

## Database Systems Journal BOARD

### Director

Prof. Ion Lungu, PhD, University of Economic Studies, Bucharest, Romania

### Editors-in-Chief

Prof. Adela Bara, PhD, University of Economic Studies, Bucharest, Romania

Prof. Marinela Mircea, PhD, University of Economic Studies, Bucharest, Romania

### Secretaries

Conf. Iuliana Botha, PhD, University of Economic Studies, Bucharest, Romania

Lect. Anda Velicanu, PhD, University of Economic Studies, Bucharest, Romania

### Editorial Board

Prof. Ioan Andone, PhD, A.I. Cuza University, Iasi, Romania

Prof. Anca Andreescu, PhD, University of Economic Studies, Bucharest, Romania

Prof. Emil Burtescu, PhD, University of Pitesti, Pitesti, Romania

Joshua Cooper, PhD, Hildebrand Technology Ltd., UK

Prof. Marian Dardala, PhD, University of Economic Studies, Bucharest, Romania

Prof. Dorel Dusmanescu, PhD, Petrol and Gas University, Ploiesti, Romania

Prof. Marin Fotache, PhD, A.I. Cuza University, Iasi, Romania

Dan Garlasu, PhD, Oracle Romania

Prof. Marius Guran, PhD, University Politehnica of Bucharest, Bucharest, Romania

Lect. Ticiano Costa Jordão, PhD-C, University of Pardubice, Pardubice, Czech Republic

Prof. Brijender Kahanwal, PhD, Galaxy Global Imperial Technical Campus, Ambala, India

Prof. Dimitri Konstantas, PhD, University of Geneva, Geneva, Switzerland

Prof. Hitesh Kumar Sharma, PhD, University of Petroleum and Energy Studies, India

Prof. Mihaela I. Muntean, PhD, West University, Timisoara, Romania

Prof. Stefan Nitchi, PhD, Babes-Bolyai University, Cluj-Napoca, Romania

Prof. Stefan Nithchi, PhD, Babes-Bolyai University, Cluj-Napoca, Romania

Prof. Corina Paraschiv, PhD, University of Paris Descartes, Paris, France

Davian Popescu, PhD, Milan, Italy

Prof. Gheorghe Sabau, PhD, University of Economic Studies, Bucharest, Romania

Prof. Nazaraf Shah, PhD, Coventry University, Coventry, UK

Prof. Ion Smeureanu, PhD, University of Economic Studies, Bucharest, Romania

Prof. Traian Surcel, PhD, University of Economic Studies, Bucharest, Romania

Prof. Ilie Tamas, PhD, University of Economic Studies, Bucharest, Romania

Silviu Teodoru, PhD, Oracle Romania

Prof. Dumitru Todoroi, PhD, Academy of Economic Studies, Chisinau, Republic of Moldova

Prof. Manole Velicanu, PhD, University of Economic Studies, Bucharest, Romania

Prof. Robert Wrembel, PhD, University of Technology, Poznan, Poland

### Contact

Calea Dorobanților, no. 15-17, room 2017, Bucharest, Romania

Web: <http://dbjournal.ro>

E-mail: [editordbjournal@gmail.com](mailto:editordbjournal@gmail.com); [editor@dbjournal.ro](mailto:editor@dbjournal.ro)

## CONTENTS

<b>Distributed algorithm to train neural networks using the Map Reduce paradigm.....</b>	<b>3</b>
Cristian Mihai BARCA, Claudiu Dan BARCA	
<b>Performance Enhancement using SQL Statement Tuning Approach.....</b>	<b>12</b>
Hitesh Kumar SHARMA, Mr. S.C. NELSON	
<b>Successful social games and their super-power: Big data analytics.....</b>	<b>22</b>
Ioana Roxana STIRCU	
<b>Explain Plan and SQL Trace the Two Approaches for RDBMS Tuning.....</b>	<b>31</b>
Hitesh Kumar Sharma, Mr. S.C. Nelson	
<b>Information security in digital trunking systems.....</b>	<b>40</b>
Claudiu Dan BARCA	

## Distributed algorithm to train neural networks using the Map Reduce paradigm

Cristian Mihai BARCA<sup>1</sup>, Claudiu Dan BARCA<sup>2</sup>,

<sup>1</sup> Electronics, Communications and Computers, University of Pitesti, Romania

<sup>2</sup>The Romanian-American University, Bucharest, Romania

*With rapid development of powerful computer systems during past decade, parallel and distributed processing becomes a significant resource for fast neural network training, even for real-time processing. Different parallel computing based methods have been proposed in recent years for the development of system performance. The two main methods are to distribute the patterns that are used for training – training set level parallelism, or to distribute the computation performed by the neural network – neural network level parallelism. In the present research work we have focused on the first method.*

**Keywords:** Artificial Neural Networks, Machine Learning, Map-Reduce Hadoop, Distributed System

### 1 Introduction

An Artificial Neural Network (ANN), usually called neural network (NN), is a mathematical or computational model that is inspired by the structure and/or functional aspects of biological neural networks. From a biological point of view, a neural network consists of an interconnected group of artificial neurons, and it processes information using a connection approach to computation. In most cases, an ANN is an adaptive system that changes its structure based on external or internal information flowing through the network during the learning phase. Specifically and technically saying, in a neural network model simple nodes (also called neurons or units), are connected together to form a directed graph hence the term neural network. While a neural network does not have to be structurally adaptive by itself, its practical use comes with algorithms designed to alter the weights of the connections in the network to produce the desired signal flow. Nowadays, neural networks are perceived as non-linear statistical data modeling tools; in the past the linearity of the neural networks was considered an inconvenience. They are usually used to

model complex relationships between inputs and outputs or to find patterns in data. Neural networks are applicable in virtually every situation in which a relationship between the predictor variables (independents, inputs) and predicted variables (dependents, outputs) exists, even when that relationship is very complex [1, 2]. In the artificial intelligence field, neural networks have been successfully applied to identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases), visualization and e-mail spam filtering [3]. For such neural networks practical applications, fast response to external events within short time are highly demanded and expected. However, the extensively used gradient descent based learning algorithms obviously cannot satisfy real-time learning needs in many applications, especially large scale application and when higher learning accuracy and generalization performance are required [4]. This drawback made artificial neural networks to lose their scientific

enthusiasm, but not for long time. As the hardware performances (multithreading, multicore, big clusters and grids) began to increase so the neural networks regained interest[5].

## **2. Challenges, Motivation and Aim**

When applied on a large scale, large scale learning system, in cases such as Netflix [6] or Google spam filters [7], machine learning mechanisms require massive amounts of computational resources. These resources are also needed when we talk about neural networks and only because of the training phase that passes through an iterative process with a very slow convergence rate. For daily problems where the training data is large, training times of the order of days and weeks are not unusual on serial computers. This has been the main problem for artificial neural network use in real-world applications and has also made its wider acceptability decrease. At first sight the graph structure of a neural network may reveal several degrees of parallelism within it, such as weight parallelism, node parallelism, and layer parallelism, but because of its strong dependencies between layers (a complete feedforward network connects weights from any node of one layer with any node of a next layer) it makes the whole problem very difficult to apply on a large scale distributed system (a network-based parallelism requires fine grained synchronization and communication – high overhead). Also, the size of the training set raises a challenge for the computational complexity of the neural network learning algorithm. This is more obvious in cases where training data is being continuously generated, and it must be absorbed quickly by the model, or else the model will become stale. One good example of such a situation is monitoring a person living at home [8, 9], with a set of sensors, in the context of AAL (Ambient Assisted Living [10]).

Assuming that we are building a classifier that announces the caregivers when the monitored person is in a dangerous situation, what is normal or not can change on a daily basis, especially when the person suffers a chronic illness with complications.

Recent research has shown the possibility of using distributed computing for machine learning, known as distributed learning. While in the past the whole attention has been drawn over the neural network training with multiprocessors or multicore computers using network level parallelism or pipeline [4, 11, 12, 13], now is pointed on using large scale distributed machines with training set parallelism [14, 15]. Basically, a training set parallelism means that the training set is split across multiple processing units. Each unit has a local copy of data-subset and collects change-values (deltas) for the given training patterns. After that, the data is updated using different aggregation methods. By far, for processing large distributed data Map-Reduce seems to be the right choice – it is highly scalable and also has a great potential in distributed learning [16]. Map-Reduce is a framework introduced by Google in 2004 for processing highly distributable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster or a grid. Computational processing can occur on data stored either in a filesystem (unstructured) or in a database (structured) [17]. It allows programmers to write functional- style (using two functions map and reduce) code that is automatically parallelized and scheduled in a distributed system. Google's Map Reduce implementation runs on a large commodity cluster (simple machines wired up in a local network) and is highly scalable: a typical Map-Reduce job processes many terabytes of data on thousands of machines. Software engineers find the system easy to use: hundreds of Map- Reduce programs have been implemented and upwards of one thousand Map-Reduce jobs are executed on Google's clusters every day [18]. However, there are also cases where researchers use in

their experiments Phoenix [19] or Hadoop [16] for the implementation of Map-Reduce and in others they developed custom frameworks using PThreads or other multicore methods [20] (multicore Map-Reduce frameworks).

### 3. Distributed Learning Algorithms

On top of the parallel architecture comes the machine learning algorithm. In order to use Map-Reduce, a machine learning algorithm must be mapped on both the map-stage and reduce-stage. One parallel programming method [20] that is easily applied to many different learning algorithms is to adapt them in a certain “summation form”, which allows easy parallelization on multicore and distributed computers. Algorithms that calculate sufficient statistics or gradients fit this model, and since these calculations may be batched, they are expressible as a sum over data points. Divide the dataset into as many pieces as there are cores, give each core its share of the data to sum the equations over, and aggregate the results at the end. This form of the algorithm is the “summation form” and allows adapting Map-Reduce paradigm to demonstrate this parallel speed up technique on a variety of learning algorithms. A brief description of backpropagation's (NN learning algorithm) Map-Reduce adaptation is presented by Cheng-Tao Chu & Co. [20]. The authors define a simple network structure (with one hidden layer) and use each mapper to propagate subsets of data through the networks (each mapper has its own network) and to compute the partial gradient for each of the weights in the network when the error is back propagated. After that, reducers sums the partial gradients from each mapper and do a batch gradient descent to update the weights of the network. In this manner backpropagation performs a batch learning phase, otherwise updating a set of values after each training example

creates a bottleneck for parallelization. The authors have implemented their adapted algorithm on top of a custom multicore Map-Reduce framework – their aim was to prove that the “summation form” of a learning algorithm allows an easy parallelization on multicore computers.

#### 3.1 Research Approach

During our research we covered many scientific papers with topics about using neural networks on large scale, read thoroughly and understood different learning algorithms and also dug for several comercial neural training engines' insights. However, it is clear by now that we are going to focus on the idea of shaping a batch neural network training algorithm into a Map-Reduce form – applying the same map phase (split the data between mappers and process the partial gradient of its local network's weights) and for sure calling a complete reduce stage, instead of combiners, with the appropriate weight update function, which depends on the algorithm (we acheive parallel batch-training and parallel update). The approach followed by this research is to:

- have a custom extensible/modular engine for neural network training (for feedforward networks) that will be easy to adapt with Hadoop's framework (we would have used Encog for this phase, but it is well encapsulated and a reverse engineering would have taken a long time)
- add backpropagation, batch backpropagation and resilient backpropagation support in the engine, error plotting mechanism and object based run configuration (set up some specific members of an object, eg. input path, maximum number of epochs to train, etc, and send it to the base class)
- adapt resilient backpropagation to a Map-Reduce form and do analysis of large data by applying multiple identical neural networks to learn several sub dataset
- use Hadoop's Map-Reduce implementation (but keeping the same interfaces, we want

both serial and distributed methods to be integrated under the same base application)

-find a different mechanism than HDFS (hadoop distributed filesystem) to pass the internal data of a neural network to a mapper/reducer (weights, errors, network structure),

e.g. Cassandra Database

-exploit at maximum capacity the reduce stage of the Map-Reduce paradigm (not just a single reducer) by performing a parallel update (almost 30% computational size) of the neural network  
 -find datasets for accuracy, consistency, convergence, speed and scalability tests; the datasets used in our experiments will be from publicly available projects, will cover many of neural network's practice usage (classification, detection, prediction, etc) and some of them will be large enough to demonstrate our framework's performances (obtain speedup with Map-Reduce).

### 3.2. Algorithms Research

As one would believe there is not a single algorithm for designing a neural network model, each of the available with its own advantages and disadvantages. The main

difference between them lies in formulation of how to alter the weights of the neurons and in the relations of the neurons to their environment [4]. Though, we restrained our view just on two of them that we encountered during our research: backpropagation and resilient backpropagation.

## 4. Resilient Backpropagation in Map-Reduce Form

*Resilient backpropagation (Rprop)* is a learning scheme that solves the convergence's issue of the batch backpropagation algorithm. The difference between these two methods resides in the update function of the weight, which has the following features [21]:

- eliminates the negative influence of the size of the gradient on the weight step
- the direction of the weight update is indicated by the sign of the gradient

We know from [21] the previous sections that the resilient backpropagation algorithm respects the summation form (batch learning), so now let us examine each stage of our rprop Map-Reduce form like we did in the word count example (Figure 1):

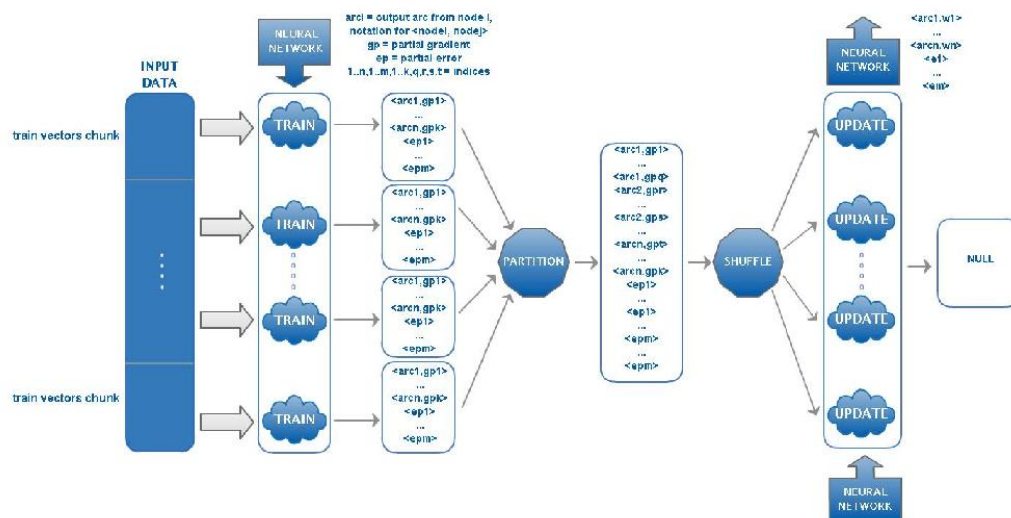


Fig. 1. Map-Reduce applied on resilient backpropagation algorithm

*Data split*

-the input data is split in equal size chunks of training vectors ('\n' separator)

*Map*

- each chunk is taken by a mapper and independently trained with the updated neural network pulled from the system – partial gradients are computed for every arc of the neural network and also the partial sum of the output neurons' squared error; each mapper outputs pairs of values  $\langle \text{arc } i, \text{gp} \rangle$  - where  $\text{arci}$  is an output arc from node  $i$  to the node  $j$  ( $\text{arci} = \langle n_i, n_j \rangle$ ),  $\text{gp}$  the partial gradient associated with arc  $i$  - and pairs of values  $\langle \text{epx} \rangle$  meaning the partial sum of the output neuron  $n_x$  squared errors

*Partition*

- when the map stage is finished the collection of output pairs  $\langle \text{arci}, \text{gp} \rangle$  is partitioned by the neuron  $n_i$  defined in  $\text{arci}$ ; we will have as many list of pairs  $\langle \text{arci}, \langle \text{gp}, \dots \text{gp} \rangle \rangle$  as different neurons  $n_i$  are in the collection; analogue for  $\langle \text{epx} \rangle$

*Shuffle*

- each list of  $\langle \text{arci}, \langle \text{gp}, \dots \text{gp} \rangle \rangle$  is sent to only one available reducer

*Reduce*

- sums (aggregates) the partial gradients from the given list to compute the total gradient of its correspondent arc; the final gradient is used to update the arc's weight to a new value that will be (along with the rest of the results from the other reducers) saved into the system in order to be used in the next training epoch.

## 5. The classes Driver – Mapper – Reducer

Basically, our three main classes (Driver – Mapper – Reducer) respect the same structure like those presented above, but it is obvious that they are more complex (figure2). We are now going to briefly explain them from a technical point of view in the following paragraphs:

*Driver*: Our driver reads the run parameters XML file uploaded by the user on the cluster (manually or using the GUI client) and then creates a network object structure (using the same RunParams class that fnn uses) based on the specified values from the file (input neurons, number of middle layers, etc). In the same time a shorter XML file with just the names of the experiment and network is copied to Hadoop's Distributed Cache [22]. Mappers need to know from where they are going to read the network structure and the updated weights' values during their phase (the network and experiment names are names of Cassandra column families). On start, the neural network is initialized with random values and is pushed into the NetStruct Cassandra's column family for further use in the next phases. One step before pushing the network we try to establish a connection to the database and also we check the presence of the key space we use. If there is no such keyspace a new one with the default name 'mrtsdb' is created automatically. After that, our driver starts the map-reduce cycle by launching computational jobs until it achieves the desired network's error or the maximum number of epochs (both values are taken from the XML file). At the end of the cycle we save the resulted neural network in Cassandra's NetSave column family.

*Mapper*

In the current framework's version our mapper takes the input from HDFS (in a future version it may also take the input directly from Cassandra Database). We use the default *TextInputFormat* to get chunks delimited by the carriage return symbol ('\n'). Each chunk represents an equal set of training vectors delimited by “#” symbol – the dataset is manually parsed prior to its upload on HDFS. Before reading the input data we use the file copied to the Distributed Cache to know the location of the network structure with the updated values from Cassandra (or the initial ones if is the first map-stage run). From an input chunk we extract the training vectors and we train them with the updated local network (using

fnn component in a lightweight version – without plotting features). Once an epoch has finished we collect the partial gradients and the sum of squared errors, values that we pass in a pair  $\langle \text{key}, \text{value} \rangle$  formation to the following stage.

#### Reducer

Our reducer receives the list  $\langle \text{key}, \langle \text{values} \dots \rangle \rangle$  partitioned and sorted by the shuffle and sort stage. A key means a node and the values represent serialized internal objects which contain information (weight value, gradient, and so on) about the arcs that start from the correspondent key- node. The only things that a reducer does are: aggregates data to obtain a global value (upon all the mappers) and updates the weights or the output node squared error, using the previous arc values and the global computed value (we query Cassandra to retrieve those data).

## 6. Workflow

The initial workflow was based on a simplified diagram modelled with a natural language notation (figure 3). After we went deeper into the implementation, we reshaped the workflow using proper technical notations (pair notations for the get and put operations implemented in cassdb component – figure 4). It helped us to define exactly the way we want our software framework to work, without knowing the hardware infrastructure we are going to experiment on. We also used this diagram to separate concerns between custom components, for example: client (mrts-client), database support (cassdb), import-export (mrts-io), driver – mapper – reducer (mrts), and so on.

In a real scenario our project works as follows:

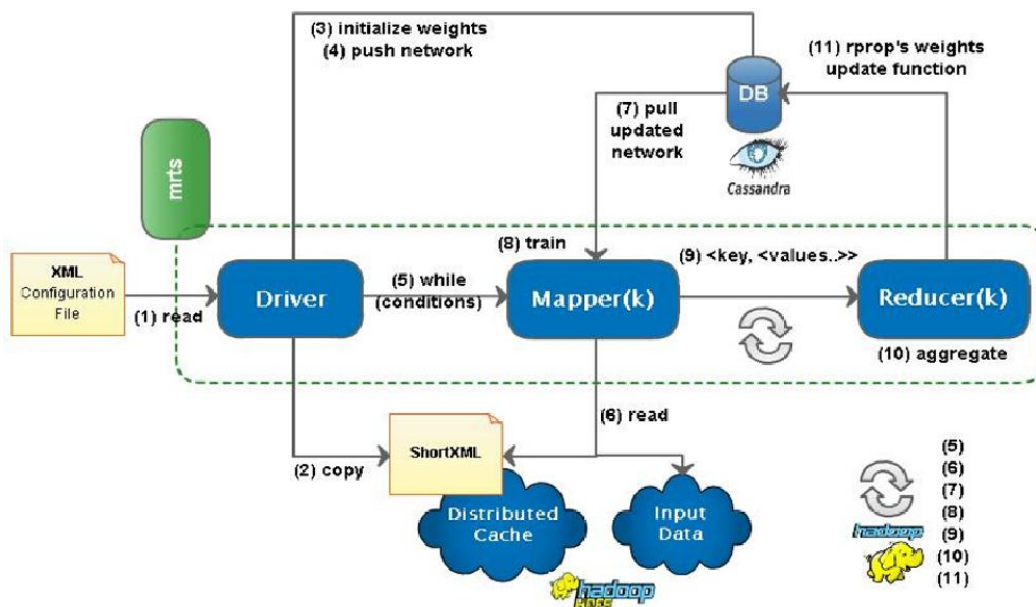


Fig. 2. Mrts component (internal workflow, driver-mapper-reduce cycle)



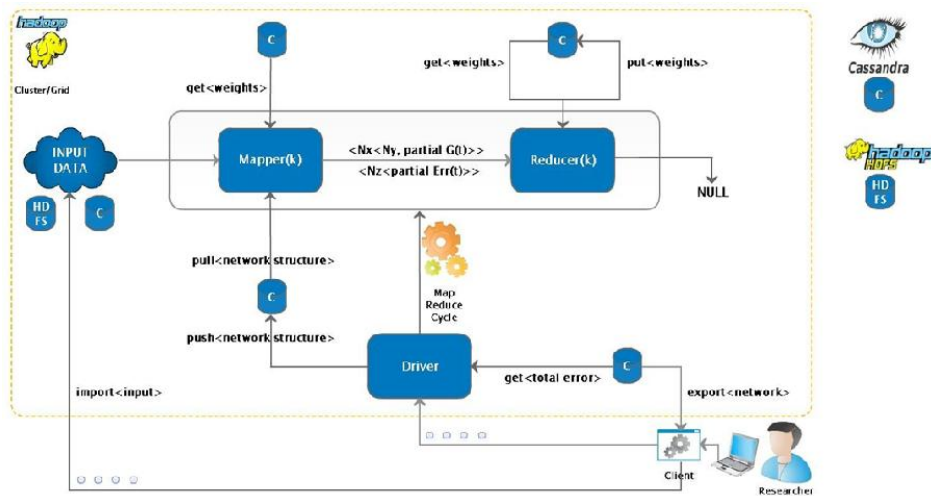


Fig. 3. Basic Workflow – without technical notations

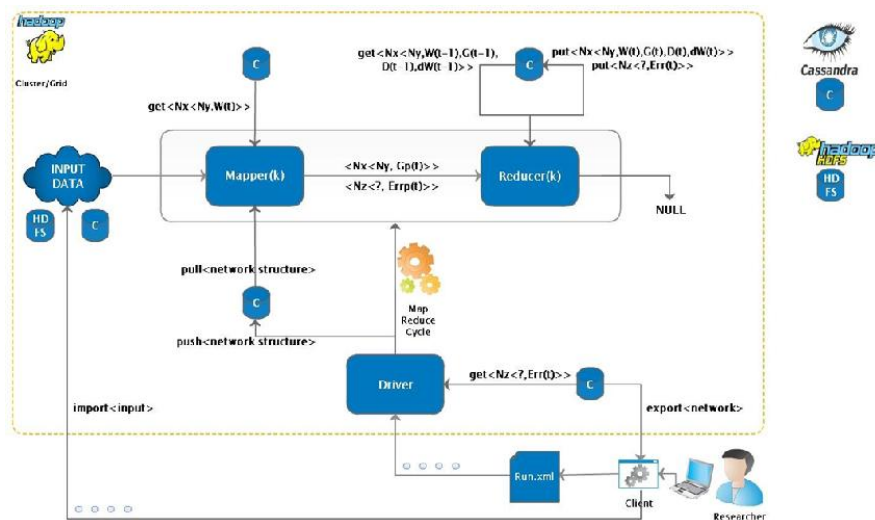


Fig. 4. Extended Workflow – with specific get/put notations (Symbol “?” represents a wildcard integer value bellow or equal to 0)

- A researcher sets an experiment and starts the training process. The experiment is defined using a GUI – we set the neural network structure (number of layers and neurons), training parameters (maximum number of epochs, neural network target error, etc.), the data input path and the data output path (for extracting and saving the network on local computer – in CSV [23] or a serialized format). All of the experiment’s configuration, including the

network’s structure, will be written to a .xml formatted file, which will be sent to the Hadoop cluster along with the start command.

- The request to start the training process is received by the Driver which will have the job to:
  - push the network structure in Cassandra
  - initialize the network’s weights and save them in Cassandra

- start the Map-Reduce cycle and run it while we have get the expected network error or the maximum number of epochs

- A Map-Reduce cycle, as we already know, has two components, each with its own set of tasks to perform:

Mapper:

- creates a local neural network using the structure and the weights' values from Cassandra:

- receives the training vectors chunk

- trains the current neural network over the given data

- outputs the partial gradient values and errors

Reducer:

- receives the partial gradient values or the partial output error and sums them up to compute the total values (total gradient or sum of squared errors)

- now that we have the final values, the weights and the output errors (all determined by the given list of pairs), the reducer updates and saves them in Cassandra

- the HDFS output of a reducer is set to NULL (we used Cassandra for output)

## Conclusions

We developed an extensible framework (client-server application) for training neural networks using the Map-Reduce paradigm.

Our serial implementation of resilient back propagation achieved similar accurate results compared with Encog.

Our convergence tests applied on the researched algorithms showed that the resilient backpropagation algorithm was the right choice for our purposes: batch training and good convergence rate.

## References

[1] Statsoft – Neural Network. Url: [www.statsoft.com/textbook/neural-networks/button=2](http://www.statsoft.com/textbook/neural-networks/button=2)

[2] Wikipedia – Artificial Neural Network.

Url:[en.wikipedia.org/wiki/Artificial\\_neural\\_network](http://en.wikipedia.org/wiki/Artificial_neural_network)

[3] Neural Networks Applications. Url: [www.peltarion.com/doc](http://www.peltarion.com/doc)

[4] Kiran Kumar Kaki. Parallelized Backpropagation Neural Network Algorithm using Distributed System. Master thesis, Thapar University, Patiala, January 2009.

[5] Lungu Ion, Bâra Adela, Căruțașu George, Pîrjan Alexandru, Oprea Simona-Vasilica, Prediction intelligent system in the field of renewable energies through neural networks, Journal of Economic Computation and Economic Cybernetics Studies and Research, Vol. 50, No. 1/2016, pp. 85-102, ISSN online 1842– 3264, ISSN print 0424 – 267X.

[6] Netflix. Url: [www.netflixprize.com](http://www.netflixprize.com)

[7] Google spam. Url: [www.google.com/mail/help/fightspam/spamexplained.html](http://www.google.com/mail/help/fightspam/spamexplained.html)

[8] Vikramaditya Jakkula and Diane J. Cook. Detecting Anomalous Sensor Events in Smart Home Data for Enhancing the Living Experience. Artificial Intelligence and Smarter Living — The Conquest of Complexity: Papers, AAAI Workshop, 2011.

[9] Vikramaditya R. Jakkula, Diane J. Cook, and Aaron S. Crandall. Temporal pattern discovery for anomaly detection in smart homes. Proceedings of the 3rd IET International Conference on Intelligent Environments (IE 07), Germany, 2007.

[10] Wikipedia - Ambient Assisted Living. Url: [en.wikipedia.org/wiki/Assisted\\_living](http://en.wikipedia.org/wiki/Assisted_living)

[11] Jigisha Gandhi, Shitanshu Perekh. Deployment of Neural Network on Multi-Core Architecture. International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 3, May 2012.

- [12] Markus Scholz, Prof.Dr.rer. nat. Peter F. Stadler. Development of an artificial neural network on a heterogeneous multicore architecture to predict a successful weight loss in obese individuals. Bachelor thesis, Leipzig, March 2008.
- [13] Honghoon Jang, Anjin Park, Keechul Jung. Neural Network Implementation using CUDA and OpenMP. Dicta, 2008.
- [14] Kritsanatt Boonkiatpong and Sukree Sinthupinyo. Applying Multiple Neural Networks on Large Scale Data. International Conference on Information and Electronics Engineering IPCSIT vol.6, IACSIT Press, Singapore, 2011.
- [15] Quoc V. Le and Co. Building High-level Features Using Large Scale Unsupervised Learning. Appearing in Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.
- [16] Dan Gillick, Arlo Faria, John DeNero. MapReduce: Distributed Computing for Machine Learning. 2006.
- [17] Wikipedia – Map-Reduce. Url: [en.wikipedia.org/wiki/MapReduce](http://en.wikipedia.org/wiki/MapReduce)
- [18] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Google Inc. OSDI, 2004.
- [19] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, Christos Kozyrakis. Evaluating MapReduce for Multi-core and Multiprocessor Systems. Computer Systems Laboratory, Stanford University, 2007.
- [20] Cheng-Tao Chu et al. Map-Reduce for machine learning on multicore. Neural Information Processing Systems (NIPS) Conference, 2007.
- [21] Martin Riedmiller. Rprop – Description and Implementation Details. Technical Report, January, 1994.
- [22] Hadoop Map-Reduce Tutorial. Url: [hadoop.apache.org/common/docs/r0.20.2/mapred\\_tutorial.html](http://hadoop.apache.org/common/docs/r0.20.2/mapred_tutorial.html)
- [23] OpenCSV. Url: [opencsv.sourceforge.net](http://opencsv.sourceforge.net)



**Cristian Mihai BÂRCĂ** graduated from Faculty of Automatic Control and Computer Science, University POLITEHNICA of Bucharest in 2012, and holds a master degree in Parallel and Distributed Computer Systems, Vrije Universiteit Amsterdam since 2014. His scientific fields of interest and expertise include database systems and Web Technologies



**Claudiu Dan BÂRCĂ** graduated from Faculty of Computer Science for Business Management, Romanian American University in 2007, and holds a master degree in Economic Informatics since 2008 and a PhD in the field of Engineering Sciences since 2013. He is an assistant lecturer within the Faculty of Computer Science for Business Management having nine years of teaching experience. He also has good research and publishing activity: he was a member of the research teams of international and national projects. His core competences are in software programming and connected areas.

## Performance Enhancement using SQL Statement Tuning Approach

Hitesh Kumar SHARMA<sup>1</sup>, Mr. S.C. NELSON<sup>2</sup>

University of Petroleum & Energy Studies, Dehardun, Uttarakhand, India  
[hksharma@ddn.upes.ac.in](mailto:hksharma@ddn.upes.ac.in), [cnelson@ddn.upes.ac.in](mailto:cnelson@ddn.upes.ac.in)

*Tuning your SQL statements may be one of the most important tasks you can do to improve the performance of your Oracle system. By tuning your SQL statements to be as efficient as possible, you use your system to its full potential. Some of the things you can do to improve the efficiency of your SQL statements may involve as little effort as rewriting the SQL to take advantage of some property of your database or perhaps even changing the structure of the database itself.*

### 1 Introduction

Tuning SQL really falls into two separate categories:

*Tuning an existing application.* This approach involves less flexibility in terms of changing the structure of the application and the database, but may provide performance improvements anyway.

*Designing a new application.* With a new application, you have the flexibility to design the application and perhaps even the database itself. With this approach, you can take advantage of indexes, clustering, and hashing.

This paper emphasis at both of these categories. The amount of changes you can do and the flexibility you have in changing the design of the application and database depends on your particular situation. The more flexibility you have in designing the database along with the application, the better your overall results will be. Tuning an Existing Application

Tuning an existing application can be easier in some respects and harder in others. It is easier in terms of determining the data access patterns and the specific problem areas because the application is already running and can be profiled very easily with SQL Trace and EXPLAIN PLAN. However, fixing the problems can be a challenge because you may not have the flexibility to do so. With an existing application, you may or may not have the flexibility to fix the problem. For example,

you may have system performance problems, but the system is still stable enough and performs well enough for the users to get their work done. It is hard to justify the downtime involved in reconfiguring the system when it is still in a somewhat functional state. In this type of situation, it is important to plan far in advance and take advantage of scheduled downtime to implement system enhancements and fix any current problems and anticipated additional growth in user activity. This may be done by changing Oracle or OS parameters or by adding more disk drives, memory, and so on. If you can afford to build in an additional 20 to 30 percent growth, you may save yourself some reconfiguration time down the road.

Over the years, I have found that you take advantage of additional capacity much faster than most people anticipate and plan for. I remember the old days of the PC industry, when it was inconceivable that anyone would ever need more than 64 megabytes of memory or even think of needing a gigabyte of disk space on a desktop PC. Today, high-end PC servers support more than a gigabyte of RAM and are getting close to supporting a terabyte of disk space.

### 2. Problem Analysis

To tackle the problem of tuning an existing Oracle application, I recommend using

some kind of methodology. Here are the steps that will hopefully lead you to the problem's resolution:

1. **Analyse the system.** Decide if there is a problem; if there is one, what is it?
2. **Determine the problem.** What do you think is causing the problem? Why?
3. **Determine a solution and set goals.** Decide what you want to try and what you think will result from your changes. What is it you want to accomplish? Do you want more throughput? Faster response times? What?
4. **Test the solution.** Try it. See what happens.
5. **Analyse the results.** Did the solution meet the goals? If not, you may want to return to step 1, 2, or 3. By following a plan of this sort, you will find it much easier to resolve the problem—or determine if there is a problem. It is possible to spend a lot of time trying to solve a performance problem that may not even exist.

By looking at the performance of the system as it is and carefully examining its characteristics, you may be able to determine whether any action is necessary to fix the problem and how much effort is involved. We like to classify performance problems into one of three categories:

- **It's broken.** Performance is severely handicapped because of a configuration error or an incorrect parameter in the OS or RDBMS. Problems that fall into this category cause a performance swing of 50 percent or more. Problems that fall into this category are usually oversights during the system build, such as incorrectly building an index or forgetting to enable asynchronous I/O. This category of problem may indicate a more serious problem such as insufficient disk drives or memory.
- **It's not optimized.** Performance is slightly degraded because of a small miscalculation

in parameters or because system capacity is slightly exceeded. These types of problems are usually easily solved by fine-tuning the configurations.

- **Not a problem.** Don't forget that sometimes there isn't a problem; you are just at the capacity of the system. This "nonproblem" is easily solved by upgrading or adding more capacity. Not all problems can be solved with tuning.

In the first case, you may have to perform some drastic fix that probably involves rebuilding the database in some fashion. The solution may be as drastic as having to rebuild from scratch to add more disk drives; it may be as simple as adding an additional index.

In the second case, you may be able to tune the system with an OS or database configuration parameter. This is usually very easy to do and does not involve much risk. However, this solution does not usually result in a huge increase in performance.

The final case may involve adding an additional CPU (if you have an SMP or MPP machine) or upgrading to new hardware. Perhaps you will find that there really isn't a problem after all and that everyone is happy with the performance of the system. One thing to remember: You rarely see the end users if everything is going fine. It's only when there are performance problems that you hear from them.

### 3. Tuning the Application

When analysing the SQL statement, you should look for two things:

- **The SQL statement.** What does it do?
- **The effect of the SQL statement.** What is it doing it to? How does it fit into the big picture?

By looking at the SQL statement from these different angles, you may find a problem that you wouldn't find by just looking at it from one viewpoint.

For example, consider an application that does not use cached sequences to generate a primary key value. By itself, there is nothing wrong with this approach and the application is probably very efficient. But add a thousand users executing the same application and the problem is quite apparent: You have contention getting the value for the primary key value.

**The SQL Statement**

The best way to go about tuning the SQL statements of an existing application is to follow these few steps:

1. Familiarize yourself with the application. You should be familiar not only with the specific SQL statements but with the purpose of the application and what it does.
  2. Use the SQL Trace facility to analyse what the particular SQL statements are doing, what features of the RDBMS are being used, and how well those features are being used.
  3. Use EXPLAIN PLAN within SQL Trace to analyse how the optimizer is executing those SQL statements.
- Now take a look at some specifics of these steps.

**Familiarize Yourself with the Application**

Look not only at the SQL statements themselves but at the effect of those statements. Make a chart of the different SQL statements and determine the number of accesses each SQL statement makes to each table in the database. This chart can give you an effective visual idea of which tables are being accessed most frequently. Consider the example shown in Figure 1. This chart is a good quick reference for which SQL statements are affecting which tables. You can take this a step further and split the chart into different types of statements such as SELECTs, INSERTs, UPDATEs, DELETEs, and so on. Depending on your system and whether your application is shrink-wrapped or

developed in-house, this may be or may not be practical.

**Use SQL Trace to Analyse the SQL Statements**

By running SQL Trace on the SQL statements, you can gather much valuable information about the specific operation of each of the SQL statements. SQL Trace provides such valuable information as the following:

- Parse, execute, and fetch counts
- CPU and elapsed times
- Physical and logical reads
- Number of rows processed
- Library cache misses

You can use this information to determine which SQL statements are efficient and which ones are not. Look for the following indications of inefficient statements:

*SQL Trace Output Comments*

CPU and elapsed time. If the CPU or elapsed times are very high, this SQL statement is a good candidate for tuning. It doesn't make much sense to spend time tuning statements that don't use many resources. Executes Focus on SQL statements that are frequently executed. Don't spend time on SQL statements that are infrequently used.

SQL Statement #	Acct 1	Acct 2	Finance 1	Finance 2	History 1
1	R W	R W	R W	R W	W
2	R W	W		R W	W
3	R	R W		W	W
4	R	W	R	R	W
5	W	W		R	W

R = Read  
W = Write

**Fig. 1.** An example of an SQL statement-analysis chart.

Rows Processed. An SQL statement with a high number of rows processed may not be using an index effectively.

Library Cache. A large number of library cache misses may indicate a need to tune the shared pool or to change the SQL statement to take advantage of the shared SQL area. These clues may point you in the direction of the SQL statements that need to be tuned. You may have to alter these SQL statements to improve their efficiency. By using EXPLAIN PLAN, you may find addition areas that can be improved.

### Use EXPLAIN PLAN to Analyse Statement Execution

By running EXPLAIN PLAN as part of the SQL Trace report, you can get a better idea of how the SQL statement is actually being executed by Oracle. This information (and the information supplied by SQL Trace) helps you judge the efficiency of the SQL statement. Here is a list of some of the things to look for:

Are the table's indexes being used when they should be? If not, the statement may not be supplying the correct parameters in the WHERE clause.

- Are indexes being used when they should not be? In cases when you are selecting too much data, you may want to use the FULL hint to bypass the index.
- What is the cost of the SQL statement? This value is given in the *position* column of the first row of the table returned by EXPLAIN PLAN.
- What is the amount of overhead incurred from SELECT or UPDATE operations?
- Is the statement being parallelized? You may have to provide a hint to effectively take advantage of the Parallel Query option.

You should ask these questions and your own specific questions as you review the EXPLAIN PLAN output. By knowing what your application is supposed to do,

you may find important information about the efficiency of your statements by looking at this information.

### The Effect of the SQL Statement

In addition to looking at the SQL statement itself, you should also look at the effect of the SQL statements. In many cases, some detail that is unimportant by itself can become a problem when the application and SQL statements are run by hundreds or thousands of users at the same time. The effect of this can be a bottleneck on a specific table or even a specific row.

Here is a list of some things to look for when analysing the effect of the SQL statements:

- Is the SQL statement updating a specific row? If you update a specific row as a counter, it may cause a bottleneck.
- Where is the majority of the table activity? Is a specific table being heavily accessed? This could indicate an I/O bottleneck.
- Is there significant INSERT activity? Is it all to one table? This may indicate a contention problem on a certain table.
- How much activity is there? Can the system handle it? You may find that the SQL statements overload your particular system.

These are just a few of the things to consider when you are looking at the effects of the application on the system. I have seen cases in which an application, fully tested in the lab, moves into production and fails because it was tested with only one or two users. It is important to take into account the effect of hundreds or thousands of users simultaneously accessing the application.

### Review of How to Tune an Existing Application

Tuning an existing application can be quite a challenge. Determining whether the system is in need of optimization and figuring out how to do it is not always

easy. The task may be easier if you take a methodical approach like this one:

1. **Analyse the situation.** It may be that your system is not in need of adjustment. I do not recommend making any changes to a stable system unless you have to.
2. **Familiarize yourself with the application.** Look at the SQL statements as well as the overall application. Understand the purpose of the application.
3. **Make an analysis chart.** Look at the table accesses being generated by the application.
4. **Run SQL Trace with EXPLAIN PLAN.** See what the SQL statements are really doing. Choose the statements to focus on based on how often they are used and how many resources they use.
5. **Understand how these SQL statements affect the server system.** Look at Oracle and the OS. Determine which disks may be overused and where contention could occur when many users run the application. With an existing application, you may or may not have the flexibility to fix the problem. I do not recommend making any changes to an existing application or a functioning system unless some specific performance problems are affecting users or limiting the capacity of the system. Of course, if you have the flexibility to make changes and there is a need, any of the design and application changes described in the following section, “Designing a New Application,” also apply to an existing application.

### **Packages, Procedures, and Functions**

Another way to improve performance of your SQL statements is by using packages, procedures, and functions. *Packages* can help improve performance by storing together procedures and functions that are often used together. By storing these elements together, you can reduce the I/O required to bring them into memory from disk. Because these elements are often used together, they can also be loaded from disk together. By using *stored procedures*,

you benefit in several ways. Stored procedures allow you to reduce the amount of data sent across the network. The stored procedure requires fewer instructions to be sent to the server; in many cases, less data must be sent back to the client from the server.

A second benefit of a stored procedure is the increased chance that the SQL statement can be used by other processes. Because the SQL statement is defined and used by many processes, chances are good that the SQL statement will already be parsed in the shared SQL area and available to other users.

### **Optimization Approaches**

Oracle offers several options for optimization techniques. Among these are a cost-based approach and a rule-based approach. The approach you take depends both on your application and your data. In most cases, the cost-based approach is recommended because it determines an execution plan that is as good or better than the rule-based approach. The following sections look at the optimization approaches available from Oracle and when each approach is appropriate. Remember that you can use hints to specifically tell Oracle how you want the SQL statement to be executed. There are several ways you can indicate your preference, as described later in this chapter.

### **Rule-Based Approach**

The rule-based approach to Oracle optimization is straightforward and consistent. In the rule-based approach, the execution plan is derived by examining the available paths and ranking them against a list of predetermined values for these paths (see Figure 2).



Rank	Access Path
1	Single row by ROWID
2	Single row by cluster join
3	Single row by less index/cluster key with unique or primary key
4	Single row by unique or primary key
5	Cluster join
6	Full index/cluster key
7	Indexed cluster key
8	Composite key
9	Single-column indexes
10	Bounded range search on indexed column
11	Unbounded range search on indexed column
12	Sort-merge join
13	MAX or MIN on indexed column
14	ORDER BY on indexed column
15	Full-table scan

**Fig. 2.** The rule-based optimization rankings.

With the rule-based optimization approach, the optimizer determines the ways to execute the SQL statement. If there is more than one way to execute the SQL statement, the table in Figure is used to choose the approach with the lowest ranking.

### Cost-Based Approach

The cost-based approach to optimization uses existing knowledge of the database to choose the most efficient execution plan. During the normal operation of the RDBMS, statistics are gathered on the data distribution and storage characteristics. The optimizer uses this information to determine the most optimal execution plan. This optimization approach takes three steps:

1. The optimizer generates a set of possible execution plans, just as it does with the rule-based optimization approach.
2. The cost of each plan is determined based on statistics gathered about the database. This cost is based on CPU time and the I/O and memory necessary to execute the plan.
3. The optimizer compares the costs and chooses the execution plan with the lowest cost. The cost-based approach is usually preferred. In some cases, the rule-based approach may be more appropriate.

### Discrete Transactions

Discrete transactions are an optional way of processing SQL statements. They may help performance in certain situations. Discrete transactions can be used only in a limited set of circumstances, but if you can take advantage of them, they can help performance. Discrete transactions work with small, non-distributed transactions and improve performance by deferring the writing of redo information until the transaction has been committed. Discrete transactions cannot and should not be used for all transactions.

### How Do Discrete Transactions Work?

With discrete transactions, all changes made to data are deferred until the transaction has been committed. Even though redo information is saved, it is not written to the redo log until the transaction has been committed. Until the commit, the redo information is stored in another area of memory. When the transaction is committed, the redo information is written to the redo log and the changes are made to the data block. This arrangement causes the rollback segments to be bypassed. Because the changes are deferred until the commit, the undo information does not have to be saved. With normal transactions, the undo information must be saved in the rollback segments because the data blocks have already been changed. With discrete transactions, because the data blocks have not been changed until the commit, it is unnecessary to save that information. Discrete transactions should not be used on data accessed with long-running queries because those queries will not be able to access any undo information.

### When Should Discrete Transactions Be Used?

Discrete transactions should be used only under certain circumstances and with certain transaction types. Here are some guidelines for when to use discrete transactions:

- Discrete transactions should be used to modify only a few data blocks. The amount of data and the amount of time it takes to perform the transactions should be as small as possible.
- Discrete transactions should not be used on data that might be accessed by long-running queries. Those queries will not be able to access any undo information.
- Discrete transactions should not be used on any rows containing LONG data. Because of these restrictions, discrete transactions can be used only with a small subset of transactions. These transactions should be small and quick. If you can take advantage of discrete transactions, you should see a fairly good performance benefit.

Discrete transactions work with small, non-distributed transactions and can improve performance by deferring the writing of redo information until the transaction has been committed. Discrete transactions cannot and should not be used for all transactions. Before making modifications to your application design, determine whether discrete transactions can benefit your application. Determine whether your transactions fit the profile that can benefit from discrete transactions; also determine whether these changes will be significant enough to warrant the use of discrete transactions. If your system is under a heavy load of mostly small transactions, it may be worth investigating discrete transactions. Try implementing them and see what kind of benefit they return.

### Tuning Considerations

The data warehouse is tuned to allow several large processes to run at maximum throughput. There is usually no concern for response times. We may have to tune both Oracle and the server operating system. The following sections look first at Oracle and then at the server operating system.

### Server OS Tuning

We may have to tune the server OS to provide for a large number of processes (if we are using the Parallel Query option) and optimal I/O performance. Some of the things we may have to tune in the server OS are listed here; remember that some OSes may not require any tuning in these areas:

- **Memory.** Tune the system to reduce unnecessary memory usage so that Oracle can use as much of the system's memory as possible for the SGA and server processes. We may also need significant amounts of memory for sorts.
- **Memory enhancements.** Take advantage of 4M pages and ISM, if they are available. Both features can improve Oracle performance in a data warehouse environment.
- **I/O.** If necessary, tune I/O to allow for optimal performance and use of AIO.
- **Scheduler.** If possible, turn off preemptive scheduling and load balancing. In a data warehousing system, allowing a process to run to completion (that is, so that it is not pre-empted) is beneficial.
- **Cache affinity.** We may see some benefits from cache affinity in a data warehousing system because the processes tend to run somewhat longer. The server operating system is mainly a host on which Oracle does its job. Any work done by the operating system is essentially overhead for Oracle. By optimizing code paths and reducing OS overhead, we can enhance Oracle performance.

### Hardware Enhancements

For a data warehouse, there are several hardware enhancements that can improve performance. These hardware enhancements can be beneficial in the area of CPU, I/O, and network, as described in the following sections.

### CPU Enhancements

Enhancing the CPUs on our SMP or MPP system can provide instantaneous performance improvements, assuming that we are not I/O bound. The speed of CPUs is constantly being improved as are new and better cache designs. For SMP or MPP machines, the process of enhancing the CPU may be as simple as adding an additional CPU board. Before we purchase an additional processor of the same type and speed, however, consider upgrading to a faster processor. For example, upgrading from a 66 MHz processor to a 133 MHz processor may provide more benefit than purchasing an additional 66 MHz CPU with the added benefit that we now have the option of adding more 133 MHz CPUs. Because of the complexity and run time required by these queries, we can benefit from more and faster CPUs. SMP and MPP computers provide scalable CPU performance enhancements at a fraction of the cost of another computer. When upgrading our processors or adding additional processors, remember that our I/O and memory needs will probably increase along with the CPU performance. Be sure to budget for more memory and disk drives when we add processors.

#### **I/O Enhancements**

We can enhance I/O by adding disk drives or purchasing a hardware disk array. The data warehouse can benefit from the disk striping available in both hardware and software disk arrays.

Using Oracle data file striping can also help the performance of our data warehouse.

If our system performs only one query at a time and we are not taking advantage of the Oracle Parallel Query option, we may not see a benefit from a hardware or software disk array. In this specific case, we do not recommend OS or hardware striping; we should use traditional Oracle striping. Because we are executing only one query at a time without using the Parallel Query option, the I/Os to the data files are purely sequential on the table

scans. This scenario is somewhat rare; any variance from “pure table scans” results in degraded performance. Hardware and software disk arrays have the added benefit of optional fault tolerance. We should first choose the correct fault tolerance for our needs and then make sure that we have sufficient I/O capabilities to achieve the required performance level. If we use fault tolerance, we will most likely have to increase the number of disk drives in our system. Another benefit of hardware disk arrays is caching. Most disk arrays on the market today offer some type of write or read/write cache on the controller. The effect of this cache is to improve the speed of writing to the disk; the cache also masks the overhead associated with fault tolerance. If our queries often perform table scans, we may see good improved performance with disk controllers that take advantage of read-ahead features. Read-ahead occurs when the controller detects a sequential access and reads an entire track (or some other large amount of data) and caches the additional data in anticipation of a request from the OS. Unlike an OLTP system in which this is just wasted overhead, in the data warehouse where we are performing DSS queries, it is likely that we will need that data soon; if we do, it will be available very quickly. Enhancements to the I/O subsystem almost always help in a data warehouse environment because large amounts of data are accessed. Be sure that we have a sufficient number of disk drives, properly configured. An I/O bottleneck is usually difficult to work around. As with all types of systems, a well-tuned application is very important.

#### **4. Conclusion**

Tuning your SQL statements is one of the most important tasks you can do to improve performance.

In fact, you should tune your SQL statements before tuning your RDBMS server. By tuning your SQL statements to be as efficient as possible, you can use

your system to its full potential. Some of the things you can do to improve the efficiency of your SQL statements may involve as little effort as rewriting the SQL to take advantage of a property of your database

## References

- [1] Hitesh Kumar Sharma, Aditya Shastri, Ranjit Biswas, "A Framework for Automated Database Tuning Using Dynamic SGA Parameters and Basic Operating System Utilities", Database Systems Journal", Academy of Economic Studies-Bucharest, Romania.
- [2] Hitesh Kumar Sharma, Sandeep Kumar, Sambhav Dubey, Pawan Gupta, "Auto-selection and management of dynamic SGA parameters in RDBMS", Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference.
- [3] Hitesh Kumar Sharma, Aditya Shastri, Ranjit Biswas, "SGA Dynamic Parameters: The Core Components of Automated Database Tuning", Database Systems Journal", Academy of Economic Studies-Bucharest, Romania.
- [4] S. Elnaffar, W. Powley, D. Benoit, and P. Martin, "Today's DBMSs: How Autonomic are They?", Proceedings of the 14<sup>th</sup> DEXA Workshop, Prague, 2003, pp. 651-654.
- [5] D. Menasec, Barbara, and R. Dodge, "Preserving Qos of E-Commerce Sites through Self-Tuning: A Performance Model Approach", Proceedings of 3<sup>rd</sup> ACM-EC Conference, Florida, 2001, pp.224-234.
- [6] D. G. Benoit, "Automated Diagnosis and Control of DBMS resources", EDBT Ph.D Workshop, Konstanz, 2000.
- [7] B. K. Debnath "SARD: A Statistical Approach for Ranking Database Tuning Parameters" 2007.
- [8] K. P. Brown, M. J. Carey, and M. Livny, "Goal-Oriented Buffer Management Revisited", Proceedings of ACM SIGMOD Conference, Montreal, 1996, pp. 353-364.
- [9] P. Martin, H. Y. Li, M. Zheng, K. Romanufa, and W. Poweley, "Dynamic Reconfiguration Algorithm: Dynamically Tuning Multiple Buffer Pools", Proceedings of 11<sup>th</sup> DEXA conference, London, 2002, pp.92-101.
- [10] P. Martin, W. Poweley, H. Y. Li, and K. Romanufa, "Managing Database Server Performance to Meet Qos Requirements in Electronic Commerce System", International Journal of Digital Libraries, Vol. 8, No. 1, 2002, pp. 316-324.
- [11] S. Duan, V. Thummala, S. Babu, "Tuning Database Configuration Parameters with iTuned", VLDB '09, August 24-28, 2009, Lyon, France.
- [12] H. K. Sharma, A. Shastri, R. Biswas "Architecture of Automated Database Tuning Using SGA Parameters" , Database Systems Journal vol. III, no. 1/2012.
- [13] A. G. Ganek and T. A. Corbi, "The Dawning of the Autonomic Computing Era", IBM Systems Journal, Vol. 42, No. 1, 2003, pp. 5-18.
- [14] H. K. Sharma, A. Shastri, R. Biswas "A Framework for Automated Database Tuning Using Dynamic SGA Parameters and Basic Operating System Utilities", Database Systems Journal vol. III, no. 4/2012.

**Dr. Hitesh Kumar Sharma:** Author is an Assistant Professor (Senior Scale) in University of Petroleum & Energy Studies, Dehradun. He has published 40+ research papers in International Journal and 10+ research papers in National Journals.

**Christalin Nelson. S:** Author is an Assistant Professor (Selection Grade) in University of Petroleum & Energy Studies, Dehradun. He has published 40+ research papers in International Journal and 12 research papers in National Journals. He is Head of Department of Analytics.

## Successful social games and their super-power: Big data analytics

Ioana Roxana STIRCU  
The Bucharest University of Economic Studies  
roxana.stircu@gmail.com

*This article is a short presentation of big data analysis and game analysis. The paper describes the case of social games, and observes the huge improvement that big data analysis has on social games and their success. It also contains a presentation of Pokémon GO, and its evolution on the market, from launch until today. A set of metrics and algorithms are proposed, that can be used to improve game features and monetization. In the last section, I apply a Naive Bayes classifier, using WEKA, on a set of data collected from social media networks, to predict how using a game that implies walking influences the amount of daily steps a player makes.*

**Keywords:** big data analysis, Naive Bayes classifier, WEKA, monetization, Pokémon GO case study, social networks, social games

### 1 Introduction

In the recent years, together with other industries that have experienced an exponential growth, the gaming industry has also evolved incredibly well. One of the main reasons, aside introduction and fructification of the social gaming on social networks, using new platforms (such as consoles and mobile cell phones), has been introduction of game analytics. There are many gaming companies on the market using game analytics in the most important steps of a gaming project. The most frequent cases are: define user behaviour, identify patterns in the process of acquisition and retention, test and optimize social media ads, calibrate player experience in game, improve game design in order to be suitable for every type of player, predict and optimize players' retention and monetization.

#### 1.1 Big Data and Analytics

Big data is a term generally used to denote a huge amount of data, of a large variety of types, which are processed and analysed to gain all the information and insights resulted from it. One of the defining characteristics of big data is the fact that, in order to get the value from data fast enough to make the corresponding decisions and

improvements, all the existing platforms and tools have to be updated and re-designed in order to meet the new business needs. The four V's of big data are the main properties and problems in the field ([1], [2]):

- volume – the needed space capacity is incredibly big, and is exponentially growing;
- variety – data is structured and unstructured, from a lot of different sources, like applications, websites, social networks, emails, news, etc. to video, audio, texts, logs;
- velocity – big amounts of very variate data can be processed in a long time; sometimes only the process of gathering, cleaning and formatting data from different sources can take too long, add the analysing time and the decisions are made too late;
- veracity – when exploring the universe of big data, errors and mistakes can be very hard to track and fix, so the process should be made very carefully.

In order to meet the requirements of big data, has been developed new solutions for storage, called NoSQL (not only SQL) ([3],

[4]). Document storage is a NoSQL method, and is made using some types of documents, such as XML, YAML, JSON, BSON. Another storage method is graphs database, where the graphs properties like nodes, edges and orientation are used to describe data and relations between data nodes. One of the widely used solutions is columnar-based storage, where data is represented based on columns (or lines), that can be grouped logically [5].

The standard solution for big data management is represented by Apache Hadoop, a software platform coded in Java, which runs on a cluster of servers. This properties explains the performance and efficiency of Hadoop, as it is scalable according to business needs, due to MapReduce ([6], [7]).

### 2.1 Gaming Industry

The gaming industry can be divided into three big categories, according to the type of games developed within that category. The traditional video and MMO games doesn't seem to be the most important category of games anymore, and is represented predominantly by male players, with payments made via cash and credit card. Social games have the advantage that are offered at smaller prices via social networks, and are also played on mobile devices. This category include approximately 54% female players, using mobile payments and PayPal accounts to make purchases, and are playable across multiple platforms and devices. The mobile games represents a fundamental change in gaming industry, with all new methods to redefine the landscape: with mobile gaming, now anyone can play games, anywhere they want, and having a huge list of games to choose from, as the majority of games had adopted freemium monetization.

### 2.2 Game Analytics

Some very big and successful gaming companies understood that game

analytics, and gaining insights from players' generated data, is a very powerful tool, which can ease their path to success in their project. Many companies have a lot of projects based entirely on game analytics, pattern recognition or predictions of the best mix to use in different business processes. This step forward also changed the project of making games, from the structure of the previous projects where a big teams worked at a game for a few years, and then released the disc/CD on the market, waiting for the sales numbers. The game has changed: now, all the smart gaming companies make brainstorming and prototypes for a few games then, using game analytics with targeting and player segmentation, analysis and prediction, they can decide which of the projects ideas are more successful, and develop only the selected projects. From this point, all the features added to the initial projected are tested, optimized and designed to meet the desired properties, according to patterns and insights gained during testing and prototyping process.

The trend of using data analysis to improve the game development, combined with the importance and usage of social factor and social networks in some types of games and, most important, the huge development of mobile cell phones in the last few years, has projected the game analytics in the direction of big data analysis. Mobiles casual games or MMORPGs with millions of players, with hundreds of missions, maps, puzzles and sessions, with dedicated players that access the app on a daily basis, with years of history and progress, all together represent the growing picture of big data in gaming.

For example: every day, *Pokémon GO* have over nine millions of daily active users only from the USA, who are logged into their preferred social game [8], continuously generating data, while social and mobile gaming is still a fraction of the \$80 billion video game market. The entire industry is based on data and insight. Because a big part of the revenue comes from a small percent of the initial players, is crucial to understand player preferences and behaviour when even

the smallest improve has a fundamental impact on monetization and revenue.

### 2.3 Monetization and KPIs of Gaming

The process of game monetization [9] represents the moment when a video game starts producing money for the company and for the people involved in the creation and development. Monetization should start long before this moment, and the monetization strategies and models should be created at the beginning of the project. This process had a great evolution in the past decades. In the initial model of monetization, in the 80-90's, the user simply bought the disc or the console from the store. The developers followed the sales and fans, and had no other information about the game's success chances before launching the disc. In the last few years the monetization changed radically, as people are increasingly logging into social gaming apps through mobile phones and Facebook. The new projects are developed having a monetization model in mind, and the developers launch a small test part of the game. Using test results and insight gained, the company develops the next features and calibrate their monetization methods to be successful in the final part of their project. Monetization can be made by the application, but also by events related, campaigns, products, and merchandise.

For the game monetization, also called game subscription, a lot of subscription models are developed by gaming companies, according to their game types and platforms. Most important models are *pay-to-play* (players have to pay a monthly subscription fee to be able to play the game), *free-to-play* (usually involves an upfront cost and no additional payments), and *freemium* (game access and game content is granted for free for all users, but offers the option to pay for additional access and content). This models of monetization are applied via retail purchase, in-game micro

transactions, digital download, indirect monetization (like product placement, banner advertisement or commercial breaks), and even the newest trend of monetization, called crowdfunding (funding independent gaming projects – the developers can raise enough money before the development process starts).

Zynga is credited for starting social gaming in 2009, when they introduced *Farmville* for Facebook. From this moment, users started spending real money to gain credits in the virtual world of social gaming in no time. Other projects that successfully combined social gaming and great monetization models are *Candy Crush*, *Clash of Clans* and *Clash Royale*, *Pokémon GO*.

In social games, the most used metrics in analysing players' activity, also known as KPIs (key performance indicators) are:

- *DAU* is the number of daily active users, *MAU* is the number of monthly active users, *DAU* divided by *MAU* ratio;
- *sessions* are determined every time the player opens the game;
- *retention* represents the number of players still retained in game after a number of days, most used types are *second day retention*, *seven days retention*, *twenty-eight days retention*;
- *engagement* can be defined as the amount of time spent playing a game;
- *K-factor* measures the growth and “going viral” property of a game, based on virality actions and their success;
- *ARPU* is the average revenue per paying user;
- *LTV* is the lifetime value, which can be calculated based on in-game purchases, but also other



monetization-related actions, such as influence on virality, shares.

### 3.1 Case Study: Pokémon GO

*Pokémon GO* success started last summer, in July 2016, right after its launch on USA market and a few other selected countries. In the first week of availability, the game became the most downloaded mobile application ever, surpassing *Candy Crush* and *Clash Royale*. On February 27, 2017, Niantic announced that the game has surpassed 650 million downloads and that players have walked from here to Pluto since launch [11]. The app is a location-based augmented reality game, developed by the studio called Niantic – in collaboration with Nintendo, both for iOS and Android devices.

The players use the mobile device' GPS to locate virtual creatures, called Pokémon (pocket monsters), who appear on the screen as if they were in the same real-world location as the player. Once located, the player has to walk to reach the selected Pokémon, and then he can capture, battle, and train the virtual creatures. The main goal is to catch as many monsters as you can, and to complete your Poke dex, as you level up and are able to catch more complex and powerful creatures (the dex can be completed by hatching Pokémon eggs, and a Pokémon can also evolve by catching more of the same type). This way you can become a Pokémon trainer, take on Gym battles and defend your Gym, for the prestige and ownership of Gyms. By visiting Poke stops, located in interesting places like museums, bars, parks or coffee shops, the player can stock up on Poke balls and other very interesting and helpful items.

### 3.2 Gameplay and Game Mechanics

After creating a new account, the player chooses his customized avatar. This avatar is then displayed on the game's map – generated and updated based on player's

current geo-location. The geo-map is updated with Poke stops, which can provide eggs, Poke balls, berries, potions or lure modules (used to attract rare Pokémon, and Poke Gyms, which are battle locations, displayed in places of interest (like parks, coffee shops, bars, public institutions). The main feature is that player's movement in real life makes his avatar progress on the game's map. Every time a player encounters a Pokémon, viewed in augmented reality mode or on a generic background, and generated using the camera and gyroscope – displayed as it were in real world, he can take screenshots or catch it directly with a Poke ball by flicking it from the bottom of the screen up toward the monster. This way the Pokémon is caught and added to the inventory, but the player also gets two type of in-game currencies, candies and stardust, that are used to evolve a Pokémon. Depending on the success of the action is calculated the capture rate, determined by the timing and the type of the Poke ball used.

The monetization model implemented in this project is free-to-play, but the game also supports in-app purchases for additional in-game items, where players can purchase additional Poke balls or other in-game items (like incense – attracts Pokémon to you as you move for 30 minutes, lure models – attracts Pokémon to a fixed location, lucky eggs – double XP points gained for 30 minutes).

### 3.3 Proposed Metrics and Data Structures

In order to define and optimize the monetization model, we have to first define a set of metrics that can be used in the case of Pokémon GO. The data system should contain at least the following elements:

- user and initial setup: unique user ID (numeric), account created date (date and time)
- game version (numeric/string) and GPS related data: current location (GPS coordinates - string), city

- (string), country (string), time zone (string)
- responses to pop-ups: allow access to location (true/false) and camera (true/false) to be able to play the game, allow notifications (true/false), accept terms of service (true/false)
- birth date (date)
- account type (string): sign-up with Google or just access the “Pokémon trainer club”
- avatar selection (current avatar configuration, may be string) and updates (new avatar configuration, updated string)
- nickname (string)
- actions in game: like avatar update, Pokémon catch, check out a Poke stop or visit a gym, change location, check out menu (Pokémon, items, Poke dex, Shop, Settings, Tips, etc) and actions done in sections of the menu, like change settings, check out information, buy a package from the shop, level up, and sessions.

To the data described above, generated from triggers implemented in game code, we can add external information, such as data imported from Facebook (posts, photos articles about Pokémon GO). If validated and successfully used, Facebook information can be very

important to test and measure virality and general feelings and trends about the game. Very important are the articles from the media related to the game, Google searches and all the emails and messages addressed to the development team via customer support or the social networks. All this data is very important, even though we cannot do a direct correspondence between data and players IDs in the database. Using the information gathered from this channels, the company can improve the current game build, and update it with fixes mentioned by users, or even implement new features that seem to be much needed; and all this based on players’ feedback and news related to the game.

For data storage, most probably various types of NoSql solutions could be used, like document storage, but it seems more suitable to use columnar based storage. Considering that fact that new data will be added continuously for every player, probably best choice is to process streams of data. This can be done by processing “time windows” of data, to get more velocity in the process, or use the similar approach of managing data at rest using Hadoop.

### 3.4 Architecture for Data Processing

After defining the data structure and a proposal of metrics that could be used for game analytics, we can define the process as seen in **Fig.1**. In the previous section, we defined the main sources as game data (defined by triggers implemented directly in game), and the data that can be imported from other external sources, such as Facebook, media and news.



**Fig.1.** Game analysis with Big data

After gathering data from the available sources, the process continues with Big data management. This part represents

the processing and preparation of data, and in the case of the metrics described above, this can be made by splitting some columns,

in order to make some information more accessible and clear. Another processing step can be adding new data calculated as the age of players, using their birth date known in game and the current date. The data discovery is made by exploring the available data and understand it better before the data preparation step. During this part the data is prepared to be used in analysis, which can help us make predictions of successful monetization mix. The last step of the process is visualization of the data and the results of predictions algorithms.

### 3.5 Data Analysis and Prediction Algorithms

In the data analysis process, considering the data structures described above, we can use various analysis tools and prediction algorithms to make the best monetization for the game. For players' segmentation, we can use clustering or classification algorithms, such as k-means algorithms or neural networks and decision trees, to make sure we address similar players with the corresponding methods.

For monetization, prediction algorithms can be used to optimize the monetization mix according to each of the players' segmentations described above, or to estimate future sales based on the previous purchases made in game. For example, based on evolution of the player in game, the number of sessions and the current number of in-game coins, we can predict using regression, when will the player remain without any coins and action accordingly. At that point, in-game store offers and a friction area could be introduced in order to make the player need to buy more items in order to pass that point.

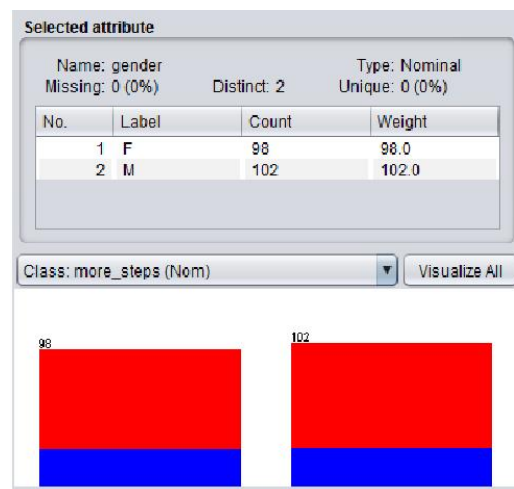
All these are small differences in implementation but can truly change the monetization of a product, and help the company to get the most profit out of its products and services.

### 4.1 Data Analysis using Pokemon Data

The data set used in this section was gathered from the social media networks, using articles and posts from the last months, created by players of Pokemon GO. The user specific data (such as age, gender, etc) was approximated using player's looks, in case this information wasn't available on their public profile.

The data set has 200 instances, with the following values and attributes, which were analysed during the pre-processing step, using WEKA:

- *player\_id*, given by the order of the list;
- *name*, representing their name used on this particular social network;
- *gender*, which was setted according to each player's profile; according to pre-processing steps, data is distributed as 98 females and 102 males (see **Fig.2**):



**Fig.2.** Gender distribution

- *age*, calculated using their birth year – when it was available on the social network, or approximated using their looks, and according to pre-processing step, the data is distributed as follows, in **Fig.3**, with users having an average age of 31.805 years:

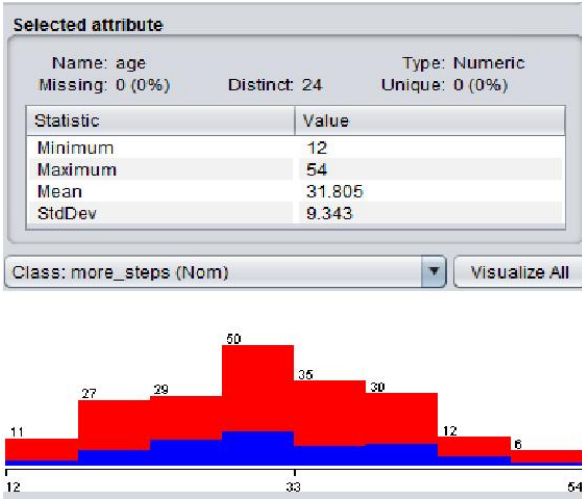


Fig.3. Age distribution

- *height*, approximated using player's looks;
- *weight*, approximated using player's looks;
- *BMI or the body mass index*, calculated field, using the values of player's height and weight, and the BMI formula:

$$BMI = \text{weight} / \text{height}^2,$$

where weight was transformed in kilograms and height was transformed in meters, and according to pre-processing steps, having the following distribution, as seen in Fig.4:

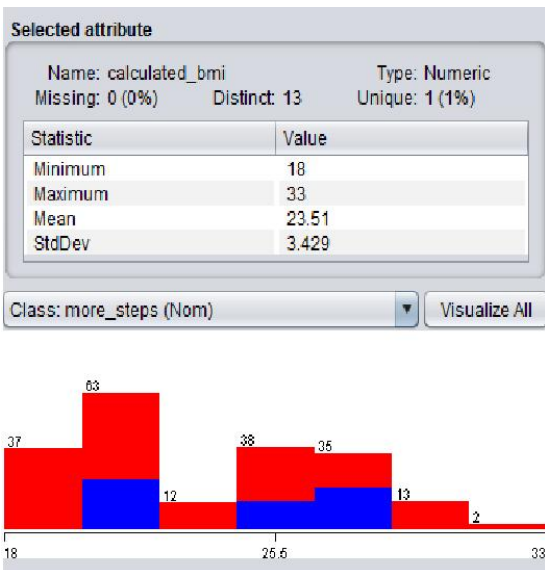


Fig.4. BMI distribution

- *days\_since\_playing*,
- *daily\_avg\_steps\_before*,

*daily\_avg\_steps\_after*, *more\_steps*, information specific to this case, of a game that implies to walk in order to proceed.

### 4.2 Naive Bayes Classifier

The Naive Bayes Classifier represents a supervised learning method, and a statistical method for solving classification problems. The classifier assumes an underlying probabilistic model and allows to capture uncertainty about the model by determining probabilities of the outcomes. This model can be used to solve diagnostic problems, and also predictive problems.

This classifier is called after Thomas Bayes (1702-1761), who proposed the Bayes Theorem, with the “naive” assumption of independence between every pair of features, as follows: given a class variable  $y$ , and a dependent feature vector  $x_1$  through  $x_n$ , Bayes’ then:

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)}$$

Using the naive independence assumption that

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

for all  $i$ , this relationship is simplified to

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)}$$

Since  $P(x_1, \dots, x_n)$  is constant given the input, we can use the following classification rule:

and can use Maximum A Posteriori estimation to estimate  $P(y)$  and  $P(x_i | y)$ , the former is then the relative frequency of class  $y$  in the training set.

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of  $P(x_i | y)$ .

In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the

necessary parameters. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution.

Naive Bayes classifiers are mostly used for

- text classification, where the Bayesian classification is used as a probabilistic learning method;
- spam filtering- the best known use, classifier is used to identify spam e-mail;
- recommendation system - filtering unseen information and predict whether a user would like a given resource;
- online applications - using a training set of examples which reflect nice, nasty or neutral sentiments.

### 4.3 Classification of social media data

After pre-processing, calculating and formatting the needed variables, a Naive Bayes classifier (scheme: `weka.classifiers.bayes.NaiveBayes`) is applied using 12 attributes, in WEKA on the 200 instances of data gathered from the social network, using as test option the cross-validation with 10 folds:

=== Run information ===

```
Scheme:
weka.classifiers.bayes.NaiveBayes
Relation: data_csv
Instances: 200
Attributes: 12
player_id
name
gender
age
height
weight
height^2
calculated_bmi
days_since_playing
daily_avg_steps_before
daily_avg_steps_after
more_steps
```

Test mode: 10-fold cross-validation

After running the model, the results are as follows:

=== Stratified cross-validation ===  
 === Summary ===

Correctly	Classified	Instances
175	87.5 %	
Incorrectly	Classified	Instances
25	12.5 %	
Kappa		statistic
0.6951		
Mean	absolute	error
0.1141		
Root	mean	squared
0.301		error
Relative	absolute	error
28.5286 %		
Root	relative	squared
67.3965 %		error
Total	Number	of Instances
200		

=== Detailed Accuracy By Class ===

Precision	Recall	TP Rate	FP Rate	MCC
ROC Area	PRC Area	Class	F-Measure	
		0.818	0.103	0.750
0.818	0.783		0.696	0.968
0.918	N			
		0.897	0.182	0.929
0.897	0.912		0.696	0.968
0.989	Y			
Weighted Avg.		0.875	0.160	0.879
0.875	0.877		0.696	0.968
0.969				

=== Confusion Matrix ===

```
a b <-- classified as
45 10 | a = N
 15 130 | b = Y
```

The correctly and incorrectly classified instances show the percentage of test instances that were correctly and incorrectly classified. The raw numbers are shown in the confusion matrix below, with a and b representing the class labels. The percentage of correctly classified instances is called accuracy or sample accuracy. Given this data set, the accuracy of classifier is about 87%. To get better results will have to try different classifiers or preprocess data even further. Kappa is a chance-corrected measure of agreement between the classifications and the true classes. It's calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum

possible agreement. A value greater than 0 means that the classifier is doing better than chance. The paper shall include an introduction on the current research in the papers field, original solutions, experimental results analysis, conclusions and references.

### References

- [1] Martin Hilbert, Internet: "Big Data for Development: A Review of Promises and Challenges. Development Policy Review.", martinhilbert.net, Retrieved 2015-10-07
- [2] Mark Beyer, Internet: "Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data", Gartner, Archived from the original on 10 July 2011, Retrieved 13 July 2011
- [3] C. Mohan, Internet: "History Repeats Itself: Sensible and Nonsensical Aspects of the NoSQL Hoopla", <http://openproceedings.org/2013/conf/edbt/Mohan13.pdf>, 2013
- [4] Internet: "Amazon Goes Back to the Future With 'NoSQL' Database", WIRED, <https://www.wired.com/2012/01/amazon-dynamodb/>, 2012-01-19, Retrieved 2017-03-06
- [5] Stephen Yen, Internet: "NoSQL is a Horseless Carriage" (PDF), NorthScale, Retrieved 2014-06-26
- [6] Jeffrey Dean, Sanjay Ghemawat, Internet: "MapReduce: Simplified Data Processing on Large Clusters", <https://static.googleusercontent.com/media/research.google.com/ro/archive/mapreduce-osdi04.pdf>
- [7] Jeff Bertolucci, Internet: "Hadoop: From Experiment To Leading Big Data Platform", Information Week, 2013, Retrieved on 14 November 2013
- [8] Kurt Wagner, "How many people are actually playing Pokémon Go? Here's our best guess so far", Internet: <http://www.recode.net/2016/7/13/12181614/pokemon-go-number-active-users>, 2016
- [9] Jeremy Liew, Internet: "29 business models for games", <http://lsvp.com/2008/07/02/29-business-models-for-games/>, July 2, 2008
- [10] Anders Drachen, Christian Thureau, Julian Togelius, Georgios N. Yannakakis, and Christian Bauckhage, Internet: "Game Data Mining", <http://julian.togelius.com/Drachen2013Game.pdf>, 2013
- [11] Samit Sarkar, Internet: "Pokémon Go hits 650 million downloads", <http://www.polygon.com/2017/2/27/14753570/pokemon-go-downloads-650-million>, 2017



**Ioana Roxana STIRCU**, graduated the Faculty of Mathematics and Informatics of the University of Bucharest in 2010, and gained the Master title in Cryptography and Codes Theory at the same university.



## Explain Plan and SQL Trace the Two Approaches for RDBMS Tuning

Hitesh Kumar SHARMA, Mr. S.C. NELSON

University of Petroleum & Energy Studies, Dehardun, Uttarakhand, India

[hksharma@ddn.upes.ac.in](mailto:hksharma@ddn.upes.ac.in), [cnelson@ddn.upes.ac.in](mailto:cnelson@ddn.upes.ac.in)

*Probably the best way to determine whether your SQL statements are properly optimized is by using the Oracle SQL Trace facility and the EXPLAIN PLAN command. You can use the SQL Trace facility and the Oracle program TKPROF, which is used to translate trace files, to trace production SQL statements, and gather statistics about those statements. You use SQL Trace to gather information into a trace file; the Oracle program TKPROF formats the trace information into useful, understandable data.*

### 1 Introduction

The EXPLAIN PLAN command is used to display the execution plan chosen by the Oracle optimizer for SELECT, UPDATE, INSERT, and DELETE statements. By analysing the execution plan the Oracle optimizer has chosen, and knowing your data and application, you should be able to determine whether the optimizer has chosen the correct execution plan for your application. After using EXPLAIN PLAN, you can rewrite your SQL statements to take better advantage of such things as indexes and hash keys. By analysing the output, you may be able to provide hints that the Oracle optimizer can use to take better advantage of your knowledge of your data. By using hints, you may be able to take better advantage of features such as the Oracle Parallel Query option. By the end of this chapter, you should be able to execute SQL statements using both SQL Trace and EXPLAIN PLAN and be able to analyse the output from these statements. You should also understand the value of registering applications for later use when tracking performance problems. These Oracle options can greatly improve the stability and performance of your system.

### 2. Tuning Considerations

The data warehouse is tuned to allow several large processes to run at maximum throughput. There is usually no concern for response times. We may have to tune both

Oracle and the server operating system. The following sections look first at Oracle and then at the server operating system.

### 3. Server OS Tuning

We may have to tune the server OS to provide for a large number of processes (if we are using the Parallel Query option) and optimal I/O performance. Some of the things we may have to tune in the server OS are listed here; remember that some OSes may not require any tuning in these areas:

- **Memory.** Tune the system to reduce unnecessary memory usage so that Oracle can use as much of the system's memory as possible for the SGA and server processes. We may also need significant amounts of memory for sorts.
- **Memory enhancements.** Take advantage of 4M pages and ISM, if they are available. Both features can improve Oracle performance in a data warehouse environment.
- **I/O.** If necessary, tune I/O to allow for optimal performance and use of AIO.
- **Scheduler.** If possible, turn off preemptive scheduling and load balancing. In a data warehousing system, allowing a process to run to completion (that is, so that it is not preempted) is beneficial.
- **Cache affinity.** We may see some benefits from cache affinity in a data warehousing system because the

processes tend to run somewhat longer. The server operating system is mainly a host on which Oracle does its job. Any work done by the operating system is essentially overhead for Oracle. By optimizing code paths and reducing OS overhead, we can enhance Oracle performance.

#### **4. Hardware Enhancements**

For a data warehouse, several hardware enhancements can improve performance. These hardware enhancements can be beneficial in the area of CPU, I/O, and network, as described in the following sections.

##### **4.1. CPU Enhancements**

Enhancing the CPUs on our SMP or MPP system can provide instantaneous performance improvements, assuming that we are not I/O bound. The speed of CPUs is constantly being improved as are new and better cache designs. For SMP or MPP machines, the process of enhancing the CPU may be as simple as adding an additional CPU board. Before we purchase an additional processor of the same type and speed, however, consider upgrading to a faster processor. For example, upgrading from a 66 MHz processor to a 133 MHz processor may provide more benefit than purchasing an additional 66 MHz CPU with the added benefit that we now have the option of adding more 133 MHz CPUs. Because of the complexity and run time required by these queries, we can benefit from more and faster CPUs. SMP and MPP computers provide scalable CPU performance enhancements at a fraction of the cost of another computer. When upgrading our processors or adding additional processors, remember that our I/O and memory needs will probably increase along with the CPU performance. Be sure to budget for more memory and disk drives when we add processors.

##### **4.2. I/O Enhancements**

We can enhance I/O by adding disk drives or purchasing a hardware disk array. The

data warehouse can benefit from the disk striping available in both hardware and software disk arrays.

Using Oracle data file striping can also help the performance of our data warehouse.

If our system performs only one query at a time and we are not taking advantage of the Oracle Parallel Query option, we may not see a benefit from a hardware or software disk array. In this specific case, we do not recommend OS or hardware striping; we should use traditional Oracle striping. Because we are executing only one query at a time without using the Parallel Query option, the I/Os to the data files are purely sequential on the table scans. This scenario is somewhat rare; any variance from “pure table scans” results in degraded performance. Hardware and software disk arrays have the added benefit of optional fault tolerance. We should first choose the correct fault tolerance for our needs and then make sure that we have sufficient I/O capabilities to achieve the required performance level. If we use fault tolerance, we will most likely have to increase the number of disk drives in our system. Another benefit of hardware disk arrays is caching. Most disk arrays on the market today offer some type of write or read/write cache on the controller. The effect of this cache is to improve the speed of writing to the disk; the cache also masks the overhead associated with fault tolerance. If our queries often perform table scans, we may see good improved performance with disk controllers that take advantage of read-ahead features. Read-ahead occurs when the controller detects a sequential access and reads an entire track (or some other large amount of data) and caches the additional data in anticipation of a request from the OS. Unlike an OLTP system in which this is just wasted overhead, in the data warehouse where we are performing DSS queries, it is likely that we will need that data soon; if we do, it will be available very quickly. Enhancements to the I/O subsystem almost



always help in a data warehouse environment because large amounts of data are accessed. Be sure that we have a sufficient number of disk drives, properly configured. An I/O bottleneck is usually difficult to work around. As with all types of systems, a well-tuned application is very important.

### 5. Fault Tolerance Consideration

Because the data warehouse contains so much data, we can take one of two approaches to dataprotection:

- **Protect everything.** Because there is so much data and so many disks in use, everything must be protected. The large number of disks in use increases the possibility of a disk failure. The massive amount of data increases the time needed for backup and recovery.
- **Conserve cost.** Because there are so many disks involved, it may be cost prohibitive to use RAID-1 or disk mirroring. When we mirror the disks, we double the number of disks.

In a data warehousing system, a good compromise is to use a fault tolerant method such as RAID-5 for the data files. We can be somewhat selective and use RAID-1 on volumes with heavy update activity and RAID-5 on volumes with more read activity. Remember that the performance penalty for RAID-5 is only on writing; we can achieve excellent read performance from RAID-5.

### 6. Hardware Considerations

When choosing hardware to use for a data warehousing system, consider these factors:

- **Low user load.** Not many concurrent processes/threads simultaneously access the system unless we take advantage of the -Parallel Query option.

- **High I/O load.** I/Os are concurrent and heavy, with mostly random I/O.
- **Huge amounts of data.** Data warehousing systems typically involve massive amounts of data. We must make sure that our system can support the high volumes of data we will be using.
- **Low network traffic during runtime, possibly high during load.** During the execution of typical decision support queries against our data warehouse, there is very little network activity. When data is being loaded or updated from other sources (possibly our OLTP systems), the network activity can be quite high.

If we can take advantage of the Oracle Parallel Query option, many different processes will use the machine at once; an SMP or MPP machine should scale very well. Because an SMP architecture uses CPUs based on the processes that are available to be run, if we always have a runnable process available for each CPU, we should see good scaling by adding additional processors. With an MPP machine, we see a similar effect but on a much larger scale. Because there is much random access to the disks, we can benefit from a disk array. We prefer hardware striping to OS striping because hardware striping does not incur any additional overhead for the operating system and does not take up valuable CPU cycles. If hardware striping is not available, OS striping is adequate. Network traffic may or may not be an issue to our data warehousing system. If necessary, segment the network or add faster network hardware. A network bottleneck is an easy problem to solve: simply add more and faster hardware.

### 7. SQL Trace

The SQL Trace facility and the Oracle program TKPROF are designed to give performance information about individual SQL statements. You can use this information to determine the characteristics of those statements. You can enable SQL Trace for a session or for an entire instance. Of course, because this facility gathers an abundance of information about SQL statement functionality and performance, SQL Trace has an effect on the performance of the system. If you use SQL Trace on a single session, the effect is fairly minimal, but if you use SQL Trace on an entire instance, you will see a substantial effect on the performance of the system. Avoid running SQL Trace on an entire instance for this reason.

### 7.1. SQL Trace Initialization

Before you run SQL Trace, you must make sure that certain Oracle initialization parameters are set:

#### *Parameter Description*

<i>Parameter</i>	<i>Description</i>
TIMED_STATISTICS	Setting TIMED_STATISTICS to TRUE enables SQL and some of the dynamic performance tables to collect timed statistics such as CPU and elapsed times. Enabling timed statistics incurs significant overhead because Oracle operations are now being timed; avoid this parameter except when necessary.
MAX_DUMP_FILE_SIZE	Specifies the maximum size of trace file dumps in blocks. Set this fairly low to avoid filling up the file system with trace files. If the SQL Trace output file is being truncated, increase this value.
USER_DUMP_DEST	This parameter specifies the destination for the trace files. The default destination is the same as for system dumps on your OS.

### 7.2. Controlling SQL Trace

You can enable the SQL Trace facility on a per-session basis or for the entire instance. The following sections explain how to enable and disable SQL Trace for both of these cases.

#### **Enable SQL Trace for a Session**

To enable SQL Trace for a session, use this Oracle command:

```
ALTER SESSION
SET SQL TRACE = TRUE;
```

Alternatively, you can use the Oracle procedure:

```
RDBMS_SESSION.SET_SQL_TRACE.
To enable SQL Trace for a session other
than your own, you can use the Oracle
procedure
RDBMS_SYSTEM.SET_SQL_TRACE_IN_
SESSION with the arguments SID,
Serial#, and TRUE. To determine
the values for SID and Serial#, use the
following SQL statements:
```

```
SQL> SELECT sid, serial#,
osuser
2 FROM v$session
3 WHERE osuser = 'Ed Whalen';
SID SERIAL# OSUSER
-----
7 4 Ed Whalen
```

To turn SQL Trace on for that session, use the Oracle stored procedure as follows:

```
SQL> EXECUTE
RDBMS_system.set_sql_trace_in_
_session(7, 4, TRUE);
PL/SQL procedure successfully completed.
```

#### **Disable SQL Trace for a Session**

To disable SQL Trace for a session, use this Oracle command:

```
ALTER SESSION
SET SQL TRACE = FALSE;
```

The SQL Trace facility is also disabled when your session disconnects from Oracle. To disable SQL Trace for a session other than your own, use the Oracle procedure RDBMS\_SYSTEM.SET\_SQL\_TRACE\_IN\_SESSION with the arguments SID, Serial#, and FALSE as shown here:

```
SQL> EXECUTE
RDBMS_system.set_sql_trace_in_
_session(7, 4, FALSE);
PL/SQL procedure successfully completed.
```

#### **Enable SQL Trace for an Instance**

To enable SQL Trace for your instance, set the Oracle initialization parameter `SQL_TRACE` to `TRUE`. Doing so enables SQL Trace for all users of this instance for the duration of the instance.

#### ***Disable SQL Trace for an Instance***

The SQL Trace facility cannot be disabled for the entire instance without shutting down the Oracle instance and setting the Oracle initialization parameter `SQL_TRACE` to `FALSE`. Alternatively, you can remove the parameter because its default value is `FALSE`. When SQL Trace is enabled for the entire instance, it is still possible to disable it on a per session basis. You can disable SQL Trace on a per-session basis with the SQL statement shown in the preceding section.

### **7.3. SQL Trace Functionality**

Once SQL Trace is enabled, it gathers the following information:

- **Parse, execute, and fetch counts.** These counts can give you vital information about the efficiency of the SQL statements.
- **CPU and elapsed times.** This information can tell you which statements take the most time to execute.
- **Physical and logical reads.** This information can help you determine the effectiveness of the database buffer pool.
- **Number of rows processed.** This information can be used as an indication that more rows are being processed than you expected, thus indicating a problem.
- **Library cache misses.** This information can show you the effectiveness of the shared SQL area and how well you are reusing already parsed SQL statements.

SQL Trace puts this information into a trace file in an unreadable form. You then use the Oracle program `TKPROF` to format

the trace information into useful, understandable data.

### **7.4. Interpreting SQL Trace**

This section looks at some of the statistics available from SQL Trace and how to interpret them. For each SQL statement executed, SQL Trace provides the following information:

<i>Parameter</i>	<i>Description</i>
<code>count</code>	Number of times the OCI procedure was executed. (The OCI interface is the standard set of calls used to access the Oracle database.)
<code>cpu</code>	CPU time in seconds executing. This value is the amount of time Oracle user <sub>i</sub> to process the statement.
<code>elapsed</code>	Elapsed time in seconds executing. This value is equivalent to the user's response time.
<code>disk</code>	Number of physical reads of buffers from disk. This value tells you how many reads actually missed the buffer cache and had to go to physical disk.
<code>query</code>	Number of buffers gotten for consistent read. This value represents the number of buffers retrieved in consistent mode. Consistent mode guarantees consistent reads throughout the transaction; it is used for most queries.
<code>current</code>	Number of buffers gotten in current mode (usually for update). In current mode, the data blocks gotten reflect the value at that instant in time.
<code>rows</code>	Number of rows processed by the fetch or execute call. This value gives you an idea of how many instructions have been executed.

By looking at each of these parameters, you can get an idea of how your SQL statements are being processed and which statements are taking the most time. By analysing which statements are taking the longest, you may be able to find some inefficiencies you can correct. The SQL Trace facility was enabled in a session by using this Oracle command:

```
EXECUTE
RDBMS_system.set_sql_trace_in
_session(7, 3, TRUE);
```

In another session (with `SID = 7` and `Serial# = 3`), the following SQL statement was executed:

```
SELECT
SUBSTR(dogname,1,20)      "Dog
Name",
SUBSTR(description,1,20) "Breed",
SUBSTR(owner,1,20)      "Owner"
FROM
dogs, breeds
WHERE
dogs.breed = breeds.breed
```

```
ORDER BY
dogs.breed;
```

The SQL Trace facility was later disabled using this Oracle command from the first session:

```
EXECUTE
RDBMS_system.set_sql_trace_in
_session(7,3,FALSE);
```

Following the execution of the SQL statements, the trace file was translated by running TKPROF as follows:

```
tkproforclshad.trctrace.out
sys=no explain=ed/ed
```

In this syntax, the following are true:  
 orclshad.trc Trace file generated by SQL Trace trace.out Where I want the output to go sys=no Indicates that no SYS or recursive SQL statements should be printed explain=ed/ed Specifies that I also want to generate EXPLAIN PLAN output TKPROF generated the output file shown in Listing

### **The EXPLAIN PLAN Command**

The EXPLAIN PLAN command shows you the execution plan that the Oracle optimizer has chosen for your SQL statements. With this information, you can determine whether the Oracle optimizer has chosen the correct execution plan based on your knowledge of the data and the application. You can also use EXPLAIN PLAN to determine whether any additional optimization should be done to your database (for example, the addition of an index or the use of a cluster). The EXPLAIN PLAN command is used to display the execution plan chosen by the Oracle optimizer for SELECT, UPDATE, INSERT, and DELETE statements. After using EXPLAIN PLAN, you can rewrite your SQL statements and see whether the new SQL statement is better optimized than the original statement. By analysing the output, you may be able to provide

hints that the Oracle optimizer can use to take better advantage of the data.

### **EXPLAIN PLAN Initialization**

When you run SQL statements with the EXPLAIN PLAN command, the output of EXPLAIN PLAN is put into a table with the default name plan\_table. You must create this table before you can run EXPLAIN PLAN. The table can be created in one of two ways:

Using the UTLXPLAN.SQL script provided by Oracle.

Creating the plan\_table table by hand.

The plan\_table table is defined as follows:

```
SQL> describe plan_table
Name Null? Type
-----
STATEMENT_ID VARCHAR2(30)
TIMESTAMP DATE
REMARKS VARCHAR2(80)
OPERATION VARCHAR2(30)
OPTIONS VARCHAR2(30)
OBJECT_NODE VARCHAR2(128)
OBJECT_OWNER VARCHAR2(30)
OBJECT_NAME VARCHAR2(30)
OBJECT_INSTANCE NUMBER(38)
OBJECT_TYPE VARCHAR2(30)
OPTIMIZER VARCHAR2(255)
SEARCH_COLUMNS NUMBER(38)
ID NUMBER(38)
PARENT_ID NUMBER(38)
POSITION NUMBER(38)
OTHER LONG
```

You do not have to name the table plan\_table. You can direct EXPLAIN PLAN to use a table of another name if you want.

### **Invoking EXPLAIN PLAN**

Invoke the EXPLAIN PLAN command with the following Oracle command sequence:

```
EXPLAIN PLAN
```

```
SET STATEMENT_ID = 'Testing
EXPLAIN PLAN' INTO plan_table
FOR
SQL Statement;
```

STATEMENT\_ID should reflect the statement's function so that you can recognize it at a later time. The plan\_table parameter is the name of the table you created as described in the preceding section. If the INTO clause is omitted, the command defaults to the name plan\_table.

Here is an example of a completed command:

```
SQL> EXPLAIN PLAN
2 SET STATEMENT_ID = 'Testing
EXPLAIN PLAN'
3 INTO plan_table
4 FOR
5 SELECT
6 SUBSTR(dogname,1,20) "Dog
Name",
7 SUBSTR(breed_name,1,20)
"Breed",
8 SUBSTR(owner,1,20) "Owner"
9 FROM
10 dogs, breeds
11 WHERE
12 dogs.breed = breeds.breed
13 ORDER BY
14 dogs.breed;
Explained.
```

The results of the EXPLAIN PLAN are written into the table plan\_table. The following section explains how to retrieve the information in that table.

### ***Extracting EXPLAIN PLAN Results***

The output of EXPLAIN PLAN is written to the table specified in the EXPLAIN PLAN command (by default, to the table named plan\_table). You must extract this information in order to look at the results of EXPLAIN PLAN. The results can be displayed with a query such as this:

```
SELECT SUBSTR(LPAD('
',2*(LEVEL-
1))||operation,1,30)
```

```
||' ||SUBSTR(options,1,15)
||'
' ||SUBSTR(object_name,1,15)
||' ||SUBSTR(DECODE(id, 0,
'Cost = '||position),1,12)
"Statement Execution Plan",
SUBSTR(optimizer, 1, 10)
"Optimizer"
FROM
plan_table
START WITH
id = 0 AND statement_id =
'Testing EXPLAIN PLAN'
CONNECT BY PRIOR
id = parent_id
AND
statement_id = 'Testing
EXPLAIN PLAN';
```

This query results in the following output:

### **Statement Execution Plan Optimizer**

```
-----
-----SELECT STATEMENT Cost =
CHOOSE
MERGE JOIN
SORT JOIN
TABLE ACCESS FULL BREEDS
SORT JOIN
TABLE ACCESS FULL DOGS
6 rows selected.
```

If the optimizer had chosen a cost-based approach, the cost of the query would have been reflected in the first line of the optimization plan. Any features such as parallel query are also reflected here. With this information, you can tell whether your SQL statements take advantage of indexes, clusters, or hash clusters. If you use EXPLAIN PLAN, you can see precisely how your SQL statement is being executed and what effect any changes you make to the SQL statements have on the execution plan. If you change your SQL statements to take advantage of an index or a cluster, for example, you can see an immediate improvement. EXPLAIN PLAN output is ideal for pointing out your execution plan and may indicate that where you thought

you were taking advantage of an index, you actually were not.

## 8. Registering Applications

When you register an application, the name and the actions performed by that application are stored in the database to assist with debugging and performance tuning efforts. When an application is registered, its name and actions are recorded in the `V$SESSION` and `V$SQLAREA` views. This information can be used later to track problems. To register an application, use the following procedures, available in the `RDBMS_APPLICATION_INFO` package:

### *Procedure Description*

**SET\_MODULE** Used to set the name of the module currently being run.

**SET\_ACTION** Used to set the name of a certain action currently being performed.

**SET\_CLIENT\_INFO** Used to set up information for the client information field.

**READ\_MODULE** Reads the current values of the module and action fields for the current session.

**READ\_CLIENT\_INFO** Reads the current client information field for the currently running session.

By registering the application, you can track many different parameters. Some of the values available through `V$SQLAREA` are given here:

- Memory used
- Number of sorts
- Number of executions
- Number of loads
- Number of parse calls
- Number of disk reads
- Number of buffer gets
- Number of rows processed

These parameters can provide valuable information when you are trying to debug various modules within your application. The information is enhanced by the addition of actions, which can further identify sections of your application.

## 9. Conclusion

Determining whether your SQL statements are properly optimized can be as important as anything else you can do to tune your system. An improperly tuned SQL statement can nullify any work you have done to optimize the database system. A well-tuned server system that is handling hundreds or thousands of unnecessary SQL statements can be perceived to have poor performance when, in reality, there is just an abundance of excess work being done.

The Oracle SQL Trace facility and the `EXPLAIN PLAN` command can be valuable tools in debugging inefficient SQL code. The SQL Trace facility and its companion program `TKPROF` can give valuable information into such areas as these

## References

- [1] Hitesh Kumar Sharma, Aditya Shastri, Ranjit Biswas, "A Framework for Automated Database Tuning Using Dynamic SGA Parameters and Basic Operating System Utilities", *Database Systems Journal*, Academy of Economic Studies-Bucharest, Romania.
- [2] Hitesh Kumar Sharma, Sandeep Kumar, Sambhav Dubey, Pawan Gupta, "Auto-selection and management of dynamic SGA parameters in RDBMS", *Computing for Sustainable Global Development (INDIACom)*, 2015 2nd International Conference.
- [3] Hitesh Kumar Sharma, Aditya Shastri, Ranjit Biswas, "SGA Dynamic Parameters: The Core Components of Automated Database Tuning", *Database Systems Journal*, Academy of Economic Studies-Bucharest, Romania.
- [4] S. Elnaffar, W. Powley, D. Benoit, and P. Martin, "Today's DBMSs: How

- Autonomic are They?”, Proceedings of the 14<sup>th</sup> DEXA Workshop, Prague, 2003, pp. 651-654.
- [5] D. Menasec, Barbara, and R. Dodge, “Preserving Qos of E-Commerce Sites through Self-Tuning: A Performance Model Approach”, Proceedings of 3<sup>rd</sup> ACM-EC Conference, Florida, 2001, pp.224-234.
- [6] D. G. Benoit, “Automated Diagnosis and Control of DBMS resources”, EDBT Ph.D Workshop, Konstanz, 2000.
- [7] B. K. Debnath “SARD: A Statistical Approach for Ranking Database Tuning Parameters” 2007.
- [8] K. P. Brown, M. J. Carey, and M. Livny, “Goal-Oriented Buffer Management Revisited”, Proceedings of ACM SIGMOD Conference, Montreal, 1996, pp. 353-364.
- [9] P. Martin, H. Y. Li, M. Zheng, K. Romanufa, and W. Poweley, “Dynamic Reconfiguration Algorithm: Dynamically Tuning Multiple Buffer Pools”, Proceedings of 11<sup>th</sup> DEXA conference, London, 2002, pp.92-101.
- [10] P. Martin, W. Powely, H. Y. Li, and K. Romanufa, “Managing Database Server Performance to Meet Qos Requirements in Electronic Commerce System”, International Journal of Digital Libraries, Vol. 8, No. 1, 2002, pp. 316-324.
- [11] S. Duan, V. Thummala, S. Babu, “Tuning Database Configuration Parameters with iTuned”, *VLDB '09*, August 2428, 2009, Lyon, France.
- [12] H. K. Sharma, A. Shastri, R. Biswas “ Architecture of Automated Database Tuning Using SGA Parameters” , *Database Systems Journal* vol. III, no. 1/2012.
- [13] A. G. Ganek and T. A. Corbi, “The Dawning of the Autonomic Computing Era”, IBM Systems Journal, Vol. 42, No. 1, 2003, pp. 5-18.
- [14] H. K. Sharma, A. Shastri, R. Biswas “A Framework for Automated Database Tuning Using Dynamic SGA Parameters and Basic Operating System Utilities”, *Database Systems Journal* vol. III, no. 4/2012.
- [15] P. S. Yu, M. S. Chen, H. U. Heiss, S. H. Lee, “On Workload Characterization of Relational Database Environments”, IEEE Transactions on Software Engineering”, Vol. 18, No. 4, 1992, pp.347-355.
- [16] J. Seok Oh, S. Ho Lee, “ Resource Selection for Autonomic Database Tuning”, Korea Research Foundation .

**Dr. Hitesh Kumar Sharma:** Author is an Assistant Professor (Senior Scale) in University of Petroleum & Energy Studies, Dehradun. He has published 40+ research papers in International Journal and 10+ research papers in National Journals.

**Christalin Nelson. S:** Author is an Assistant Professor (Selection Grade) in University of Petroleum & Energy Studies, Dehradun. He has published 40+ research papers in International Journal and 12 research papers in National Journals. He is Head of Department of Analytics.

## Information security in digital trunking systems

Claudiu Dan BARCA

The Romanian-American University, Bucharest

*In Europe, most countries have implemented its digital trunking systems to meet the communication needs of various public safety organizations. The development of these systems will be closely correlated with the evolution of operational requirements and communications services needed by users. For all these digital trunking systems, information security has proven to be an essential aspect. In this paper we present some aspects of the security functions of the Tetra system because this system was imposed as an open standard ETSI and is used as a Schengen cooperation system.*

**Keywords:** digital trunking systems, Tetra, algorithm, encryption, interoperability

### 1 Introduction

Digital systems are trunking radio communications networks used for public safety services necessary for organizations such as police, fire, emergency medical services and critical infrastructure services (nuclear, energy, gas, etc.). Most security and civil protection organizations use dedicated systems based on telecommunication standards developed especially for public safety communications such as the ARCP-25 (P25) Tetrapol and Tetra Project, which use narrowband technology [1]. In Europe the frequency spectrum used for these services is 380-400 MHz.

The Tetra system has been developed since the beginning by ETSI (The European Institute for Telecommunication Standards). This has led to rapid adoption by manufacturers, operators, government users. The second generation of the system, TEDS (Tetra Enhanced Data Service), is currently being implemented. TEDS is also known as Tetra Release 2 or Tetra Broadband. This generation uses frequency bands of 25, 50, 100 and 150 kHz, and several modulation schemes:  $\pi / 4$ -DQPSK,  $\pi / 8$ -D8PSK, 4-QAM, 16-QAM and 64-QAM. Depending on the frequency band and the modulation scheme used, net transfer speeds can reach 500 kbit / s [2].

The system has various operating modes and allows radio communication on large areas with a single frequency. Tetra is the radio communication system with high spectral efficiency due to the use of TDMA (Time Division Multiple Access)

To meet the needs of the users were offered numerous voice and data services. Of these, the most important are considered to be: voice services, group call, emergency call, priority calls, Dynamic Group Assignment (DGNA), ambient listening, dispatcher authorized call, selection of the area, data services, data packets.

From the point of view of technology, ETSI develops the standard through modern reception techniques, dynamic power emission control techniques, optimization of protocols, and the development of encryption algorithms certified by authorized bodies.

### 2. Tetra digital trunking system architecture

The digital trunking system is based on standardized technical architectures consisting of functional structures (Figure 1) [3], namely:

- Tetra radio terminal - Trt - represents a fixed or mobile radio terminal in Tetra technology: each radio terminal is identified in the network by two parameters: SIM (Subscriber



Identity) and TEI (Tetra Equipment Identity)

- Tetra Network Management TNM - includes a set of Tetra network base stations, Tetra network interfaces with other networks (ISDN - Integrated Service Digital Network, PSTN), terminal databases
- Network management control unit - UTNM - represents the structure that performs the programming of the network functions
- External networks - fixed, mobile, analogue, digital communications networks
- Operators dispatchers - are the dispatchers / operators that coordinate the different sub-networks made on the system infrastructure - the police network, the fire network, the public utility operator network.

Depending on the complexity of networks Tetra digital trunking, manufacturers can provide standardized interfaces. The main standardized Tetra system interfaces are: [4]

- AI - air interface - via this interface is established the connection between a radio terminal and the base station or directly between two radio terminals (DMO - Direct Mode Operation establishes the direct connection between the two terminals without interacting with the base station)
- ENGI - External Network Gateway Interface- standardized interface that connects to other communication networks
- UTNMI - Interface that links TNM to UTNM
- PEI - makes the connection between the radio terminal and an external device
- ISI -the interface that makes possible the connection between Tetra networks of different infrastructures
- RCI-interface between TNM and dispatchers / operators

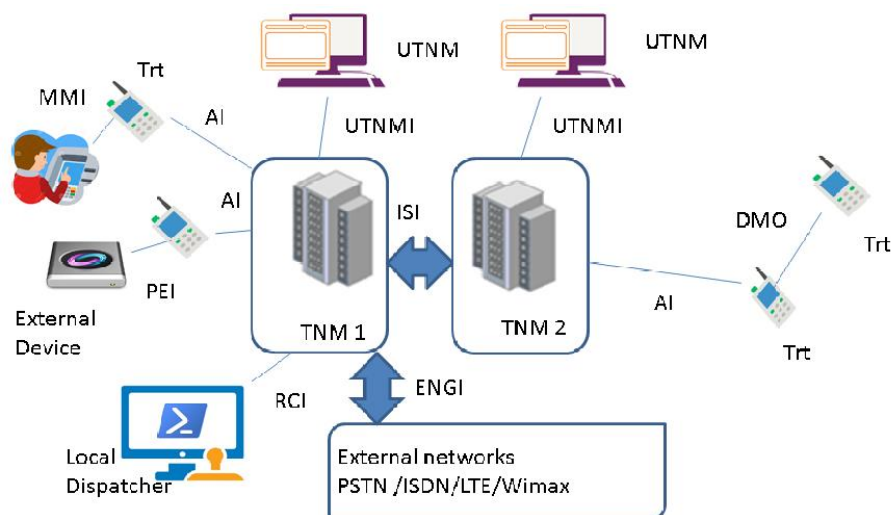


Fig. 1. Architecture of the Tetra digital trunking system

### 3.Security features in the Tetra digital trunk system

The digital trunk system Tetra, especially for governmental organizations, required that the information transmitted on this

system be secured. This system is designed and implemented to provide high-security services[5].

The system presents three security classes [6] presented in Table 1.

**Table 1.** Tetra Security Classes

	Authetification	Air Interface Encryption	Enable/Disable	End to End Encryption	OTAR
Security Class 1	optional	not use	optional	optional	not used
Security Class 2	optional	mandatory	optional	optional	optional
Security Class 3	mandatory	mandatory	optional	optional	mandatory

A radio terminal may have:

- 1, 2 or 3 class
- classes 1 and 2
- classes 1 and 3
- classes 2 and 3
- classes 1,2 and 3

A radio cell can have:

- 1, 2 or 3 class
- classes 1 and 2
- classes 1 and 3

The Tetra standard specifies security mechanisms for the protection of various protocols, interfaces, and applications [7].

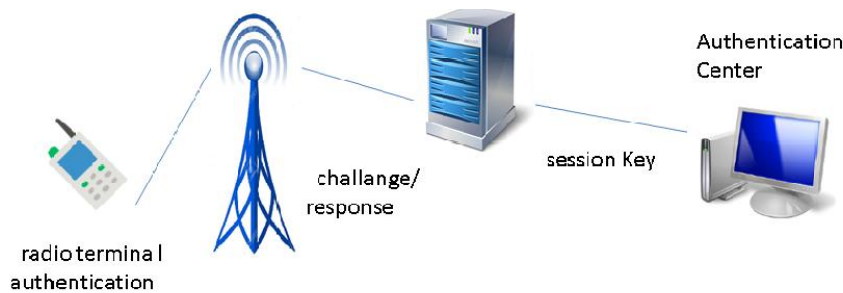
In general, these security mechanisms are implemented in the functional blocks of the communication network.

The main system security mechanisms [7] are

- authetification
- air interface encryption
- enable /disable
- end to end encryption

#### Authetification

Authetification means recognizing the radio terminal in the system by an authentication center to allow it to communicate with the network. (The identity of the transmitter is verified by the receiver) (Figure 2).

**Fig. 2.** Authentication process

The two parameters by which a radio terminal is identified are: SIM (Subscriber Identity) and TEI (Tetra Equipment Identity).

In the Tetra Authentication Procedure, two main entities appear - Authentication Center and Authentication Keys.

Generally use two classes of algorithms for key generation:

- symmetric algorithms - between the transmitter and receiver are the same keys
- asymmetric algorithms - different keys are used between the transmitter and receiver

In Tetra, authentication is based on symmetric keys, and the length of the

secret keys is 128 bits regardless of how they are generated.

In all procedures, radio terminal are registered in the network for the first time with a User Authentication Key (UAK). This key will be recorded in the handheld as well as database authentication center

The authentication process between the radio terminal and the infrastructure is carried out according to the procedures:

- authentication of a radio terminal by the infrastructure (which may include the base station and the authentication center)
- infrastructure authentication by the radio terminal
- mutual authentication

Authentication provides proof identity of all radio's attempting use of the network. A session key system from a central authentication centre allows key

storage (Secret key need never be exposed). Authentication process derives air interface key (TETRA standard) (Figure 3).

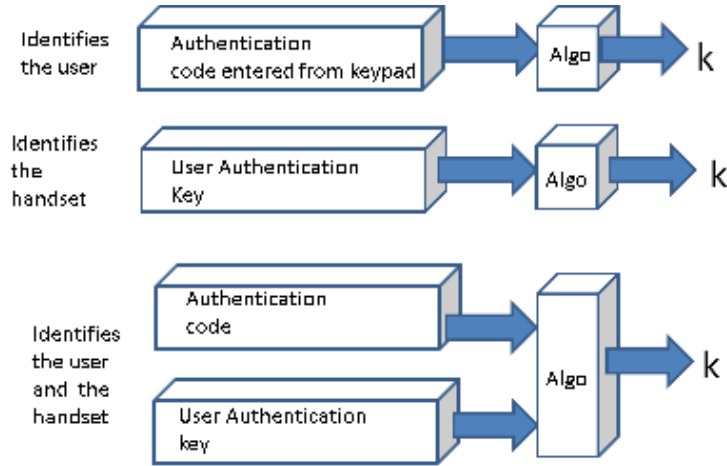


Fig. 3. Authentication key generation

*Encryption*

In the wireless communications system the most vulnerable interface is the air interface because it has the role of ensuring the communication between the radio terminal and the base station. Through this interface, some important security features can also be achieved.

By using the security keys required for the air interface, encryption will be performed between the radio terminal and the infrastructure (Figure 4). This encryption is valid for group or individual communications, for Direct Mode Operation, for voice and data (Figure 5).

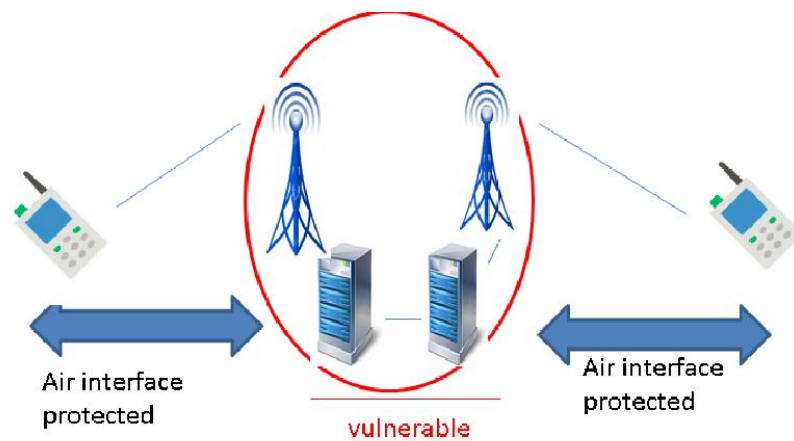


Fig. 4. Standard air interface

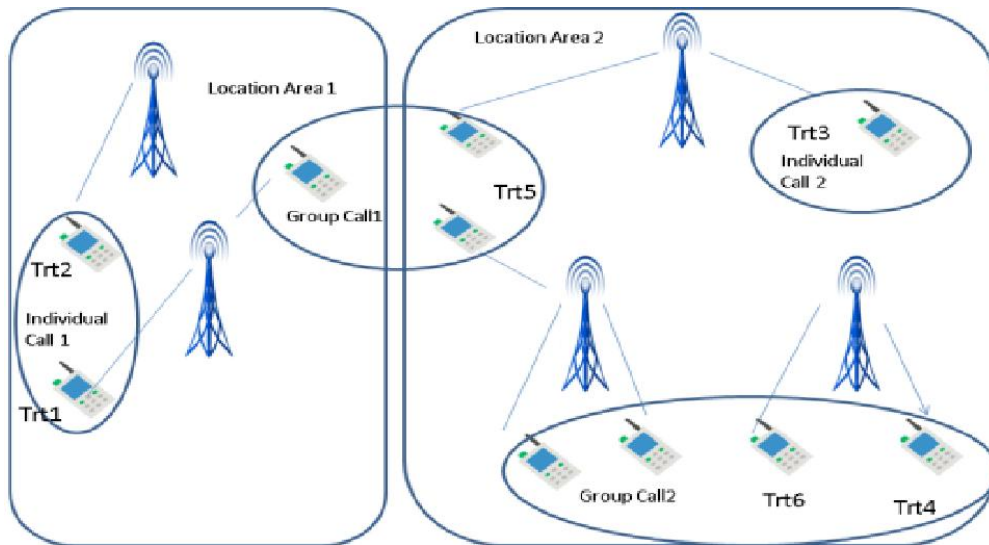


Fig. 5. Communications individual / group DMO

From the point of view of communication for Tetra system users, encryption of the air interface can be:

- dynamics (in trunking mode)
- static (in DMO mode)

In static mode air interface uses a fixed encryption key -SCK (Static ciphering key) that encrypts signaling and voice. In this case class classification system is second class.

The method by which to transfer the secret keys from the infrastructure to the radio terminal is called OTAR (Over The

Air Re Keying), the standardized procedure [ 8 ]

The DCK -Derived Ciphering Key is used in dynamic mode. These are generated during authentication (Figure 6). Each part of the authentication process produces a part of the DCK, DCK1 and DCK2. These are combined by algorithm TB4. DCK is used wherever possible as it is the most secure. It only has a life equivalent to the authentication period and is unique to the terminal.

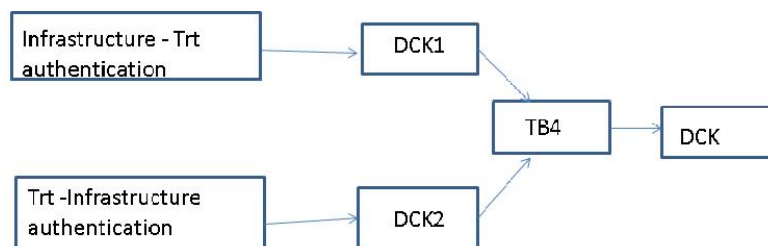


Fig. 6. Deriving DCK from mutual authentication

The keys are still in use:

-CCK - Common Cipher Key - are generated by the infrastructure server and distributed to all radio terminals

-GCK - Group Cipher Key - are generated by the infrastructure server and distributed only to radio terminals belonging to a closed group

The use of the encryption keys is shown in Figure 7.

Tetra digital trunked radio system supports both standard encryption algorithms as well as owners, which are used depending on the applications you made system.

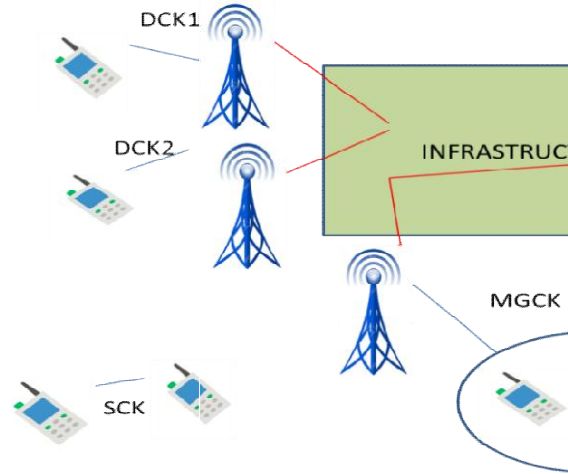
The air interface secret algorithms allow a secure connection between the infrastructure and In order to interoperability of equipment from different manufacturers have implemented standard algorithms (TEA1, TEA2, TEA3, TEA4), which are subject to

regulations ETSI and SAGE Security Algorithm Group of Experts Group- so:

- TEA1, TEA3, TEA4 - regulatory authority is ETSI

- TEA2 - the regulator is the Public Safety Organization in Schengen

These algorithms are written in strict licensing rules (Restricted Export Algorithms)



**Fig. 7.** Air Interface Encryption – the Keys

*Secret key management and air interface authentication*

Generally, a single authentication key standard attached to a key management is specified. This algorithm is TAA1 and is controlled by ETSI.

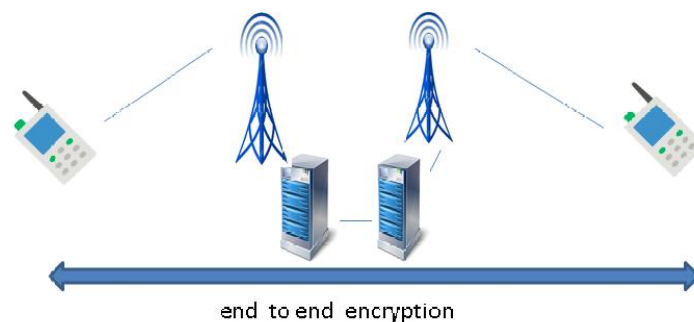
EN 392-7 document and has two levels: temporary or permanent. Each radio equipment has its own code called TEI which makes it possible to enable / disable the radio terminal in the network.

*Enable/Disable*

Through this mechanism, the access or prohibition of the radio terminal to the Tetra communications network functions is achieved. This is defined in the ETSI

*Encryption End to End*

Encryption is taking place between radio system terminals, the connection between them being achieved by infrastructure (Figure 8).



**Fig. 8.** End to End Encryption

For this encryption there are no 'standard' algorithms defined by SFPG, but use algorithms IDEA (International Data Encryption Algorithm) and AES (Advanced Encryption Standard) IDEA was defined as a good candidate 64 bit block cipher algorithm for use with

TETRA and test data and an example implementation was produced. AES is a block encryption algorithm in which block and key lengths could be: 128 bits, 192 bits, or 256 bits. The AES specification restricts the block length to 128 bits. Thus, the input and output of



encryption and decryption algorithms is a 128-bit block. AES operations are defined as matrix operations, where both the key and the block are matrixed. At the beginning of the cipher roll, the block is copied to a state-of-the-art table, the first four bytes on the first column, then the next four on the second column, and so on until you complete the dashboard. The algorithm modifies this array of numbers at every step, and then supplies it as an output [9].

The Tetra standard does not have security specifications for two interfaces: the peripheral equipment interface (PEI) and interfaces with other Tetra systems (ISI).

- PEI - the interface realizes the communication between a mobile or fixed radio terminal and a peripheral terminal (mainly the transmission of data); It can virtually be considered that there is no need for encryption of the link, because in general the connection between the two equipment is made locally (distance of about 1m)
- ISI – Is the interface that allows the connection between two Tetra trunk networks.

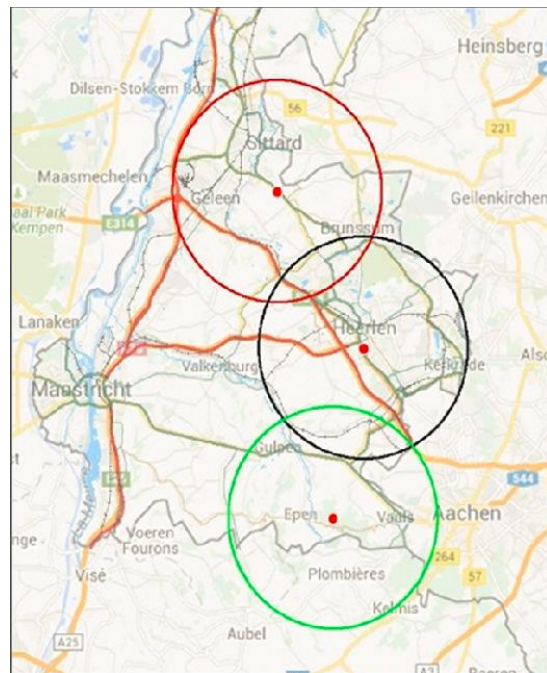
Cross-border cooperation has to be achieved through the ISI interface specifications. In the past few years, various projects have been carried out to achieve interoperability between Tetra systems belonging to Schengen States.

A first test was conducted in Finland where there was a national Tetra network (Vivre)

Used by various public security organizations (police, border health services, fire brigades) and is perfectly interoperable. From the interoperability tests with neighbouring countries, Sweden and Norway, it has become necessary to standardize the interoperability architecture for applications (e.g. command and control)

and infrastructure (e.g., interface gateways, mobile unit) [10].

Another project had been in Aachen (Germany) -Liège (Belgium) -Maastricht (The Netherlands) to demonstrate a proof of concept for ISI. Operational research was applied during the operational field trials. The functional requirements for the TETRA ISI were based on Technical TETRA ISI standard and Interoperability profiles, operational scenarios defined for the three-country pilot scenarios and recommendations from the users (ASTRID, BMI and C2000) [6]. This test demonstrated the migration of terminals between the three countries (Figure 9).



**Fig. 9.** Tetra pilot - Aachen (Germany) – Liège (Belgium) –Maastricht (The Netherlands)

The interoperability of the Tetra digital trunking system for cross-border cooperation was demonstrated in the DACEA (Romania-Bulgaria Danube Cross-Border Earthquake Alert) project (Figure 10).



**Fig. 10.** Cross-border cooperation between Romania and Bulgaria

DACEA project general objective is to develop a cross-border system for Earthquake alerts in order to prevent the natural disasters caused by those events in the cross-border area, taking into account the nuclear power plants and other high risk facilities located along the Danube on the territories of Romania and Bulgaria[11].

#### 4. Conclusions

TETRA has historically been the digital technology with the greatest uptake in Europe

The resilience, availability and security of TETRA coupled with its efficient use of spectrum will continue to make it an attractive bearer for critical voice and dedicated data solutions.

Experience of several nations in responding to crises emphasises the crucial role of support for managing communications interoperability. This includes real-time keymanagement to define the cryptographically determined communication groups that are the fundamental feature of Tetra system, making operational choices on the 'profiles' of system and security parameters, optimising gateways between systems, and maintaining priority settings.

#### References

[1] Ramon Ferrús, Oriol Sallent, "Public Protection and Disaster Relief Communications," in *Mobile Broadband Communications for Public Safety: The Road Ahead Through LTE Technology*, Wiley,

August, 2015,

- [2] Milan Stojkovic *Public safety networks towards mission critical mobile broadband networks Master of Telematics - Communication Networks and Networked Services Submission date: June 2016 Norwegian University of Science and Technology Department of Telematics*
- [3] Peter Stavroulakis *Terrestrial Trunked Radio – TETRA - Springer-Verlag Berlin Heidelberg 2007*
- [4] Baldini, G., Karanasios, S., Allen, D., & Vergari, F. (2013). *Survey of Wireless Communication Technologies for Public Safety. Communications Surveys & Tutorials, IEEE, 16(2), 619-941*
- [5] SALUS-Security And Interoperability in Next Generation PPDR Communication Infrastructure SFP7 Project Number: 313296
- [6] ISITEP-Inter System Interoperability for Tetra-TetraPol Networks- FP7 Project Number: 312484
- [7] Roelofsen, G. (2000) "TETRA Security", *Information Security Technical Report, Elsevier Science, Vol 5, No.3*
- [8] SFPG Recommendation 01
- [9] Benjamin Dowling, Marc Fischlin, Felix Günther, Douglas Stebila, "A Cryptographic Analysis of the TLS 1.3 Handshake Protocol Candidates," in *ACM Conference on Computer and Communications Security (CCS 2015)*, February, 2016
- [10] Gianmarco Baldini -Report of the workshop on "Interoperable communications for Safety and Security"- 28-29 June 2010 – Ispra, Italy
- [11] L. Dimitrova, D. Solakov, S. Simeonova, I. Aleksandrova- *System of Earthquakes Alert (SEA) in the Romania-Bulgaria cross border region-Bulgarian Chemical Communications, Volume 47, Special Issue B 2015*



**Claudiu Dan BÂRCĂ** graduated from Faculty of Computer Science for Business Management, Romanian American University in 2007, and holds a master degree in Economic Informatics since 2008 and a PhD in the field of Engineering Sciences since 2013. He is an assistant lecturer within the Faculty of Computer Science for Business Management having nine years of teaching experience. He also has good research and publishing activity: he was a member of the research teams of international and national projects. His core competences are in software programming and connected areas.