Review: Data Driven Traffic Flow Forecasting using MapReduce in Distributed Modelling

Vidhya N.Gavali* — *Computer Engineering, DCOER, Pune University* — vidhya.n.gavali@gmail.com

Prof. Prashant Mane — *IT Department, DCOER, Pune University* — prashant.mane@zealeducation.com

Rashmi R. Patil — *Computer Engineering, DCOER, Pune University* — rashmiashtagi@gmail.com

Abstract— ***from last decade, the use of communication and transportation technology increases in urban traffic management system. To predict the correct result forecasting technique is used. Furthermore, as more data are collected, increase in traffic data. In short, traffic flow forecasting system find out collection of historical observations for records similar to the current conditions and uses these to estimate the future state of the system. In this paper we focus on data driven traffic flow forecasting system which is based on MapReduce framework for distributed system with Bayesian network approach. For probability distribution of data between two adjacent node i.e. data used for forecasting(Input node) and data which is forecasted (output node) used a Gaussian mixture model (GMM) whose parameters are updated using Expectation Maximization algorithm. Finally focus on model fusion, main problem in distributed modelling for data storage and processing in traffic flow forecasting system.***

Keywords— Distributed modelling, MapReduce, Gaussian mixture model, model fusion, traffic flow forecasting

I. INTRODUCTION

URBAN traffic control systems (UTCSs) and freeway management systems around the world are collecting large amount of traffic condition data every day. Typical data include volume, flow rate, occupancy, and speed. Development of systems that put these data to good use for traffic control and management has become an active area of ongoing transportation research, which is usually referred to as Intelligent Transportation Systems (ITS) [1].There are various sources available from which we get the data and for processing on such a huge data required a traffic management system which convert it into desirable outputs i.e. information. With the use of communication Technology, large amount of traffic data obtained with very less efforts, traffic system is a dynamic, complex, huge system within a network. There are many approaches proposed for traffic flow forecasting .some approaches based on data mining using various methods mentioned below:

1. Kalman filter theory
2. neural network model
3. fuzzy neural model
4. type-2 fuzzy logic model
5. Bayesian network model

The learning and training process in these methods is data driven. To increase the performance of above mentioned methods need to update the parameters based on real time situation. That means uses a more recent data because traffic condition must affect any small changes in the configuration (e.g. decision support systems for improving business operation may identify and predict changes and trends).Dynamic changes should be kept in learning process. But when we are going to data processing it is very complicated on standalone mode because of computational time and efforts. Second thing, storage capacity of standalone system is not much larger. The overall result degrades the performance of traffic forecasting system. To effectively manage the computation time and storage space distributed modelling approach is used using Hadoop database system [1].

In the process of model learning MapReduce technique is used which is invented by Google. It is used in many application like text processing, in the analysis of social network, etc [1].In this paper, we mainly focus on data driven traffic flow forecasting using MapReduce technique for distributed modelling and Bayesian network approaching context of network utilization. So first we describe the current techniques used for traffic flow forecasting and then see how Hadoop database system is better for traffic flow forecasting.

II. LITERATURE SURVEY

Many existing traffic forecasting algorithms were developed, considering only a single site, instead of a whole region. In the former case, the data processing part is not an issue. Recently, more and more research efforts have been devoted toward developing systems for region-wide traffic flow forecasting. When it comes to a whole region, the data issue can be very pronounced. It is very important to study the utilization of information in such a network [1].

In paper [2], Traffic flow prediction is established seasonal time series methods, especially seasonal ARIMA modelling .But it generates only one forecast value rather than range of forecast values. Second paper [3] is Kalman Filter Theory proposed for predicting short-term traffic volume which determine the traffic in the interval of only next five minutes or half and hour. Third paper uses a neural network model for data driven traffic prediction in broadband

networks. This prediction is possible for certain traffic types but not for others [4].Fourth one is Type -2 fuzzy logic model used in Day-to-day traffic information is combined with real-time traffic information to construct fuzzy rules Need to tune the parameters to optimize the performance of the model because there are only limited data sets available [5].
So to expand data storage and data processing in distributed traffic flow forecasting MapReduce framework is used. In next section, we first give a brief introduction to a MapReduce job and then will see the architecture of traffic flow forecasting using MapReduce Framework.

III. HADOOP SYSTEM

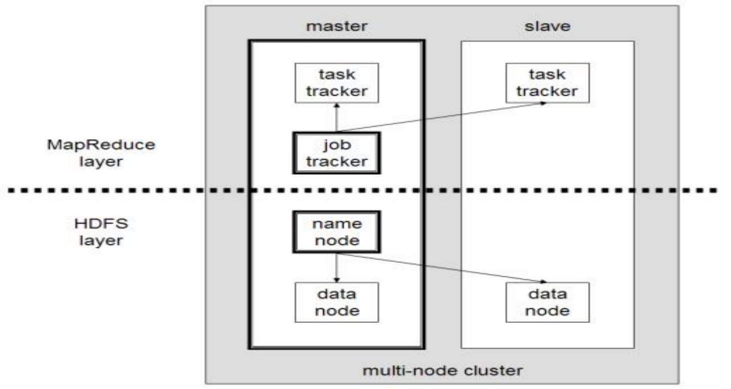

Fig. 1: Hadoop System [7]

Hadoop system consists of two layers:
1. Hadoop Distributed File System (HDFS)
2. MapReduce Layer (Execution Engine)

HDFS is mainly used for storage of huge amount of data .HDFS consists name node and data node. Centralised name node keeps the metadata means information about file. The input file is divided into number of blocks. The block size is fixed which is 64MB. In this layer, many data nodes exist. The replicated copies of blocks are stored in data node. Each block is replicated N times.
MapReduce framework consists of job tracker and task tracker. The job tracker works like master node. It receives the user's request and from that decides that how many mapper will require to run tasks and where to locate the mapper (e.g. if file is divided into 5 blocks then 5 map task will run).
Comparison between Hadoop and other Distributed system:

TABLE I
COMPARISON OF DISTRIBUTED DATABASE AND HADOOP

Characteristics	Distributed Database	Hadoop
Computing Model	Transaction is the unit for work and support for concurrency control and ACID properties	The unit for work is jobs and concurrency control is not supported.
Data Model	Uses structured data with known schema. Support for Read/Write operation	Uses structured, semi-structured, unstructured data. -Read mode
Cost model	Server cost is high	Cheap commodity machines are available.
Fault Tolerance	Failures are rare. It has Recovery mechanisms.	Because of thousands machine failures are common.

System Architecture overview for Traffic Flow Forecasting Using Hadoop System:
The system architecture focuses on three important problems that will arrive in distributed modelling not in standalone mode [1]:

a. How to select number of component from local Gaussian mixture model (GMM).

b. How effectively model fusion is used to reduce computation time and obtain a accurate result.

c. Merging of several local models into a single global model.

For solving above mentioned problem Expectation Maximization algorithm is used with MapReduce framework.

The Traffic Flow Forecasting is divided into two parts. One is data storage and second part is data processing. For storing huge amount data (e.g. petabytes or gigabyte) which is needed those application having large database Hadoop Distributed File System (HDFS) is used and for data processing MapReduce is used. This system architecture consists of two computers one is master node and other is slave node. The master nodes assigns or distribute various algorithm to slave node .In the next step slave node takes that particular algorithm and complete its execution in map phase [1].

The Expectation Maximization (EM) algorithm which is also called as model learning algorithm is used in map phase. The input for this model learning algorithm is distributed traffic data. This traffic data is used in local model learning. The nature of EM algorithm is iterative contains hundreds of MapReduce job and these iterations are controlled by master node. The reduce phase uses a global model merging algorithm to obtain a single global model which is used for traffic flow forecasting with real time traffic flow data. [1].

IV. WHAT IS MAPREDUCE FRAMEWORK?

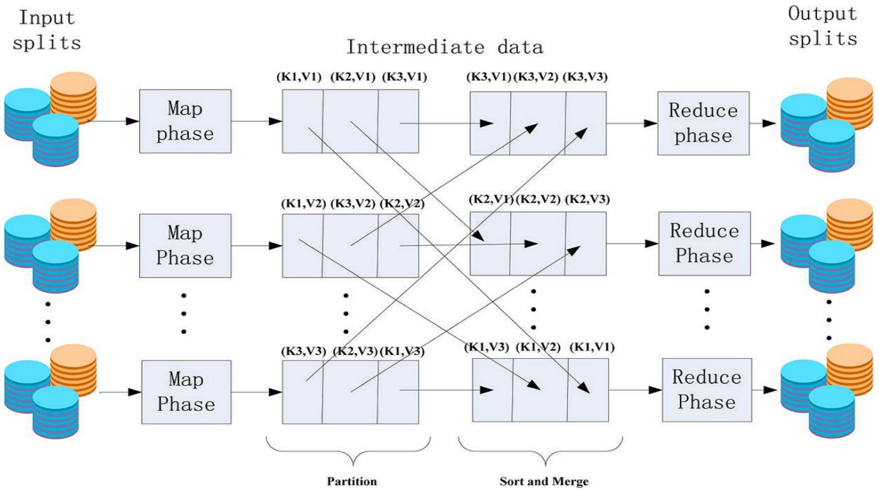

Fig.2: MapReduce Framework [1]

In Distributed System, MapReduce is a very popular parallel programming model. It has a two function:

1. Map function
2. Reduce function

In map function large program is divided into smaller subprograms and the output of these subprogram is integrated in reduce function to obtain a final result. There is an intermediate state i.e. shuffling and sorting whose implementation details are hidden from user. The Map and reduce job both have a (key, value) pair. Accordingly that shuffling and sorting will be done. Sorting method used in MapReduce:

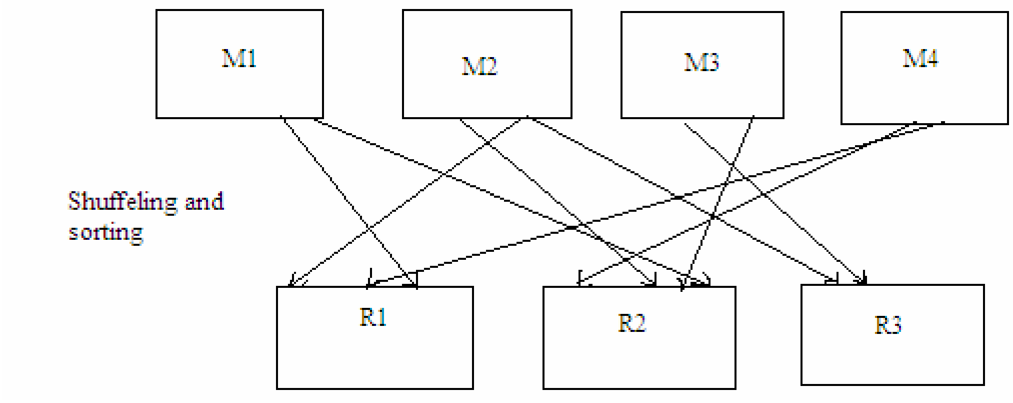

Fig. 3: Intermediate shuffling and sorting

Mapper: identity function for value

i.e. (Key_id, Value) -> (Value, _)

Reducer: identity function

i.e. (key_id’, _) -> (key_id’, “”)

(Key, Value) pair from Mapper sent to particular reducer using hash function. Hash function must be such that: key1< key2 => hash (key1) < hash (key2)

V. Distributed Model Learning

The distributed model learning consists of following steps:

1. Model selection using Gaussian Mixture Model(GMM)
2. Bayesian network for traffic flow forecasting.
3. Parameter learning with Expectation Maximization algorithm.
4. Distributed model fusion.

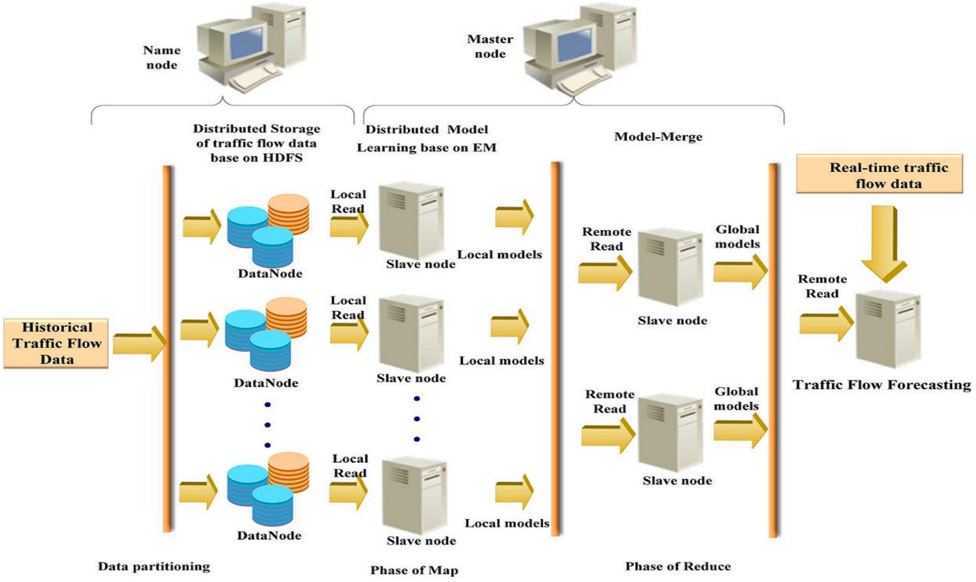

Fig.4: System Architecture for traffic flow forecasting [1]

Step 1:

GMM is a probabilistic distribution model used for probability distribution of data. The formula used for GMM is [1]:

$$p(X|\Theta) = \sum_{i=1}^{M} wipi\,(X|\theta i) \qquad (1)$$

Where

X=Training data set , M= number of mixture Gaussian models

Θ and θ = are the parameters of the total model and mixture model, respectively.

The traffic flow data can be categorized into three state:1) light 2) nominal 3) congested traffic data. Thus the GMM is used to categorize the entire data set [1].

Step 2:

A Bayesian network is known as a causal model which is a directed graphical model for representing conditional independencies between a set of random variables. It is matrimony between probability theory and graph theory. It provides a tool for solving two problems that occur through applied mathematics and engineering—uncertainty and complexity [8]. The simplest statement of conditional independence relationships encoded in a Bayesian network can be stated as follows: a node is independent of its ancestors given its parents, where the ancestor/parent relationship is with respect to some fixed topological ordering of the nodes [9]. Therefore, for a Bayesian network consisting of *n* nodes (random variables) (x_1, x_2. . . x_n), we have the representation for the joint probability distribution[8].

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(xi|xPi) \qquad (2)$$

Where p $(x_i|xP_i)$ is the local conditional probability distribution associated with node *i* and *Pi* is the set of indices labelling the parents of node *i* (*Pi* can be empty if node *i* has no parents) [8].

Step 3:

Expectation Maximization algorithm has two steps:

1) E-step to learn degree to in which model data will fit

2) M-Step to update the parameters based on old parameters.

The following figure 5 describes the processing of EM based on MapReduce [1]:

Step 4:

After dividing local models into *K* clusters, we need to get *K* global models C_i based on the weights, mean vectors, and covariance matrixes of local models C_i [1].

$$J(Ci,k) = \begin{cases} 0, Ci \in Ck \\ 1, Ci \notin Ck \end{cases} \qquad (2)$$

The local model C_i is defined as learning model ρ (C_i).When we are going to update the online global model with newly traffic flow data take into account the global model as local model for merging new local model. Thus the size of sample used for learning process is different [1].

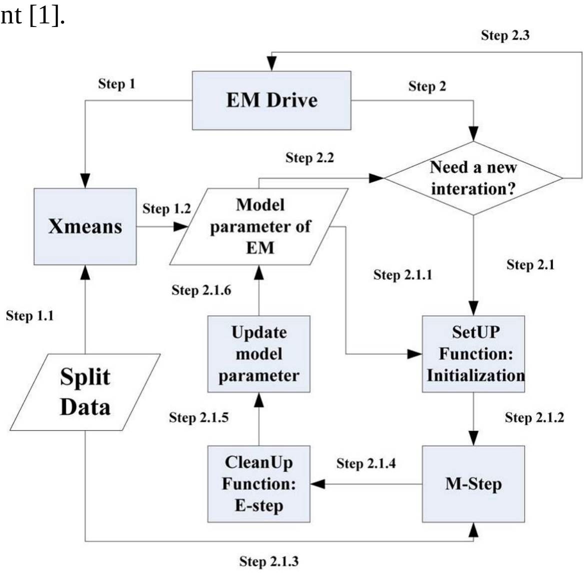

Fig 5: Processing of EM based on MapReduce [1]

VI. CONCLUSION

Data driven traffic flow forecasting application with distributed system strategies using MapReduce framework can be helpful to reduce computational efforts in standalone mode. MapReduce technique used to handle and process large amount of data in traffic flow forecasting system. Hadoop system has a high processing capacity, fault tolerance and scalability .Bayesian network is used to detect the traffic data because it shares common characteristics to utilize information within a network.

REFERENCES

[1] CHEN *et al.*,"Modelling In MapReduce Framework For Data-Driven Traffic Flow Forecasting" ,IEEE transactions On intelligent transportation systems, vol. 14, no. 1, march 2013

2] B. M. William, "Modelling and forecasting vehicular traffic flow as a seasonal stochastic time series process," Ph.D. Dissertation, Dept. Civil Eng., Univ.Virginia, Charlottesville, VA, 1999

[3] Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filter theory," *Transp. Res. Part B, Methodol.*, vol. 18B, no. 1, pp. 1–11, Feb. 1984

[4] J. Hall and P.Mars, "The limitations of artificial neural networks for traffic prediction," in *Proc. 3rd IEEE Symp. Computer. Commu.*. Athens, Greece, 1998, pp. 8–12.

[5] L. Li,W.-H. Lin, and H. Liu, "Type-2 fuzzy logic approach for short-term traffic forecasting," *Proc. Inst. Elect. Eng.—Intell. Transp. Syst.*, vol. 153,no. 1, pp. 33–40, Mar. 2006.

[6] Traffic Flow Forecasting Using Approximate Nearest Neighbor Nonparametric Regression *by* R. Keith Oswald,Dr. William T. Scherer.

[7] http://hadoop.apache.org/

[8] M. I. Jordan, *Learning in Graphical Models*. Cambridge, MA: MIT Press, 1999.

[9] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York: Springer-Verlag, 2001.