Edge-side Common Data Processing on The Computing Continuum

Background

- Data growth from edges for IoT applications
  - Evolution of 5G/Post-5G mobile networking technologies
  - Increasing demands of AI use
  - However, still much data from real worlds have not been used.
- Transferring all data to clouds?
  - It highly costs backhaul networks and cloud resources.
  - Sometimes, sensitive data cannot be transferred to clouds, due to regulations and security concerns.
- Complete everything at edges?

  Device/Edge resources are often limited.

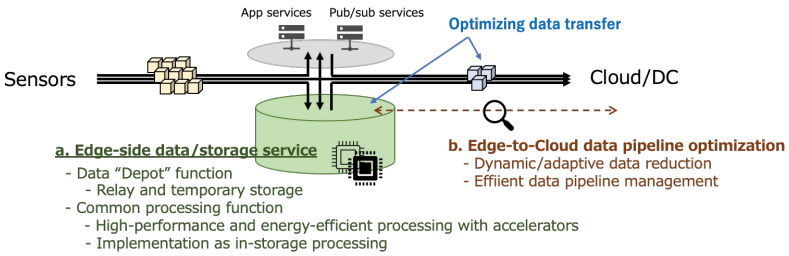
  Some insights can be obtained from multiple datasets.

➡ Computing Continuum – takes both benefits from clouds and edges and provide a seamless infrastructure

Our approach

We propose “a common data processing and storage service” at edge servers, which intermediates data transfer.

- Only necessary data is sent to clouds by proper data reduction.
- We support typical data reduction algorithms (SQL-like and storage-level) and provide their implementation with accelerators.
- We study dynamic/adaptive methods of transferring data between edges and clouds, including Federated Learning use cases.



Data processing with accelerators at edge servers

General data reduction and processing capability on edge servers make application developers free from implementing their own

- Data aware operation: filtering, aggregation, transformation
- Data agnostic operation: compression, deduplication
- Privacy processing: encryption, anonymization, pseudonymization
- AI-based processing: (some widely usable methods)

For data compression, we are evaluating compression performance with FPGA, GPU and CPU. The following is a comparison result of FPGA and CPU, on the resource-constrained environment of edge-cloud systems, where both power and processing capacity are limited. We implemented GZIP, a highly optimized lossless compression algorithm, for the FPGA card DE10-Agilex. We achieved a 48.8% and 52.5% reduction in data size for the enwik8 and enwik9 datasets, respectively. This reduction is crucial for maintaining performance in bandwidth-sensitive edge-cloud environments. Compared to the typical CPU-based compression method pigz, tested on a desktop equipped with an Intel Xeon Gold 6212U processor running at 2.4 GHz with multithreading optimization, our FPGA approach improves throughput and power efficiency by approximately 5.6x and 5.2x on average, respectively, highlighting its effectiveness in resource-constrained scenarios.

◆System implementation on FPGA and CPU

dataset	data size(KB)	compressed data size(KB)		compressed ratio		throughput (GB/s)	
		FPGA	CPU	FPGA	CPU	FPGA	CPU
enwik8	97657	49943	35706	1.96	2.74	2.87	0.45
enwik9	976563	463188	316474	2.11	3.09	3.01	0.62

Table 2. Breakdown of the throughput in the benchmark enwik (GB/s)

program	LZ77	Huffman Encoding	CRC	host-to-device	device-to-host
enwik8	7.68	3.27	8.18	5.79	6.0
enwik9	7.74	3.34	8.25	5.90	6.19

Table 3. Power consumption in the benchmark enwik (W)

	enwik8	enwik9
FPGA	13.16	13.42
CPU	18.77	120.52

✓FPGA card: DE10-Agilex; System design language: data parallel C++

✓Desktop: CPU: Xeon Gold 6212U (24 cores @2.4 GHz); RAMs: 512 GB DDR4

This work is based on results obtained from the project “Research and Development Project of the Enhanced Infrastructures for Post-5G Information and Communication System” (JPNP20017), commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

AIST SC24 Web Site
https://www.digiarc.aist.go.jp/event/sc2024/