A Framework for Data Research in GIS Database using Meshing Techniques and the Map-Reduce Algorithm

Abdoulaye SERE[1], Jean Serge Dimitri OUATTARA[2], Didier BASSOLE[3],
José Arthur OUEDRAOGO[4], Moubaric KABORE[5]
Network of Computer Science Teachers
and Scientists of Faso
Bobo-Dioulasso, Burkina Faso

***Abstract*—Everywhere, centers, laboratories, hospital and pharmacy have faced many challenges to delivery quality of health service due to constraints related to limited availability of resources such as drugs, places, equipments and specialists, often in health deficit with increasing number of patients, for instance during COVID-19 pandemic. Late information on these constraints from health service centers will play negatively on service quality because of time delayed between requesting service on place and the response to delivery safe service. All these problems don't strengthen prevention or fighting against diseases in a region. This paper proposes a data research framework in a NoSQL database based on GIS data, containing an abstract table that could be inherited or specialized to any adopted GIS solution leading to a central data management instead of installing several database sites. The central database accepts data updated in back office by data owner and allows data research based on meshing Techniques and the map-reduce algorithm in front office. Variant meshing techniques have been presented to clustering GIS data with associated definitions of the content of map-reduce in order to improve processing time. In application in health service, the experimental results reveal that this system contributes to improve drug management in pharmacies and could be also used in others fields such as Finance, Education and Shopping through agencies spread over the territory, to strengthen national information systems and harmonised data.**

Keywords*—Map-reduce; big data; digital health; classification; Geographic Information System (GIS); COVID-19; Spark; MongoDb; NewSQL;NoSQL

I. Introduction

The Sustainable Development Goals (SDG) reaffirm international commitment to achieve Universal Health Coverage (UHC) by 2030. The quality of health services is a global imperative in view universal health coverage, according to World Health Organization (WHO) in [12] and OCDE in [13]. Thus, a framework to measure quality of health service has been proposed by Arah, and others in [14]. Quality of Health Services provided by Hospital, pharmacies, Health Centers has played essential roles in fighting and prevention of diseases.

Health centers, Governments have faced many challenges due to many constraints related to availability of resources such as specialists, equipments, places, drugs during diseases. Among the need of human capacity building, materials for radiography contribute to improve Health services. For instance, according to Abdulrahman M. Qahtani and others in [11], since the beginning of the COVID-19 in 2019, WHO also faced many challenges in increasing the global healthcare and Hygiene awareness to overcome COVID-19 pandemic.Thus, the needing of noze cover and washing hands regularly have been strongly recommended by Governments to prevent the COVID-19 disease.

The question is why going to a saturated clinic for health care and leaves others unsaturated. Information on available resources from Health centers might be opened somewhere and should indicate to patients, the way to follow in order to take the best decisions related to safe health. That will reduce forward death rate.

Today's technologies indicate the scale and speed at which technology is transforming traditional socio-economic sectors such as Health to reach digital Health.

Thus, digital transformation through software based on processing data related to Health has been proposed by engineers and scientists. For instance, applications based on disease diagnostic have contributed to support specialists in disease research.

Classification techniques are used in Big Data to identify groups in order to accelerate data processing and to take best decisions in smart system.

In classification, criteria can be taken into account to have data in the same groups. In Machine Learning, classification techniques such as supervised classification or not supervised classification, support Vector Machine (SVM), Decision Tree, Fuzzy Classification, Multi-Label Classification [1] could be used to establish relations between data.

Many techniques of tiling a space have been also developed by scientists to obtain cells in different grids such as Voronoi diagrams, Triangulation Delaunay, quasi-affine transformations, presented in [2], [3] and [15]. Fortune's algorithm also gives a way to build voronoi diagrams with a given set of continuous points. All these methods allows to get either regular grids or irregular grids in the image space, leading to data classification.

In a regular grid, each cell has the same geometric shape and the same size while in the irregular grid, the cells have different sizes or shapes. For instance, Vacavant's thesis in [4] presents different techniques of mesh generation used in simulation, that lead to irregular grids. Several tools as in [16] and [17] generate meshing models.

The Map-Reduce framework in [6], [7], [8], [10] performs speedly a large volume of data, in using the parallelism of map and reduce. For instance, a survey on performance comparisons of different frameworks such as Hadoop, Spark, Phoenix++, Marissa, Mariane, Sasreduce, Bitdew, Mr4c and Themis, has been presented by Zeba Khanam and others in [9]. An application of the map-reduce algorithm to improve the Hough Transform method processing has been introduced by SERE and others in [5]. It has been extended by Mateus Coelho and others in [19] to deal with circle recognition. SERE and others in [18] have also used the map-reduce algorithm to extract speedily posts from social networks.

Our work concerns with the problem to find out a particular data in Big Data distributed on different clusters. The proposed method is represented by an architecture that searches a data in a grid of clusters with algorithms introduced into the functions map and reduce. The generated clusters takes the concept of classification based on k-neighborhood into account.

This paper is organized as follows: The Section II named preliminaries introduces the problem specification and the concepts related to meshing techniques and the map-reduce algorithm. The Section III explains the proposed method with the applications of meshing techniques and the map-reduce algorithm. Experimental results deal with the case study of drug management in pharmacies, illustrated by the Section IV.

II. Preliminaries

This section brings informations on the problem specification, the description of the map-reduce algorithm and the meshing techniques used in Discrete Geometry.

A. Problem Statement

Let W be the universal set of database sites distributed in a space. Let S be a subset of database sites such as $S \subset W$. Suppose that $S = \{S_1, S_2, \ldots, S_{n-1}, S_n\}$ where S_i is a database site. All the data in database site $S_i \in$ S, together possesses the characteristics of Big Data through data volumetric.

Consider d as a specific data such as $d \in u$ and $u \in W$. u could be a member of S or not. Let M be a point. The problem is to find out database sites $S_i \in S$ where database sites S_i must be the neighbors of the point M and $d \in S_i$. That leads to a decisional problem in calculability.

There are two manners to verify if the data d is in the database S_i. That means if $d \in S_i$? :

- the first one is to search data sequentially in each $S_i \in S$ in going from 1 to n.
- the second one is to proceed by a parallel verification with research in each group of S_i.

Our hypotheses is that the parallel verification is speeder than the first one. Suppose that α is the time taken to perform S_i.

In the first case, the effective execution time will be $n\alpha$ in the worst case. The worst case corresponds to the conditions (if $d \in S_n$ or if d is not a member of any S_i).

In the second case, let β be the number of group of S_i. That means clearly $\beta \leq n$. α will be also considered as the execution time to perform each group. The global execution time will be $\beta\alpha$. We obtain $\beta\alpha \leq n\alpha$ because $\beta \leq n$.

Thus, our analysis will focus on two axes : firstly, before searching data, we classify data into clusters; secondly, we accelerate data research in using the map-reduce algorithm on limited clusters.

Furthermore, the following sections will deal with an analysis of the map-reduce framework and meshing techniques that generate different groups to make data research speeder in the Big Data context.

B. Mesh Generation

There are also many techniques in discrete geometry to create regular or irregular grid which allows forward detection of the nearest sites. For instance, Vacavant's thesis in [4] study possible applications of regular grids and irregular grids in simulation, illustrated by the Fig. 1.

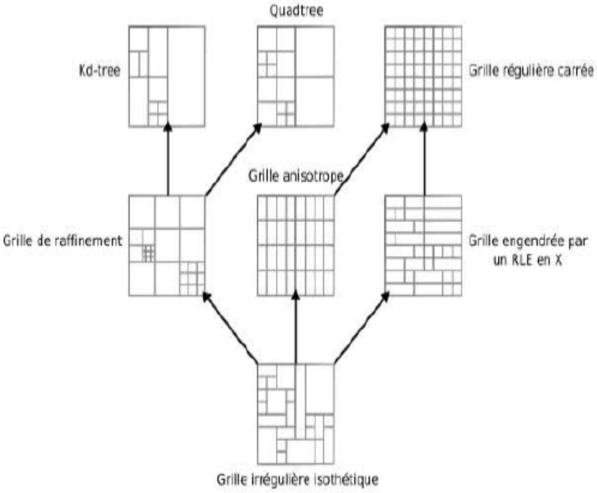


Fig. 1. Irregular Grids and Regular Grids in Vacavant's Thesis in [4].

K-neighborhood is very used in Discrete Geometry. It follows from meshing technique application and corresponds to 4-neighborhood and 8-neighborhood in a two dimensional space :

Definition 1 (4-neighborhood in [3], [15]) Let A and B be two pixels with integer coordinates respectively (X_A, Y_A) and (X_B, Y_B).

(A and B are in 4-neighborhood) $\iff$ $|X_A - X_B| + |Y_A - Y_B| = 1$

4-neighborhood uses the distance of Manhattan based on D(A, B) = $|X_A - X_B| + |Y_A - Y_B|$ in 2D.

Figure 2 shows an example of 4-neighborhood between the central pixel A and the pixels B, C, D, E.

Definition 2 (8-neighborhood in [3], [15]) Let A and B be two pixels with integer coordinates respectively (X_A, Y_A) and (X_B, Y_B).

(A and B are in 8-neighborhood) $\iff$ max $(|X_A - X_B|, |Y_A - Y_B|)$=1

But, 8-neighborhood implements the distance of Tchebychev which is defined by D(A, B)= max $(|X_A - X_B|, |Y_A - Y_B|)$ in 2D.

Fig. 3 shows an example of 8-neighborhood between the central pixel A and the pixels B, C, D, E, F, G, H, I.

Euclidian distance and K-nearest neighbor are another alternatives to define neighborhood, to build clusters and to reach classification.

Manhattan distance, Tchebychev distance and Euclidean distance lead respectively to several geometry shapes such as lozenges, squares and circles, used in computing neighbors.

These above definitions are necessary in computing nearest database sites. Others information related to Geographic Information System (GIS) might be integrated in this way.

The following sections will explain how meshing techniques can be useful in classification in order to speed up the processing through the map reduce algorithm.

C. Map-reduce Concepts

Map-Reduce content defines two important tasks, namely Map and Reduce. It describes the parallelism of the map functions followed by the parallelism of the reduce functions, as explained by Dean and others in [10] and SERE and others in [18]. The shuffle phase is executed automatically by the system, between both the map function and the reduce function.

The map function defines a transformation of pairs that accepts as inputs a single key and a value, noticed (k, v) pair and produces as outputs a set of intermediate (key, value) pairs (k_i, v_i). At the end of all the map functions, several pairs (k_i, v_i) are produced by the map functions. For instance the map function transforms the pairs (k, v) as input and produces the set of pairs (k_1, v_1), (k_2, v_2).

The shuffle phase starts after all the map functions ended and before starting the reduce functions. The shuffle phase consists of having together the value of the pairs (k_i, v_i) produced by all the map functions : it produces the pairs that have the same key. It also sort keys into correct order to prepare next computation. For instance, for the following pairs $(k_1, v_1), (k_1, v_2), (k_2, v_2), (k_2, v_3)$ produced by all the map functions, the shuffle phase returns the pairs $(k_1, < v_1, v_2 >), (k_2, < v_2, v_3 >)$. Thus, the shuffle phase carry out data classification where each class referenced by a key.

The reduce function takes as an input a (key, list of values) pair that contains a intermediate key and a set of values for that key. The reduce function produces a pair (key, result of a list of values). The key in the input is the same in the ouput. For instance, the pairs $(k_1, < v_1, v_2 >)$ and $(k_2, < v_2, v_3 >)$ become $(k_1, < v_1 + v_2 >), (k_2, , < v_2 + v_3 >)$.

Reduce functions could start before the end of all the map functions. Mixing map and reduce will reach an improvement of processing time : That will be studied in perspectives, with synchronization control between each others.

The map-reduce model consists of the parallelism of map function followed respectively by the shuffle phase and reduce functions: there is a master node that controls all the processes tasks, distributed on secondary nodes with distributed memory.

Thus, the map function, the shuffle phase and the reduce function have summarized successively in 1, 2, 3, 4, 5, 6 and 7 as follows:

$$Map(k, v) \longrightarrow \{(k_1, v_1), (k_1, v_2)\} \tag{1}$$

$$Map(k', v') \longrightarrow \{(k_2, v_2), (k_2, v_3)\} \tag{2}$$

$$\{(k_1, v_1), (k_1, v_2)\} \longrightarrow_{shuffle} \longrightarrow (k_1, < v_1, v_2 >) \tag{3}$$

$$\{(k_2, v_2), (k_2, v_3)\} \longrightarrow_{shuffle} \longrightarrow (k_2, < v_2, v_3 >) \tag{4}$$

$$Reduce(k_1, < v_1, v_2 >) \longrightarrow (k_1, < v_1 + v_2 >) \tag{5}$$

$$Reduce(k_2, < v_2, v_3 >) \longrightarrow (k_2, < v_2 + v_3 >) \tag{6}$$

Generally the reduce function is defined by :

$$Reduce(k_i, < v_1, ..., v_j >) \longrightarrow (k_i, v_k) \tag{7}$$

Where v_k =$v_1 + ... + v_j$, being the result of the operator + applied to the members of the list $< v_1, ..., v_j >$.

The map-reduce framework has applied to problems of counting the number of word in a document, computing the average of numbers, doing data selection and to sort dataset.

III. Method Description

This section is focusing on the application of meshing techniques very used in discrete geometry, database structure description, Data research algorithms and the content of map-reduce algorithm.

Due to the size of all the database sites reaching Big Data size as presented in problem statement, it will be better to have different clusters of database sites, linked by an unique structure of database in a central NoSQL database. In this manner, the central nosql database has connected on different sites in the same network : these sites, as data owners have the right to update data in back office. Users request to read data from the system in front office through their mobile phone or online with a desktop connected on the network provided by mobile operators for instance.

The network will contribute to improve the map-reduce algorithm according to the execution time, in giving easily data accessibility through data research.

Meshing techniques will define the central database structure in giving a relation between GPS references and related data. They also creates the clusters of database sites that must be together (in the same cluster) in order to improve data research with the map- reduce algorithm.

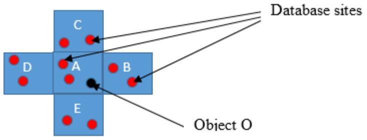


Fig. 2. 4-Neighborhood.

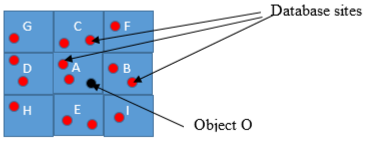


Fig. 3. 8-Neighborhood.

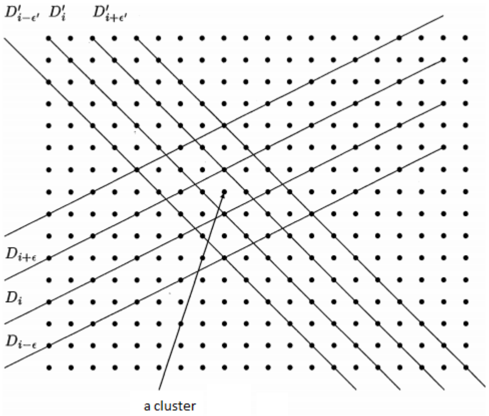


Fig. 4. Clusters Resulting of Crossing Parallel Straight Lines.

A. *Meshing Application*

Through the problem specification, all the database sites are referenced by the coordinates corresponding to its GPS reference. They have been distributed into different cells, called clusters. The center of each cluster is also referenced by the coordinates associated to its GPS reference.

A central database is introduced to take all the data of all the database sites into account. The Section III-B will discuss about database structure and the network architecture used.

The concept of k-neighborhood in discrete geometry uses the notion of distance. It determines the neighbor cells around a central cell.

4-neighborhood and 8-neighborhood are the particular cases of k-neighborhood in a two dimensional space. Fig. 2 shows an example of a regular grid with different pixels in 4-neighborhood, where A is the central pixel : there are four (04) pixels in 4-neighborhood with the pixel A, such as B, C, D, E.
While, Fig. 3 also presents an example of 8-neighborhood: the pixel A has eight (8) pixels as its neighbors such as B, C, D, E, F, G, H, I.

The pixels A, B, C, D, E, F, G, H, I indicate different clusters which contain database sites (as presented by red points in Fig. 2 and in Fig. 3). All the database sites represented by red points together has the characteristics of Big Data as illustrated in the problem statement. The problem is to find out the nearest database sites of an object O (a black point in Fig. 2 and in Fig. 3) that verify some conditions about the item d.

There exist many techniques to generate meshing grids. For instance, Quasi-affine applications leads to establish a grid on an image, to overcome pixels, called in our study, as clusters. For instance let (D_i) and (D'_i) be straight lines, respectively defined by $ax+by = w_i$ and $cx+dy = w'_i$ in a two dimensional space. Fig. 4 shows clusters, resulting of the intersection of the straight lines (D_i) and (D'_i).
Each cluster is referenced by idc_k. It contains the database sites. Each database site is referenced by $idc_{k,l}$ where k is an integer in $\{1, 2, \ldots, n-1, n\}$ and l is an integer in $\{1, 2, \ldots, m-1, m\}$. n is the number of clusters in all the GIS area while m corresponds to the maximal number of database sites in each cluster. Then the number of database sites in the GIS area defined by an image is then determined by the value $n.m$.

In others ways, let $nbd(x)$ be a function that allows to get easily the number of database sites in the cluster referenced by the index x. Then, the number of database sites in all the area will become $\sum_{k=1}^{k=n} nbd(ic_k)$.

That means :

$$\sum_{k=1}^{k=n} nbd(ic_k) \leq n.m \tag{8}$$

Moreover, if each database site has a maximal size, named s, the size of all the data named big data will be either $s.n.m$ or $s.\sum_{k=1}^{k=n} nbd(ic_k)$ where obviously

$$s.\sum_{k=1}^{k=n} nbd(ic_k) \leq s.n.m \tag{9}$$

Our purpose concerns time reduction in using the map-reduce algorithms for data research and to localize the data as inputs to these algorithms.

Then, we have:

$$s.\sum_{k=1}^{k=t} nbd(ic_k) \leq s.\sum_{k=1}^{k=n} nbd(ic_k) \leq s.n.m \tag{10}$$

where $t < n$.

Only a limited number of clusters defined by the value t, will be processed by the map-reduce algorithm.

B. *Database Structure and Architecture*

The proposed solution for data control is a network between different sites, connected on the central nosql database with permissions of data updating and data selection to each others.

Database structure takes into account the relation between database sites referenced by GPS coordinates distributed into the same cluster.

Thus, a cluster referenced by a pair (u_i, v_i) contains a set of pairs (m_i, n_i) representing database site references. For instance, the Table I shows a NoSQL database table.

TABLE I. A DATABASE TABLE

cluster references	database site references	data in sites	data in sites
(u_1, v_1)	(m_1, n_1)	Data	Data
(u_2, v_2)	(m_2, n_2)	Data	Data
(u_3, v_3)	(m_3, n_3)	Data	Data

We introduce a dynamic table to save clusters references corresponding exactly to the regular grid, associated to a region. This grid is resulting of the meshing technique application. But, to overcome the problem of empty clusters having no data inside that might happen in the central database, the solution is to establish for instance, an adapted irregular grid in following the technique of Delaunay triangulation or Voronoi diagram. Another possibility is to accept empty clusters being inserted into the nosql database Table I. The future works will focus on the strategies to transform a regular grid to an irregular grid to avoid empty clusters.

The Table II is useful for neighbor detection to get the next clusters as inputs to the entity "data research" in the following section.

TABLE II. A MEMORY TABLE OF CLUSTER REFERENCES GENERATED BY A MESHING TECHNIQUE

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)
(5,0)	(5,1)	(5,2)	(5,3)	(5,4)
(6,0)	(6,1)	(6,2)	(6,3)	(6,4)

The first column named "cluster references" of the table I contains the same data as the cell values of the Table II. While the second column named "database site reference" presents site references. Fig. 5 provides more details on the relation between the Table I and the Table II with (u_1, v_1) and (u_2, v_2) as cluster references.

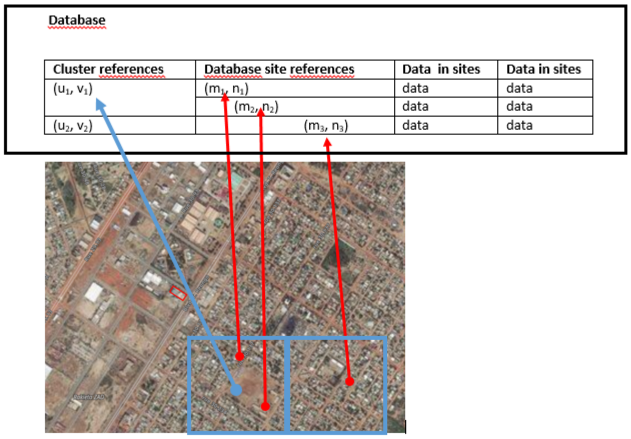
There are three layers in the architecture: the layer "database", the layer "map-reduce algorithm" and the layer "application" as presented in the Table III.

C. Data Research

In a two dimensional space, a quasi-affine application is defined by the function $F_0(x, y)$ which returns the coordinates of the central cluster that contains the initial object O of coordinates (x_O, y_O). That means $F_0(x_O, y_O) = (u, v)$ where (u, v) is the coordinates of the central cluster.

In 4-neighborhood, we'll obtain the couples $(u, v-1), (u, v+1), (u-1, v), (u+1, v)$ of the central clusters (u, v).

But, in following 8-neighborhood, the first package of neighbors around the central cluster (u, v) gives the following set of clusters $\{(u, v-1), (u, v+1), (u-1, v), (u+1, v), (u-1, v-1), (u-1, v+1), (u+1, v+1), (u+1, v-1)\ \}$.



Database

Cluster references	Database site references	Data in sites	Data in sites
(u₁, v₁)	(m₁, n₁)	data	data
	(m₂, n₂)	data	data
(u₂, v₂)	(m₃, n₃)	data	data

Fig. 5. Link Between an Image and a Database.

TABLE III. THREE LAYERS IN THE ARCHITECTURE

Layer 3 : Application
Layer 2 : Map-reduce algorithms
Layer 1 : Database

Our method will study 8-neighborhood in a regular grid, constituted of squares. Let I_1 be the set {-1, 0,+1}. It is obvious that card(I_1)=3. The first package contains clusters referenced by the set $\{(u+k, v+l)\ where\ (k,l) \in I_1\}$.

Finally, the number of clusters in the first package is determined by card(I_1^2).

As card(I_1^2)=9, there are nine clusters in the package 1, used for data research.

For generalization, suppose that

$$I_i = \{-i, -(i-1), ..., -1, 0, +1, ..., +(i-1), +i\} \quad (11)$$

The following Table IV summarizes the packages with their cluster references inside that may be used step by step in data research.

TABLE IV. PACKAGES

Packages	I_n	**cluster references**	**card(I_n^2)**
Package 1	I_1 ={-1, 0,+1}	$\{(u+k,\ v+l)\ /\ (k,l) \in I_1^2\}$	9
Package 2	I_2={-2,-1, 0, +1, +2}	$\{(u+k,\ v+l)\ /\ (k,l) \in I_2^2\}$	25
Package 3	I_3={-3,-2,-1, 0,+1,+2, +3}	$\{(u+k,\ v+l)\ /\ (k,l) \in I_3^2\}$	49
Package i	I_i= {-i, -(i-1),..., -1, 0, +1, ..., +(i-1), +i}	$\{(u+k,\ v+l)\ /\ (k,l) \in I_i^2\}$	$(2i+1)^2$

Data research deals with the processing of the package i. It begins initially with the package 1. If the value d is not found in the package i, then the next package i+1 will be processed: in fact, as (package i) ⊂ (package i+1), we are interested precisely in cluster references in the package i+1, not already performed in the package i, as illustrated in Fig. 6 by colored layers successively with the numbers 1, 2, 3, 4, and 5.

As we know that:

$$I_{i-1} \subset I_i \quad (12)$$

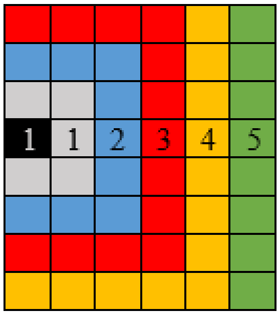


Fig. 6. Clusters Sent Respectively to the Map Functions for Analysis in the Order of Colored Packages 1, 2, 3, 4 and 5.

We have precisely:

$$I_i = I_{i-1} \cup \{-i, +i\} \tag{13}$$

That means generally:

$$I_n = I_1 \cup (\bigcup_{i=2}^{i=n} (\{-i, +i\})) \tag{14}$$

Consider

$$D_i = I_i - I_{i-1} \tag{15}$$

We conclude then

$$D_i = \{-i, +i\} \tag{16}$$

Suppose that

$$R_i = (D_i \times I_i) \cup (I_i \times D_i) \tag{17}$$

As

$$(D_i \times I_i) \cap (I_i \times D_i) = \{(-i, -i), (-i, +i), (+i, -i), (+i, +i)\} \tag{18}$$

We have

$$card(R_i) = [(2 \times (2i+1)) + ((2i+1) \times 2)] - 4 \tag{19}$$

That means

$$card(R_i) = 8i + 4 - 4 \tag{20}$$

Finally

$$card(R_i) = 8i \tag{21}$$

As card(R_i)=8i, 8i new clusters have been performed by data research, for each iteration i. For instance:

- if i=2, data research will run on 16 clusters referenced by:
  * {(u-2, v-2), (u-1, v-2), (u, v-2), (u+1, v-2), (u+2, v-2)}
  * {(u-2, v+2), (u-1, v+2), (u, v+2), (u+1, v+2), (u+2, v+2)}
  * { (u-2, v-1) (u-2, v), (u-2, v+1)}
  * { (u+2, v-1) (u+2, v), (u+2, v+1)}
- if i=3, data research will deal with 24 clusters referenced by:
  * {(u-3, v-3), (u-2, v-3), (u-1, v-3), (u, v-3), (u+1, v-3), (u+2, v-3), (u+3, v-3)}
  * {(u-3, v+3), (u-2, v+3), (u-1, v+3), (u, v+3), (u+1, v+3), (u+2, v+3), (u+3, v+3)}
  * { (u-3, v-2), (u-3, v-1), (u-3, v), (u-3, v+1), (u-3, v+2)}
  * {(u+3, v-2), (u+3, v-1), (u+3, v), (u+3, v+1), (u+3, v+2)}
- finally if i=a, data research will take in entry 8a clusters referenced by:
  * {(u-a, v-a), (u-(a-1), v-a), ..., (u, v-a), (u+1, v-a),..., (u+(a-1), v-a), (u+a, v-a)}
  * {(u-a, v+a), (u-(a-1), v+a), ..., (u, v+a), (u+1, v+a),..., (u+(a-1), v+a), (u+a, v+a)}
  * {(u-a, v-(a-1)) (u-a, v-(a-2)),...., (u-a, v), (u-a, v+(a-2)), (u-a, v+(a-1))}
  * {(u+a, v-(a-1)) (u+a, v-(a-2)), ..., (u+a, v),(u+a, v+(a-2)), (u+a, v+(a-1))}.

The coordinates (u, v) of any cluster localized in a region must verify the constraints $\begin{cases} x_{min} \leq u \leq x_{max} \\ y_{min} \leq v \leq y_{max} \end{cases}$ where $x_{min}, x_{max}, y_{min}$ and y_{max} are constants corresponding to the coordinates of clusters in the extremities of the region.

We have introduced an algorithm to create a list of cluster as follows in Algorithm 2. This algorithm takes as parameters an integer and the Table II which has its extremities coordinates defined by the constraints $\begin{cases} x_{min} \leq u \leq x_{max} \\ y_{min} \leq v \leq y_{max} \end{cases}$

The global algorithm illustrated in Algorithm 1 shows more details on the main steps of data research: it generates in each iteration a list of cluster through the method getClusterofPackage(i, m) which content is given by the Algorithm 2.

We recall that in both Algorithms 1 and 2 the coordinates (u,v) verify $F_0(x_O, y_O) = (u, v)$.

Thus, the clusters in entry to the map-reduce algorithm have been analyzed successively by iterations, as presented in Algorithm 1. Each iteration is associated to a new package.

As a conclusion, we have studied the neighbors packages, in considering 8-neighborhood based on the distance of Tchebychev. But others distances such as the Euclidean distance leading to digital disks, should be explored precisely in perspectives.

But, the concepts of distance are limited in analysis of the nearest database sites, because they might be influenced by barriers or obstacles in the environment, depending on effective presence of roads described in Geographic Information System (GIS). In reality, the distance between a point M and a database site may be short and separated by a mountain. The future works will study on how to take the presence of roads into account.

Now, the remaining question is to search precisely data in the content of each cluster.

D. Map-Reduce Algorithm and Application

This section concerns the definitions of map-reduce content, in explaining how to extract data of database sites through

Algorithm 1: GlobalAlgorithm(Object O, Table m)

```
Result: a list of data found in the form list $< data >$
d : Data ;
i : integer ;
b: boolean ;
l: list$< Cluster >$;
r: list$< Data >$ ;
l.add(Cluster(u, v));
l.addList(getClusterofPackage(1, m);
i=1;
b=False;
while (not empty(l) and b==False) do
    r =Map-reduceAlgorithm(l);
    if not empty(r) then
        b=True ;
    end
    else
        i=i+1;
        l.clean();
        l.addList(getClusterofPackage(i, m);
    end
end
if (b==True) then
    return r ;
end
else
    return NULL ;
end
```

a restricted list of cluster instead of taking all the clusters in the region.

The reference of a cluster is giving by the pair (u, v) that corresponds to the coordinates of its center or its GPS coordinates, transformed.

Firstly, the entries of the function map are the clusters of the package 1 represented, respectively by (u, v), $(u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4), (u_5, v_5), (u_6, v_6)$ $(u_7, v_7), (u_8, v_8)$ in relation in 8-neighborhood.

Due to the image size and then the cluster size, each cluster reference must verify the constraints: $\begin{cases} x_{min} \leq u_i \leq x_{max} \\ y_{min} \leq v_i \leq y_{max} \end{cases}$

For new data research, regarding to the value 8i increasing, the number of map function could change dynamically to face scalability with more users connected on the system.

Moreover, due to using a regular grid, the database will present empty clusters which contain data from any database sites: that means no database sites in these clusters. But, these empty clusters give information about the need of installing database sites in these regions.

Our method considers each map function taking one cluster in entry. Algorithm 3 shows exactly the content of the map function.

But the future works might concern the case of several clusters or the whole package as entries, being analyzed by one map function.

Algorithm 2: getClusterofPackage(Int a, Table m)

```
Result: a list of Cluster as in the form list
        $< Cluster >$
c : Cluster ;
j : integer ;
l: list$< Cluster >$;
l.clean() ;
for (j=-a; j $\leq$ a; j++) do
    if ($x_{min} \leq u + j \leq x_{max}$ and
     $y_{min} \leq v - a \leq y_{max}$) then
        c=new Cluster(u+j, v-a);
        l.add(c);
    end
end
for (j=-a; j $\leq$ a; j++) do
    if ($x_{min} \leq u + j \leq x_{max}$ and
     $y_{min} \leq v + a \leq y_{max}$) then
        c=new Cluster(u+j, v+a);
        l.add(c);
    end
end
for (j=-a+1; j $\leq$ a-1; j++) do
    if ($x_{min} \leq u - a \leq x_{max}$ and
     $y_{min} \leq v + j \leq y_{max}$) then
        c=new Cluster(u-a, v+j);
        l.add(c);
    end
end
for (j=-a+1; j $\leq$ a-1; j++) do
    if ($x_{min} \leq u + a \leq x_{max}$ and
     $y_{min} \leq v + j \leq y_{max}$) then
        c=new Cluster(u+a, v+j);
        l.add(c);
    end
end
return l ;
```

Algorithm 3: function map(Doc idcluster, Doc value)

```
Result: the pairs in the form $(k, v)$
d : Data ;
if Verification(d, value ) then
    information←getInformationSite(value) ;
    emit(idcluster, information);
end
```

The variable value represents the content of a cluster and contains on each line, the name and the data of a site, corresponding to the data of the column name "data in sites" in the table I. The output of a mapper is for instance the set of pairs $(id, info_1), (id, info_2)$, with the same id as a reference for a cluster. Each mapper works on different clusters.

The shuffle phase is automatic as illustrated in the Table V : it consists of putting together all the pairs issued by the map functions. That means:

- for the cluster id_1 in entry: the pairs $(id_1, info_{11})$, $(id_1, info_{12})$ becomes $(id_1, < info_{11}, info_{12} >)$;
- for the cluster id_2 in entry: the pairs

$(id_2, info_{21}), (id_2, info_{22})$ returns $(id_2, < info_{21}, info_{22} >)$

- for the cluster id_3 in entry: the pairs $(id_3, info_{31}), (id_3, info_{32})$ becomes $(id_3, < info_{31}, info_{32} >)$.

TABLE V. THE SHUFFLE PHASE

Pairs issued by the map functions (in entry)	Results after the shuffle phase
$(id_1, info_{11})$, $(id_1, info_{12})$	$(id_1, < info_{11}, info_{12} >)$
$(id_2, info_{21})$, $(id_2, info_{22})$	$(id_2, < info_{21}, info_{22} >)$
$(id_3, info_{31})$, $(id_3, info_{32})$	$(id_3, < info_{31}, info_{32} >)$

The reduce functions will work on the pairs in the Table V resulting of the shuffle phase, as illustrated in Algorithm 4.

```
Algorithm 4: function reduce(Docid idCluster, Iterator value)
  Result: the pairs in the form
          $(k, < v_1, v_2, ..., v_{n-1}, v_n >)$
  result : String ;
  result ← ””;
  for each $v_i$ in value do
      result= result+” ”+v ;
  end
  emit(result);
```

Finally, the outputs of all the reduce functions will give three groups of data such as, $info_{11} + info_{12}$, $info_{21} + info_{22}$, $info_{31} + info_{32}$, representing information related to the presence of the value d.

IV. SIMULATION AND DISCUSSIONS

This section deals with method applications in referring to drugs management in pharmacies as the case study of the problem specification.

Consider data about drugs in pharmacies, spread over the territory. The question is to find out the nearest pharmacies (or drugstores) accordingly to the initial position of an Object O with coordinates (x_O, y_O), requesting data research in a region.

The number of clusters 8i will increase easily in each iteration i (for instance for i going 1 to n): the value 8i corresponds to a linear function that will lead to more clusters with i increasing. If the object O is localized in a cluster in one of the extremities of the region, the worst case will consist of finding out data related to the value d in the others extremities of the region. Consider $F_0(x_O, y_O)$ =(0,0), the corresponding cluster reference.

It will be better that the structure of cluster references in the database follows 8-neighborhood, to facilitate data research in the database and to allows speedily the response to any request. The relation of 8-neighborhood between two clusters is obviously reflexive and symmetric. Thus, two clusters linked by 8-neighborhood might be juxtaposed in the database structure. that will be of course difficult because the position of the object O is not definitively fixed for all the requests sent by users. But to overcome this, we have built a cluster list (precisely a type list $< Cluster >$), in doing research on cluster references in the memory Table II, as illustrated in Algorithm 1 with the function getClusterofPackage(i, m).

Moreover, there are several nosql databases such as Mongodb, Hbase, Cassandra, CouchDB, Couchbase, Neo4j, OrientDB, Oracle Graph, and Big table in [22] that might be used to store GIS data, as mentioned in the previous table I. But, here we have decided to use Mongodb as the database layer and to connect finally spark on Mongodb to realize implementation of the map-reduce algorithm.

Simulation considers a NoSQL database under the spark layer and is based on a computer with the following characteristics:

- the layers spark, NoSQL database (Mongodb) installed in localhost;
- memory: 3,7 GB
- processor: Intel® Celeron(R) CPU B830 1.80 GHz × 2
- graphic card: Intel® HD Graphics 2000
- operating system: Ubuntu 18,04 LTS 64 bits

Consider cluster references and site references in the Table VI, specializing the Table I, and saved in a mongodb database.

Moreover, we establish a link between data in the Table VI and its associated region in the Table VII through the foreigner key “region name”: the database will contain only two tables.

Cluster references and database sites references are obviously fixed, like the region name and its image.

Dataset is constituted by data extracted from the list of drugs from Hospitals in Cameroon in [21]. Generally, pharmacies provide the same drugs. Redundancy on the drugs names and its specification area will appear in the database. Then, we have decided to repeat data in the Table VI for different clusters to have more than 100 tuples: here, the Table VI is just a sample of real data in the database.

TABLE VI. TABLE OF DRUGS

Region name	**Cluster Ref**	**Site Ref**	**Pharmacy**	**Drugs**	**Specif**	**quantity**
belle-ville	(0, 0)	(0,1)	soudia	amoxiline	500mg	40
belle-ville	(0,0)	(0,1)	soudia	amoxiline	400mg	40
belle-ville	(0,0)	(0,1)	soudia	amoxiline	300mg	40
belle-ville	(0,0)	(0,1)	soudia	amoxiline	200mg	40
belle-ville	(0,0)	(0,1)	soudia	amoxiline	100mg	40

TABLE VII. TABLE OF REGIONS

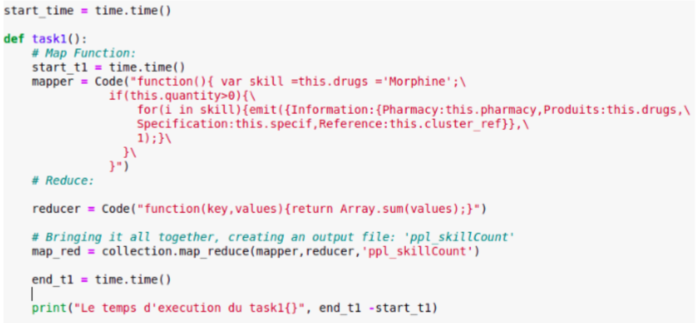
Region name	**images**
belle-ville	image blob

The architecture used for implementation consists of three layers (mongodb, spark and python application) as defined by the Table VIII, specializing the layers, proposed in the Table III.

A spark connector allows connection between spark and mongodb to get dataframes from mongodb for visualisation in python application. A sample of codes in python is implemented in Fig. 7 and gives for instance the following results:

TABLE VIII. Architecture with a Nosql Database

Python application
Spark
Mongodb



```
start_time = time.time()

def task1():
    # Map Function:
    start_t1 = time.time()
    mapper = Code("function(){ var skill =this.drugs ='Morphine';\
                if(this.quantity>0){\
                    for(i in skill){emit({Information:{Pharmacy:this.pharmacy,Produits:this.drugs,\
                    Specification:this.specif,Reference:this.cluster_ref}},\
                    1);}\
                  }\
                }")
    # Reduce:

    reducer = Code("function(key,values){return Array.sum(values);}")

    # Bringing it all together, creating an output file: 'ppl_skillCount'
    map_red = collection.map_reduce(mapper,reducer,'ppl_skillCount')

    end_t1 = time.time()
    |
    print("Le temps d'execution du task1{}", end_t1 -start_t1)
```

Fig. 7. A Sample of Codes in Python

{{“Information”: {“Pharmacy”: “Escale”, “Produits”: “Morphine”, “Specification”: “500mg/ml”, “Reference”: “(0,1)”}}, “value”: 33.0}.

We are interested in time evaluation between the sequential research step by step on selected clusters near the localized cluster and the parallelism research simultaneously on the same clusters. As mentioned in the data Table IX and its related Fig. 8, experimental results show a large difference between the curves of sequential execution and parallel execution in considering the time taken for data research and the number of clusters: The parallel execution of different clusters with the map-reduce algorithm brings interesting improvement of processing time, giving possibilities of speed responses to users.

We notice that, in Table IX and in Fig. 8, time is evaluated on each group, defined by the same drug name: that means each $group_i$ corresponds to an unique $(drugname)_i$.

TABLE IX. Time Evaluation in Millisecond (ms) with a NoSQL Database

Group N°	1	2	3	4	5	6	7	8	9	10
Clusters found	10	20	30	40	50	60	70	80	90	100
Sequential	0.4994	0.524	0.5409	0.58790	0.7335	0.8577	0.9989	1.132	1.2859	3.5537
Parallel	0.4897	0.5294	0.565	0.5004	0.4906	0.4831	0.4898	0.5099	0.5205	0.5333

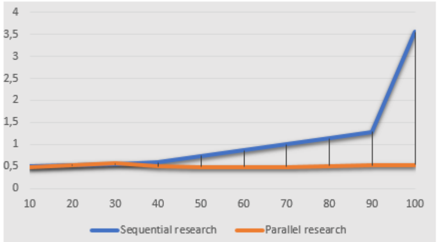


Fig. 8. Time Evaluation (in millisecond) Through the Curves of Sequential Research and Parallel Research with Spark

On the other hand, a newsql database could be substitued to a nosql database, to integrate performance of newsql databases such as viz, VoltDB, MemSQL and NuoDB. Because, newsql databases conserve the power of nosql databases such as horizontally scalable, highly available and take into account ACID properties, SQL support for SQL databases, accordingly to Irina Astrova and others in [20]. In this case, implementation will consider the layers MemSQL, Spark and Python application: spark will be connected on MemSQL through a connector. Details on comparisons with time evaluation between MemSQL and Mongodb will be studied in perspectives.

V. Conclusion and Perspectives

A GIS database structure has been proposed in taking into account a meshing technique based on a quasi affine application in order to get a regular grid and to identify clusters. A NoSQL database table has been established as the implementation of data clustering. Proposed data research uses a limited number of clusters in entry to the map-reduce algorithm, to improve processing time.

Experimental results reveal effectively an improvement of processing time with the parallel execution on selected clusters around the central cluster through the map-reduce algorithm than the sequential execution on the same clusters.

In perspectives, the remaining questions will concern others meshing techniques to create new clusters and to undertake new concepts related to neighborhood through establishment of distance definitions and in taking others criteria such as modelization of presence of roads near the region.

The regular grid leads to empty clusters with no data inside. In the future works, as an alternative to alleviate this problem, we will study the case of irregular grids adapted to real data in the database sites to eliminate empty clusters in the central database. We will analyze strategies to transform a regular grid to an irregular grid to avoid empty clusters.

Comparisons between the layers Mongodb and MemSQL through spark connector and python application will be analyzed with time evaluation to determine the best alternative in data research.

The future works will explore in more details the applications of this framework to others fields such as Finance, Education and Shopping.

Acknowledgment

The authors would like to thank the Ministry of Numeric Economy and Post (MDENP) for finding relevance in using big data opportunities for digital transformation of public administrations through several events such as FNDSI 2020 (Forum National des DSI) and SN 2020 (Semaine du Numérique 2020). These events focused on artificial intelligence, big data and cyber-security. This paper is regarding to these topics.

References

[1] Yaya TRAORE, Malo SADOUANOUAN, Didier BASSOLE, Abdoulaye SERE, *Multi-Label Classification using an Ontology*, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 10, No. 12, 2019

[2] Marie-Andree and Jacob-Da Col, *Applications quasi-affines et pavages du plan discret*, Theoretical Computer Science vol 259 page 245-269, 2001

[3] Abdoulaye SERE, *Transformations analytiques appliquées aux images multi-échelles et bruitées*, thèse de doctorat unique en informatique, Université de Ouagadougou, 2013

[4] Antoine VACAVANT, *Géométrie discrète sur grilles irrégulières isothétiques*, thèse de doctorat en informatique, Université Lumière Lyon 2, 2007,

[5] Abdoulaye SERE and Dario Colazzo and Oumarou SIE, *A Hough Transform based on a Map-Reduce Algorithm*, International Journal of Engineering Research and Applications (IJERA), 2016

[6] Jimmy Lin, *MapReduce Algorithm* Design, Tutorial, Rio de Janeiro, 2013

[7] Jairam Chandar, *Join Algorithms using Map-Reduce*, Master of Science, Computer Science School of Informatics, 2010

[8] Jesus Camacho-Rodriguez and Dario Colazzo and Ioana Manolescu, *PAXQuery : Efficient Parallel Processing of Complex XQuery*, IEEE, 2015

[9] Zeba Khanam and Shafali Agarwa, *Map-Reduce implementations : survey and performance comparison*, International Journal of Computer Science and Information Technology (IJCSIT), volume 7 , issue 4, 2015.

[10] J. Dean and S. Ghemawat, *Map-reduce: simplified data processing on large clusters*, Commun. ACM, volume 51, issue 1, page 107-113, 2008.

[11] Abdulrahman M. Qahtani, Bader M. Alouffi, Hosam Alhakami, Samah Abuayeid, Abdullah Baz, *Predicting Hospitals Hygiene Rate during COVID-19 Pandemic*, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 12, 2020.

[12] World Health Organization (WHO), *la qualité des services de santé : Un impératif mondial en vue de la couverture santé universelle*, ISBN 978-92-4-251390-5 OMS.

[13] OCDE, *Caring for quality in health: lessons learnt from 15 reviews of health care quality*, Éditions OCDE, Paris, http://dx.doi.org/10.1787/9789264267787-en, 2017.

[14] Arah, O.A., G.P. Westert, J. Hurst et N.S. Klazinga, *A conceptual framework for the OECD Health Care Quality Indicators Project* , International Journal for Quality in Health Care, Suppl. 1, pp. 513, 2006.

[15] David COEURJOLLY, *algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*, thèse de doctorat en informatique, université Lumière Lyon2, 2002.

[16] Zhi-Qiang Feng, Zhengqun Guan, Zhuowei Chen. *FER/Mesh: un logiciel de génération automatique de maillages*, 9e Colloque national en calcul des structures, CSMA, May 2009, Giens, France. hal- 01413781

[17] Vincent François, *Méthodes de maillage et de remaillage automatiques appliquées à la modification de modèle dans le contexte de l'ingenierie simultanée*, thèse de doctorat, Université Henri Poincaré - Nancy 1, 1998

[18] Abdoulaye SERE, José Arthur OUEDRAOGO, Boureima ZERBO, Oumarou SIE, *Post Classification in the Social Networks using the Map-reduce Algorithm*, (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 11, No. 12, 2020

[19] Mateus Coelho ; Dylan Sugimoto ; Gabriel Melo ; Vitor Curtis andJuliana Bezerra, *A MapReduce based Approach for Circle Detection*, In Proceedings of the 14th International Conference on Software Technologies - Volume 1: ICSOFT, 454-459, Prague, Czech Republic, 2019.

[20] Irina Astrova, Arne Koschel, Nils Wellermann, Philip Klostermeyer, *Performance Benchmarking of NewSQL Databases with Yahoo Cloud Serving Benchmark*, Springer Nature Switzerland AG 2021, K. Arai et al. (Eds.): FTC 2020, AISC 1289, pp. 271–281, 2021.

[21] *Liste nationale des médicaments et autres produits pharmaceutiques essentiels*, Cameroun, 2017.

[22] Moubaric KABORE, *Application du framework map-reduce à des groupes de données massives*, mémoire de master en Informatique, Université Joseph KI-ZERBO, 2017.