Cloud-based classification of text documents using the Gridgain platform

M. Sarnovsky*, T. Kacur*

* Department of cybernetics and artificial intelligence, Faculty of electrical engineering and informatics, Technical University in Kosice, Letňa 9/A, 042 00 Kosice, Slovak Republic
martin.sarnovsky@tuke.sk, tomas.kacur@gmail.com

Abstract—Motivation for the research effort presented in this paper is to use the cloud computing storage and computational capabilities for text mining tasks. Cloud computing is nowadays favored approach in area of data-analysis and related fields by providing data storage and computational capabilities as the services. Main aim of our research activities is to design and develop experimental cloud platform for text mining tasks. In this particular paper we describe the design and implementation of a distributed tree-based algorithm for text categorization purposes. We used our own implementation of decision tree classification algorithm and used Gridgain framework for its cloud implementation. Cloud also provides storage services for handling large data collections as well as increases computational effectiveness as the algorithm is implemented in distributed fashion. We describe the experiments we have performed on the private cloud using the two datasets and analyze the results.

I. INTRODUCTION AND RELATED WORK

Knowledge discovery in texts is a variation of knowledge discovery in databases field, and its main purpose is to find interesting patterns in data. It is a process of semi-automatic, non-trivial extraction of previously unknown, potentially useful and non-explicit information from large textual document collection. This process in general starts with text preprocessing phase, which is a transformation of the text to the appropriate internal representation that the mining algorithms can work with. One of the most common internal representations of the text document collections is vector space model [1]. Text mining is considered as a core process within the knowledge discovery in texts. It is an application of machine learning algorithms on the transformed data in order to find patterns in data. There are several types of text mining tasks, text categorizations is one of them.

Text categorization in general is the problem of assigning a text document into one or more topic categories or classes based on document’s content. It is a classification task, but in case of text documents it is usually different from traditional classification approaches (in data mining). Traditional approaches to classification problems usually consider only the unilabel classification, which means that each document in collection is assigned in one class. But in text categorization is likely, that data belongs to multiple classes, one document can be labeled by set of classes and this is the reason to explore multi-labeling problems. It is possible to use multi-label algorithms such as Naïve Bayes probabilistic classifier, or modify common classifiers to be able to use multi-label data. Most frequently used approach to deal with multi-label classification problem is to treat each category as a separate binary classification problem, which involves learning a number of different binary classifiers and use an output of these binary classifiers to determine the labels of a new example. In other words, each such problem answers the question, whether a sample should be assigned to a particular class or not. From the perspective of distributed computing it is ideal for parallelization, as buildings of particular binary classifiers are independent tasks.

Utilization of parallel and distributed computing in area of data mining and text mining was subject of numerous research activities and projects. Grid computing paradigm was used DiscoveryNet [2] and myGrid [3] projects, both provided a service-oriented computing model for knowledge discovery, allowing the user to connect to and use data analysis software as well as document collection that are made available online by third parties. The aim of these projects was to develop a unified real-time e-Science data and text mining infrastructure that leverages the distributed grid-based technologies and methods. Both projects have already developed complementary methods that enable the analysis and mining of information extracted from biomedical text data sources using Grid infrastructures, with myGrid developing methods based on linguistic analysis and DiscoveryNet developing methods based on data mining and statistical analysis. In area of biomedicine National Centre for Text Mining was also involved in research activities covering the Grid based text mining. Primary goal of this project was also focused to develop an infrastructure for text mining, a framework comprised of high-performance database systems, text and data mining tools, and parallel computing. Cloud-based data and text mining services are studied in [5, 6] and involve implementation of high-performance cloud using Hadoop and Sphere frameworks to analyze and mine large distributed datasets. Some of our text mining algorithms (classification and clustering tasks) have been also already used within the GridMiner project [4]. Grid infrastructure has already been utilized within the various text mining services. In [7] authors describes the grid services for distributed decision tree
induction, self-organizing maps for texts clustering are investigated in [8] and approach for formal concept analysis algorithm [9] was also presented in distributed environment [10].

The work presented in this article represents our activities in building of the coherent and complex system for text mining experimental purposes built upon the cloud infrastructure. Similarly to Grid, similar cloud computing infrastructure can offer computational effectiveness and data storage facilities for on-line analysis tool that comprises of various cloud services for knowledge discovery in texts and provides specific data and computing capacity. Our main motivation is to provide coherent system leveraging of cloud-based approach and providing simple user interface for users as well as administration and monitoring interface.

II. CLOUD COMPUTING AND OPEN SOURCE CLOUD FRAMEWORKS

Cloud computing paradigm is not a new one. Main idea behind using of the term “cloud” in various contexts can be traced to 1990s, when this term was used to describe networks. Cloud computing is also often compared to both grid and utility computing. Grid computing is a distributed computing paradigm that that allows from geographically distributed computational and memory resources creating a universal computing system with extreme performance and capacity [11]. Main idea of grid computing was to utilize distributed infrastructure and resources for computation intensive scientific tasks. Cloud computing leverages the main principles of this approach and extends it using virtualization principles and technologies NIST defines the cloud computing as a “model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [12].

There are four main service models of the cloud:

1. **Infrastructure as a Service (IaaS)** means on-demand providing of infrastructural resources. Examples of IaaS providers include Amazon EC2, GoGrid and Flexiscale.
2. **Platform as a Service (PaaS)** refers to provisioning of the platform resources, such as software development frameworks. In general it allows to customer deploy its own application on the cloud using the specific programming languages or development kits. Examples include Google App Engine, Microsoft Windows Azure.
3. **Software as a Service (SaaS)** represents the on-demand applications over the Internet. The applications are accessible via browsers and customer is not allowed to control the infrastructure. Examples of this concept are Salesforce.com, SAP Business ByDesign, or Google Docs.

There are various different types of clouds from the privacy perspective:

- **Public cloud** is a cloud on general public level. Service provider offers their resources as services on the Internet. Main disadvantage of using public clouds is data controlling and as well as security issues.
- **Private or internal clouds** are infrastructures designed to be used by specific organization. A private cloud may be built and managed by the organization itself or by external provider. Cloud of that type offers the highest level of security and data control.
- **Community cloud** can be viewed as a type of private cloud. The main difference is, that...
Gridgain is a Java based middleware for development of applications leveraging from cloud computing paradigm. It enables development of scalable data-intensive high performance distributed applications [13]. Main characteristic of the framework is, that Gridgain is independent from IaaS and PaaS, so applications developed using the framework can be deployed on various types of cloud. Gridgain provides native support for Java, Scala and Groovy (Groovy++) languages. Gridgain integrates two technologies [13]:

- Computational Grid
- In-Memory Data Grid

Computational grid is a technology handling distribution of process logic. It supports MapReduce type of processing which means splitting of original computation task into multiple sub-tasks executed in parallel fashion in distributed environment of any managed infrastructure and aggregating the results. In-memory data grid on the other hand presents the capability to parallelize the data storage by storing partitioned data in memory closer to the application. The goal of this approach is to provide high availability of the data by keeping it highly distributed an in-memory. We used Gridgain as a platform to create distributed text mining services, which are part of more complex data analysis tool. Algorithms are tested and evaluated on our private cloud infrastructure. One of the main advantages of Gridgain application platform is relatively small dependency on infrastructure. It means that designed and implemented approaches can be in very simple fashion modified to applications that are able to run on large public clouds such as Amazon EC2 or Windows Azure.

### III. DESIGN AND IMPLEMENTATION OF CLOUD-BASED DECISION TREE ALGORITHM

In the work reported in this paper, we used the decision trees algorithm based on the Quinlan’s C4.5 [14] implemented in JBOWL library. JBOWL - (Java Bag-of-Words Library) [15] is an original software system developed in Java to support information retrieval and text mining. The system is being developed as open source with the intention to provide an easy extensible, modular framework for pre-processing, indexing and further exploration of large text collections, as well as for creation and evaluation of supervised and unsupervised text mining models [16]. JBOWL supports the document preprocessing, building the text mining model and evaluation of the model. It provides a set of classes and interfaces that enable integration of various classifiers. JBOWL distinguishes between text mining algorithms (SVM, SOM, linear perceptron) and text mining models (rule based classifiers, classification trees, maps, etc.).

In this work we modified our implementation of decision tree algorithm using Gridgain Map Reduce technique.

The interface of the cloud-based distributed version of the service defines three main methods needed to build final model. `ClassificationExample` is main class, which defines the process of building of the classification model. Gridgain methods are implemented in order to use cloud infrastructure. All available nodes of the computing cloud are found (including the node running the code). Class implements main method `execute()`, that has two parameters. The first one is input class containing the method that divides the tasks into the separate sub-tasks as well as its distribution among particular computing nodes. The second parameter is `instances`, data read from indexed training set of documents that represents the dataset, which will be processed in selected task. Particular task (classification in this case) is then implemented as a `ClassificationTask` class. `ClassificationTask` is class used to sub-task assigning. The number of sub-tasks depends on number of available nodes in computing cloud. If available, number of sub-tasks (jobs) corresponds to number of nodes within the network and jobs are executed remotely on particular nodes (including the node running the code). If not, each node receives selected range of sub-tasks to process. The concrete amount of sub-tasks is conditioned by number of nodes, number of sub-tasks and particular sub-task complexity. When each sub-task represents the partial binary tree classifier, we can estimate its complexity by number of documents in training set assigned to the corresponding class. Indexes of particular range of binary classifiers (corresponding to particular classes) are used
in `ClassificationImpl`, the method containing the implementation the modification of JBOWL C4.5 algorithm. It contains `buildModel` method that is executed on computing node upon the assigned range of categories and builds particular binary tree model for each of them. Partial classification models are created on computing nodes. Final classification model is merged on the node running the code after the last partial model is finished.

IV. EXPERIMENTS AND RESULTS

The main goal of the experiments was to prove, that the distribution process using Map Reduce in Gridgain could reduce the time needed to build the classification model. Experiments were performed using two different datasets - Reuters-21578 and MEDLINE.

Reuters-2157811 (ModApte split) is collection of text documents that contains Reuters articles published in 1987. Collection is available publicly in SGML format and for our experimental purposes was transformed into the XML. ModApte version is split into the training and testing subsets. Both subsets contain documents in 90 different categories. Training set contains 7769 test documents and 28736 terms (lexical units).

MEDLINE12 is bibliographic database of the US National Library of Medicine. Dataset contains text documents archived since 1950 and comprises of more than 11 million articles and more than 4 800 indexed records. In this work we have used OHSUMED collection, which is MEDLINE subset, containing documents published in 1990. Documents in dataset describe the domain of heart diseases (5192 types) and contain 49580 documents in 90 categories using 171936 terms. MEDLINE dataset also has been transformed into the XML format. Fig. 5 and 6 depicts frequencies of occurrences of particular categories within the dataset (MEDLINE graph is depicted in logarithmical scale). The distribution shows, that there are 2 categories in Reuters collection that contains more than 1000 documents. Similarly, MEDLINE dataset contains several categories with significantly higher frequency of occurrence than other categories. Binary classifiers constructed upon these categories will have most significant impact on global time consumption of final model construction.

As a testbed we used our simple local private cloud infrastructure at Technical University in Kosice. The infrastructure serves for testing purposes and comprises of 12 networked computers with deployed Gridgain framework. Computers have Intel® Xeon® Processor W3550, 3.06 GHz CPU installed, 4 GB RAM memory and are connected via 1 GBit local network.

Experimental workflow for the distributed text categorization application starts with the process of data collection preprocessing. The process involves creation of vector space model of the dataset using the implementation of various preprocessing approaches. Those comprises of removal of stop words and computing of weighting scheme. Then the vectors are indexed and document – term matrix is created. This matrix represents the document collection model that classification algorithms work with.

Next phase represents the building of tree classifier. Algorithm searches for available nodes in the cloud and computes the number of categories within the data collection, which represents the number of partial models to be created in distributed fashion. Node computes the range of assigned categories for each available node.

- If number of categories is lower or equal to available computing nodes; each node will be assigned with one particular sub-task.
- If number of categories exceeds the number of available nodes; algorithm computes document frequency of particular categories and assigns the set of sub-task to particular nodes considering the complexity of sub-tasks.

Next process phase involved generation of partial classification models computed on cloud nodes. Partial models are then retrieved and merged on the node running the code into the final classification model.

We started the experiments using the sequential version of the service and then performed the first series of distributed service testing on different number of computing nodes. Series of experiments were repeated three times and values in graphs presents mean value of registered times achieved with particular number of nodes. Results show the speedup of classification model building using the distributed cloud version of tree algorithm. Examination of the workload of particular nodes and structure of datasets reveals that the speedup achieved by addition of computing nodes is heavily conditioned by occurrences of specific categories with significantly higher amount of documents. Our approach of sorting the categories by frequency of occurrence and assigning the sub-task according to the complexity of partial models helped us to achieve significant speedup.
V. CONCLUSION AND FUTURE WORK

This paper described the design and implementation of cloud-based decision tree algorithm for text classification purposes. Our approach was evaluated on our experimental testing private cloud with main objectives to prove the concept. Our testbed consists of Gridgain enabled computers used for computing and storage purposes. We performed several experiments on two different datasets and demonstrated the achieved speedup of cloud-based approach. As the testing was performed in rather smaller experimental environment, various aspects of the process of sub-tasks distribution can be further examined, also experiments performed on larger testbed.

On the other hand from perspective of knowledge discovery applications, cloud computing paradigm offers much interesting ideas. Our main motivation is to design and develop coherent and complex cloud-based information system for data-analysis tasks, which will provide cloud services for knowledge discovery in texts. Such a system will include support for various text mining algorithms, including the one described in this paper. The system will cover the preprocessing methods, clustering of text documents (implemented Growing Hierarchical Self-Organizing Maps algorithm), formal concept analysis approach, that will be also for information retrieval purposes.

ACKNOWLEDGMENT

The presented work is the result of the project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF (50%). This work was also supported by the Slovak VEGA Grants No. 1/1147/12. (50%).

REFERENCES
