DISCRETE TIME EVENTS REPRESENTATION USING MAPREDUCE

Methaq Kadhum
The University of Jordan/ ComputerSecince Department
Methaq_Kadhum@yahoo.com

Azzam Sleit, Rizik Alsayyed
The University of Jordan/ ComputerSecince Department

Abstract

The behaviour of enormous real-life systems can be represented as a sequence of discrete events and illustrated with a visual map that enables focus and context perceptions. The behaviour illustration, based on the discrete time event visualization, becomes challenging as the encountered amount of events increased substantially. The challenge to create such visual maps is two folds: First: a substantial processing is required to create the map. Second: it is difficult to gain both focus and context simultaneously in the same map. In this paper, MapReduce, framework paradigm for distributed tasks, is used for processing the behavioural-data. Then, the pre-processed data is plotted in 2D space with clear focus and context using Time-Maps scheme. The proposed technique uses simple technique to plot the data, by calculating the "time before" and "time after" for each event and projected these events into two-dimensional space that represent the calculated measures. The implementation of the proposed technique along with huge syntactic behavioural-data shows the robustness of the proposed scheme for plotting big data onto maps with enabled focus and context perceptions.

Keywords: Discrete Time Event, MapReduce, Visualization

1. INTRODUCTION

The behavior of enormous real-life systems, such as the global market, computer network, farming, are naturally represented as a sequence of events that represent the changes in these systems along time scale. Subsequently, system behaviours can be represented as discrete events, visually illustrated and used when necessary. The visual representation of discrete events is motivated by the human's perceptual abilities, by which, knowledge about the represented system can be extracted. The visual representation of discrete events allows for capturing a summary, significant and insignificant characteristics, duplication and uniqueness of the system behaviour. However, in order to facilitate the perceptual remarks, the visual illustration should enable, both context and focus over the represented data [1].

Discrete event representation depends on the time at which the events are occurred. These events, associated with time slots, are represented as a histogram of events, as illustrated in Fig. 1. An obvious problem of such naïve representation is loosing of the focus and/or context. The amount of data usually represents a challenge for plotting and displaying, as various details might be hide. Basic time-series data representation using lines and bars, subsequently, is not proper for representing large data in event-driven systems [2].

To address such problem, focus-enabled technique was proposed and utilized with various illustration schemes. Focus-enabled technique illustrates an overview of the data to enable context-aware perception and enable user-activation focus on specific data portion [3]. However, even with focus-enabled technique, perceptual-based knowledge extraction is still difficult task. Subsequently, various other sorts of data representation have been proposed. These methods, focus on the illustration map and the pre-processing of the data to generate different data form with clear focus and context. Examples of the existing techniques are distortion-oriented, overview projection and filtering [4].

Reforming the data is the process of deriving new series from the original data, which required processing the data elements. This process consumes massive time, given that these data are usually large, thus parallel and distributed process is required in-order to facilitate such processing.

{ SHAPE * MERGEFORMAT **}**

Fig. 1: Discrete-Time Representation using Histogram

In this paper, MapReduce is used for data processing in-order to generate a new form of data that can be plotting in 2D space with clear focus and context. Data plotting is implemented using Time-Maps scheme, which considered the time before and time after each event as an identification of each event [5]. The rest of the paper is organized as follows: Section 2 presents an overview of MapReduce and Hadoop deign framework. Section 3 presents the related work for time events plotting schemes. Section 4 presents the proposed work for discrete event visualization based on Time Map and MapReduce. Section 5 presents the results and Section 6 gives a conclusion.

2. MAPREDUCE

MapReduce is a framework paradigm for distributed tasks, proposed by Dean and Ghemawat in 2004 [6]. A task formed by the MapReduce paradigm should include two major steps, these are Map and Reduce. The map task actually is a set of tasks that are executed in a parallel over various processing units. The inputs for this task are independent data points represented by a key-value pair. The map task, over various processing units, implements the same task over different input data elements, regardless of its content. The reduce task, which implemented over a single or various processing units, receives the output of the map task and aggregates identical key elements together. The shuffle and sort processes, which are implemented by the MapReduce paradigm, as illustrated in Figure 2, are responsible for sorting the data elements, to facilitate easy and rapid reduce task. Hadoop architecture, illustrated in Fig. 2, is the actual implementation of MapReduce using Java, which is distributed as an open source [7].

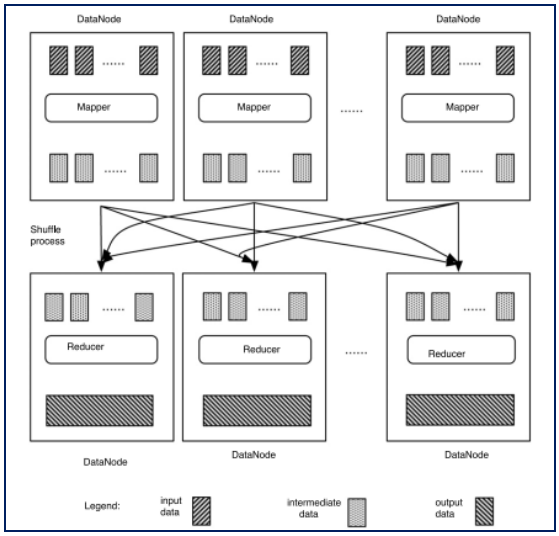

Figure 2: Hadoop MapReduce Architecture [8]

MapReduce is very efficient for processing large datasets, subsequently, it has been used in various research fields, such as, processing huge text data [9] processing huge amount of remote sensing images [10], Large Job Scheduling [11] and others.

3. RELATED WORK

Naïve illustration techniques, such as bars, lines and histogram and the focus-enabled technique did not show a robustness in visualizing large discrete event data. Thus, in-order to enhance the focus and context illustration, several illustration schemes were proposed for this type of data. Fig. 3 represents some of these shapes [2]. Cycle bins technique was proposed to enable both focus and context of the data, as illustrated in Fig. 3(a) [12]. Cycle bins scheme used 3D imagining with information hiding technique. First, the data is divided into distinctive groups, based on some specific criteria, such as the location on the map. Then, each group is represented as a cylinder of a complete season. Events in specific time scale are highlighted over the cylinder by a specific color. Note that, while this representation gives a robust view of the represented system, it cannot be used with data of single group [12].

Square Bins technique was proposed that enables context and focus perceptions using separated, yet related square bins, as illustrated in Fig. 3(b) [13]. The time-events are plotted based on three variables, these are: data range, display space and the rendering method. This technique used equal length time slots, each is represented by a single bin with a magnitude represents the density of events in that slot. The equal interval between these slots is calculated by analyzing the data and determines the significant interval to be used for data division and grouping. Later on, multiple adjacent bins can be merged if they have equal magnitude. Then, multiple resolutions images are generated by calculating the aria of bins under the determined resolution [13]. This kind of representation, depends on the ability to categorize the data into equal length slots, and it is not useful for data with irregular patterns.

{ SHAPE * MERGEFORMAT **}**

Fig. 3: Discrete-Time Representation using Various Shapes

Spiral with distinctive line styles technique was proposed to represent time-event data, as illustrated in Fig. 3(c) [14]. Spiral is drawn with distinctive color or texture, including line styles and patterns, thickness of the line or icons based on the density and duration of the events. Multi-spirals are used of the data, which can be naturally categorizes into

multiple groups. Instead of spirals, 3D-based helix can be used to support exploring the data and support multi-resolution aspects [14]. Scatter dots was also used effectively for time-event data after some pre-processing. Such illustration has proven that time events can be illustrated effectively with simple 2D-shapes and after some pre-processing [5].

As a summary, all the existing illustration techniques derive new data forms from the original data then plotting the derivate forms into various shapes, either classical, such as bars, lines and dots or innovative, such as spiral, cylinders, and squares. Regardless of the utilized shape, data elements need to be processed first to enable such visualization.

4. PROPOSED WORK

A MapReduce with Time Map scheme for visualizing time events data is proposed. The goal of the proposed technique is to enable focus, context and rapid representation of big data. The proposed technique calculates the "time before" and "time after" for each event, which causes the derivation of new data form. The derivate data is projected into two-dimensional space, related to the calculated measures. The measures are calculated in MapReduce paradigm. The processing steps, as illustrated in Fig. 4, contains, pre-processing, map, combine, reduce and visualization.

{ SHAPE * MERGEFORMAT **}**

Fig. 4: Proposed Framework for Discrete-Time Representation

Pre-Processing

Given a set of events, each determined by an occurrence time, the pre-processing step is responsible for indexing these events and forms discrete data elements. Each event is given a unique number that is the index of the event in the time-line. The index is a positive integer number that starts at 1 and increased accordingly. Besides the index, each event is linked to the time of the previous event and time of the next event. Subsequently, the output of this step is a set of data elements, each is of the form of (key, value). A single element is represented by three values, as in the following: (index, {currentEvent_time, previousEvent_time, nextEvent_time}).

Map

All the involved mappers receive the generated (key, value) pairs from the controller in the MapReduce framework, apply the same function on each input data element and produce, for each input, another (key, value) pair represented as follows ({period_before, period_after}, count). The mappers apply the two functions, given in Equation 1 and Equation 2, to get the value included in the output pair and sets the count to the value of 1. Then, each mapper produced a list of (key, value) pairs and sent it back to the controller.

period_before=currentEvent_time-previousEvent_time (1)

period_after= nextEvent_time - currentEvent_time (2)

Combine

The controller, in the MapReduce framework, as the background process, combines all the pairs with identical key in a single list. The generated list for each group of identical key is represented by a pair of (key, value list), where the key is pair of {period_before, period_after}, while the list will be a sequence of counts (e.g.: list of ones}.

Reduce

The reducer(s) receives the generated (key, value list) pairs from the controller in the MapReduce framework and combines the pairs in each list. The reducer sums up the list of ones into a single value and generates, for every list a pair of (key, value) represented as ({period_before, period_after}, number of occurrence).

Visualization

The controller collects the results of the reduce phase and outputs them to be visualized in two-dimensional space. The complete processes are implemented in Algorithm 1.

Algorithm 1: Time Event Discretization in MapReduce

```
1. index:=1
2. FOREACH (event : eventList)
3.          outputList1 += Generate (index,
{previous_event, event, next_event})
4.    index++
5. ENDFOR
6. FOREACH (pair: outputList1)
7.    MAP(pair -> Map-output-pair)
8.    Generates outputList2
9. ENDFOR
10.Create outputList3 = null
11.FOREACH (pair: outputList2)
12.   IF (outputList3.contains(pair-> key))
13.        outputList3.add (value).
14.   ELSE
15.       outputList3.add (key, value).
16.   ENDIF
17.ENDFOR
```

18. FOREACH (pair: outputList3)
19. value:= Sum(value-list)
20.ENDFOR
21.Visualize

Lines 1 to 5 represent the pre-processing phase, in which events are given and an index and re-formulated as a pair in line 3. Line 1 sets the index to its initial value and line 4 increases the indexed accordingly. Mapping phase is implemented in line 6 to line 9, in which the pair is re-formed based on Equation 1 and Equation 2. Lines 10 to 17 represent the combination process implemented at the MapReduce framework. The reduce task is implemented at line 18 to line 20. Visualization is implemented independently afterward.

As noted, the proposed scheme is based on simple pre-process steps and utilized simple visualization technique. Subsequently, it can be plotted using simple shapes, scatter plot will be used for this purpose.

5. RESULTS

In order to criticize the robustness of the proposed technique, random data sets are generated with specific characteristics, then, the data are processed and visualized and the output is criticized against human perception. The syntactic data utilized is generated so that the time between events is in the range [0-1]. The first experiment uses 3,000 events, that are generated such as 50% of the events are occurred with time interval in the range of [0-0.1], 20% in the range (0.1-0.5) and 30% in the range (0.5-1.0]. This dataset is processed and visualized and the results are represented in Fig. 5.

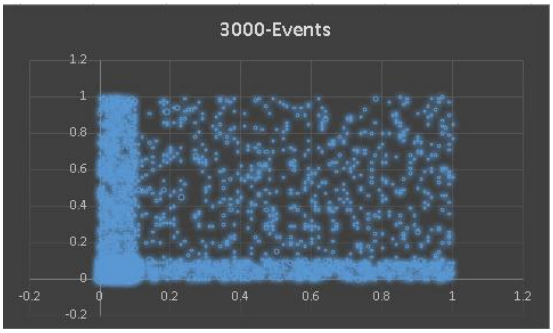

Fig. 5: The Output of the First Experiment

As noted, human can perceive that most of the events (50%) occurred after each other with small time interval, as illustrated in Figure 5 near the origin point (0,0). Besides, less events occurred in with the range (0.1 to 0.5) that formed a line-like shape in-line with the x-coordinates and y-coordinates. Finally, a considerable portion scatters in the rest of the time interval range. Subsequently, this perception is identical to the conditions by which the data were generated.

The second experiment uses more events, with 30,000 events generated under the same conditions that were used in the first experiment, such as 50% of the events are occurred after each-other with time interval in the range of [0-0.1], 20% in the range (0.1-0.5) and 30% in the range (0.5-1.0]. This dataset is processed and visualized and the results are represented in Fig. 6. Again, human can perceive the same as it was observed in the first experiments with obviously more events compare to the first experiments. Subsequently, this perception is identical to the conditions by which the data were generated.

The second experiment, is formed of 30,000 events generated under the different conditions, compared to the first and the second experiments. The generated set contains 33% of the events have time interval in the range of [0-0.1], 33% in the range (0.1-0.5) and 33% in the range (0.5-1.0]. This dataset is processed and visualized and the results are represented in Fig. 7.

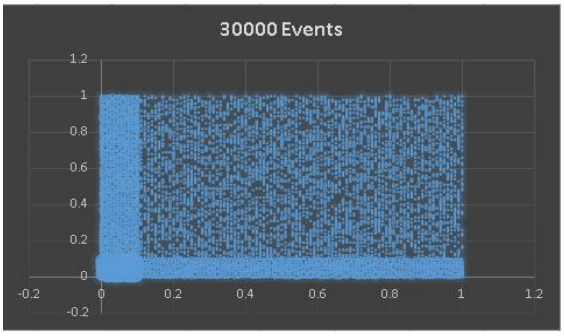

Fig. 6: The Output of the Second Experiment

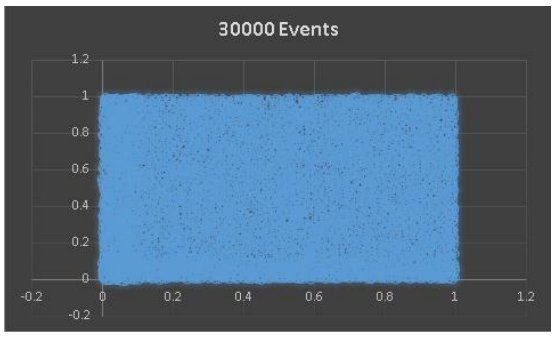

Fig. 7: The Output of the Third Experiment

As noted, there is no patter to be distinguished in the visualization of this dataset as it has been generated with un-unique data events.

6. CONCLUSION

In this paper, a visualization technique for discrete time events is proposed. The proposed technique processes the input data to derive a new series from the original data using MapReduce paradigm. In MapReduce, a new data form is derived, by calculating the time boundaries that are determined in the Map task and quantified in the Reduce task. The derived data is illustrated based on the Time-Maps scheme, which consists of two-dimensions, for the time before and time after each event. The results generated is a simple 2D space with scatter plots that reveal the characteristics of the original data. As proved by the conducted experiments, there is strong match between the properties of the generated data and the perception about that data from its visualization. Future work will consider more experiments, on real data, using different pre-processing techniques and various visualization shapes.

REFERENCES

[1] [1] Zeigler, Bernard P., Herbert Praehofer, and Tag Gon Kim. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. Academic press, 2000.

[2] [2] Kosara, Robert, Helwig Hauser, and Donna L. Gresh. "An interaction view on information visualization." State-of-the-Art Report. Proceedings of EUROGRAPHICS (2003).

[3] [3] Dean F. Jerding and John T. Stasko. The information mural: A technique for displaying and navigating large information spaces. IEEE Transactions on Visualization and Computer Graphics, 4(3):257–271, July/ September 1998.

[4] [4] Kosara, Robert, Helwig Hauser, and Donna L. Gresh. "An interaction view on information visualization." State-of-the-Art Report. Proceedings of EUROGRAPHICS (2003).

[5] [5] Watson, Max C. "Time maps: A tool for visualizing many discrete events across multiple timescales." Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2015.

[6] [6] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.

[7] [7] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: a flexible data processing tool." Communications of the ACM 53.1 (2010): 72-77.

[8] [8] Wang, Lizhe, et al. "G-Hadoop: MapReduce across distributed data centers for data-intensive computing." Future Generation Computer Systems 29.3 (2013): 739-750.

[9] [9] Lin, Jimmy, and Chris Dyer. "Data-intensive text processing with MapReduce." Synthesis Lectures on Human Language Technologies 3.1 (2010): 1-177.

[10] [10] Lv, Zhenhua, et al. "Parallel K-means clustering of remote sensing images based on MapReduce." International Conference on Web Information Systems and Mining. Springer Berlin Heidelberg, 2010.

[11] [11] Zaharia, Matei, et al. "Job scheduling for multi-user mapreduce clusters." EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-55 (2009).

[12] [12] Tominski, Christian, Petra Schulze-Wollgast, and Heidrun Schumann. "3d information visualization for time dependent data on maps." Ninth International Conference on Information Visualisation (IV'05). IEEE, 2005.

[13] [13] Hao, Ming C., et al. "Multi-resolution techniques for visual exploration of large time-series data." EUROVIS 2007. 2007.

[14] [14] Weber, Marc, Marc Alexa, and Wolfgang Müller. "Visualizing Time-Series on Spirals." Infovis. Vol. 1. 2001.