Railway Big Data Real-time Processing Based on Storm

Shihang Guo, Lichen Zhang

School of Computer Science and Technology Guangdong University of Technology Guangzhou 510000, China

381997826@qq.com

Keywords: Storm, Big Data, Railway, GIS

Abstract. Train Operation Control System is a typical Cyber-physical System which is combined with sensors, decision-making control unit and actuators. There are more and more sensors are used in railway, as a result these sensors produces Big Data stream data. In view of this situation, we propose the architecture for railway Big Data processing based on Storm. On the base of the architecture, we focus on the real-time stream processing which is the most important for railway Cyber-physical. We use Storm to process the GIS data of trains and prove the feasibility of the method.

1. Introduction

Big data driven railway cyber-physical system needs real-time stream processing because train operation control system is a safety-critical system where data processing and decision making is real time[1].The most popular bid data processing method is MapReduce based on Hadoop, but it builds mappers and reducers everytime. Moreover, MapReduce needs to reaload hostory data and can not meet the continuous stream data from sensors. As a result, MapReduce is unable to use the previous processing result and there are gaps between stream data. So, Hadoop MapReduce is not a proper processing method for railway cyber-physical real-time processing. Storm is a good real-time stream data processing method opened by Twitter company. Storm assures that every stream data produced by souces can be processed, so we decide to use Storm to meet the real-time demand in railway field.In addition, the distribution of Storm workers which are the Storm main processing processes provises the realiability of system for railway.

2. The Architecture of Railway Big Data Procssing

2.1 Data Collection Subsytem.

Data collection subsystem can be divided into two specific parts.

1) The Flume architecture of data collection sybsystem

Flume uses the distribution architecture and is depolyed in different servers.The Flume process in every server monitors the specified folders and every Flume instance is an Agent composed with data source, data channel and data sink. When there is new data input in folders, data source components read the data and put it into a channel. Channel is a buffer realized through memory.After that, the sink gets data from channels and sent it into Kafka clusters.

2) Data buffer module

The meaning of buffer module is that at some moment the stream data of railway is large and the amount of data needing process exceed the process ability of data processing module. As a result, there is possible that data processing module abandon some data. Therefore, the data buffer module protect the data processing module and avoid the risk of system breakdown. The data buffer module use Kafka cluster which is a producer-consumer model. Kafka firstly divides the cluster into many Brokers and then builds the corresponding Partitions which are buffer arrays. Producers put the data into Partitions, then the Consumers has subscribed specific topic can take the data from the Partitions with some topics.

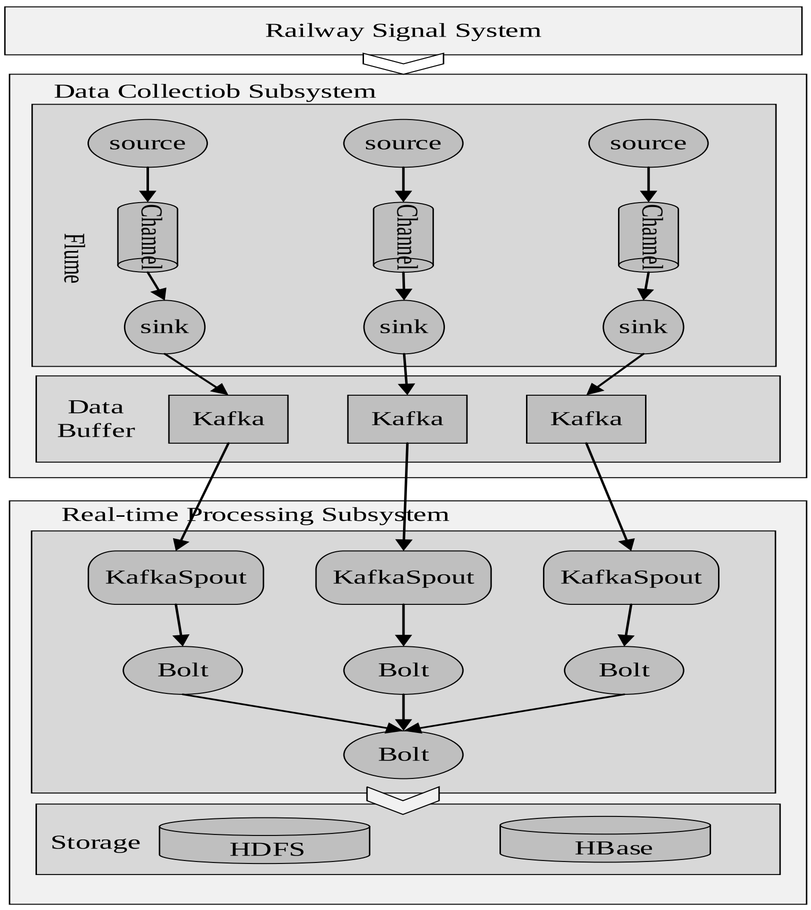

Fig.1 The Architecture of Railway Big Data Procssing

2.2 Real-time Processing Subsystem.

Real-time processing subsystem adopts Storm's distribution proccesing architecture to process railway real-time stream data. The nodes of a Storm cluster are classied into Nimbus and Supervisor. Supervisors can decide the amouts of Workers. The corresponding main processing processes of Workers are called Bolt. Bolts are reaponsible for specific working. The chain of Bolts called Topology in Storm. The first Bolt of Topology has a special name, Spout. Spouts read data from Kafka, which means the start of Topology processing. Topologies can guide the output of Bolt to input the different data for different Bolts and every Bolt works by definition. The last Bolt of a Topology is in charge of the final results output. Then, there is an example of GIS data to state the real-time processing of railway data below.

3. Train GIS Data Processing

The train's GIS data illustrates the real-time location and speed of a tain, so it can be used to track a train running and to calculate the distance of two train in same rail. The distance we acquired can be used to alarm the risk of the approach of two trains.

A. Train location correction

Because of the deviation of train location technology, sometimes the train's GIS data does not match the rail location data. That is the train is not on the rail it runs when showed on map. Thereofor, it is need to correct the GIS data, so the GIS data is matching to the rail data in databases.

Step 1: Spouts get GIS data from subscribed Topic in Kafka clusters. Storm will dispatch Spouts constinuously during running and push data into Topologies which process data by definitely.

Step 2: A Bolt called SplBolt splits the data into fileds following the specific data structure. If the amounts of fields splited are less than the sepified amounts, it means that this piece of GIS data is not integrated. So, the not integrated GIS data needs to be abandoned. The SplBolt Objectifies the GIS objects using the fields and output the object to CorBolt by the Tuple data form. At the same time, SplBolt outputs raw GIS data to Hbase.

Step 3: CorBolts after SplBolts get the Tuple of GIS data and use the traditional GIS location correcstion method to correct the raw GIS data. The meaning of this process is to match the GIS data to the corresponding rail the train runs. Fig.2 shows the whole Bolts procedure of GIS data correction.

Fig.2 The Inner Architecture of Data Collection Subsystem

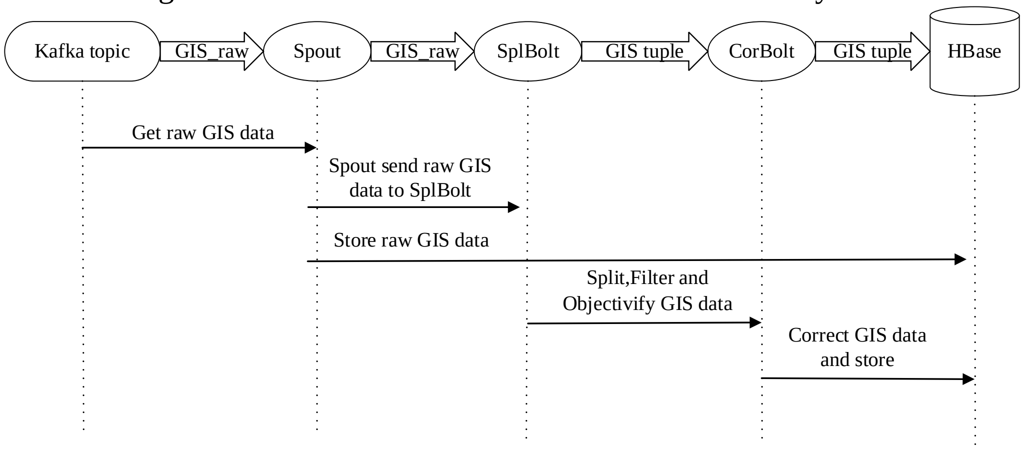

Fig.3 The Topology Structure of GIS Location Data Correction

B. Train location correction

Train's real-time location can help staff to look after the train and split two trains with safe distance, which can protect trains from collisions. So, to calculate the real-time distance of two trains after getting real-time location of trains is a good approach for railway safety. Obviously, we only need to calculte the trains on the same rail not all the trains at same time because trains on different rails can not run into each other.

The location of train Ω is $GIS_{\Omega}(t)$ at time t. Because of the latency of transmission, if the system does not reveive the GIS data for σ time, output the alarm of this train information. So, the distance of two trains Ω and Ψ is as follows:

$$Distance(\Omega, \Psi) \approx Distance(GIS_{\Omega}(t), GIS_{\Psi}(t)). \quad (1)$$

When the distance of two trains is narrow, we can think the Earth as a circle approximately. Moreover, the camber of rails can be ignored because there is no large camber on rails. Therefore, we can use following method to calculate the distance of two trains where R represents the radius of Earth.

$$\begin{aligned} C_{\Omega,\Psi} = {} & sin(latitude_{\Omega}(t)) \times sin(latitude(t)) \times \\ & cos(longitude_{\Omega}(t) - longitude_{\Psi}(t) + \qquad (2) \\ & cos(latitude_{\Omega}(t)) \times cos(latitude_{\Psi}(t)); \end{aligned}$$

$$Distance(GIS_{\Omega}(t), GIS_{\Psi}(t)) = R \times arccos(C_{\Omega,\Psi}(t)) \times \pi \div 180; \qquad (3)$$

The steps of train safe distance calculation is as followed:

Step 1: SortBolts get the GIS data has been corrected from CorBolts and then it sort trains into different groups according to train location. Fields Grouping functions sorts the trains with same rail into a same DisBolt. DisBolt is the bolt which is responsible for distance calculation.

Step 2: DisBolts adopt HashMap B+ data structure to store GIS objects. The HashMap B+ data structure is showed in Fig.4 . TrainNo is used as the key of the HashMap and the corresponding value is the GIS object. The GIS objects having the same key are stored in a same B+ tree. The DisBolt gets the newest GIS objects of every HashMap key every time and these GIS objects form the set S. The following step are as followed:

```
1) For each $s$ in $S$ do
2) if $Now() - s.getTime() > \sigma$ do
3) 输出{s};
4) For each $s1, s2$ in $S$ do
```

```
5）$D = Distance(s1.getPosition, s2.getPosition)$；
6）$if\ D < \rho\ do$
7）输出{s1, s2}；
```

Fig.4 illustrates the Topology architecture of train safe distance calculating.

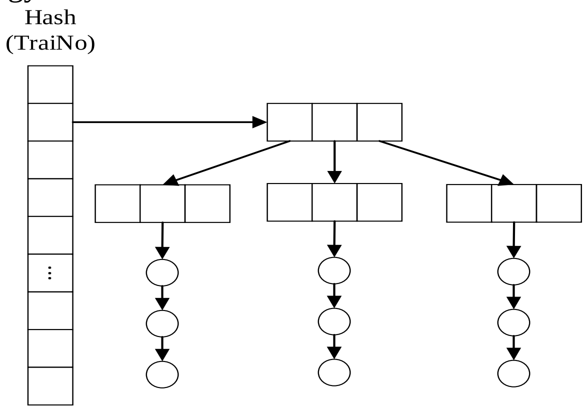

Fig.3 The Data Structure of the DisBolt

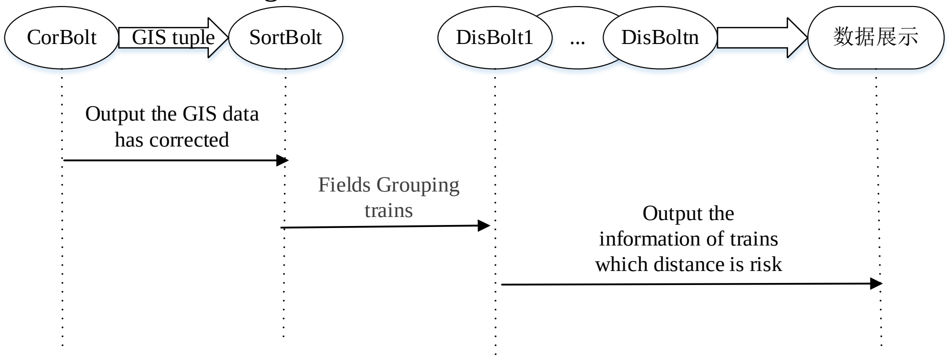
CorBolt
GIS tuple
SortBolt
DisBolt1
...
DisBoltn
数据展示
Output the GIS data has corrected
Fields Grouping trains
Output the information of trains which distance is risk

Fig.4 The Topology Structure of Train Safe Distance Calculating

4. Summary

Big Data drivend railway Cyber-physical system not only needs to focus on the traditional CPS modeling and verificationg, but also the Big Data storage and processing. railway Cyber-physical system is a safety-critical system in which realtime is really an important factor. So, our real-time approach based on Storm is a good try for railway Big Data stream real-time processing.

References

[1]. Guo Danqing, Lv Jidong, Wang Shuling, et al. Formal analysis and verification of Chinese train control system[J]. China Science, 2015，45(3):417-438.

[2]. Dun Lilong, Xu Haishui. Research on Applied Models Based on Storm[J]. Journal of Guangdong University of Technology, 2014, 31(3):114-118.

[3]. Dillon T S，Zhuge H，Wu C，et al. Web-of things framework for cyber-physical systems[J]. Concurrency and Computation，2011，23(9):905-923.

[4]. A Taherkordi，F Eliassen. Towards Independent In-Cloud Evolution of Cyber-Physical Systems[C]. IEEE International Conference on Cyber-physical System，2014:19-24.

[5]. Shahrivari S，Jalli S. Beyond Batch Processing: Towards Real-Time and Streaming Big Data[J]. Computers，2014，3(4):117-129.