Research of methods for constructing message-passing interprocess communication based system for railroad situation analysis

Daria Kobyakova
Software Engineering School
National Research University Higher School of Economics
Moscow, Russia
dskobyakova@gmail.com

Scientific Advisor: Prof. Sergey Avdoshin
Software Engineering School
National Research University Higher School of Economics
Moscow, Russia
savdoshin@hse.ru

***Abstract*—This paper briefly outlines the research of parallel system constructing methods for railroad situation analysis and emergencies prediction. Within the scope of the research various paralleling method are considered and analyzed in terms of application for railroad emergencies prediction.**

***Keywords:* parallelism, emergencies prediction, real-time data mining, exceedingly large volumes of data processing.**

I. INTRODUCTION

Nowadays in railway cargo transportation area contingencies such as crime, theft, locomotive breakdown emergencies are controlled and regulated mostly by dispatcher's offices. And in most cases contingencies become known only after they have happened, what results in the loss of money, time and client confidence. The solution is to collect data both structured and unstructured, originated from a wide variety of sources such as cameras, sensors, news feeds, VoIP and traditional databases, analyze and thereby identify and prevent possible contingency situations or emergencies on the railways. It is clear that the volume of such source data can be enormous and is estimated in terabytes or even petabytes. In order to achieve effective and timely processing of enormous volumes of data real-time and with extremely low latency it is necessary to use parallel algorithms and indeed it is critical to choose an appropriate paralleling method. In this study various parallel system building methods will be analyzed in terms of application in railway situation analysis.

This work is being performed within the scope of the research on the topic "Research and development of innovative unifying models of intelligent systems for the situational response and safety control on the Russian railways", state contract 07.514.11.4039 on September 26, 2011 at lot № 2011-1.4-514-045 "Development of algorithms and software systems for solving problems of exceedingly large scientific data sets storage and processing and data streams collection in real-time" as part of the federal target program activity 1.4 " Research and development in Russian scientific-technological system 2007-2013 evolution priority directions".

II. SPECIFIC TASKS AND OBJECTIVES OF THE RESEARCH

A. Tasks

Main subtasks in my research include the following steps:

- Identify kinds of situations to be predicted

On this stage of the research only one kind of situation is considered: accident due to technical failure

- Generate data sets that are necessary for prediction
- Build a predictive model based on historical facts of

railway situations using special data mining software IBM SPSS Modeler. The accuracy of predictive algorithm must be not less than 75%.

- Identify type of parallelism that is typical of the task and choose appropriate paralleling method. It is also supposed that on this stage various paralleling methods will be analyzed in terms of application for railway situation analysis.
- Implement parallel predictive algorithm (parallel algorithm development using special software IBM InfoSphere Streams, testing, debugging).

Preprocessing data preparation steps, including refinement, cleaning, aggregation and transformation are beyond the scope of the paper.

B. Objectives

The result of the research is expected to be a parallel algorithm targeted to analyze situations on the railways particularly:

- Accident due technical failure

III. TYPES OF PARALLELISM

In different kinds of tasks the following types of parallelism usually occur [6]:

- data parallelism

This type of parallelism is typical of tasks that include the repeated execution of the same algorithm with different input data. Such calculations can obviously be done in parallel. If the problem has a parallel data, parallel program should be organized as a set of identical programs, each of which runs on its own processor from the main program. Such a program is usually a coarse grained one. Paralleling method based on data parallelism is called data decomposition.

- functional parallelism

This kind of parallelism is based on different functional blocks in an application. It can be split into separate processing units, that communicate with a fixed number other units in such a way that the output of one part serves as the input of another part. Method of parallelization based on functional parallelism is called functional decomposition.

- geometric parallelism

It requires that the problem space should be divisible into sub-regions, within which local operations are performed. The difference between the geometric parallelism and data parallelism is that in the first one subtasks of processing in each of the subareas must be interconnected. Parallelization based on geometric parallelism is called domain decomposition method.

- algorithmic parallelism

Stands for a type parallelism, which is detected by identifying in the algorithm the fragments, which can be performed in parallel. Algorithmic parallelism rarely generates coarse-grained (large-block) parallel algorithms and programs. The paralleling method based on this type of parallelism is called algorithmic decomposition.

- pipelined parallelism

This type of parallelism is typical of task in which input data must go through several stages of processing. In this case it is natural to use the pipeline decomposition of a task.

- «disorderly» parallelism

Often occurs in classes of algorithms where the possible number of parallel branches and the computational complexity are a priori unknown and depend on a specific task.

IV. PARALLEL PREDICTIVE ALGORITHM DEVELOPMENT

A. The process of model training

All information (data about precedents) required for the model construction is going to be extracted from operational sources to a special file containing of a table with facts hereinafter referred to as full set. The rows in the table represent precedents, and the columns are attributes of each precedent. In the last column there are losses suffered as a result of each situation. Then, in order to build accurate losses- prediction model, two random samples from full set must be selected:

- training (learning) sample;
- control (testing) sample

Building the predictive model on the learning sample will result in a certain function F, mapping X (a set of attribute values for each precedent) to Y - the predicted value of possible losses. Class of the function depends on the learning technique. Some of such techniques are supposed to be considered during the analysis:

- Logistic regression
- Neural Networks
- QUEST
- Decision-tree
- C&R Tree

To assess the quality of the obtained model it must be run on the testing sample, then the predicted losses will be compared to the actual damage by calculating special metric ROC, which shows the accuracy of predicted values or average forecast error [2]. However there is still the risk that the obtained error may be strongly dependent on how the full sample has been split in learning and control ones. Therefore, the next step should be so called cross-validation. Cross validation is a statistical method of evaluating and comparing learning algorithms that assumes that full sample should be divided into subsets for training and validation randomly multiple times [3]. 10-fold cross validation is commonly used [4]. For each split must be calculated ROC. Then averaging will be used in order to estimate prediction error of each learning technique. The algorithm that comes out best (minimal) average ROC is considered as superior to the other one.

B. Parallelization in model training process

In the posed task parallelism occurs not only during real situations prediction but also on the model developing stage.

1. Predictive model development

In real practice the size of the full sample is usually too large

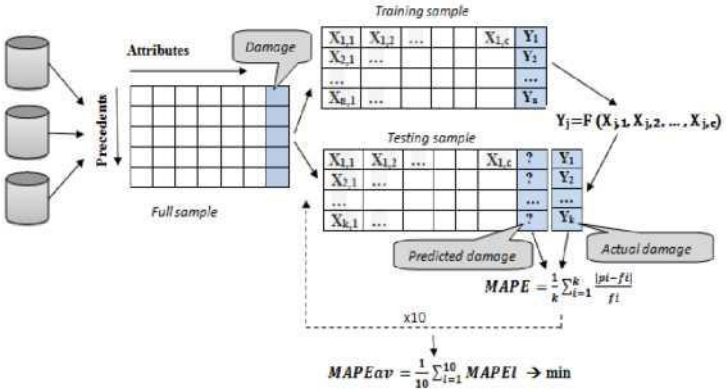

Fig. 1. Predictive model development process

therefore all the calculations can take a long time, what does not meet the target of timely response. The solution is to compute ROCs for each split in parallel. Thus *data parallelism* is typical of the posed task as it includes the repeated execution of the same algorithm with different input data. Algorithm parallelizing should be performed using data decomposition method. The process on the predictive model development will be organized as a set of identical processes, each of which runs on his slave processor from the main program that runs on the master processor.

C. Predictive model selection

The initial set of features is as follows:

(<Part of the railroad>,< the average weight of trains passed through the area over the past day>,< device 1 serviceability>,< device 2 serviceability, temperature>)

We will train the model and select the most important for prediction features simultaneously. Training of the models is supposed to be carried on in 4 steps progressively complicating the relationship among the data.

Step 1. Firstly it's necessary to generate the initial set of five features. Dependencies should not be trivial, so assume that information about the serviceability of the device 2 is not always reliable (the device fails): sometimes the device 2 serviceability indicates “true”, but really it has “false”. In 80% of cases with such a failure railway accidents happen. The rest of the accidents occurs due to unknown reasons. Failure of the device in 50% of cases are due to too low or high temperatures. The remaining data do not have any explicit dependencies. The results of training the model on such data set are presented in table below.

	Decision tree	Neural Networks	Logistic regression	QUEST	C&R Tree
Training accuracy	0.331	0.821	0.792	0.842	0.848
Testing accuracy	0.333	0.815	0.787	0.829	0.837

According to the results decision tree algorithm will be excluded of the further selection process, because it's results on

the very first sample are unsatisfactory.

Step 2. On this step it's necessary to complicate the task of modes learning by increasing the number of predictors. Select most important for the prediction features from those available. In almost any problem of forecasting the question arises: what signs to use, and what not. The problem of features selection often arises from the fact that at the stages of formulation of the problem and the generation of data is not yet clear what the signs are prediction-useless or duplicate each other. The challenge for feature selection in its exhaustive search nature. If the number of sign is n, the number of non-empty subsets of 2n -1. Direct enumeration of all subsets is impossible if n is the order of 20 even in the most modern machines. Attributes synthesis (also called features extraction) is the approach to reduce dimensionality. It consists of finding a transformation of the original feature space into a new space of substantially smaller dimension. One of the well-known and frequently used methods is the sequential addition of features - ADD method. This method assumes adding to an existing set of one additional feature, and the choice of ones, which leads to the greatest predictive error decrease (or predictive model accuracy increase) on a testing sample. It should be noted that the ADD method reduces the complexity of brute force but sometimes it tends to include a set of extra (noise) features. In our case, the duplication of features can be neglected, because we have just the model of the real situation so it's simplicity assumes the minimum number of data set. All features available are:

(<Part of the railroad>, <average mass of trains passing through particular sector per day>, <device 1 serviceability>, <device 2 serviceability>, <temperature>, <device 3 serviceability>, <ware of contact wire>, <DISC - sensor reading>, <maneuverable light signal>, <input light signal>, <occupation intensity per day>).

For best features selection, we take the predictive algorithm, which showed the best results in the first step –C&R tree.

After adding to the existing data set “device 3 serviceability” feature algorithm re-training came up with following results:

	C&R Tree
Training acuracy	0.848
Testing accuracy	0.845

Note that the accuracy of the model on the training set on average has not changed, but the accuracy of prediction on the testing sample increased. We conclude that this feature is not excess and leave it in the set of predictors. The next feature that we will check - the “wear of contact wire”. The result of the experiment is shown in the following table:

	C&R Tree
Testing accuracy	0.875
Training accuracy	0.874

The results indicate that this predictor is surely quite informative so leave it in the sample too.

After the sequential adding of the next 5 features the accuracy was practically unchanged, but after adding station battery indicator feature, the prediction accuracy was reduced, so this feature will not be considered in further process of algorithms training. Final sample get the following attributes:

(<Part of the railroad>, <average mass of trains passing through particular sector per day>, <device 1 serviceability>, <device 2 serviceability>, <temperature>, <device 3 serviceability>, <ware of contact wire>, <DISC - sensor reading>, <maneuverable light signal>, <battery voltage>, <input light signal>, <occupation intensity per day>).

The results of the re-training on the obtained sample are as follows:

	Neural Networks	Logistic regression	QUEST	C&R Tree
Training accuracy	0.862	0.790	0.849	0.875
Testing accuracy	0.857	0.786	0.848	0.875

Logistic regression came up with less accuracy than on the previous step. Since the following training stages are supposed to have more complex relationships this model will not be considered.

Step 3.Then complicate our relationships, and suppose that an accident occurs in only 50% of the device 2 failures, other emergencies do not have explicit dependencies.

	Neural networks	QUEST	C&R Tree
Training accuracy	0.884	0.892	0.928
Testing accuracy	0.886	0.890	0.926

Step 4. On the last step of models learning assume that 50% of cases when the device 2 fails, and the maneuverable light signal is "blue" (prohibitive), an accident occurs. The remaining data dependencies are not explicit.

	Neural networks	QUEST	C&R Tree
Training accuracy	0.932	0.949	0.874
Testing accuracy	0.928	0.947	0.849

As a result of 4-step models training on available data set, the algorithm showed the best result is QUEST. That's why It is selected for export in the Streams application as a scoring operator.

D. *Railroad situation analysis system parallelism*

*A*fter predictive model is selected and trained it should be translated into InfoSphere Streams application as a user-defined scoring operator for real-time processing of exceedingly large sets of raw data. This raw data is expected to come from various sources such as cameras, RFID sensors, GPS sensors, etc. and be assimilated by Streams. Then initial data will be filtered and divided into several sets each for specific kind of situations to be predicted. This is algorithmic parallelism therefore dividing into 3 parts is referred to as algorithmic decomposition. On this stage of research only one kind of emergency is considered. Since the supposed volume of raw data is enormous, even after dividing it into 3 parts we will still have the bottleneck problem as the size of data for the particular situation can still be too large to meet the target of timely response. In order to avoid this problem data decomposition need to be applied - data need to be split into smaller parts depending on the part of the railroad after that it will proceed to the scoring operators distributed among different processing nodes. Each kind of emergency of course requires it's own set of features, therefore there must be different predictive model (scoring operator) for each one. Thus we have that data parallelism is nested in algorithmic parallelism.

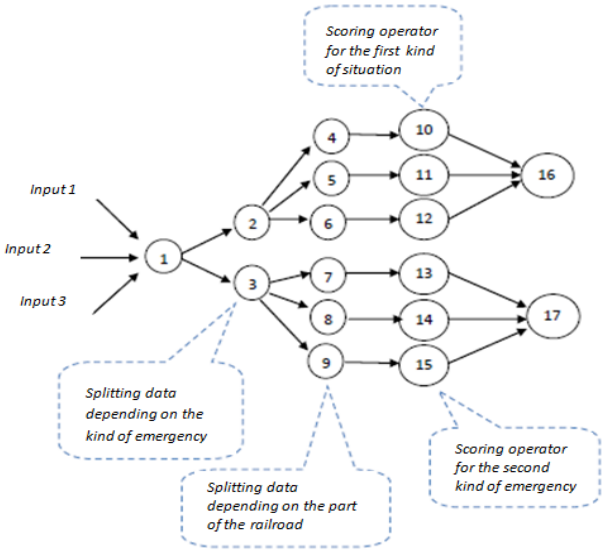

Fig. 2 Railroad situation analysis system parallelism

Parallelism pattern in Infosphere Streams application can be implemented using special operator *Split*. The Split Operator is used to split an input stream into various output streams. These output streams typically route each tuple based on Attribute characteristics.

V. DEVELOPMENT TOOLS

The predictive model will be elaborated with the use of the special data mining software IBM SPSS Modeler.

For parallelization will be used IBM InfoSphere Streams.IBM InfoSphere Streams is a software platform that intended for the development and execution of applications, processing data streams in parallel [5].

The platform provides:

- Streams Processing Language (SPL) consisted of a *programming language* interface that enables end- users operating on data streams and *runtime framework* that can execute the applications on a single or distributed set of hosts in parallel. Streams runtime implements its own message-based interprocess communication model [6] but enables to create applications without needing to understand the lower-level stream-specific operations.
- An integrated development environment (IDE) for Streams applications. Integrating SPSS Model Scoring in InfoSphere Streams makes possible leveraging the powerful predictive models in a real-time scoring environment.

CONCLUSION

According to the research by company DISCOVERY Research Group at the present time in Russia 83% of cargo transportation accounts for the railways. That's why railways security is a key priority for the Russian Government for many years ahead [7]. However current technologies are unable to support predictive detection and prevention of emergencies on railways due to extra large volumes of data and due to the lack of technical means for intellectual data mining in real time. The research conducted by IBM along with NRU-HSE within the state contract is targeted to facilitate russian intellectual rail transport system development.

REFERENCES

[1] Karpenko, A. *Parallel computing.* Retrieved January 29, 2012, from Bauman's MSTU educational base website: http://bigor.bmstu.ru

[2]Wikipedia, Free Encyclopedia, *Mean absolute percentage error.* Retrieved March 3, 2012, from Wikipedia website: http://en.wikipedia.org/wiki/Mean_absolute_percentage_error

[3] Liu, Ling & Ozsu, M. Tamer (Eds.) (2009). Cross Validation. *Encyclopedia of Database Systems*

[4] McLachlan, Geoffrey J. & Do, Kim-Anh & Ambroise, Christophe (2004). *Analyzing microarray gene expression data.* Wiley.

[5] Ballard, C. & Farrell, D. & Lee, M. & Stone P. & Thibault, S. & Tucker, S. (2010). *IBMInfosphere Streams: harnessing data in motion* (pp. 128-130). Texas: IBM International Technical Support Organization.

[6] IBM Corporation. *Transport options.* Retrieved March 25, 2012, from IBM InfoSphere Streams Information Center: http://publib.boulder.ibmcom/infocenter/streams/v2r0/index.j sp

[7] Avdoshin, S. & Gorbatovskiy, M. & Chernov, A. (2011). The concept of the of situational response and modern russian railways safety intellectual system. *Business-informatics*, 4(18), 8-15.