StoreEdge RippleStream
Versatile Infrastructure for IoT Data Transfer

Presenter: Peter Corbett
Authors: Madhumita Bharde, Annmarie Justine K, Suparna Bhattacharya, Dileep Deepa Shree
Affiliation: HPE HCI Engineering

USENIX HotEdge, July 10, 2018
Motivation: Versatile Edge Infrastructure

• Benefits of traditional software defined infrastructure
  • End to end management across a federation of edges + core
  • In-built data protection with asynchronous WAN optimized replication
  • High efficiencies for data-at-rest
    • compression (short range redundancies) + de-duplication (long range redundancies)
    • efficiencies improve over time when data has redundancies over days, weeks, months, ..

• Benefits of an edge computing optimized stack
  • Optimized for streaming data flows
  • Low latency action
  • Application defined filtering

What would it take to leverage the best of both worlds?
Motivation: Efficient IoT Data Transfer

• Most data generated at edge will never transmit to the core
• It is necessary to transmit significant information to the core
  • High value data with low redundancy
• Storage replication techniques provide a useful mechanism for high-value information transfer
  • Already well-established in commercial storage and HCI products
  • Eliminate redundancy via compression & deduplication of transmitted data
• Study considers combining storage replication with other (semantic) data reduction techniques to increase the information density of transmitted data
• Technique is suitable when significant latency is tolerable in transmitting edge data to the core
Storage Replication for Streaming

EDGE (Real time analytics)

CORE (Deep analytics)

Application Layer Streaming
Storage Replication for Streaming

**EDGE** (Real time analytics)

**CORE** (Deep analytics)

**EDGE Server**
- Sensors
- Edge Gateway
- Streaming Framework
- Storage

**Storage Managed Streaming**

**CORE**
- Streaming Framework
- Storage

**WAN EFFICIENT DATA FABRIC**
Storage Replication for Streaming

EDGE (Real time analytics)

- Sensors
- Edge Gateway
- Edge Server
- Streaming Framework
- Storage

CORE (Deep analytics)

- Streaming Framework
- Storage

Bandwidth in Mbps

- Application streaming without compression
- Application streaming with compression
- Storage replication with compression

- 72KB record, 62 sensors 100ms

WAN EFFICIENT DATA FABRIC
Storage Replication for Streaming

EDGE  (Real time analytics)

CORE  (Deep analytics)

WAN EFFICIENT DATA FABRIC

1.6KB record, 1350 sensors, 100ms
Similarity Awareness: Time Series

**Edge Server**
- Early analytics and Compute

**Similarity Service Plugin**
- Intercept and feed stream to check similarity
- Apply Haar Wavelet Transform
- Lookup Coefficients (signature) In dictionary
- Replace/Link to similar data to make it dedup friendly

**Storage**
- Transfer Unique Data or Signatures

**Core**
- Deep analytics and compute

**Similarity Service Plugin**
- Reconstruct stream with error bound
- Apply Inverse Wavelet transform

**Storage**

Optional steps if transmitting signatures instead of data
Similarity Awareness: Time Series

Sensors

- Traffic Simulator 1
- Traffic Simulator 2
- Traffic Simulator 1350

Edge Server

- Moving Average
- Similarity plugin

Core

- Hotspot Analysis
- Archiver

Haar Wavelet Transform & compare coefficients to dedup similar data
Similarity Awareness: Time Series

**Sensors**
- Traffic Simulator 1
- Traffic Simulator 2
- Traffic Simulator 1350

**Edge Server**
- Moving Average
- Similarity plugin

**Core**
- Hotspot Analysis
- Archiver

**Graph:**
- Similarity-aware data reduction for time series
  - Original
  - Reduced

Metrics:
- GreenD
- Traffic
- Pollution

Tools:
- WIRESHARK
- TCPDUMP
- kafka
- rsync
Similarity Awareness: Autonomous Car Video

Video pre-processing
- Object Identification*
- Scene segmentation
- ...
- SLAM

Deep learning training (steering, throttling, ...)

Core
- Deep learning training (steering, throttling, ...)
- Route / navigation training

Car(Edge)
- Similarity Service Plugin
- Leverage Semantic pipeline for similarity check*
- Replace/Link similar frames to make them de-dup friendly

Storage
- WAN efficient mirror replication

Core
- Similarity Service Plugin
- Post-processing* (stitching/smoothing)
- Storage

Video pre-processing
- Object Identification*
- Scene segmentation
- ...
- SLAM

Deep learning training (steering, throttling, ...)

Core
- Deep learning training (steering, throttling, ...)
- Route / navigation training

Car(Edge)
- Similarity Service Plugin
- Leverage Semantic pipeline for similarity check*
- Replace/Link similar frames to make them de-dup friendly

Storage
- WAN efficient mirror replication

Core
- Similarity Service Plugin
- Post-processing* (stitching/smoothing)
- Storage

Video pre-processing
- Object Identification*
- Scene segmentation
- ...
- SLAM

Deep learning training (steering, throttling, ...)

Core
- Deep learning training (steering, throttling, ...)
- Route / navigation training

Car(Edge)
- Similarity Service Plugin
- Leverage Semantic pipeline for similarity check*
- Replace/Link similar frames to make them de-dup friendly

Storage
- WAN efficient mirror replication

Core
- Similarity Service Plugin
- Post-processing* (stitching/smoothing)
- Storage
Data Reduction for Autonomous Cars

- Kitti (6.3 GB): 2% (Gzip), 42% (Frames), 13% (Video)
- UMich Downtown (78 GB): 26% (Gzip), 66% (Frames), 60% (Video)
- Umich Ford (119 GB): 27% (Gzip), 66% (Frames), 51% (Video)
- CCSAD* (8 MB): 17% (Gzip), 38% (Frames), 23% (Video)
Co-design framework and infrastructure layers for versatility
  • Apply storage replication to facilitate high-value data transmission from edge to core
  • In conjunction with other data reduction techniques that work on different scopes of data, therefore different timescale ranges of input values
  • Needs separable streams, Application consistent triggers, Semantic similarity plugin
  • Balance Edge-Core compute distribution to enable detection of semantic redundancies

Managing accuracy vs data reduction
  • Adjusting similarity metrics (application guided or automatic learning)
  • Exact vs semantic views
  • Shifting exact data sensitive computation
  • Post-process correction

Achieves efficiency at expense of latency and/or accuracy
  • Under what conditions is this sufficient? When would this category of approach fail?