In-network data consistency enforcement framework to support multiple consistency-related function accelerations, e.g., concurrency control, replica, data sharding, synchronization, and etc.

Challenge 1: Put many INDAC functions into one chip?

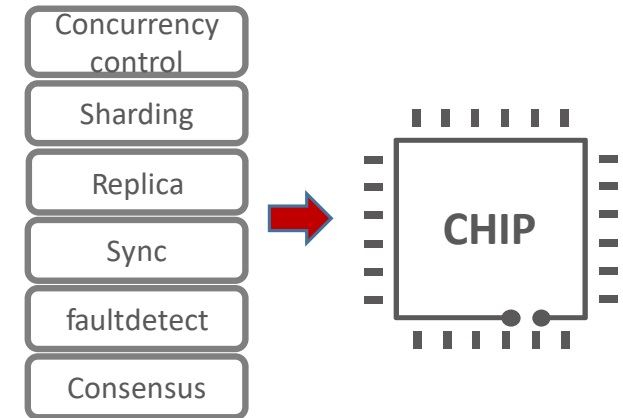
- The available memory resource is limited, usually less than MB.
- The logic resource is also limited and should not affect the line-rate forwarding.

Challenge 2: Coordinate server and switch to realize every INDAC function?

- Different INDAC functions have different data handling process, different match action table, different packets and etc.

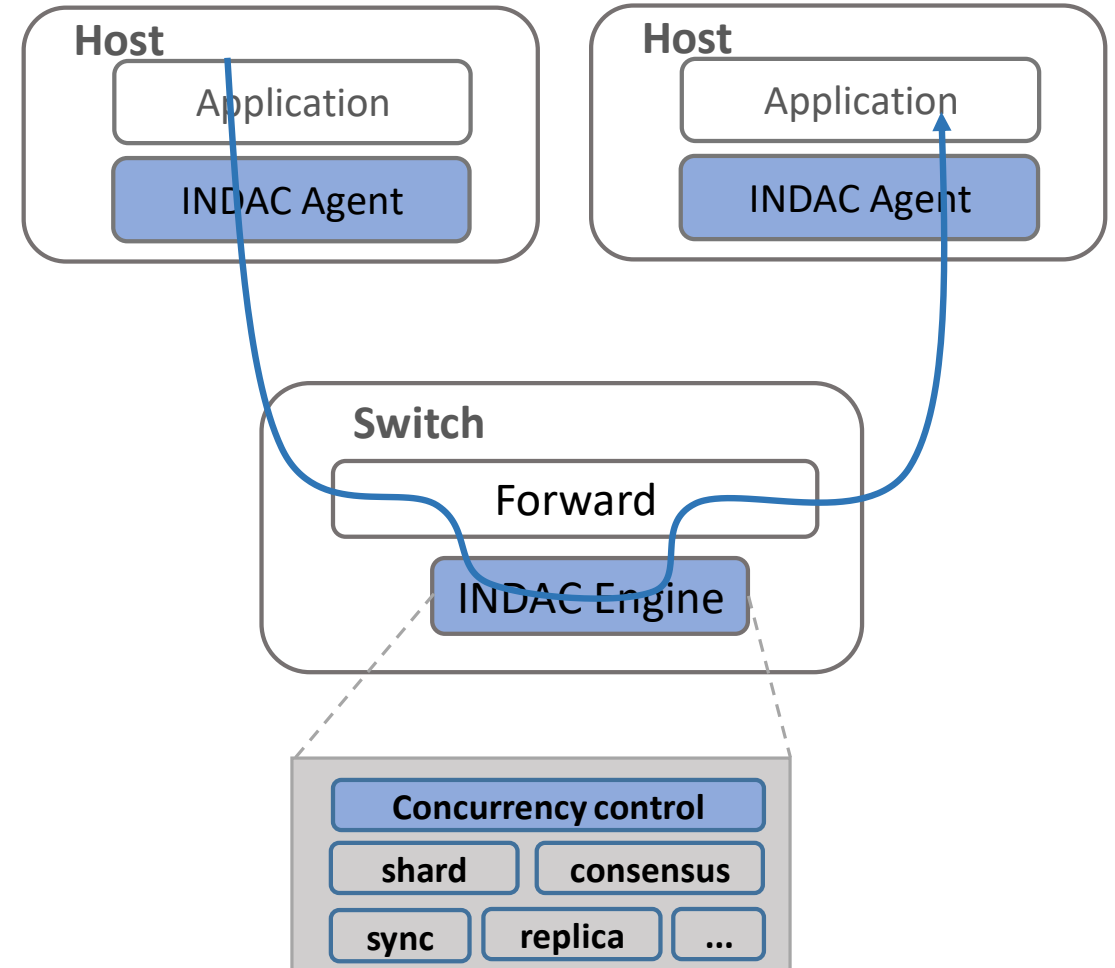
Challenge 3: How to support different applications?

- Different application has different software API, lib, data structure and etc.



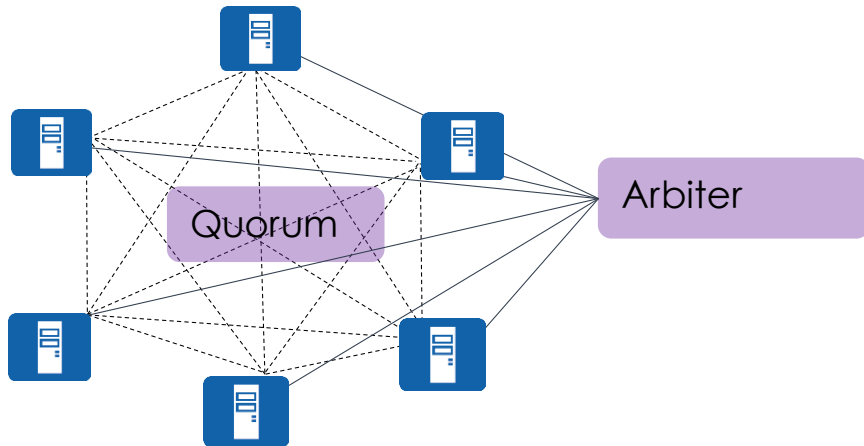
INDAC framework to support multiple DAC functions

- ✓ **Co-design system:**
 - **INDAC engine in switch** provides INDAC functions
 - **INDAC agent in host** provides common API to support different applications
- ✓ **Protocol between INDAC agent and engine:**
 - Identify INDAC packet
 - Identify the specific INDAC function
- ✓ **Hardware Resource:** Function abstraction and memory compression facilitate the use of small memory to support multiple INDAC functions.
- ✓ **Flexibility:** Programmability of the INDAC engine provides opportunities to implement new functions in the future.
- ✓ **Fallback plan:** In case of switch down and memory used up, INDAC supports fallback to original consistency enforcement mechanism.



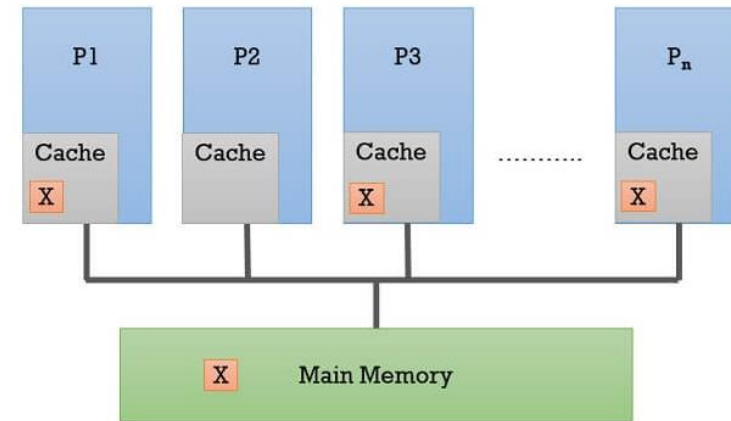
Future work: more INDAC function

Network Arbitration



- In case of node failure including split-brain, distributed system needs a new leader for arbitration.
- Switch is a good candidate for INDAC arbitration function.

Cache Coherence



- Multiple CPUs can read/write shared data in cache, which needs to ensure consistency. Supported in CXL and etc.
- Possibly implemented as INDAC function in other devices like DPU.

Thank you