A Generic COIN framework in controlled environments

draft-yao-coinrg-generic-framework-00

Presenter: Kehan Yao (China mobile)
Introduction

- Programmable Network Devices (PNDs) have inspired a lot of research work like LBs\FWs in the area of COIN. Technically, they are not strictly “computing” in the network, but hardware implementations of network functions.

- We think that 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.
Introduction

• Academic researches and industrial practice in COIN area are lack of generality and scalability.
  - mostly case-by-case design
  - rely heavily on PNDs

• Application development-friendly framework should be designed to help App developers better leverage the network capabilities provided by operators.

• To increase generality and promote the standardization of COIN
  • summarize some general computing primitives/operations/semantics that can be implemented inside normal network devices rather than PNDs.
  • Promote them to be standardized.
  • propose a generic framework of COIN in the controlled environments.
Generic COIN Framework

• The generic COIN framework contains three logical layers:

  • **Scheduling layer(S):** decomposes jobs into host tasks and COIN tasks

  • **Control layer(C):** Host/COIN task deployment and control, management, routing, End-Network Collaboration

  • **Infrastructure layer(I):** hosts and network equipment
Enabling Technologies-Scheduling Layer

• Schedulers follow a scheduling policy to manage job resource allocation for better efficiency.
• The scheduling policies vary for different workloads:
  • deadline-aware policies for Deadline-Aware (DA) jobs
  • heuristic policies for Best-Effort (BE) jobs
  • ......
• 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
Enabling Technologies-Control Layer

• Both host controller and COIN controller can be centralized or distributed. They cooperate to achieve End-Network collaboration.

• **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.

• **Network side:**
  * Network equipment management (status, load condition, computing capacity and resource, etc.)
  * topology management
  * Routing
Enabling Technologies-Infrastructure Layer

- Network equipment implements the standard COIN primitive.

<table>
<thead>
<tr>
<th>Type</th>
<th>Data Structure</th>
<th>Primitives</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValStr_Agg</td>
<td>Array</td>
<td>Map.get, Map.add, Map.clear</td>
</tr>
<tr>
<td>Asyn_Val_Agg</td>
<td>Map</td>
<td>Map.get, Map.add, Stream.modify</td>
</tr>
<tr>
<td>K-V</td>
<td>Map</td>
<td>Map.get, Map.add</td>
</tr>
<tr>
<td>consensus</td>
<td>Integer</td>
<td>Map.get, Map.add, Map.clear</td>
</tr>
</tbody>
</table>

- ValStr_Agg: used in applications like distributed machine learning training
- Asyn_Val_Agg: used in big data analysis applications where map-reduce is needed.
- K-V is used for caching
- consensus: used for synchronization within distributed systems

- COIN transformation of application program on host side.
  - the host applications need to be COIN aware
  - flexibly process the data that has been in-network processed or not
Research challenges and other considerations

• End and network collaboration
  • fallback mechanisms --- when tasks cannot be fully accomplished within the network, they should be finished at the end devices. Relative algorithms, protocols should be considered for implementation.

• COIN reliability and correctness
  • correctness mechanisms --- to maintain that the COIN results is consistent with that when tasks are fully accomplished at end devices.
  • reliable data transmission --- to fullfill strict QoS requirements.
Thanks!