Heart Disorder Detection with Menard Algorithm on Apache Spark

Lorenzo Carnevale, Antonio Celesti, Maria Fazio, Placido Bramanti, Massimo Villari

Heart Disorder Detection with Menard Algorithm on Apache Spark

Lorenzo Carnevale[1 3], Antonio Celesti[2], Maria Fazio[1], Placido Bramanti[3] and Massimo Villari[1 3]

(1) Department of Engineering, University of Messina, Italy - {lcarnevale,mfazio,mvillari}@unime.it,
(2) Scientific Research Organisational Unit, University of Messina, Italy - acelesti@unime.it,
(3) IRCCS Centro Neurolesi "Bonino Pulejo", Messina, Italy - {lcarnevale,mvillari,pbramanti}@irccsme.it

Nowadays, healthcare is facing Big Data processing in order to support medical staff by means of decision making tools. In this context, a challenging topic is the storing and analysis of data in the cardiology field. Electrocardiogram produces signals about the heart health that need to be processed in order to detect a possible disorder. In this paper, we discuss an Apache Spark based tool and that uses the Menard algorithm. In order to validate our solution, we performed experiments on a use case in which the algorithm has been implemented in order to detect heart disorder. Experiments prove the goodness of our approach in terms of performance.

Key words: Big Data, healthcare, cardiology, Heart, ECG, arrhythmia

1.1 Introduction

Currently, the healthcare industry is looking at the adoption of Big Data due to the volume, velocity and variety properties of health data. Big Data solutions have been adopted so far in different healthcare fields including biotechnology [1], clinical analysis [2], and so on. Therefore, a careful study of clinical data performed by means of decision making tools helps doctors to make diagnosis.

In this regard, hospital facilities are relying on external providers or internal staff to manage and analyze clinical data. Apache Spark is a licensed framework designed to support distributed applications for creating batch applications, interactive queries and stream processing. Spark adopts the MapReduce software framework, created by Google to support distributed data computing on cluster.

In this scientific work, we used Apache Spark in order to analyze electrocardiogram (ECG) signals. Specifically, goal of our analysis was to detect heart disorder. To this end, we planned to use the Menard algorithm.

The rest of the paper is organized as follows. Related work are summarized in Section 1.2. An overview about heart physiognomy and functionality and ECG is presented in Section 1.3 and 1.4, whereas a Menard algorithm description and its Spark implementation for hearbeat peaks detection are presented in Section 1.5. Experiments and

evaluation results are presented in Section 1.6. In the end, conclusion and lights to the future are summarized in Section 1.7.

1.2 Related Work

The scientific community has proposed several scientific works on heart disorder analysis in order to improve the accuracy, prevent diseases and reduce mortality. In the following, we report some of these works.

Many scientific works combine ECG signal analysis with the most famous Big Data framework: Apache Hadoop. A tele-ECG system has been proposed in [3]. The authors aimed to process Big Data in order to detect and monitor heart diseases. They proposed a cluster which takes advantage of Apache Spark framework. Specifically, this system classifies data using decision tree and random forest.

An analysis of arrhythmias has ben proposed in [4], in which the authors proposed an automatic detection of P-wave in an ECG. More specifically, they worked a improved method based on local distance transform, such as horizontal segments and rising or declining segment. As result, they proved the simplicity and efficiency of the algorithms for transplanting to wearable medical devices whose processing ability is weak.

In order to facilitate the data migration from medical devices to Cloud storage, we mention the work proposed in [5]. The authors described the first step of an architecture able to manage the Big Data Acquisition and Integration workflow for storing health data coming from several medical instrumentations. This scientific work has proved the goodness of the method used in terms of time performance.

A Cardiovascular Disease (CVD) detection algorithm was proposed in [6]. The algorithm uses patient demographic data as input, along with several ECG signal features automatically extracted through signal processing techniques. The algorithm has been integrated into a web based system that can be used at anytime by patients to check their heart health status. Signals are sent from the ECG sensor attached on the patient's body to the detection algorithm via an Android device. Cross-validation results showed the 98.29% accuracy.

We aim to enrich the scientific literature by proposing a work that uses the Menard algorithm to perform the distributed calculation of an ECG signal through the Apache Spark framework, in order to detect the most common heart diseases.

1.3 The Heart. How does it work?

The heart is passively filled with blood that comes from veins and actively pushes blood through the body. A complete contraction and relaxation sequence represents the heart cycle, which normally is repeated about 75 times per minute. The figure 1.1 shows this cycle.

Specifically, in the first phase (diastole) the heart is completely relaxed and blood flows into its four cavities because of the atriovascular valves opening. The second phase (systole) begins with a short limbs contraction that completely fills ventricles

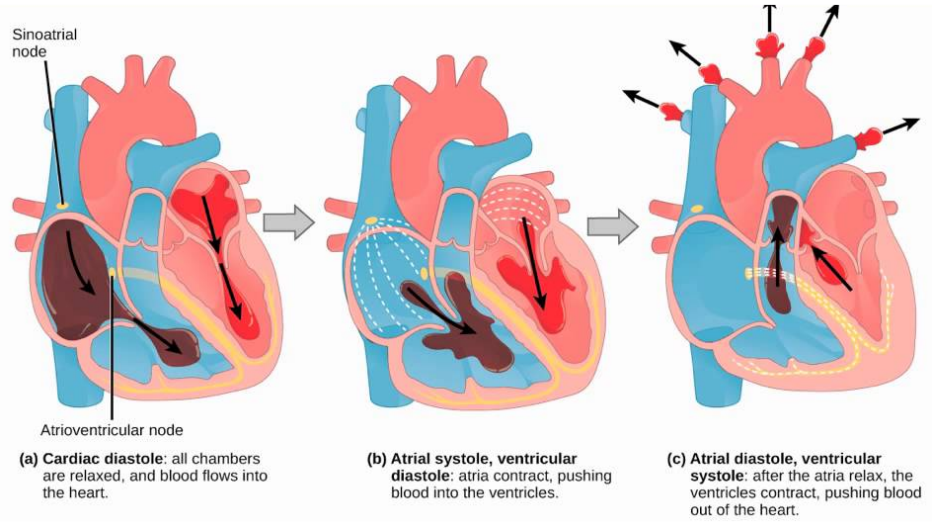

Fig. 1.1. The heart cycle [The Mammalian Heart & Cardiac Cycle]

with blood. Afterwards, the ventricles will contract for about 0.3 seconds. The force of their contraction closes the atrioventricular valves, opens the semilunar valves and pumps blood into the large arteries. During the last phase, blood flows into the atrioventriculars.

1.4 Heart Medical Instrumentation: the ECG

The electrocardiogram (ECG) is an instrumental diagnostic test that graphically records the rhythm and electrical activity of the heart. This allows the cardiologist to detect health disorder such as the presence of heart arrhythmias, ischemia, myocardial infarction or outcomes of a previous heart attack.

Indeed, heart pathological phenomena creates abnormal conditions in the muscle fibrocells, generating a different pattern from the standard. However, a standard pattern does not represent a proper heart condition, and vice versa, healthy people can have abnormal ECG outcomes. In these conditions, medical opinion is always mandatory.

The ECG test produces positive and negative waves, according to the signal position compared to the baseline, called isoelectric. Each wave is the graphic representation of an electric phenomenon that occurred in the heart. In particular, we can distinguish five signal period as reported in the figure 1.2. For our purpose, here we specify the QRS complex as the stimulus propagation to the ventricular muscle.

1.5 Application Design

Over the years, many algorithms have been developed for recognizing the QRS complex, which can be classified according to their complexity and performance. Specif-

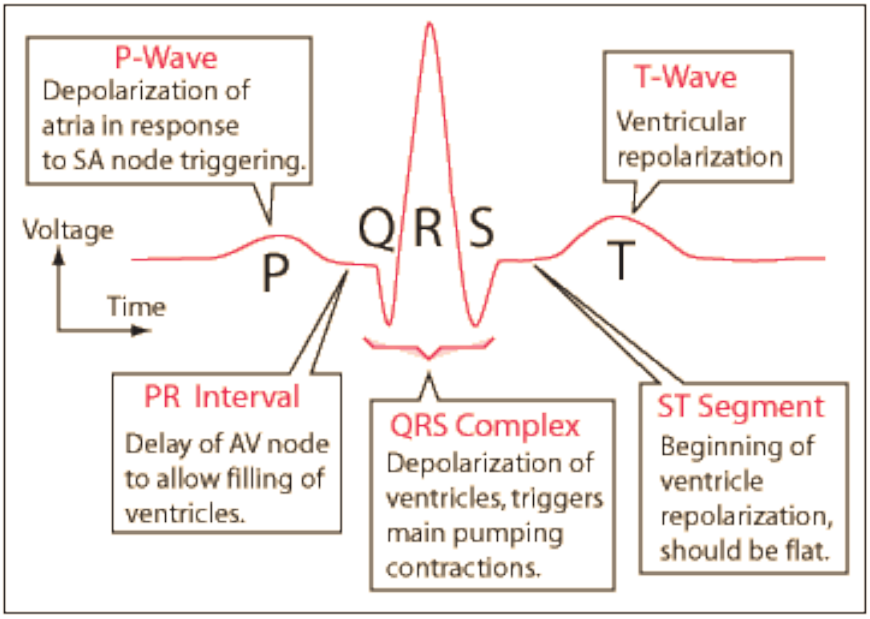

Fig. 1.2. ECG outcomes [hyperphysics.phy-astr.gsu.edu]

ically, most efficient and complex algorithms are based on appropriate techniques for filtering and processing ECG signals, whereas less complex algorithms are based on statistical thresholds. In the following, we are going to focus on a specific algorithm based on ECG signal derivation: the Menard algorithm [7]. This is calculated using the following equation:

$$Y(n) = -2X(n-2) - X(n-1) + X(n+1) + 2X(n+2) \tag{1.1}$$

Afterwards, a ζ threshold of 70% of the maximum $Y(n)$ value is chosen:

$$\zeta = 0.7 * max[Y(n)] \tag{1.2}$$

Finally, the algorithm adopts the following decision rule to detect the QRS complex:

$$Y(i) > \zeta \Rightarrow QRS \tag{1.3}$$

In this scientific work, the Menard algorithm has been used for calculating the QRS complex thorugh the utilization of Apache Spark. The dataset used for the following experiments comes from the Physionet.org European ST-T Database. It includes a signal acquired by a digitizer with sampling rate equal to $f_s = 250Hz$. In order to process it, the file must be properly formatted. For this purpose, two preliminary steps were needed.

Primarily, having to act on multiple samples at the same time, we needed to organize an appropriate set of samples on one file line because Apache Spark treates each of

them as a strings RDD. Moreover, Spark distributes workload in tasks, which involves multiple lines processement of the RDD. Neverthess, the Menard algorithm implementation performs the derivation through the formula 1.1, which shows how a continuous set of data is needed. Indeed, in order to determine the nth element of the derivative, we needed to know the two previous and subsequent elements of the nth ECG signal. Thus, we overlap content introducing row by row redundancy (except the last one). This avoids the information losing during the cluster distribution phase. Moreover, during the source file formatting process, each line is indexed for tracking the reference samples.

However, how many samples should form an RDD element? How many values should be placed on a row of the file? Let's consider that an electrocardiogram typically oscillates between -20mv and 20mv, the calculation of the Menard algorithm threshold may not take into account these variations using a signal large portion. Therefore, it may not correctly detect heartbeat peaks. The proposed solution was to implement a version of the algorithm with an adaptive threshold, which is calculated differently for each sample block.

Thus, our implementation uses a set of samples with a duration equal to 10 seconds. To this end, if we indicate with f_s the sampling frequency of the ECG signal, all the file lines (except the last one) have $n = (f_s * 10) + 4$, where 4 is due to the abovementioned overlap.

The only information required for calculating the QRS complex is represented by the detected peak index because, multiplying it by the sampling frequency reciprocal, it is useful to trace the beat time. Moreover, we had to determine which peak signals above the threshold may be considered a heartbeat. Indeed, these values are more than one around a QRS complex. In order to simplify it, we chose the first value above the threshold. The figure 1.2 shows the peak signals of an ECG derivative.

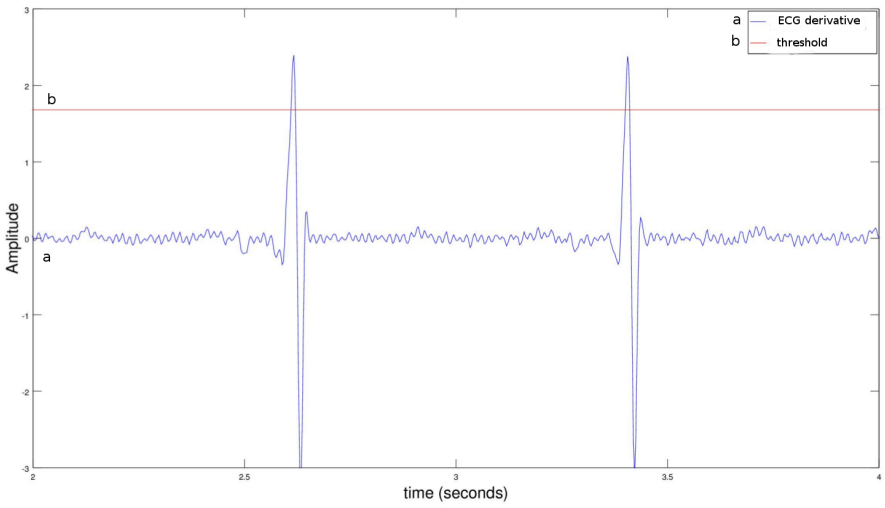

Fig. 1.3. ECG derivative and threshold

In order to distribute the RDD to the cluster's nodes and create a elements list on the driver, it is necessary to use the *collect()* method. It is the first action performed by the application. Indeed, until now, we have only talked about transformations. Therefore, the *saveAsTextFile()* method examines all the peaks' RDD trasformations in order to save it on a file.

What if the threshold values of a ECG signal section were between two blocks (or between two nodes)? Both the first index above the threshold of the first block and the first index above the threshold of the second block would be selected as peaks. The proposed solution requires that application knows the values found, and recognizes the extremely close peaks.

1.6 Experiments

The testbed for performing our experiments was configured into a three nodes (1 master, 2 slaves) docker distributed environment with the following hardware specifications: Quad-Core and RAM 8 GB. Each node was configured with Ubuntu 14.04, OpenJDK 7, Spark 1.6.1 and Scala 2.11.8. Moreover, the Apache Spark framework used its scheduling process, without relying on third party cluster manager, such as YARN. Specifically, using a 5GB input file, Spark distributes the workload in 157 tasks among cluster's nodes.

The test included a hour ECG signal with sampling rate equal to 360Hz. With reference to section 1.5, actions carried out by the Spark application are *collect* and *saveAsTextFile*. In this regard, we show the *collect* tests outcomes in the figure 1.4.

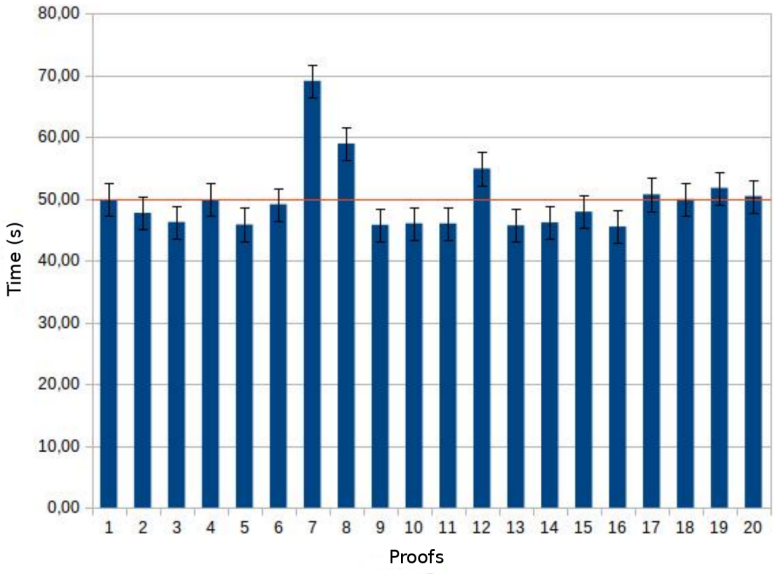

Fig. 1.4. *collect* action for 5 GB file. 20 proofs average and confidence interval equal to 95%

The temporal outcomes are quite similar, as highlighted by the small confidence interval. The *collect* average time value is about 23 seconds.

Now consider the case of the *saveAsTextFile* action, shown in the figure 1.5. Again, the input size increasing causes a nonlinear increasing of the action execution time. Specifically, the average execution time is about 5.6 seconds. In this specific context, the *collect* benefits more from parallelization than *saveAsTextFile*.

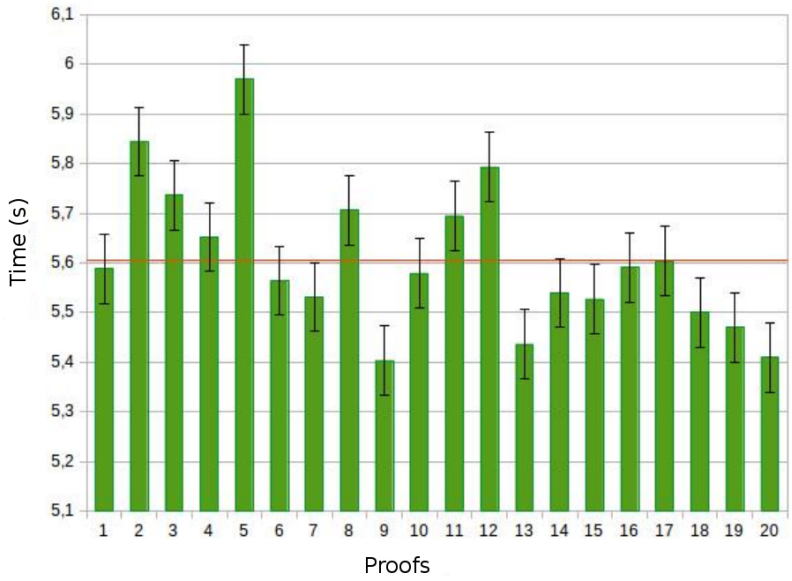

Fig. 1.5. *saveAsTextFile* action for 5 GB file. 20 proofs average and confidence interval equal to 95%

The analyses carried out are related to a specific ECG signal. Clearly, each ECG differs from others in terms of peaks and abnormalities, therefore these outcomes are not considered indicative for generic ECG signal.

1.7 Conclusion and Future Work

This scientific work has addressed the problem of the ECG signals distributed processing. Specifically, the algorithm implemented for the determination of heartbeats and arrhythmias required a preprocessing phase. Each Big Data framework bases its computing philosophy on task independence, wherefore this context is not in the ideal conditions for exploiting distributed processing. To this end, a local files preprocessing must be made. Thus, large files could represent the bottleneck of the entire application.

On the other hand, the performance offered by this computing paradigm is definitely important. As a conseguence, solved the problem of local preprocessing of the ECG

signals, the application would only get benefits from processing on a cluster. For this purpose, an idea might be to use an ad hoc device for recording the electrocardiogram. Specifically, if we suppose to make a device that during the recording of the ECG signal introduces the overlapping required for the algorithm, any limitation due to the nature of the data itself would be eliminated.

From the algorithm point of view, the ECG signal analysis was performed based on the heart rhythm obtained by calculating the R-R intervals. Indeed, more complex operations could be implemented by making elaborations based on the shape of the waves that make up a heartbeat. Thus, exploiting the compute parallelization, it would be possible to implement computationally highly costly algorithms by obtaining relevant performance by processing on a cluster.

Finally, Spark Streaming could be used to perform continuous and Real Time processing. In this regard, an ad-hoc device for sending electrocardiogram sections should be implemented, allowing continuous monitoring of the patient's health status.

ACKNOWLEDGMENT

This work has been supported by Cloud for Europe (C4E) Tender: *REALIZATION OF A RESEARCH AND DEVELOPMENT PROJECT (PRE-COMMERCIAL PROCUREMENT) ON "CLOUD FOR EUROPE"*, Italy-Rome: Research and development services and related consultancy services Contract notice: 2014/S 241-424518. Directive: 2004/18/EC. (http://www.cloudforeurope.eu/). Authors would like to thank Fabio Pandolfo for his valuable technical support in this scientific work.

References

1. A. Celesti, F. Celesti, M. Fazio, P. Bramanti, and M. Villari, "Are next-generation sequencing tools ready for the cloud?" *Trends in Biotechnology*, vol. 35, no. 6, pp. 486 – 489, 2017.
2. A. Celesti, F. Maria, A. Romano, A. Bramanti, P. Bramanti, and M. Villari, "An oais-based hospital information system on the cloud: Analysis of a nosql column-oriented approach," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, no. 99, pp. 1–1, 2017.
3. M. A. Ma'sum, W. Jatmiko, and H. Suhartanto, "Enhanced tele ECG system using hadoop framework to deal with big data processing," in *2016 International Workshop on Big Data and Information Security (IWBIS)*. IEEE, oct 2016.
4. Y. Wang, L. Wang, X. Chen, and W. Zhu, "P wave detection and delineation based on distances transform," in *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, aug 2016.
5. L. Carnevale, A. Celesti, M. Fazio, P. Bramanti, and M. Villari, "How to enable clinical workflows to integrate big healthcare data," in *2017 IEEE Symposium on Computers and Communications (ISCC) (ISCC 2017)*, Heraklion, Greece, Jul. 2017.
6. H. Alshraideh, M. Otoom, A. Al-Araida, H. Bawaneh, and J. Bravo, "A web based cardiovascular disease detection system," *Journal of Medical Systems*, vol. 39, no. 10, 2015.
7. A. Menrad, Ed., *Dual microprocessor system for cardiovascular data acquisition, processing and recording*, vol. Inr. Con5 Industrial Elect. Contr. Instrument. IEEE, 1981.