Edge-side Common Data Processing on The Computing Continuum

Background

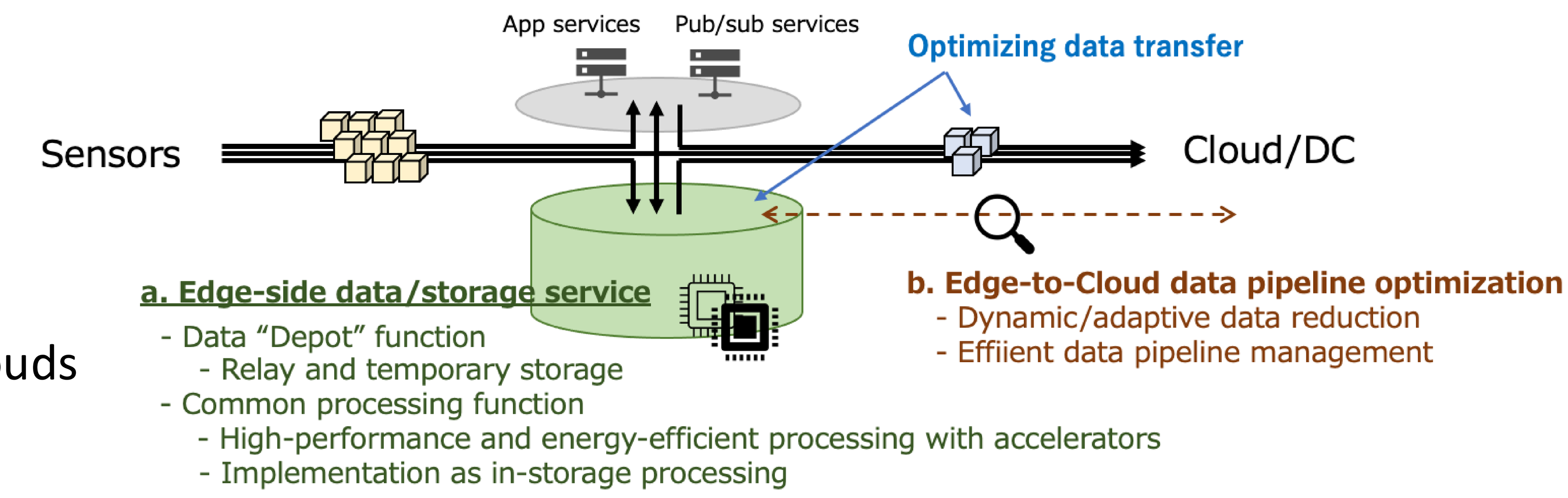
- Data growth from the edge for IoT applications
  - Evolution of 5G/Post-5G mobile networking technologies
  - Increasing demands of AI use
  - However, many data from real world scenarios are not used
- Transferring all data to clouds?
  - It increases resource cost of backhaul networks and cloud
  - Regulations and security concerns prevent transfer of sensitive data to clouds
- 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
- Supports typical data reduction algorithms (SQL-like, storage-level) and provides their implementation with accelerators
- Researching dynamic/adaptive methods of transferring data between edges and clouds, including Federated Learning use cases



Data processing with accelerators at edge servers

Objectives: The aim is to evaluate and compare the performance of **CPU-based** and **GPU-based data compression algorithms** for diverse data types (numerical, image, and audio). The goal is to identify optimal algorithms for different deployment scenarios—balancing **compression efficiency, speed, energy consumption, and memory usage**.

Implementation:

- CPU algorithms: Deflate, Gzip, Snappy (single/multi-threaded)
- GPU algorithms: LZ4, Deflate, Snappy, Zstd (NVCOMP)

Datasets:

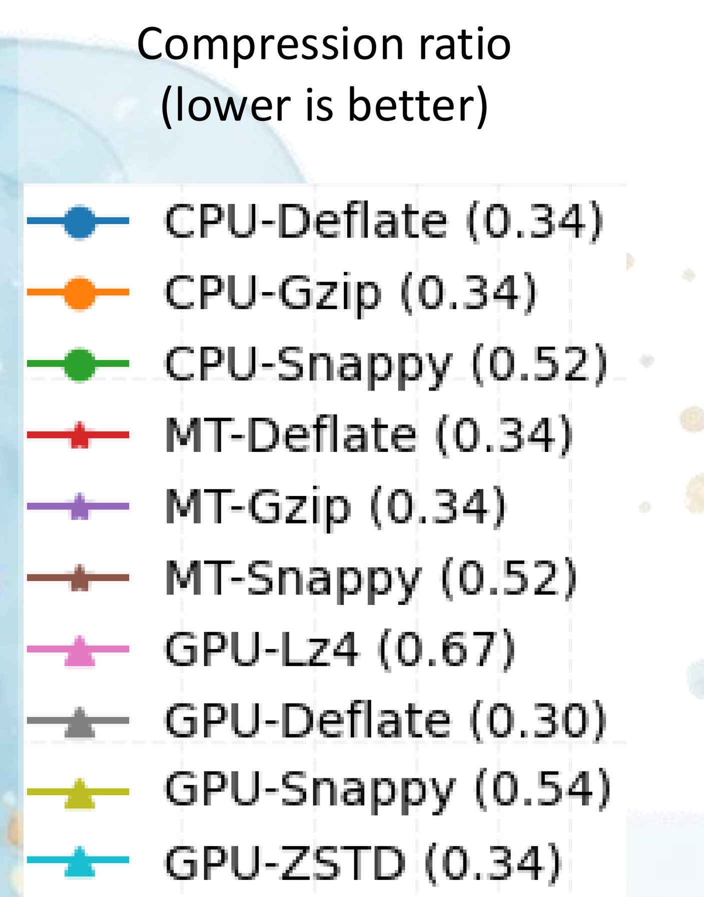
- *Numerical:* ECG time-series
- *Audio:* FSD50K WAV files
- *Image:* Cosmic (NumPy arrays)

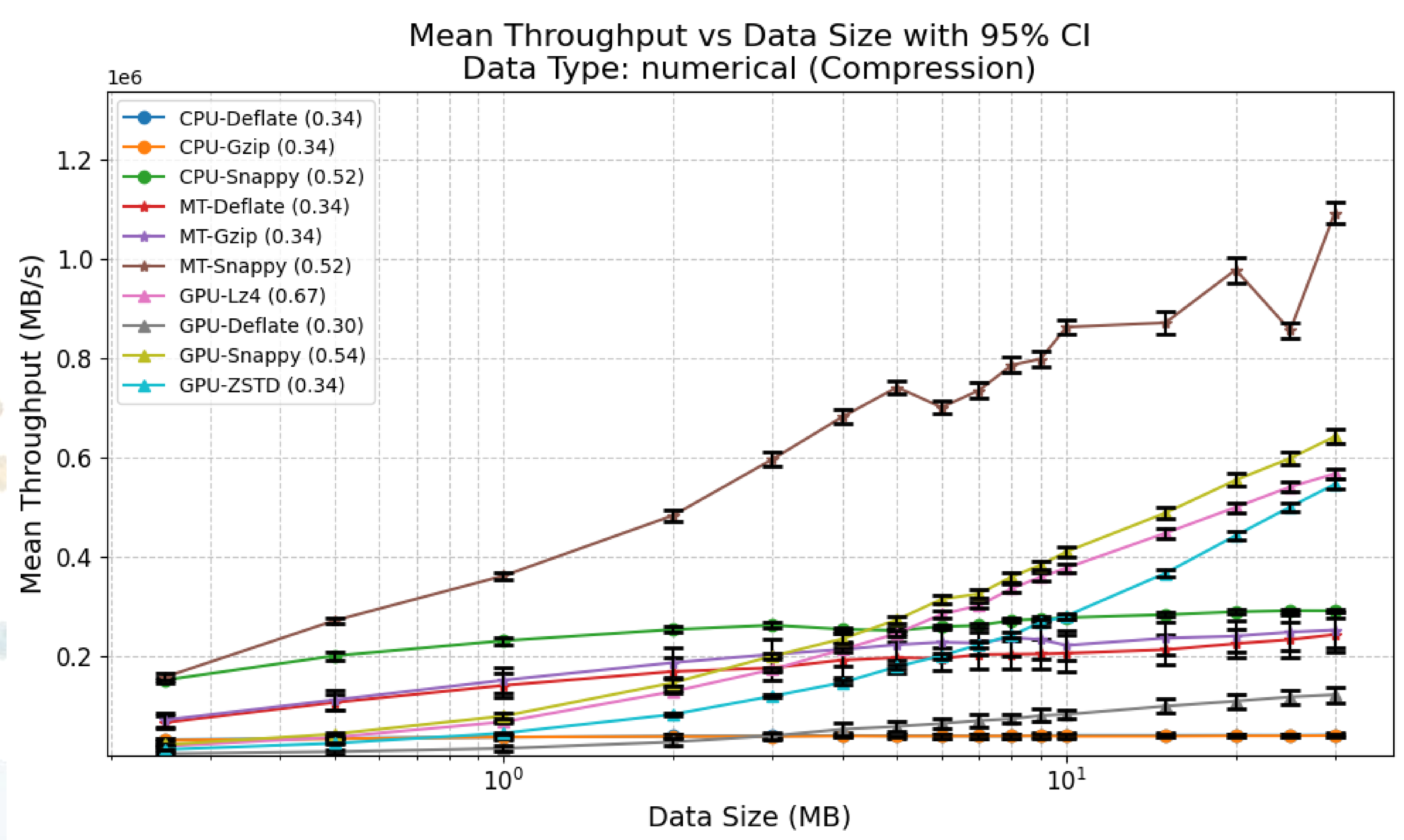
Metrics:

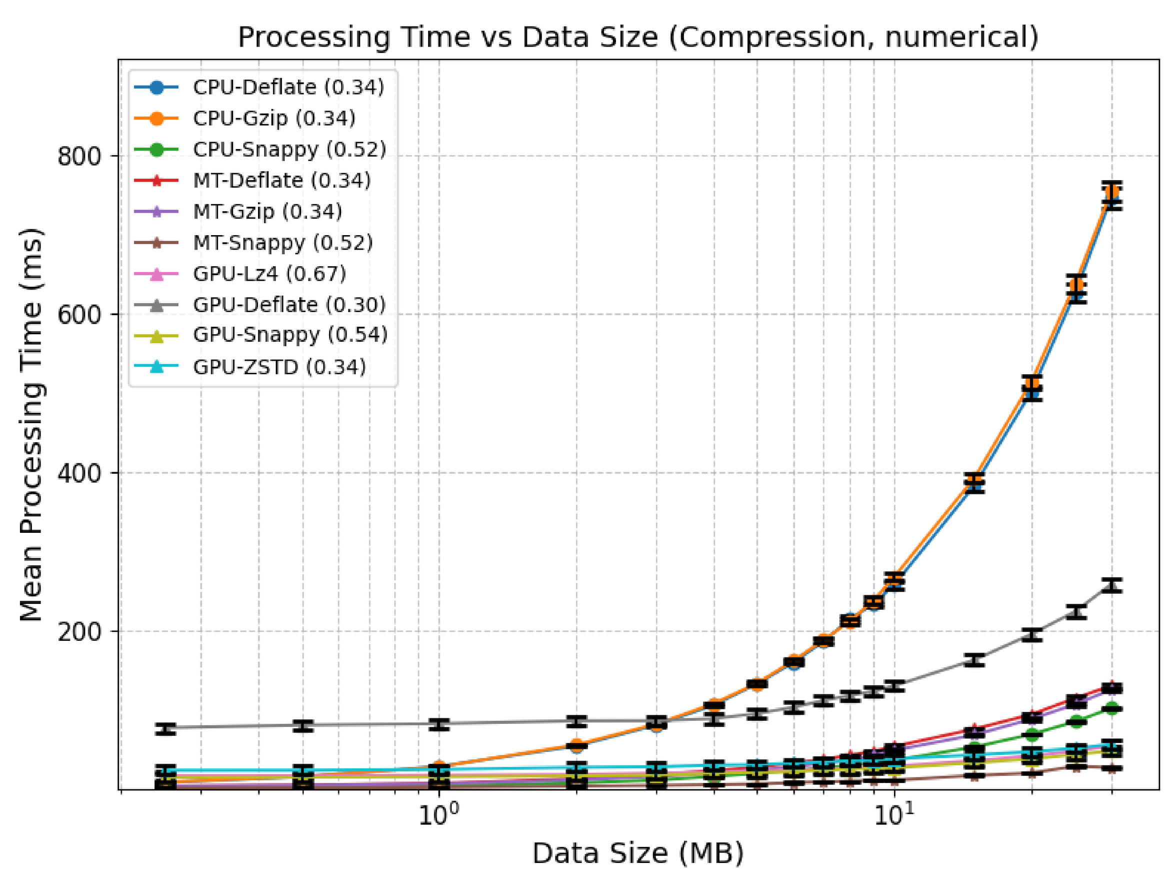
- Compression ratio, throughput, and energy efficiency

Hardware:

- 12-core Intel Xeon w3-2423 CPU; NVIDIA RTX A4000 GPU







Speed & Latency:

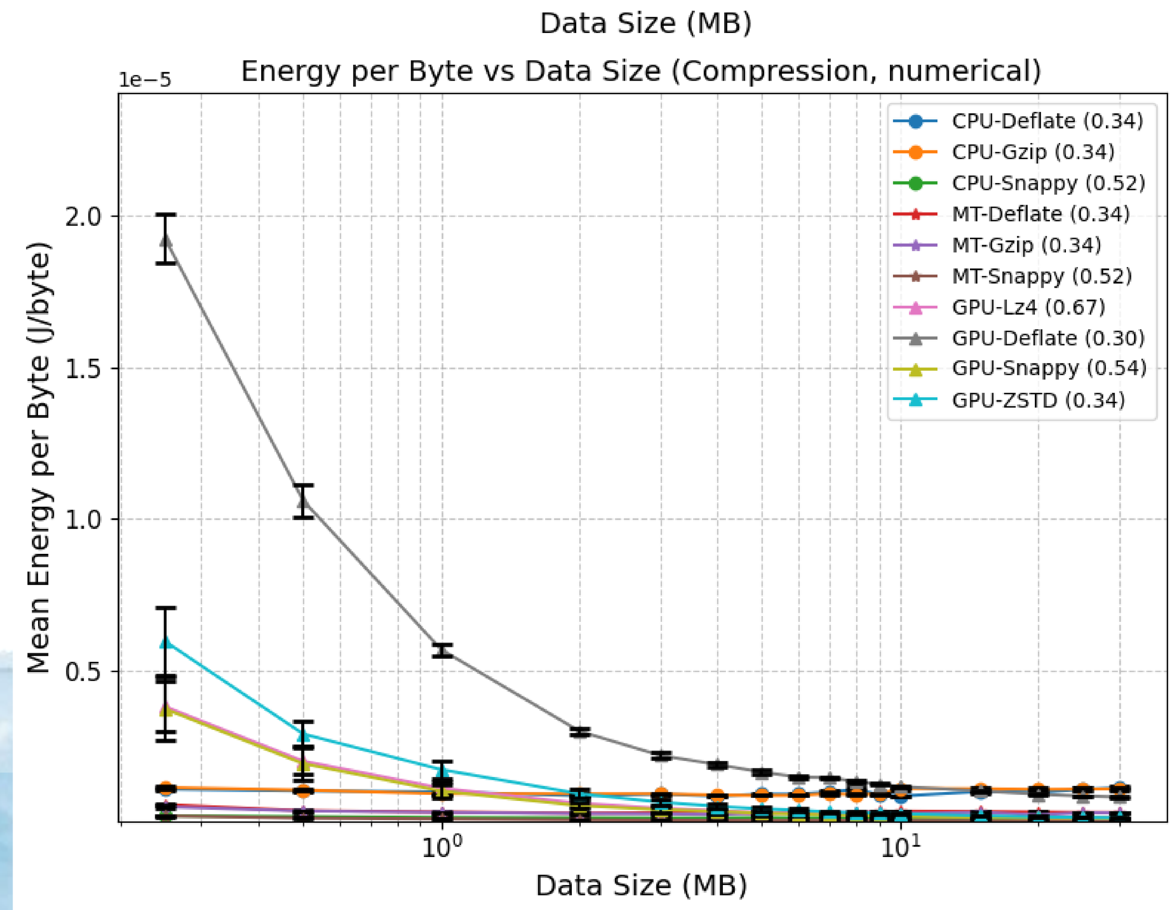
- GPU algorithms faster for large datasets (>10MB)
- Multi-threaded CPU competitive for smaller data
  Snappy excels for audio and streaming

Throughput:

- Multi-threaded CPU achieves highest throughput
- GPU algorithms good for large data

Energy Efficiency:

- Multi-threaded CPU offers the best energy efficiency
- GPU more energy efficient with larger datasets
- Single-threaded CPU consumes most energy



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 SC25 Website ➡