TTL-based Approach for Data Aggregation in Geo-Distributed Streaming Analytics

Dhruv Kumar, Abhishek Chandra
University of Minnesota
Twin Cities, MN
{kumar434, chandra}@umn.edu

Jian Li, Ramesh K. Sitaraman
University of Massachusetts
Amherst, MA
{jianli, ramesh}@cs.umass.edu

1. Motivation and Problem Statement

Geo-distributed data streams
• Video streaming: client logs
• CDN: server logs
• Smart homes: sensor data

Real-time analytics
• Monitor QoS metrics
• Web content access
• Energy consumption

WAN Bandwidth: Scarce or Expensive
• Pure Streaming: High Traffic, Low Delay
• Pure Batching: Low Traffic, High Delay

Proposed Approach:
TTL-based Approach for jointly minimizing delay and traffic

2. Problem Formulation

Model:
• Edge modeled as a TTL (Time-to-live) Cache
• TTL is the time for which a key is kept in cache

Intuition:
• High TTL ↔ More Delay, Less Traffic
• Low TTL ↔ Less Delay, More Traffic

Objective:
Find optimal TTL for every key

3. Optimization Model

Main idea: Compute TTL based on the arrival rate of any key

Input: Trade-off parameter(α), Arrival rate of each key(λ_k), Unit delay cost(d_k), Unit traffic cost(c_k)

\[ C = \alpha c_{delay} + (1 - \alpha) c_{traffic} \]

Method: \( C \) is convex in \( TTL_k \) and has a closed form solution.

Output: TTL for each key

\[ TTL_k = \frac{2(1 - \alpha)c_k}{\alpha d_k \lambda_k} \]

4. Implementation and Evaluation

Implementation
• Implemented TTL Cache primitive in Apache Flink

Experimental Setup
• Data set: Month-long Akamai download analytics trace
• Deployment: Geo-distributed test bed using AWS EC2 as well as emulated test bed using local clusters.

5. Other Optimization Models

Minimize delay with a traffic constraint:
• Initial Solution: Allocate bandwidth in the order of decreasing delay cost (d_k)
• Enables differentiated QoS for multi-class traffic

Minimize traffic with a delay constraint:
• Initial Solution: Stream keys with lower traffic cost (c_k)
• Enables guaranteed data delivery time

Work in Progress:
1. Incorporate fairness into the models to avoid starvation
2. Dynamically modify TTL based on changing arrival rates
3. More extensive experimentation on AWS EC2 sites.

Acknowledgement: This research was sponsored by the NSF under Grant CNS-1717834, 1717179, and by the U.S. ARL and the U.K. MoD under Agreement Number W911NF-16-3-0001.