

Workgroup:

Computing in the Network Research Group

Internet-Draft:

draft-yao-coinrg-generic-framework-00

Published: 13 March 2023

Intended Status: Informational

Expires: 14 September 2023

Authors: K. Yao S. Xu Z. Li
 China Mobile China Mobile China Mobile
 W. Wu
 Peking University

A Generic COIN framework in controlled environments

Abstract

There have been a lot of academic research and industrial practice in the area of COIN, but most of them are case-by-case design and currently they also rely heavily on programmable network devices, which lacks some generality and scalability, thus will impede the development of COIN. This document summarizes the computing primitives/operations/semantics that can be implemented inside the network, through analysis of different COIN use cases, and proposes a generic framework of COIN in the controlled environments. Enabling technologies related to the framework and the standardization landscape are also analyzed in the document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 September 2023.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Conventions Used in This Document](#)
 - [2.1. Terminology](#)
 - [2.2. Requirements Language](#)
- [3. Generic Framework](#)
- [4. Enabling Technologies](#)
 - [4.1. The Scheduling Layer](#)
 - [4.2. The Control Layer](#)
 - [4.3. The Infrastructure Layer](#)
- [5. Research challenges and other considerations](#)
- [6. Security Considerations](#)
- [7. IANA Considerations](#)
- [8. Normative References](#)
- [Authors' Addresses](#)

1. Introduction

Programmable network devices(PNDs) including programmable switches and SmartNICs have inspired a lot of research work in the area of COIN. Like In-band Network Telemetry(INT), Network functions offloading(LBs, Firewalls), etc. However, technically, we argue that these use cases are not strictly “computing” in the network, since they are hardware implementation of network functions which traditionally implemented in servers so as to accelerate or enhance these network functions. The “network” in COIN is also ambiguous. Narrowly, it refers to network devices like PNDs, but broadly, it refers to network elements in different contexts. In edge computing or fog computing, these network elements refer to ubiquitous heterogeneous edge devices, but in controlled environments like data centers, network elements refer to normal network devices. And in this draft, we just limit the scope of the discussion inside the controlled environment, which is consistent with most of the existing work.

To make the work in COIN move further, there is a need to reach a consensus on the definition of COIN. Despite there is an ongoing draft about the terminology of COIN in the group, we want to share

our thoughts. Computing in the network is “to offload application-specific functions to network elements, so as to accelerate applications”. These application-specific functions are described by series of computing primitives/operations/semantics that could be supported by network elements, and they explain about what to “compute” in the network. A very illustrative example is In-network Aggregation(INA) for distributed machine learning model training. The aggregation operation is implemented in network devices, which could accelerate the entire model training process. A lot of research have investigated what kind of computing primitives can be offloaded to network devices, but there still lack a systematic summarization of these application-specific primitives. We think that application-specific functions can be generalized to be several types of computing primitives which could be further standardized, thus COIN will not depend on PNDs for implementation, but normal network devices that support these general primitives could take the work.

Further, current research on how COIN could accelerate applications usually depend on a case-by-case hardware software co-design scheme, which lacks generality and scalability for the development of COIN. There is a need to design a generic framework of COIN, for one thing, to make COIN a common capability of the network, for another, to lower the application development barriers.

Based on the analysis above, this document classifies several kinds of computing primitives which could be standardized, and proposes a generic framework of COIN, which can be scaled and promoted in the controlled environment.

2. Conventions Used in This Document

2.1. Terminology

PND Programmable Network Device

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14[RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Generic Framework

The generic COIN framework contains three logical layers: Scheduling layer(S), Control layer(C), and Infrastructure layer(I).

The infrastructure layer (I) includes the host and network equipment, including the relevant routing protocols and reliability protocols to realize COIN.

4. Enabling Technologies

4.1. The Scheduling Layer

Task decomposition is the first step to achieve end-network collaborative in-network computing. Through appropriate scheduling policy, reasonable resource allocation can be achieved and better task performance can be achieved. With the addition of in-network computing technology, it is necessary to consider not only the host resources, but also the in-network computing resources.

4.2. The Control Layer

End-network collaborative control realized by the host controller and the COIN controller.

Network side:

- * Network equipment management, including network equipment status, load condition, network equipment computing capacity and resource, etc.
- * Network topology management, including network topology update, link status monitoring, etc.
- * Routing, selecting an optimal path for in-network computing and forwarding.

Host side:

- * Cooperate with the host application to do the COIN processing, including completing the overall calculation task with the network side, and reliability control.

4.3. The Infrastructure Layer

Network equipment implements the standard COIN primitive.

A set of unified COIN primitives makes COIN more easier to achieve docking and promotion. Some research work [[NetRPC](#)] [[Netcompute](#)] summarize common COIN primitives and data structures. We refer to these research work and choose some major COIN primitives out of these work. ValStr_Agg is used in applications like distributed machine learning training, Asyn_Val_Agg is used in big data analysis applications where map-reduce is needed. K-V is used for caching, and consensus is used for synchronization within

distributed systems. Heterogeneous network devices can have different internal implementations of the same COIN primitives, but the services provided externally need to be unified. There is a need to standardize these COIN primitives for generic use cases. Of course, due to equipment differences, there may be differences in calculation accuracy for some primitives. These differences need to be considered in task decomposition and routing.

Type	Data Structure	Primitives
ValStr_Agg	Array	Map.get, Map.add, Map.clear
Asyn_Val_Agg	Map	Map.get, Map.add, Stream.modify
K-V	Map	Map.get, Map.add
consensus	Integer	Map.get, Map.add, Map.clear

Figure 2: Figure 2: COIN Primitives

COIN transformation of application program on host side.

Network cannot guarantee that the computing task can be completed during each transmission process, so the host side applications need to be COIN aware and be able to flexibly process the data that has been in-network processed or not.

5. Research challenges and other considerations

* End and network collaboration. Due to the limited resources within network devices, there is a need to design some fallback mechanisms when tasks cannot be fully accomplished within the network, and they should be finished at the end devices. Relative algorithms, protocols should be considered for implementation.

* COIN reliability and correctness. On the premise that tasks can be offloaded to network devices for computing, the correctness and reliability of the work should be considered. There should be some mechanisms designed to maintain that the COIN results is consistent with that when tasks are fully accomplished at end devices. Besides, reliable data transmission in COIN should be elaborately designed, since many applications have very strict QoS requirements.

6. Security Considerations

TBD.

7. IANA Considerations

TBD.

8. Normative References

[Netcompute] Dan R. K. Ports, Jacob Nelson, "When Should The Network Be The Computer?", May 2019, <<https://doi.org/10.1145/3317550.3321439>>.

[NetRPC] Zhao, B., Wu, W., & Xu, W., "NetRPC: Enabling In-Network Computation in Remote Procedure Calls", December 2022, <<https://doi.org/10.48550/arXiv.2212.08362>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Kehan Yao
China Mobile
Beijing
100053
China

Email: yaokehan@chinamobile.com

Shiping Xu
China Mobile
Beijing
100053
China

Email: xushiping@chinamobile.com

Zhiqiang Li
China Mobile
Beijing
100053
China

Email: lizhiqiangy@chinamobile.com

Wenfei Wu

Peking University
Beijing
100871
China

Email: wenfeiwu@pku.edu.cn