Streaming of Global Navigation Satellite System Data from the Global System of Navigation

Liliana Ibeth Barbosa-Santillán[1]
Computer Science Department
University of Guadalajara
Guadalajara, México

Juan Jaime Sánchez-Escobar[2]
Research department
Technical and Industrial Teaching Center
Guadalajara, México

Luis Francisco Barbosa-Santillán[3]
Mechatronics Department
University Technological of Puebla
Puebla, México

Amilcar Meneses-Viveros[4]
Computer Science Department
CINVESTAV,México City

Zhan Gao[5]
School of Computer Science and Technology
Nantong University, Nantong, China

Julio César Roa-Gil[6]
Computer Science
ITESM, Guadalajara, México

Gabriel A. León-Paredes[7]
GIHP4C Research Group
Universidad Politécnica Salesiana
Cuenca, Ecuador

***Abstract*—The Big Data phenomenon has driven a revolution in data and has provided competitive advantages in business and science domains through data analysis. By Big Data, we mean the large volumes of information generated at high speeds from various information sources, including social networks, sensors for multiple devices, and satellites. One of the main problems in real applications is the extraction of accurate information from large volumes of unstructured data in the streaming process. Here, we extract information from data obtained from the GLONASS satellite navigation system. The knowledge acquired in the discovery of geolocation of an object has been essential to the satellite systems. However, many of these findings have suffered changes as error vocalizations and many data. The Global Navigation Satellite System (GNSS) combines several existing navigation and geospatial positioning systems, including the Global Positioning System, GLONASS, and Galileo. We focus on GLONASS because it has a constellation with 31 satellites. Our research's difficulties are: (a) to handle the amount of data that GLONASS produces efficiently and (b) to accelerate data pipeline with parallelization and dynamic access to data because these have only structured one part. This work's main contribution is the Streaming of GNSS Data from the GLONASS Satellite Navigation System for GNSS data processing and dynamic management of meta-data. We achieve a three-fold improvement in performance when the program is running with 8 and 10 threads.**

Keywords*—GLONASS; streaming; extraction; satellites data; observation files; metadata

I. Introduction

The data collection does not present a problem. However, handling these volumes of information poses a challenge to the industry. The fundamental challenge regarding large volumes of data from different sources is identifying new uses that have not been found. Companies' challenge is to develop methods of realizing the real value of this mine of terabytes of data. Big Data is the medium through which these large volumes of information acquire significant value [1] [2].

Many types of problems can occur during the different stages or processes involving Big Data. For example, in the transformation step, the storage process and the extraction process in streaming from other sources, there are significant challenges related to linking variables such as the speed, volume, and variety of data extracted and processed. Similar effects arise in the preprocessing of the data. The hardware capability plays a vital role in a system's ability to clean data in the shortest possible time. When analyzing and visualizing information, it should be presented in the easiest and simplest possible way so that anyone can understand it. One challenge in this vein is the use of techniques and methodologies that summarize and display information clearly and accurately [2] [3] [4].

The Global Navigation Satellite System (GNSS) combines several existing systems for navigation and geospatial positioning, including the Global Positioning System (GPS), GLONASS (Global System of Navigation), and Galileo (a European radio-navigation program) [5] [6].

GPS was the first system GNSS. It was released at the end of 1970 by the Department of Defense of the United States; it uses a constellation between 24 and 32 satellites and provides global coverage. The Ministry of Defense of the Russian Federation operates GLONASS; this consists of 31 satellites. Twenty-four are active; three are in backup, two in maintenance, and two more in testing. GALILEO is the European radio navigation and satellite-positioning program developed by the European Union in conjunction with the European Space Agency and expected to be officially available for civil use by 2020. In November 2016, four new satellites launched, giving 18 satellites already in orbit. These systems are composed of Space-Based Augmentation Systems (SBAS) or Ground-Based Augmentation Systems (GBAS). Examples of SBASs are the US-based Wide-Area Augmentation System (WAAS), the European Geostationary Navigation Overlay Service (EGNOS), and the Japanese Multi-functional Transport Satellite (MTSAT) based on SBAS. The GNSS signal radio is involved with frequencies close to 1.5 GHz (1.5 billion

cycles per second). GNSS signals operate at higher frequencies than FM radio signals but lower than those of a microwave oven; when GNSS signals reach land, they are fragile. An electromagnetic wave arriving from space must traverse three distinct zones before going a receiver on the Earth's surface: the vacuum, the ionosphere, and the troposphere. The signal delay increases with the propagation of time.

It arises from two factors: the propagation speed and the increase in the trajectory's length due to bending by refraction. In the vacuum, this delay is negligible, and is the programming time proportional to the distance depending on the light, whatever the frequency of the wave is. In the ionosphere (at altitudes of 100 to 1000 Km), ultraviolet, solar, and other radiation types, ionize gaseous molecules and release electrons. The number of free electrons per cubic meter varies between 10-16 and 10-19. The delay is proportional to the number of free electrons encountered by the signal along its path and is dependent on the inverse of the square of wave frequency. It varies for each particular point, according to its latitude, direction, and observation moment. The delay may change in the zenith by between 2 ns. and 50 ns. For frequencies in the L-band, the delay can reach up to 2.5 the factor due to the trajectory's inclination. Its effect at midday is up to five times between midnight and dawn. The last area that the wave traverses is the troposphere and the other regions of the upper atmosphere. Although this area extends to heights of up to 80 km, significant delays are incurred only in the lower 40 km. This delay corresponds to increments in the distance of the order between 1 m at the zenith and up to 30 m and five elevation grades (advanced GPS). GNSS systems continuously transmit signals at two or more frequencies within the L band. These signals contain range codes and navigation data. The main components of the signal are:

- A carrier, a sinusoidal radio signal, is a specific frequency.
- A ranging code consists of sequences of zeros and ones that allow the receiver to determine the Satellite radio signal's travel time to the receiver. These are called PRN sequences or PRN codes.
- Navigation data consists of a message that provides information on the satellite's ephemeris (pseudo- Keplerian Elements or the satellite's position and velocity), clock parameters, and error margins (a set of low-precision ephemeris data), the satellite status, and additional information.

Frequency-band mapping is a complex process since multiple users and services can access the same range. In other words, the same frequencies are for different purposes in different countries. The International Telecommunication Union (ITU) is a United Nations agency that coordinates the radio spectrum's shared global use. ITU divides the electromagnetic spectrum into frequency bands, with different radio services assigned to particular bands. Two band segments are given to the Aeronautical Radio Navigation Service (ARNS) at the primary level world- wide. These bands are for the safety of life (SoL) applications, and no other use of these bands can interfere with GNSS signals. These segments are the upper L band (containing the GPS bands $L1$, Galileo $E1$, GLONASS $B1$, and Beidou L), and the lower L band (including the $L5$ band of GPs, $G3$ of GLONASS, $E5$ of Galileo, and B2 of Beidou). Receiver Independent Exchange Format (RINEX) was developed by the Institute of Astronomy at the University of Bern to enable the exchange of the GPS data collected during the Europe Reference Frame (EUREF), which included more than 60 GPS receivers from four different manufacturers. In the development of this format, it was taken into account that most software for geodesic processes for GPS data use a well-defined set of observables, including:

- Measurement of the carrier signal phase is a measure of the receiver's satellite carrier signal frequency, as shown in Equation 1.

$$\begin{aligned} Phase(tight) = Phase(r) \\ -RealTime(r) \times frequency \end{aligned} \tag{1}$$

  where: r = clock

- Measurement of the pseudo-range is the difference in the reception time (expressed in the receiver's time frame) and the transmission time (described in the satellite's temporary framework) of a different satellite signal.

$$\begin{aligned} PR = distance + c \times Shiftingreceiverclock \\ +Satelliteclockshifts + Otherbiases \end{aligned} \tag{2}$$

  where: PR = PseudoRange and c = cycles

- The observation time is reading the receiver's clock at the moment of the phase carrier's validity and code measurements.

Version 3 of the RINEX format consists of three types of ASCII files: an observation data file, a navigation message file, and a meteorological data file [7].

Each file type contains a header section and a data section. The header section contains global information for the file at the beginning of it. This header section contains labels in columns 61-80 for each line in the area; these tags are mandatory and must appear as required by the format. RINEX requires a minimum amount of space, regardless of the number of different observables, the specific receiver used, or the satellite system. It indicates in the heading the types of observations recorded by each receiver and satellite system observed. There is not a maximum length for each record to limit these observations. Each meteorological data and observation file contains data from a site and a session. In RINEX version 3, navigation message files can contain messages from more than one satellite system (e.g., GPS, GLONASS, Galileo, or SBAS). GNSS observables require two fundamental quantities to be defined: The time and phase.

The timeof measurement is the time recorded by the receiver of the signals. It is similar for phase and range measurements and is identical for all satellites observed. For single system data files, expressed by default in the respective satellite's time system: otherwise, the actual time (for mixed files) is in the start time header log.

Phase involves the carrier wave and its complete cycle measures. The semi-cycles measured by quadrant-type receivers must be converted into full cycles and marked with the respective observation code—the phase changes in the same direction as

the range (a negative Doppler effect). Observable ones are incorrect for external influences such as atmospheric refraction and satellite offsets. Phase changes between phases of the same frequency but tracked in a different carrier channel are not corrected.

The knowledge acquired in the discovery of geolocation of an object has been essential to the satellite systems. However, many of these findings have suffered changes in error localization and many data. The Global Navigation Satellite System (GNSS) combines several existing navigation and geospatial positioning systems, including the Global Positioning System, GLONASS, and Galileo [8]. We focus on GLONASS because it has a constellation with 31 satellites.

The motivation of the proposed work is to extract information in real-time based on the Glonass positioning system. Research gaps consist of the GLONASS navigation file defines the orbits of the satellites by their coordinates inserted from the central bases at certain times and indicating the age of said information.

The definition of GLONASS time has also given its problems, being necessary to indicate the origin of the observations' reference time. On average, the navigation files consist of 150 lines and the observation files of 33,500 lines. The RINEX observation files could contain a receiver-derived clock offset.

The data (epoch, pseudo interval, phase) have been previously corrected or not for the reported clock shift. RINEX Versions 2.10 onwards requests a clarifying header record: RCV CLOCK OFFS APPL. Then it would be possible to reconstruct the original observations, if necessary.

Our research's difficulties are (a) To handle the amount of data that GLONASS produces efficiently and (b) to accelerate data pipeline with parallelization and dynamic access to data because these have only structured one part. This work's main contribution is the Streaming of GNSS Data from the GLONASS Satellite Navigation System for GNSS data streaming processing and dynamic management of metadata implemented within the database. We achieve a three-fold improvement in performance when running the program with 8 and 10 threads. Our research questions are as follows:

P1 Is it possible to automatically identify and download RINEX data sources from GLONASS?

P2 How can RINEX files be identified on semantics?

Our research hypothesis is as follows:

H1 It is possible to discover RINEX files in GLONASS based on semantics.

The paper is structured as follows: Section 1 a brief theoretical framework. Section 2 describes related work. Section 3 discusses stream extraction of GNSS data based on the GLONASS satellite navigation system. Sections 4 and 5 present the details of our data sets, evaluation metrics, and our results. Finally, Section 6 presents the conclusions.

II. Related Work

Several works related to this research are:

- GLONASS data stream processing,
- a parallelization mechanism for the ETL module, and
- Managing dynamic structures in a database for Big Data tasks in GLONASS for data mining and satellite data processing.

From the perspective of stream processing in GLONASS, prior works have focused on positioning and kinematic processing through GPS. Some examples of these are studied by Li et al. [9], Wang et al. [10], and Rieke Matthes et al. [11]. Some of these works have been applied to atmospheric measurements [12], [13], in which the main idea is to provide accurate positioning in real-time with minimum error. These systems have evolved to establish services at the cloud computing level, as reported by Karimi et al. [14] and Liu et al. [15]. Several approaches are to optimize the ETL module. These works include an optimization involving the environment (a distributed system), a dynamic design, and the ETL module's parallelization. The ETL module's optimization process allows processing times and efficiently designing the data warehouse structure. In [16] the authors combine GPS, GLONASS and Galileo in order to obtain precise points positioning in real-time. In [17], Koyptov et al. present a system based on data stream collection and processing to determine the geographic coordinates of Earth's ionospheric regions. Kakooei and Tabatabaei [18] develop a hybrid-heterogeneous parallel GPS acquisition algorithm working with a GPU and a multicore CPU.

Distributed parallel architectures proposed in which the ETL module works in the stream with large data volumes. An example of this is the works of Agrawal et al. [19], Boja et al. [20], Ding et al. [21], who focus on the distributed file systems and the ETL module operating in the distributed environment. Bala et al. [22] present a distribution model in order to get fine-grained data for the ETL process.

There are several works on the parallelization of the ETL module. Xiufeng et al. [23], and Radonić et al. [24] present a programming framework that uses Map-Reduce to achieve scalability. This framework is called ETLMR, and it is on a data warehouse; it constructs star schemes and snowflakes and works in dimensions that change over time. This work is evolving to include cloud computing support through the CloudETL framework [25]. In Bala et al. In [26], develop P-ETL, an ETL module that operates on a data warehouse. It runs in parallel in a cluster under the MapReduce paradigm. Masouleh et al. [27] develop an optimization for the execution time of the ETL module using parallelization methods in the shared memory cache of the distributed system. Thomsen et al. [28] propose a framework that allows the parallelization of the ETL module in terms of the three phases that it involves (extraction, transformation, and load). The framework parallelize s the task level as the data level, depending on the stage worked on. This framework works on a node with a multicore processor. Diouf et al. [29] give a review of several speedup ETL process methods.

The use of dynamic structures in the database is implemented in the last few years; the main idea is to handle various types of data transparently. In Big Data, this can help by adapting the stream's structure to incorporate the data sent

for analysis. Han et al. [30] survey the different database technologies for handling large volumes of data and high-performance techniques for cloud computing tasks in real-time. Wu et al. [31] examine the fundamentals of the dynamic characteristics required of a data model for Big-Data tasks. Ji et al., [32] and Fan et al. [33] present a review of the Big Data schemes of the different phases, emphasizing the use of dynamic schema in the database.

Currently, the vast amount of information stored in organizations in all market sectors represents a potential source of knowledge that can be explored and extracted. Positioning satellites is a significant source of information for various companies, especially those working on research geospatial data [34].

Satellites accumulate records in the form of telemetry data points. The telemetry frames format records several hundreds of thousands of data points at each time step, stored in knowledge bases for analysis. These points formed the basis of accumulated historical data and were extracted to give relevant information. Evidence of this includes the study of telemetry satellite data [35] using data mining processes, identifying and categorizing the parameters carried out within the data warehouse, where the data are prepared (standardized and reformatted). After processing, these data are metadata tags, through which experts and users can find categorized information of interest.

In satellite data applications, we know that telemetry data are the only source from which to identify and predict anomalies in artificial satellites. Although there are people who specialize in analyzing these data in real-time, these datasets' large size makes this analysis extremely difficult. Therefore, clustering algorithms are applied to help traders and analysts perform the task of analyzing the telemetry data [36].

Two real cases of anomalies in satellites on space missions are in Brazil. It was possible to evaluate and compare the effectiveness of the two clustering algorithms of K-means and Expectation-Maximization (EM). Their effectiveness was in several telemetry channels, which tended to include outliers; in these cases, they could support satellite operators, allowing for the anticipation of anomalies. However, for silent problems, in which there was only a small variation in a single channel, the algorithms were less efficient. Current cyclone detection techniques and monitoring through models and field measurements do not provide truly global coverage, unlike remote satellite observations. However, it is impractical to use a single satellite orbit to continuously detect and monitor these events due to the limited spatial and temporal coverage.

One solution to alleviate this problem is to use data sensors on multiple orbiting satellites. This approach addresses the unique challenges associated with knowledge discovery and mining of heterogeneous data stream satellites. It consists of two main components [37]: feature extraction from each sensor measurement to discover a set of cyclones, and knowledge sharing between the different remote sensor measurements, based on a linear Kalman filter, to track the predicted storms. Experimental results using historical hurricane data have demonstrated this approach's superior performance compared to other works. Other satellite television broadcasting applications have also been shown [38] (TV broadcasting). An improved algorithm has to identify the most frequent episodes over the broadcasting-satellite service. Frequent episodes at a specific scale of the alarm data extract to summarize the models obtained. Spatial data mining involves extracting implicit knowledge, spatial relations, or other patterns not stored in explicit form in the spatial databases. Based on this approach, the focus of spatial data mining is on deriving information from spatial datasets.

The geographic coordinates of the "hot spots" in the forest fire regions, extracted from satellite images, are studied and used to detect possible points or locations of fires [39]. It found that these applications may give false alarms. Thus, by comparing the brightness detected in several bands, this false information can be identified, and clustering and Hough transform are used to identify regular patterns in access points and applications classified as false alarms. This implementation demonstrates a spatial data mining application to reduce false alarms based on the set of points obtained from the images. Finally, it considers a data analysis based on data from positioning satellites. This work [40] develops an analytical real-time distributed environment in which analysis and simulation are closely coupled, integrating high-performance implementations of image mining run on dedicated servers. It was possible to simulate earthquakes at both the micro and macro levels based on images (Imageodesy) and historical data.

The header registers report the orientation of the antenna's zero direction and the direction of its vertical axis (hole view) if it is mounted and tilted at a fixed station. Header records can also be used for vehicle antennas. The comparison with other systems is closed since the manufacturers have commercial interests. However, the RINEX file is proven to arrive complete with INEX Viewer and RTKLIB and selected these tools because they are freely accessible.

III. Stream Extraction of GNSS Data based on the GLONASS Satellite Navigation System

In this section, the proposed framework for carrying out the stream analysis of data and its respective architecture are detailed.

The system's overall architecture for transfer and extraction of GNSS knowledge is into four main layers: external components, communication, software, and storage, as shown in Fig. 1.

- **External components**: These are all the elements of the physical system, such as GLONASS satellites, receiving antennas, control stations, and data broadcasters.
- **Communication**: This layer allows the transmission of data streams through the network.
- **Software**: consists of all software elements used and developed to extract, process, and store data.
- **Storage**: This layer consists of a logical meta-model and database, in which relevant information saves from downloaded data.

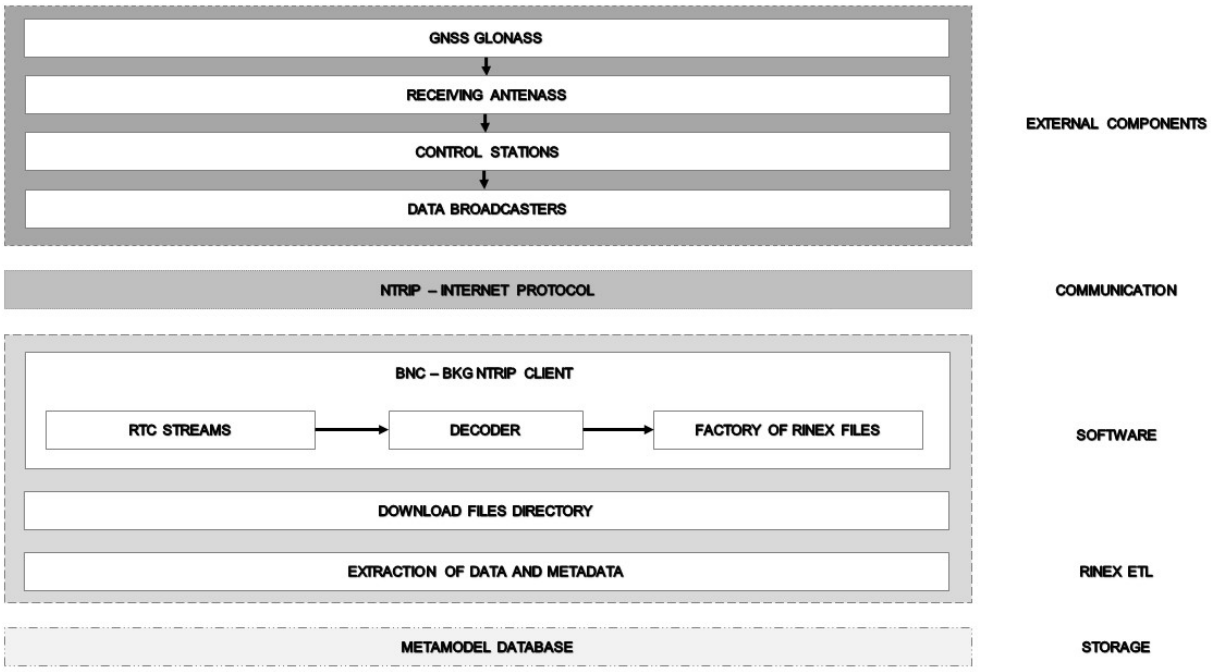

Fig. 1. Stream Extraction of GNSS Data based on the GLONASS Satellite Navigation Architecture.

1) External components: The elements of this layer of the architecture are below.

- **GLONASS:** This is a GNSS constellation with 31 satellites (24 active, three spares, two in maintenance, one in service, and one is undergoing testing) located in three orbital planes with eight satellites each. They produced the primary source from which the data in the RINEX files.
- **Receiving antennas:** These are antennas that receive signals from GLONASS data. They act as an intermediary between GLONASS communication and control stations.
- **Data control stations:** These are the stations in charge of the operation, control, and monitoring of GNSS, and data transmitted by GLONASS are stored here. Information is exchanged with GNSS if a synchronization or reconfiguration event then it is performed for any satellite.
- **Agencies/data broadcasters:** These are agencies, organizations, or institutions that collect, store, process, and investigate the data sent by any GNSS, and in turn relay this data over the Internet to anyone interested in the scientific investigation and processing of this information. The transmission of data takes place over civil frequencies that are open to the public.

2) Communication: This component involves the NTRIP protocol, which allows the transmission of the data streams generated by GNSS over the Internet to client software that receives the information.

3) Software components:

- **BKG Ntrip Client (BNC):** is one of the essential elements, and is a program that simultaneously retrieves, decodes, converts, and processes the data stream from any GNSS system in real-time. It also has some post-processing functions for the RINEX or SP3 files generated by the application. This client software is composed of three main elements, which are:
  - ***RTCM Streams*:** These are the main inputs and consist of data streams downloaded from the agencies or organizations that belong to different networks such as the International GNSS Service (IGS). These data streams arrive in RTCM format.
  - **Decoder:** This element's function is to decode the data streams arriving in the RTCM format and transform it to RINEX version 2.11.
  - **RINEX File Generator:** Once the decoder has completed the transformation, it is responsible for storing the RINEX files in the specified directory.
- **File Directory:** This is the backbone of the storage process for the client software. A well-defined structure is necessary to distinguish between the different types of files generated.
- **Extraction, Transformation, Load (ETL) Tool:**

TABLE I. GENERAL SPECIFICATIONS FOR TESTING

Item	Specifications
Number of files to process	40775
Size of data	80.9 Gigabytes
Maximum error percentage	2
Number of executions	3

TABLE II. GENERAL METRICS FOR TESTING

Metric	Specification
Number of files processed successfully	natural number
Number of processed files flawed	natural number
Execution time	seconds, minutes and hours

A developed specific software application to fulfill the purpose of extracting and transforming data. The data loaded into the database. This application is called RINEX ETL. This application's primary goal is to read the Observation Files and carry out the database's objective data's removal and insertion. The application can run in either serial or parallel mode using the processors in each core of the multiprocessor.

- **The Parser:** is the primary layer of the application and is responsible for reading, extracting, and transforming data from the Observation files. This layer implements the *Runnable* Containers interface, which enables parallel processing through the use of *threads*.
- ***Data Access Object (DAO)*:** This layer provides the standard interface between the application and the database (storage component), allowing communication between the storage and application components.

4) Storage: This section describes the database with a meta-model that is defined to store and retrieve relevant information.

IV. EXPERIMENTS

The objective of the experiments is to analyze and describe the meaning of RINEX files obtained from GLONASS. First, the specifications that apply to all the tests performed on the downloaded files' data are conducted. These specifications are listed in Table I.

The metrics used in the development of the experiments are in Table II.

The tests performed on the data were in four stages: sequential, parallel with multiprocessors, clustering, and queries.

1) **Sequential:** The RINEX ETL application was run in serial mode or with a single thread of execution.
2) **Parallel with multiprocessors:** The RINEX ETL application runs in parallel mode by using threads, making use of the multiprocessor cores. This test was performed with: 8 *Threads*, 10 *Threads*, 16 *Threads*, 32 *threads*, and 50 *Threads*.
3) ***Clustering*:** The application is linked to the database defined with the meta-model. After the database query process, the data loaded into the data mining tool. The clustering process to identify abnormalities in the LLI or find any patterns in the downloaded data from which we could infer and interpret possible improvements. The K-Means algorithm was applied because it is highly parallelizable. K-means was with four clusters.
4) **Queries:** Through the semantics, the following aims will be achieved:
   - Identification of the number of failures in the *Epoch Dates*.
   - Identification of the number of possible cycles slips for observation type $L1$.
   - Identification of the number of possible cycles slips for observation type $L2$.
   - Determination of the frequency of possible cycle slips for observation type $L1$.
   - Determination of the frequency of possible cycle slips for observation type $L2$.

The Rinex file observations are displayed, and it is possible to select one by one the observed satellite constellation in the measurements view, as shown in Fig. 2. If at any point an N appears, it means that the satellite is not observed.

V. RESULTS

This section presents the above experiments' results; we first describe the downloaded files in our sample and then report on the experimental results.
The distribution of the different files downloaded through the BNC-NTRIP client software is in Table III. The total size of the downloaded files was 95.5 GB.

Table IV shows the distribution of Observation Files between correct or readable files and corrupt or unreadable ones due to network or communication problems between the client and the broadcasters or electrical failures on the client-side.

Less than 1% of the files obtained were unreadable and excluded from the experiments. It was notable that at the end of the download time for the data, a longer run time was when implementing the application to process more data to reach a sizeable sample of data in less time.

TABLE III. DISTRIBUTION OF DOWNLOADED FILES

File types	Size (MB)
Observation Files	80,900
Ephemerids Files	105
Raw Data	14,500
Log Files	4,25

TABLE IV. DISTRIBUTION OF THE *Observation Files*

Item	Quantity	Percentage
Total files	40,792	100
Total correct files	40,775	99.96
Total corrupted files	17	0.04

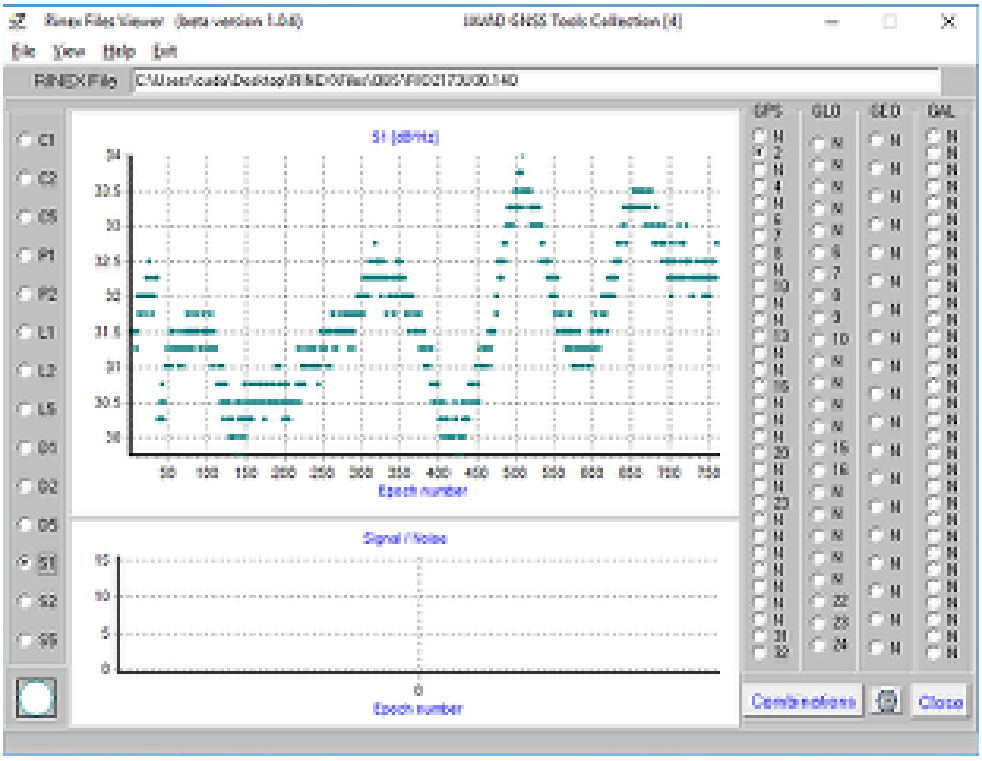

Fig. 2. Measurements view the Observations of the Rinex File.

The maximum number of observations with possible cycle slippage was 5,107,444 for $L1$ and 1,105,203 for $L2$, while the minimum values were 199,511 and 46,604 for $L1$ and $L2$, respectively.

Based on this information, we can infer that one of the more common abnormalities in the $L1$ and $L2$ types of observations is cycle sliding. It is not a failure of the observed value and merely tells us that we must perform a correction process to find the correct value observed. These corrections made using various methods that do not form part of this research.

The results are obtained in the various tests carried out here. The execution time results are described and graphically illustrated for different performance metrics such as CPU usage and memory. The container application, a tool that provides detailed information on containers running on the Virtual Machine (VM), was used to determine and monitor the different metrics.

Table V shows the number of rows saved into the METADATA database, OBSERVATION DATA, and LLI tables of the meta-model.

TABLE V. DISTRIBUTION OF RECORDS IN THE TABLES OF THE META-MODEL

Table	Quantity of records
METADATA	40,775
OBSERVATION_DATA	34,200,319
LLI	12,089,001

TABLE VI. POSSIBLE NUMBERS OF RECORDS WITH CYCLE SLIP, L1 AND L2

Observation type	Quantity of records
L1	2,517,286
L2	1,431,922

Table VI shows the number of records relating to LLI with a possible cycle slip in the $L1$ and $L2$ observation types stored in the LLI table. Within these records, no faults occurred between the epoch dates. It means that no event would alter the observed value when taking the time stamp from GNSS.

The test results for the clustering of values in the observation files for days in which a cycle slip occurred in the $L1$ and $L2$ observations types were as follows: 3,874,181 instances; 28 iterations; and 227.85 seconds to build the model.

Table VII details the epoch date at each of the available satellites according to the cycle's number on $L1$ and $L2$, respectively.

The values reported here correspond to the average of the observed values, as the percentage of instances. For example, there are 36 clustered instances in Cluster 2, as shown in Table VIII.

Table IX shows the clustering test results, in which the data grouped by days. The results were as follows: number of instances: 134, number of iterations: 5, and time is taken to build the model: 0.02 seconds.

Table X shows the percentage of clustered instances. For

TABLE VII. RESULTS, CLUSTER CENTROIDS

Attribute	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Epoch date	2017-04-30 22:35:13	2017-04-30 23:32:31	2017-04-30 22:16:12	2017-04-30 22:16:07
Available satellites	19.0311	15.6724	18.1718	15.7596
GLONASS satellites	8.1365	6.8342	8.3962	6.1779
GPS satellites	10.8946	8.8382	9.7746	9.5816
SBAS satellites	0	0	0.0005	0
Number of cycle slips on $L1$ and $L2$	55.6182	59.5535	18.9372	21.3757

TABLE VIII. RESULTS, CLUSTERED INSTANCES

Cluster	Clustered Instances	Percentage
1	821,240	21
2	1,394,169	36
3	758,616	20
4	900,156	23

instance, Cluster two has 39 %.

Table XI shows a summary of all the results obtained from the different tests.

Fig. 3 shows that the heap size required to download data in serial mode is between 0 and 200 MB.

We observe from Fig. 4 for one thread, the size of the available is used almost in its entirety,

Especially for the period, 11:30 to 11:40 required 120 MGB of data.

Fig. 5 to 9 show the performance of Heap Memory for each one of the tests.

The execution with eight threads in parallel between 10% and 60% of the CPU used.

The heap size reached a maximum of 1500 MB, as shown in Fig. 5.

In the configuration of 10 threads at the beginning was 85% to achieve stability of 50%. The heap size is large at the

TABLE IX. RESULTS, CLUSTER CENTROIDS GROUPED BY DAYS

Attribute	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Epoch date	2017-05-08	2017-05-01	2017-05-21	2017-04-30
Available satellites	9,83	17.731	18	16.35
GLONASS satellites	8.33	8.37	8.56	6.27
GPS satellites	1.5	9.37	9.44	10.08
SBAS satellites	0	0	0	0
Number of slips on $L1$ and $L2$	801,016.67	496,652.85	2,431,711.03	1,396,281.27

TABLE X. RESULTS, INSTANCES GROUPED BY DAYS

Cluster	Clustered Instances	Percentage
1	6	4
2	52	39
3	27	20
4	49	37

TABLE XI. TEST RESULTS

Test	Number of threads	Average execution time (minutes)	Speedup
Sequential	1	101,15	1X
Parallel	8	48,34	2,09X
Parallel	10	46,27	2,19X
Parallel	16	67,06	1,51X
Parallel	32	474,58	0,21X
Parallel	50	522,86	0,19X

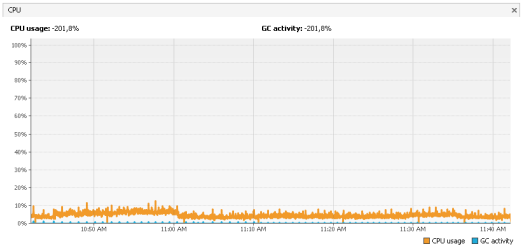

Fig. 3. CPU Performance for One Thread.

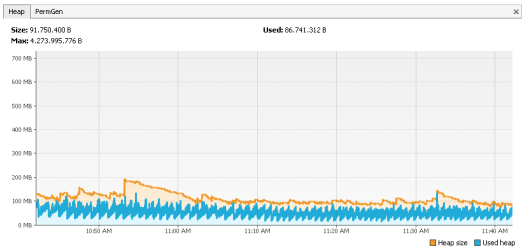

Fig. 4. Heap Memory Performance for One Thread.

Fig. 5. *a)CPU Performance- b) Heap Memory Performance*, Eight Threads.

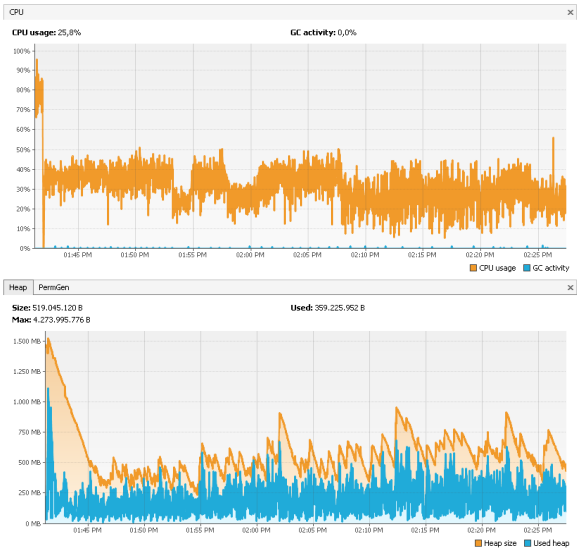

Fig. 6. *a) CPU Performance; b) Heap Memory Performance* , 10 Threads.

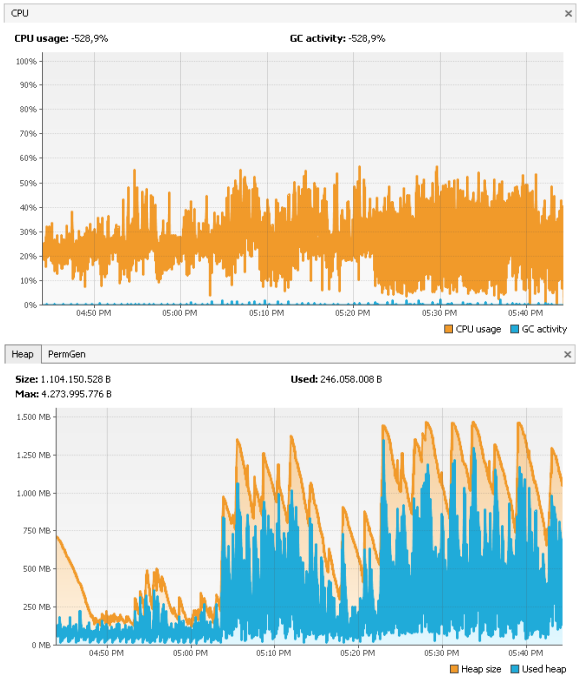

Fig. 7. a) CPU Performance, b) Heap Memory Performance, 16 Threads.

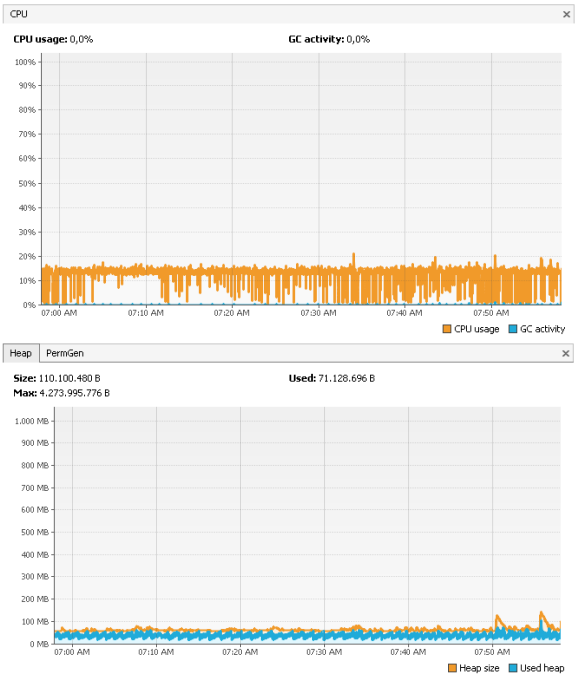

Fig. 8. *a) CPU Performance; - b) Heap Memory Performance*, 32 Threads.

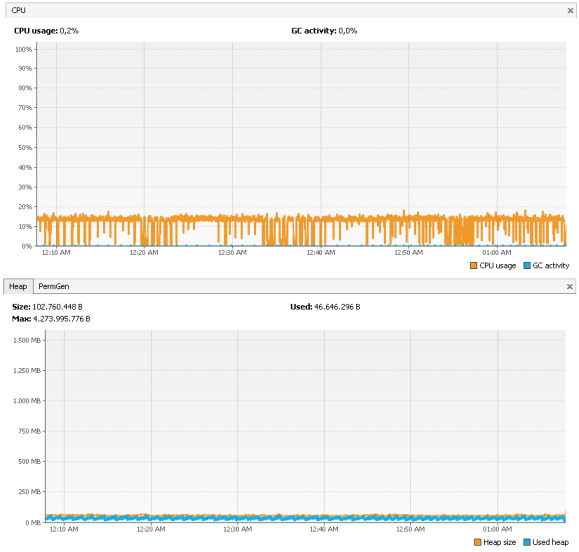

Fig. 9. *a) CPU Performance; b) Heap Memory Performance*, 50 Threads.

outset and stabilizes at around 2 pm when the heap used is lower by a difference less than 100 MB, as shown in Fig. 6.

The use of the CPU with 16 threads was 53% on average. It remained low until after 5 pm when it reached 1500 MB, giving a ratio of close to one for the heap used, as shown in Fig. 7.

Less than 20% of the CPU was for 32 threads.

A very close to 20 MB download between the heap size and the heap used, as shown in Fig. 8.

For 50 threads, the CPU usage was between 1% and 15%, and the heap size and memory used were correlated, as shown in Fig. 9.

Based on the results, the application's performance did not improve even for a large number of threads. However, an improvement was the program with eight or ten threads increased the yield by a factor of almost three. It is because parallel programs depend on finding the best balance between the available hardware and the compiler.

It was evident that when more cores used and the workstation's capacity exceeded, more time was required to execute the processes to create dynamic resource allocation queues.

The permanent space memory remained mostly unchanged throughout these tests, with peaks at 25 MB when the hardware was optimized. That is when the effective use of the multi-processors was optimized. In contrast, the heap space showed various changes in the tests. In serial mode, it showed an almost regular size of around 128 MB. In parallel mode with eight to 10 threads, it reached a peak at the beginning of the execution of a little over 1000 MB.

It was due to the operating system's initial allocation of resources as threads distributed among the cores.

An average load of about 450 MB was then required for balance, with several peaks of up to almost 700 MB due to the files' different sizes.

For 16 threads, there was a considerable amount of memory usage, including high consumption of space for three-quarters of the running time and an average of approximately 1000 MB. The process queue achieved rapid allocation of resources by the operating system. It handled many objects in the memory since memory recovery was slower than the instantiation of objects in the application. When testing more than 16 threads, constant values obtained of around 90 MB in heap space, which we interpreted as indicating that the process queue was massive. The operating system, therefore, distributed the workloads between several cores without allocating higher priorities. The outcome was that these values did not change throughout the implementation and showed lower performance than the sequential mode.

VI. Conclusions

This article proposes a method for extracting observations from several global positioning systems in real-time; these data are valuable for various science areas. Our method features a four-layer architecture. Experiments comprising a sequential test, a parallel test with multiprocessors, a clustering test, and a semantics queries test were designed and conducted. The results show a performance improvement of three-fold when running the program with eight or ten threads. Our dataset was about 100 GB in size, and retrieval was achieved in less than 60 minutes.

The speed and size of the downloaded files depend on the communications network and the availability of different broadcasters, and data extraction, therefore, has an external dependency. The development of applications using programming language optimizations gives more excellent reliability and efficiency.

It is mostly useful for applications with a high burden and a high communication level with a database.

The efficient administration of drivers and meta-data makes a difference in terms of performance.

The size of the dataset used in this study was approximately 100 GB; Big Data generally deals with much larger datasets and may also include structured or unstructured datasets. However, finding and correcting observations from several global positioning systems in real-time is a Big Data problem since it contains the four aspects of Velocity, Variety, Volume, and Veracity. where (i) velocity is the speed with which the satellites publish their results in seconds (ii) variety within existing systems for navigation and geospatial positioning, and their results are heterogeneous; (iii) volume: every second has new data, so we are talking about terabyte level; iv) veracity, in theory, is real because we are analyzing the results of satellites. RINEX files to be identified with their semantics is a challenging task.

The scalability of the method depends on the storage infrastructure for the RINEX files. The percentage of errors in the downloaded files was not more significant than 2%. Given that RINEX is an information exchange file, it complies with the conditions imposed on an exchange file. Interoperability between the various operating systems, non-redundancy of data, possibility of adding new observations excepts with a fundamental one: the great length of its files.

Initially, The method may have opted for reducing its size by choosing a binary format but at the cost of losing access to its content and availability for the user.

Nowadays, file compression programs reduce the RINEX file by a factor of three or more. For example, a file of half a day of observation, with times of 30 seconds, can occupy 1.5-2 Mb and compacted to 500-600 kb. The recording of these files has a maximum of 80 characters per line, but they contain thousands of lines.

Conflict of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

The data used to support the findings of this study are available from the corresponding author upon request. This work was supported by the Sciences Research Council (CONACyT) through the research project number 262756 http://navigationgnssproject.net/index.html.

References

[1] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: from big data to big impact," *MIS quarterly*, pp. 1165–1188, 2012.

[2] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "Big data technologies: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018.

[3] S. Li, S. Dragicevic, F. A. Castro, M. Sester, S. Winter, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein *et al.*, "Geospatial big data handling theory and methods: A review and research challenges," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 119–133, 2016.

[4] R. Tardío Olmos, A. Maté, J. Trujillo *et al.*, "An iterative methodology for defining big data analytics architectures," 2020.

[5] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–global navigation satellite systems GPS, GLONASS, Galileo, and more*. Wien; New York: Springer, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-211-73017-1

[6] S. Yalvac and M. Berber, "Galileo satellite data contribution to gnss solutions for short and long baselines," *Measurement*, vol. 124, pp. 173–178, 2018.

[7] C. Zhou, S. Zhong, B. Peng, J. Ou, J. Zhang, and R. Chen, "Real-time orbit determination of low earth orbit satellite based on rinex/doris 3.0 phase data and spaceborne gps data," *Advances in Space Research*, vol. 66, no. 7, pp. 1700–1712, 2020.

[8] K. Maciuk, "Gps-only, glonass-only and combined gps+ glonass absolute positioning under different sky view conditions," *Tehnički vjesnik*, vol. 25, no. 3, pp. 933–939, 2018.

[9] X. Li, M. Ge, X. Dai, X. Ren, M. Fritsche, J. Wickert, and H. Schuh, "Accuracy and reliability of multi-gnss real-time precise positioning: Gps, glonass, beidou, and galileo," *Journal of Geodesy*, vol. 89, no. 6, pp. 607–635, 2015.

[10] J. Wang, "Stochastic modeling for real-time kinematic gps/glonass positioning," *Navigation*, vol. 46, no. 4, pp. 297–305, 1999.

[11] M. Rieke, T. Foerster, J. Geipel, and T. Prinz, "High-precision positioning and real-time data processing of uav systems," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 119–124, 2011.

[12] X. Li, G. Dick, C. Lu, M. Ge, T. Nilsson, T. Ning, J. Wickert, and H. Schuh, "Multi-gnss meteorology: real-time retrieving of atmospheric water vapor from beidou, galileo, glonass, and gps observations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 12, pp. 6385–6393, 2015.

[13] G. Wübbena, A. Bagge, G. Seeber, V. Böder, P. Hankemeier *et al.*, "Reducing distance dependent errors for real-time precise dgps applications by establishing reference station networks," in *Proceedings of Ion Gps*, vol. 9. Institute of Navigation, 1996, pp. 1845–1852.

[14] H. A. Karimi, D. Roongpiboonsopit, and H. Wang, "Exploring real-time geoprocessing in cloud computing: Navigation services case study," *Transactions in GIS*, vol. 15, no. 5, pp. 613–633, 2011.

[15] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. Loureiro, and Q. Wang, "Energy efficient gps sensing with cloud offloading," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 85–98.

[16] D. Lyu, F. Zeng, X. Ouyang, and H. Zhang, "Real-time clock comparison and monitoring with multi-gnss precise point positioning: Gps, glonass and galileo," *Advances in Space Research*, vol. 65, no. 1, pp. 560–571, 2020.

[17] V. Kopytov, A. Shulgin, N. Demurchev, P. Kharechkin, and V. Naumenko, "High-speed stream data collection and processing system of the earth's ionospheric sounding," in *IOP Conference Series: Materials Science and Engineering*, vol. 450, no. 2. IOP Publishing, 2018, p. 022005.

[18] M. Kakooei and A. Tabatabaei, "A fast parallel gps acquisition algorithm based on hybrid gpu and multi-core cpu," *Wireless Personal Communications*, vol. 104, no. 4, pp. 1355–1366, 2019.

[19] D. Agrawal, S. Das, and A. El Abbadi, "Big data and cloud computing: current state and future opportunities," in *Proceedings of the 14th International Conference on Extending Database Technology*. ACM, 2011, pp. 530–533.

[20] C. Boja, A. Pocovnicu, and L. Batagan, "Distributed parallel architecture for" big data"," *Informatica Economica*, vol. 16, no. 2, p. 116, 2012.

[21] X.-w. DING, S.-l. XIE, and J.-y. LI, "Parallel etl based on spark," *Computer Engineering and Design*, p. 09, 2017.

[22] M. Bala, O. Boussaid, and Z. Alimazighi, "A fine-grained distribution approach for etl processes in big data environments," *Data & Knowledge Engineering*, vol. 111, pp. 114–136, 2017.

[23] X. Liu, C. Thomsen, and T. B. Pedersen, "Mapreduce-based dimensional etl made easy," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 1882–1885, 2012.

[24] M. Radonić and I. Mekterović, "Etlator-a scripting etl framework," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017, pp. 1349–1354.

[25] X. Liu, C. Thomsen, and T. B. Pedersen, "Cloudetl: scalable dimensional etl for hive," in *Proceedings of the 18th International Database Engineering & Applications Symposium*. ACM, 2014, pp. 195–206.

[26] M. Bala, O. Boussaid, and Z. Alimazighi, "P-etl: Parallel-etl based on the mapreduce paradigm," in *Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on*. IEEE, 2014, pp. 42–49.

[27] M. F. Masouleh, M. A. Kazemi, M. Alborzi, and A. T. Eshlaghy, "Optimization of etl process in data warehouse through a combination of parallelization and shared cache memory," *Engineering, Technology & Applied Science Research*, vol. 6, no. 6, pp. 1241–1244, 2016.

[28] C. Thomsen and T. B. Pedersen, "Easy and effective parallel programmable etl," in *Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP*. ACM, 2011, pp. 37–44.

[29] P. S. Diouf, A. Boly, and S. Ndiaye, "Performance of the etl processes in terms of volume and velocity in the cloud: State of the art," in *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*. IEEE, 2017, pp. 1–5.

[30] J. Han, E. Haihong, G. Le, and J. Du, "Survey on nosql database," in *Pervasive computing and applications (ICPCA), 2011 6th international conference on*. IEEE, 2011, pp. 363–366.

[31] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[32] C. Ji, Y. Li, W. Qiu, U. Awada, and K. Li, "Big data processing in cloud computing environments," in *Pervasive Systems, Algorithms and Networks (ISPAN), 2012 12th International Symposium on*. IEEE, 2012, pp. 17–23.

[33] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future," *ACM sIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1–5, 2013.

[34] R. R. Vatsavai, A. R. Ganguly, V. Chandola, A. Stefanidis, S. Klasky, and S. Shekhar, "Spatiotemporal data mining in the era of big spatial data: algorithms and applications," in *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial@SIGSPATIAL 2012, Redondo Beach, CA, USA, November 6, 2012*, 2012, pp. 1–10.

[35] L. Self, "Use of data mining on satellite data bases for knowledge extraction." in *FLAIRS Conference*, 2000, p. 149–152. [Online]. Available: http://www.aaai.org/Papers/FLAIRS/2000/FLAIRS00-029.pdf

[36] D. R. Azevedo, A. M. Ambrosio, and M. Vieira, "Applying data mining for detecting anomalies in satellites." IEEE, May 2012, pp. 212–217.

[37] S.-S. Ho and A. Talukder, "Automated cyclone discovery and tracking using knowledge sharing in multiple heterogeneous satellite data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, p. 928–936. [Online]. Available: http://dl.acm.org/citation.cfm?id=1402001

[38] Y. Li, Y. Wang, J. Yan, and Y. Qi, "The application of data mining in satellite TV broadcasting monitoring." IEEE, Apr. 2009, pp. 357–359.

[39] S. C. Tay, W. Hsu, K. H. Lim, and L. C. Yap, "Spatial data mining: Clustering of hot spots and pattern recognition," in *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, vol. 6, 2003, p. 3685–3687.

[40] Y. Guo, J. G. Liu, M. Ghanem, K. Mish, V. Curcin, C. Haselwimmer, D. Sotiriou, K. K. Muraleetharan, and L. Taylor, "Bridging the macro and micro: A computing intensive earthquake study using discovery net." in *SC*. IEEE Computer Society, 2005, p. 68.