

# Applications of the Real Time Streaming Processing with Edge Computing Gateways

Chao-Yih Hsia(夏肇毅)

[CubicPower.idv.tw](http://CubicPower.idv.tw)

Taipei, Taiwan

chaoyihxia@gmail.com

## ABSTRACT

Edge Computing Gateways are situated between cloud and local networks. It converts high-speed local network to low-speed remote network. Data must be reduced by extracting meaningful data only before sending out to the cloud. Real-time stream data from cloud must be disseminated to all local nodes efficiently. All value added functions must be performed in this gateway. This paper demonstrates some applications suitable for the streaming edge computing environment.

Keywords: Edge Computing, Complex Event Processing, Real-Time, Data Dissemination, Data Collection.

## 1. Introduction

An edge computing gateway[1] plays multiple roles at the same time. It receives incoming data from the cloud, processes it and disseminates them to the local nodes. On the other hand, it collects data from local nodes, summarizes the data and send them to the cloud centers. Key components of the edge computing gateway are: a Complex Event Processor (CEP), an Message Queue (MQ) system and a fast real-time DataBase.

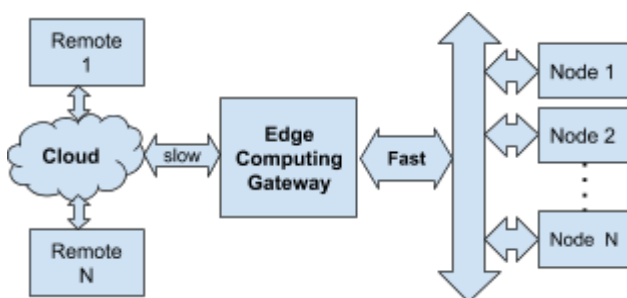


Figure: Edge Computing Gateway in Between Two Systems

## 2. Data

### 2.1 Streaming data

Streaming data is the data generated continuously from multiple sources. Normally these data come with arrival timestamps to process them sequentially. Since data keeps pouring into the system, so we have to use the sliding window technique to remove all expired data outside the sliding window. A typical example of streaming data is stock prices. Stock quotes, prices keep coming tick by tick. Each tick includes price, time stamp and other related information. Normally we keep these tick data in two places: the CEP sliding window and a database. The CEP processes data and throw them away, but a database always tries to keep them forever.

### 2.2 Tick by Tick data

Market data from the data feed is sent in a tick by tick way. A tick from the data feed must be shown on all users desktop immediately. The high frequency trading system applications usually expect a very low latency environment. Industry gateway could offer a service level of less than 10 us latency. Top-notch products made by FPGA can reach a level of 3 us latency.

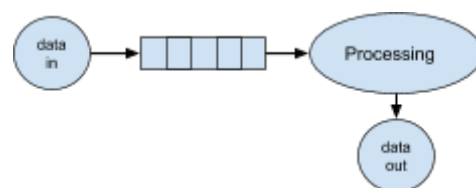


Figure: Tick Data Processing from a queue

### 2.3 Periodical Data: Minute Data

Periodical data like minute data, hourly data or even daily data could be generated from tick data in the sliding

window. These types of data keep only the open, high, low, close data of each period. Therefore, converting data from tick to period can offer a very high compression rate.

## 2.4 Tick by Tick or Micro-Batch Processing

If a process can handle 1000 tasks per second. When we do a task with tick by tick basis, then the maximum throughput will be 1000 tasks/sec. If each task can handle 1 message, the throughput will be 1000 messages/sec.

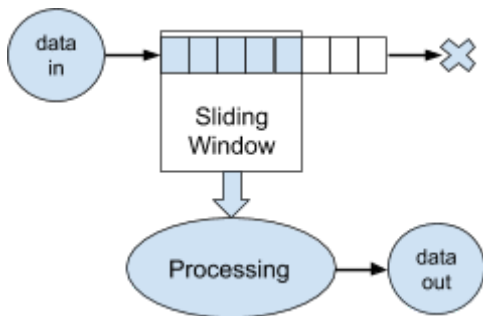


Figure: Batch Data Processing from a Sliding Window

But if we only do 100 tasks, but each task handles 20 messages, then the overall throughput becomes 2000 messages/sec. One task handles one message is a tick by tick process. One task handles multiple messages is a micro-batch process. The ratio of variable cost and fixed cost of a task decides which type of process performs better. The smaller ratio is the variable cost, and the micro-batch performance will be better.

## 3. Framework

### 3.1 Complex Event Processing

“Complex event processing, or CEP, is event processing that combines data from multiple sources to infer events or patterns that suggest more complicated circumstances.” [2] For instance, if we heard church bells ringing and saw a group of people with a white gown woman, then we can guess there should be a wedding. This method depends on some key steps:

- Collect real-time data from different sources
- Keep data in a sliding window
- Match with the existing patterns
- Generate an output signal
- Distribute signal to all entities

### 3.2 Message Processing Queue

Data flows between nodes should be handled by a Message Queue system (MQ) in a real-time streaming processing system. Messages should be disseminated between nodes and the gateway in real-time with minimum latency. Also, not all but only selected data would be sent to the nodes as requested. Popular MQ systems like JMS, ZMQ, RabbitMQ[3] and MQTT have been widely used in enterprise systems to exchange messages between units. MQ can be configured as queue, worker, rpc or publish / subscribe mode depending on the function we need. Some MQ system can handle persistent messages, but some can not. We need to choose correct system based on the application characteristic.

### 3.3 Market Data Hosted on the Edge Gateway

The most common real-time streaming data is the market data feed of stock prices in the financial institutions. Data feed from stock exchanges pours real-time broadcasted market data to the edge gateway. This data should be kept in the edge database and sent to all nodes depending on requested topics from MQ pub/sub channel. If there is any missing message, any node can re-request it from the edge gateway again thru the MQ rpc channel. All added value added data like VWAP, TA signal or AI price prediction should only be produced once by the gateway, and then be sent to all subscribing nodes.

## 4. DataFlows

### 4.1 Real-time Data Flow on the Office

Exchange data feed sends market data to the edge gateway, and then the gateway sends them to the local nodes. Most of time, the local nodes of the office are desktop PC or notebook workstations. People often use an Excel spreadsheet to generate reports with market data. Another tool called DDE server will be installed on each PC to transfer data into an Excel spreadsheet in real-time. Therefore, we can see that all desktop PC spreadsheets change their stock prices almost at the same time when any stock price changed.

### 4.2 Incoming Data Dissemination

MQ Publish / Subscribe mode should be used to handle data dissemination task to all local nodes. Each node is allowed to subscribe its owned topics. A publisher sends out the complete set of data, but only the subscribed topics will be received by subscribers.

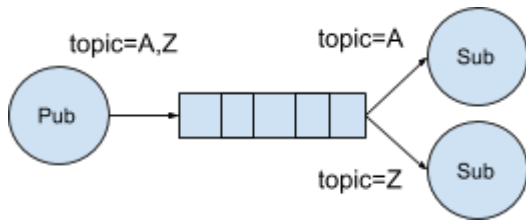


Figure: MQ Topic Pub/Sub mode

### 4.3 Outgoing Data Collection

The typical role of an edge computing gateway is to collect data from nodes periodically, such as heart beat rates, the water level of the river and measures of machines. Algorithms on the gateways check if there is anything wrong. If yes, it reports as and an issue event immediately. If no, it summarizes the data into statistics report and send them back to cloud hosts periodically. This computation power of the edge computation gateway could avoid sending huge amount of useless data to the hosts thru cloud.

## 5. Applications

### 5.1 Pattern Detection

Predefined patterns could be detected on the streaming data by CEP, such as up, down changes of stock prices, the convolution of the stream data like Code Division Multiple Access (CDMA), Volume Weighted Average Prices (VWAP) of the stock prices and trading volumes.

### 5.2 Over Threshold Detection

The CEP can compute the moving average(MA) of the data in the sliding window and then check whether any data exceeds the level of MA times a constant. We can also compute the standard deviation (SD) of data and use the SD times a constant as a threshold. This is a way to detect the abnormal condition.

### 5.3 Cumulate Data

The edge computing gateway will buffer the input data in micro-batch processing mode and start processing after a small interval of waiting time. During this period, data can be kept in the CEP sliding window.

### 5.4 Edge AI Network Center

An edge gateway could be the center of the AI network. Each node of the AI network runs its algorithm on its own machine and then sends the result to the gateway. The gateway computes all necessary statistics from data aggregation. Based on this overall statistic of all nodes, problems could be found with a better chance.

### 5.5 Order Matching System

If a gateway performs the order matching server function like stock matching or bidding server, we can select either tick by tick or micro-batch execution way. First, we sort the buy-side and sell-side orders and then match two lists to see if any execution price satisfies both sides.

### 5.6 Financial Data FIX Protocol Adaptor

If the application of edge computing gateway is used in a financial institution for trading, normally a FIX protocol adaptor must be implemented in the gateway. In this case, the latency will be the top issue to consider. To cut the latency, some steps were taken frequently:

- Do less in the critical path
- Reduce layers
- Optimize Physical Location and access

### 5.7 Credit Card Transactions Monitoring

We can use this structure to check personal credit card transactions on different shops of a mall to check whether any person who used many cards in different shops has exceeded a certain ratio threshold. We can collect the transaction data from shops as (card, shop, product, amount) and add them together with a “Select ..., SUM( amount) ... Group By card, shop, product” command. Then we can get amounts of all different (card, shop, product) combination. And then we can also compute the amounts of the (card, shop) combination with similar way. By checking the ratios the these two results, we will know the selling ratio of products in shops. Some applications check whether these ratios are too high, and some applications check these ratios are too low.

If we have the card owner data (person, card), we can also combine the card transactions into personal transactions by mapping (card, shop, product) data into (person, shop, product) before summation.

## 5.8 Stock Trades Monitoring

A stock trading surveillance system could be implemented in a similar method. First, cumulate the values of Level 4:(account, broker, stock, amount) and Level 3:(account, broker, amount). Then compute the ratio of these two values Level 4/Level 3. Finally, set an alarm if this ratio exceeds a threshold. It means this account has bought too many of certain type of stocks in the broker. All ratios of different levels can be checked and the order of data fields in the tuple can also be switched for different meanings.

## 5.9 Server Status Monitoring

Node Machines can send their status to their gateway like loading of CPU, memory, disk and network, temperature of CPU and Disk. The gateway computes their statistics and detect abnormal conditions. Summarized report and alarm will be sent to the remote hosts thru the cloud.

## 5.10 Intrusion Detection

As we mentioned in the CEP case, if we can check the pattern of first A, then B and finally C, then we can get a conclusion that some event happens. If we know the normal attacking method of hackers, by searching different logs for these patterns, we can detect intrusion attempts actively.

## 5.11 Summarizing Access Logs

A web site normally consists of many web and application servers. The director dispatches web requests randomly to any server. Access and execution logs are situated in each server. We can join these logs into the gateway to see if any client exceeds allowed quota.

## 6. Conclusion

The key role of the edge computing gateway is to generate real-time value added signals from cumulate data and disseminate them. We've tested the edge computing gateway performance on streaming stock data processing. Our experiment with some fast streaming data processor can finish a GROUP BY aggregation command of around 13,000,000 rows within a few seconds[5]. The aggregation function reduces the data to less than 1,000,000 buy or sell signals. From here we can see that a fast CEP data aggregation ability is the key to implement the real-time applications of this paper successfully.

## References:

- [1] wikipedia,"Edge Computing", [https://en.wikipedia.org/wiki/Edge\\_computing](https://en.wikipedia.org/wiki/Edge_computing)
- [2] wikipedia,"Complex Event Processing", [https://en.wikipedia.org/wiki/Complex\\_event\\_processing](https://en.wikipedia.org/wiki/Complex_event_processing)
- [3] "RabbitMQ", <https://www.rabbitmq.com>
- [4] hortonworks,"Apache storm design pattern micro batching", <https://hortonworks.com/blog/apache-storm-design-pattern-micro-batching>
- [5] Chao-Yih Hsia,"Fast Program Trading Strategies Performance Evaluation", <https://chaoyihhsia.blogspot.tw/2017/09/fast-program-trading-strategies.html>

