# Leveraging In-Memory Computations for Climate Analytics

Roshni P, Surekha Mariam Varghese

**Abstract**— Big-data analysis entered a new way for climate analytics using real time data. Variations or changes in the weather can be continuously monitored and analysed using big data techniques. Identifying the heavy thunderstorms from satellite data can prevent hazardous situations. And they are also important for climate and agricultural studies. The cloud elements that cause heavy thunderstorms are observed from infrared satellite data. These features are identified on infrared satellite data via their cloud size, shape and temperature. The cloud elements that cause thunderstorms are identified using graph automated theory called Grab 'em Tag' em Graph 'em method. The specified method is parallelized in the system to improve the performance by handling enormous amount of data and their performance against conventional identification system is also performed.

**Index Terms**— Bigdata, Cloud Elements, Distributed, Graph, Real Time, Satellite data, Thunderstorm

## 1 Introduction

ANALYZING massive real-time data has brought new knowledge and information to the world. As such, analyzing data from continuously producing weather data in real time can help in disaster management and studies in climate studies, agricultural and hydrological studies. The automated algorithms helps us to see patterns before human can analyses. This provide us a better understanding about the situation. In weather data analysis, finding out information from weather characteristics can helps to plan for the upcoming disasters. But since the data are continuously generated in real time from satellite sensors, we need a fast processing framework which makes weather data analysis easy. The method introduces a distributed framework which automates the graph analysis faster. A group of thunderstorms can spread over a large area and can cause heavy rainfall or storms. This makes it difficult for area like agriculture. Identifying the cloud elements that cause these thunderstorm can be analysed with the satellite data. The system proposes a parallel graph-based automated method for identifying these thunderstorms that can cause severe weather events. [1] Since the scientific data are increasing enormously, parallel execution of data processing is needed.

Leveraging the in-memory computations rather than batch processing is proposed to analyse the satellite data. The Grab 'em, Tag 'em, Graph 'em (GTG) method which automates identification of a particular weather phenomenon by searching for cloud elements in consecutive time data frames is distributed to support the above. The method uses Apache Spark framework for satellite data computations. The data to be analysed is extracted from the National Centre for Environmental Prediction. [3] The data location was selected as India. The data contains brightness temperature used to collect information about the height and thickness of the cloud.

- *Roshni P is currently pursuing masters degree program in Computer Science and Engineering in Mar Athanasius College of Engineering,Kerala E-mail: roshniravi1993@hotmail.com*
- *Surekha Mariam Varghese is currently heading the Department of Computer Science and Engineering, M.A. College of Engineering, Kothamangalam, Kerala, India. E-mail: surekha.laju@gmail.com*

The consecutive data frames are searched for the cloud elements and the graph is constructed so that climate scientist can analyse the graph to predict the upcoming disasters. [2] The cloud elements are identified via a criteria based on shape, size and absolute values of brightness temperatures of continuous points within a data frame, and are denoted as the vertices in the graph. Consecutive frames are then checked for overlapping cloud elements. Two cloud elements that overlap are represented as an edge in the graph. The graphs are then analysed via graph methods to determine the type of weather phenomenon. In the proposed system the GTG method is parallelized to preserving the sequential method. This is presented as the parallel-GTG.

In parallel GTG, the NCEP/CPC 4km Global (60N - 60S) IR Dataset of location 'India' are fed into the local system.[9] The extracted information is converted to resilient distributed dataset as satellite data. Then the sequential approach used in the Grab them, tag them and graph them approach is made parallel but by preserving the sequential manner. Thus fast processing of the computation is done in the in memory itself. A graph is constructed from the result to find the correlation between the cloud elements. Its performance is compared against the traditional system for finding the cloud elements. Figure 1 shows the architecture of the system.

## 2 Motivation

Analysing weather data by data scientist in real time before some hazardous situation occurs and taking proper steps against it can help to avoid big loss to human life and also to the environment. There are automated method to analyse the weather data generated from satellite. But since massive amount of satellite data are generated from satellite in consecutive times, the automated sequential for finding the group of thunderstorms takes too much time. Implementing the method as Map Reduce job increases performance, but at the same time overhead will be more since it involve disk usage.

The Spark framework is used for parallelizing the sequential GTG approach. It involves in-memory computations rather than disk usage. The GTG algorithm is distributed for

identifying the cloud elements from the consecutive frames of the satellite data by preserving the sequential manner. Spark is an open source cluster computing framework. In contrast to Hadoop MapReduce paradigm, Spark's multi-stage in-memory primitives provides performance up to 100 times faster for big data processing applications.

## 3 Existing System

The automated Grab 'em, Tag 'em, Graph 'em (GTG) algorithm can be used for thunderstorm collection identification in a time series of highly resolved infrared satellite images.[2] The algorithm is based on graph theory. The main step involved are the identifying cloud elements of prescribed size and temperature as the graph's vertices, then determining the areas of overlap between cloud elements which is used to define the graph's edges. Last, a graph is drawn defining the cloud clusters that are correlated. It utilizes infrared satellite data and depends on brightness temperature and area thresholds to identify regions of interest that could develop into various MCSs. These regions are referred to as cloud elements (CEs). The GTG algorithm uses a temperature threshold of 241 K for CE identification such that values warmer than the threshold are discarded and those colder are equal to it are maintained for analysis. The GTG algorithm also uses an area cut-off of 2,400 $km^2$ in conjunction with the TB threshold. Each frame is a function of latitude, longitude and brightness temperature and contains a number of cloud elements (CEs).

The basic method uses the sequential approach for the Graph 'em Tag 'em Grab 'em algorithm. The mesoscale convective complexes represents the collection of thunderstorms. These thunderstorm can persist for hours. The thunderstorms are collected in each frame of the satellite data. Each time (image) of IR data is referred to as a frame. Each frame, Ft, is a function of latitude, longitude and brightness temperature (BT), where t=1 hr. Within each frame, there exists a number of CEs. The properties of a CE are specified by a brightness and area criteria where the warmest temperature considered for Cloud Element identification is 241 K. The GTG utilizes the area-overlapping method. In the area overlapping method, the area overlap between consecutive cloud elements are considered and overlap are represented by the graph edges. Thus it considers the current frame and the next frame for finding the cloud elements. The approach is sequential because first it considers the frame at t and $t+\Delta t$, Then it takes $t+\Delta t$ as the current frame and $t+2\Delta t$ as the next frame.

## 4 Method

The proposed system takes satellite datasets that are continuously generated in real time in distributed manner. They are parallelized using the capabilities of scientific resilient distributed datasets. The data is then processed to analyze and identify the cloud elements that can cause rainfall etc. The automated Grab 'em Tag 'em Graph 'em (GTG) [14] algorithm which uses sequential frame processing approach is parallelize in the proposed system. In order to speed up the parallel processing, the big data processing framework called Spark is used. The dataset location are given through the spark shell in the proposed system. The user has the choice to select from where the dataset has to be taken and he can specify it in the terminal where the proposed spark program is built and executed. The spark driver gets initiated and then parallel computations are performed. The identified cloud elements details are stored in the file and corresponding graphs are drawn. The proposed system uses the Apache Spark framework for distributed in-memory computations.

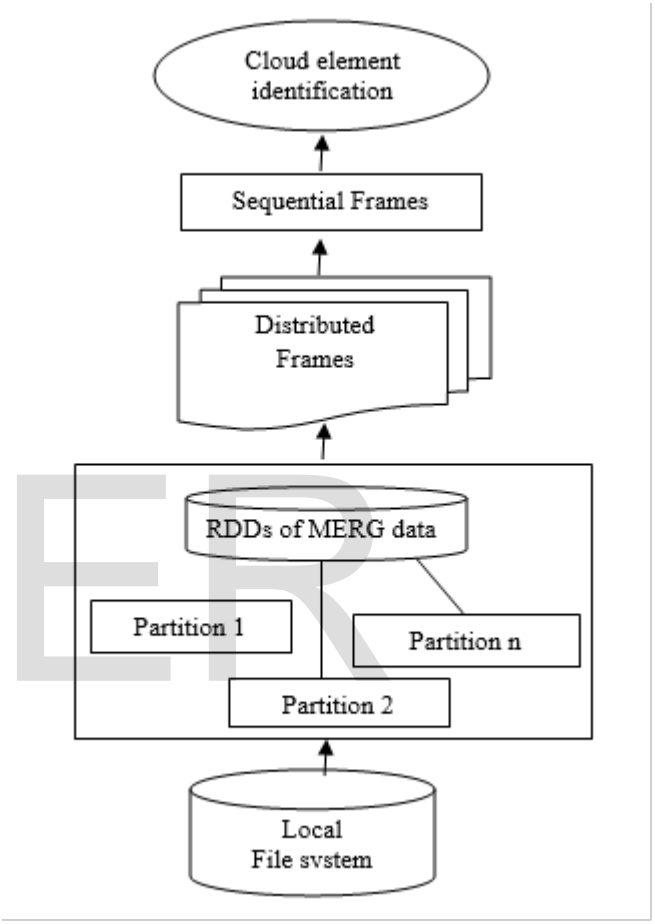


Fig. 1 Architecture of the system

### 4.1 Data Input

The data that are continuously generated from satellite sensors are stored in local file system. Thus the user provides a set of data files with unique identifier through the local filesystem in the proposed system and is given to the spark context. Then Resilient Distributed Datasets are created from the dataset. They are the distributed-computing data structure. It consist if self-document array-collection developed for RDD transformations. The Resilient Distributed Dataset (RDD) couples operations on multi-dimensional arrays and distributed in-memory processing. Each frames that contain information are extracted into labelled component frame.

### 4.2 Parallel Method

The parallel GTG method uses shuffle-sort method to achieve

parallelism and sequential frames. Using shuffle and sort method, [1] the distributed frames are made sequential. In the shuffle-sort approach, making a copy of each frame to map to the next frame is done by creating pairs between original frame and the next frame. The pairs were then sorted using the quicksort algorithm by considering the first value in each pair. Then after sorting, the values are in sorted order. Their first element value in each pair is deleted to get the consecutive frames. Thus the consecutive pairs are achieved. The pairs are Fi, Fi ) and ( Fi, Fi+1). The Fig.2 presents the Shuffle and Sort method. The a, b, c, d are the frames in sequential order. They are the distributed frames. To make it in the sequential order, quick sort is used.

(e, (e,f)) (c, (c,d)) (b, (b,c)) (d, (d,e)) (a, (a,b))

(a, (a,b)) (b, (b,c)) (c, (c,d)) (d, (d,e)) (e, (e,f))

(a, b) (b, c) (c, d) (d, e) (e, f)

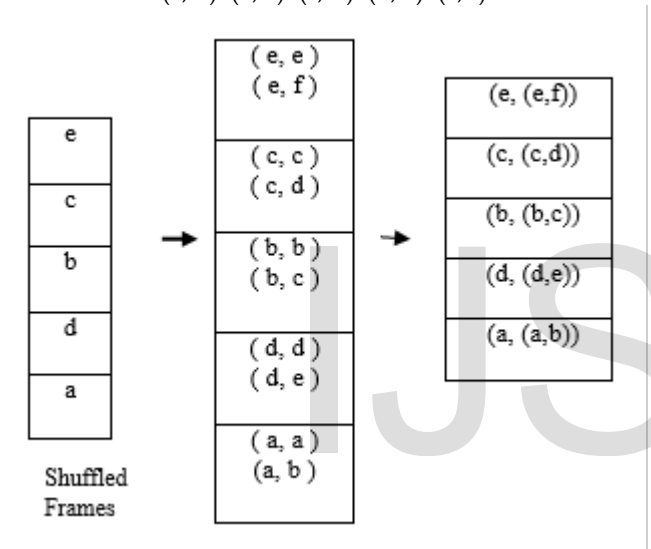


Fig. 2 Shuffle-Sort Method

### 4.3 Finding Cloud Elements and Correlating

The next steps involve defining the task that identify the cloud elements [3]. The labelled component array for the first frame is considered and is searched for non-zero values. The positions having non-zero values are only considered. Then the next sequential frame's component labelled array is taken. The second component frame is searched for non-zero values in the positions considered from the first frame. If it contain zero value, that position is wiped out. Thus we will get the overlapping clouds which contain non zero values.

In Fig.3

(i) Considering only the non-zero values in Frame 1

(ii)Consider only the positions which have non-zero values in Frame 1.

(iii)Get the cloud elements which overlap

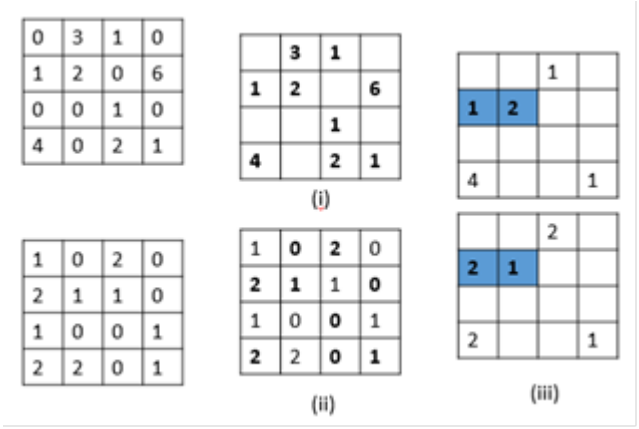


Fig 3. Cloud Element Correlation

The identified cloud elements are represented as vertex. They are uniquely identified with (frame ID, Cloud Element ID). The correlations between the cloud elements in each frame are identified and are marked as edges as in Fig.4. The edge list is a paired list containing the frame ID and the cloud element ID of two vertices. The format will be ((FrameIDi, CEi) , (FrameIDj, CEj)). The indices I and j can be same or not. If they are same, the cloud elements overlap in the same frame. Otherwise both are from two different frames.

## 5 Experimental Result

The network graph is constructed from the edge and vertex list, which shows the cloud clusters and the correlation between them. It is shown in the Fig 4. The red circle shows the vertex which is the cloud element identified in the IR frame. The edges between the vertices represents the correlation between cloud elements. The graphs are drawn using matplot tool which is used to create 2D graph images. The edge-vertex which have longest length is identified.

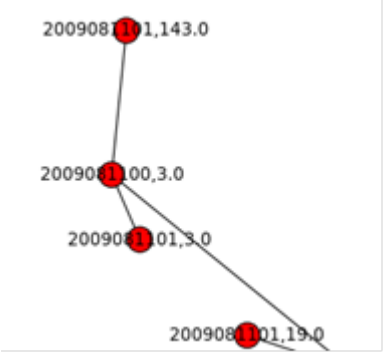


Fig 3. Graph Representing Correlation

## 6 Performance Evaluation

The performance analysis is based on the time taken to find the cloud elements. The time taken has drastically reduced in

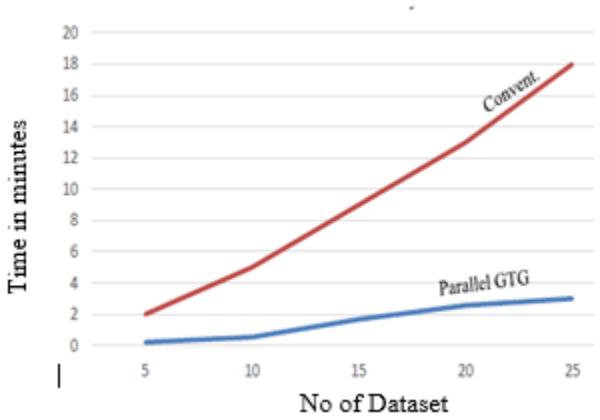


Fig 5. Performance Analysis

parallel GTG comparing with the conventional GTG Method. The experiment was done using the dataset [3] which is 300MB in size. Each frame was approximately about 25MB size. The system was deployed in the cluster mode of Apache framework. (Fig. 5)

The time taken to process the dataset are noted down and the graph is plotted. The time taken in conventional method ranges from minutes to hours, where as in the parallel GTG, it takes only minutes to process the dataset. The green line represent the parallel GTG and other represent the conventional GTG.

## 7 Conclusion

Big data analysis has opened up a new way for weather analysis using real time satellite and atmospheric data. Continuously monitoring the factors related to weather, it can be analyzed constantly with big data, so that variations in weather can be captured. The proposed method achieves parallel ingestion and partitioning of satellite datasets. The Grab 'em Tag 'em Graph 'em (GTG) algorithm which uses sequential frame processing approach is parallelized and performance based on execution is evaluated. The satellite data frames are partitioned and distributed to many nodes for parallel processing. The shuffle-sort method is used to make the frames in chronological order from distributed partitions. The cloud elements are then identified which forms the collection of thunderstorms. The performance analysis is done by comparing the conventional and proposed method on the basis of execution time. The execution time for processing the satellite data is drastically reduced. Thus leveraging the in-memory processing and map reducing capabilities, the sequential GTG algorithm which process scientific data is parallelized to achieve high performance.

## References

[1] Rahul Palamuttam, Renato Marroquín Mogrovejo, Chris Mattmann, SciSpark: Applying In-memory Distributed Computing to Weather Event Detection and Tracking, IEEE International Conference on Big Data 2015

[2] Kim Whitehall & Chris A. Mattmann & Gregory Jenkin Exploring a graph theory based algorithm for automated identification and characterization of large mesoscale convective systems in satellite datasets Earth Sci Inform

[3] MERG Dataset NCEP 4km Global (60N–60S) IR ftp://ftp.cpc.ncep.noaa.gov/precip/global_full_res_IR

[4] Matei Zaharia, Tathagata Das, Haoyuan Li, Scott Shenker, and Ion Stoica. 2012. Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. In Proceedings of the 4th USENIX conference on Hot Topics in Cloud computing (HotCloud'12). USENIX Association, Berkeley, CA, USA

[5] Buck, Joe B., Noah Watkins, Jeff LeFevre, Kleoni Ioannidou, Carlos Maltzahn, Neoklis Polyzotis, and Scott Brandt. SciHadoop: Arraybased Query Processing in Hadoop." Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis on - SC '11, 2011.

[6] Mattmann, C.A. "SciSpark: Interactive and Highly Scalable Climate Model Analytics". Presentation. Earth Science Technology Office, 2015.

[7] Robert A. Houze Jr. Mesoscale Convective System,Department of Atmospheric Sciences, University of Washington Seattle, Washington, USA 31 December 2004

[8] Kempler, Steve. "NCEP/CPC 4km Global (60N – 60S) IR Dataset Product Description". http://mirador.gsfc.nasa.gov/collections/MERG__001.shtml (accessed September 30, 2015).

[9] Rew, Russ, and Glenn Davis. "NetCDF: an interface for scientific data access."Computer Graphics and Applications, IEEE 10, no. 4(1990): 76-82.

[10] Loikith, P. C., B. R. Lintner, J. Kim, H. Lee, J. D. Neelin, and D. E. Waliser. "Classifying reanalysis surface temperature probabilitydensity functions (PDFs) over North America with cluster analysis." Geophysical Research Letters 40, no. 14 (2013): 3710-3714.

[11] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters." Communications of the ACM 51, no. 1(2008): 107-13.

[12] Nicole Giggey Identifying and Visualizing Mesoscale Convective Complexes in West Africa Howard University, Washington, District of Columbia

[13] Kim Whitehall & Chris A. Mattmann & Gregory Jenkins & Mugizi Rwebangira Exploring a graph theory based algorithm for automated identification and characterization of large mesoscale convective systems in satellite datasets, Springer-Verlag Berlin Heidelberg 2014

[14] Arnaud Y, Desbois M, Maizi J (1992) Automatic Tracking and Characterization of African Convective Systems on Meteosat Pictures. J Appl Meteorol 31(5):443–453