A Real-Time Business Analysis Framework Using Virtual Data Warehouse

Partha Ghosh[1], Deep Sadhu[2], and Soumya Sen[1]
[1]A.K.Choudhury School of Information Technology, University of Calcutta, India
[2]Department of Data Science, Ernst and Young, India

Abstract: *Data Warehouse (DW) is widely used in industries over decades to perform the analysis on data to expedite decision-making process. However, the traditional DW is slower in execution due to the huge time overhead of pre-processing stages of Extraction-Transformation-Loading (ETL). On the other hand, often the situations arise where the decision-making are required in real time. Data virtualization is one of the robust approaches over traditional data warehouse that avoids the costly steps of ETL processing. Virtual Data Warehouse (VDW) allows specific analysis for quick decision making even on the unprocessed data. Moreover, VDW could be used by the organizations that maintain DW to take some immediate business decisions for some abrupt changes. This research work performs business trend specific analysis based on VDW to generate business intelligence even in the catastrophic situations. Experimental results reveal, the proposed methodology based on VDW is around thousand times faster than traditional warehouses.*

Keywords: *Data warehouse, virtual data warehouse, data analytics, business intelligence, real-time analysis, z-score, query execution.*

Received May 19, 2020; accepted December 15, 2020
https://doi.org/10.34028/18/4/11

1. Introduction

Data warehousing has provided a successful foundation for the organizations to invest in treating their high-value information as an asset. However, in the context of today's big-data [7] era, traditional data warehousing is a slow choice, as in traditional data warehouse transformation and modelling of data are performed through Extraction-Transformation-Loading (ETL) [12] tools. Modern-day business applications consist of huge volumes and varieties of data and these require processing in real-time. In enterprise data warehouses, the data are processed in batch mode to prepare the data for business analysis in order to perform decision making in real-time [20]. In these types of conventional approaches real-time decision making is possible only for the known business cases. However, with the rapid and dynamic changes in the business environment sometimes the decision-makings are required in real-time for unknown/uncertain business cases. Traditional data warehouses are not capable to handle these types of emergency business changes. Hence, a solution is required, which gives the benefit of analytical processing and decision making as a traditional data warehouse but incorporates real-time processing capability. For example, in the case of a sudden crash of share market traditional data warehouse may fail to do the real-time analysis, as it requires situation handling and decision making in real-time sometimes for the unprecedented cases. Virtual Data Warehouse (VDW) is one of the possible solutions to address the problem described above. The time requires to process data in VDW is reduced by eliminating the pre-processing steps of ETL tools. In practice, the ETL process may take up to 80% of the time [3, 12, 16] to create a data warehouse. Moreover, VDW supports working in a distributed manner [20], hence no centrally located server requires like traditional warehouses. Each site logically views VDW as a local site consists of relevant and recent data. In addition to this, if the company's vision changes over the time we could incorporate these changing requirements in VDW in real-time. Due to the adaptive and business trend specific nature, VDW is suitable for analysing abnormal business situations like abrupt government policy change, accidental events, sudden share market crash etc. In these cases, the decisions take place on the fly, which is beyond the scope of the traditional data warehouse. Hence, a modified methodology that is capable enough to analyse relevant data as well as managing catastrophic situations quickly is indispensable. Some salient contributions of this paper are:

a) We proposed business trend specific virtual data warehouse architecture that provides around thousand times faster query processing than traditional techniques.
b) This methodology is capable to recognize and manage abnormal business cases in real-time.
c) This research work identifies new business cases and decides whether to include these in the main data warehouse, or not.

Organization of the paper is as follows. Related work in section 2, followed by the problem definition and methodology selection in sections 3 and 4 respectively. Proposed methodology is in section 5. Sections 6 and 7 presents case studies and comparative analysis respectively. In section 8 we conclude.

2. Related Work

Business organizations use specific pre-processing and integration methodologies for managing historical data and implementing data warehouses [3, 13, 14, 16]. They combine data into a single, centralized data repository from heterogeneous operational applications for reporting and analysis purposes. Warehouse typically consists of 5 to 10 years of historical data that provide a solid foundation for business trend analysis, risk management, and overall managerial decision-making [18]. Data warehouses are often termed as ETL [12, 13] tools which refers to the capability of filtering, structuring, normalizing and modeling data. However, it has been found as a slow technique [15] as the time required for the pre-processing of data is very high. In addition, ETL does not support data reusability that results in iterative ETL processing [11, 14]. In today's big data [7] era, where data produced in a prodigious size, ETL processing becomes a laborious task. Although, several work tries to incorporate time-bound decision making [13] ability in the data warehouse environment, but real-time decision making remain unaddressed for data warehouse/OLAP applications. Another inflexibility of the traditional data warehouse is that if a company's business strategy changes frequently, it fails to generate Business Intelligence (BI) on the fly. Generally, BI [4, 19] refers to identifying, extracting and analysing business trend specific data for cost and/or income prediction and act accordingly. BI is also required in data analysis for providing current and predictive views of business operations. Several alternative structures [6, 11] are proposed over the time in order to make warehouses more acquainted with the present era. A modified real-time data warehouse structure [6] proposed by integrating both Online Transactional Processing (OLTP) and Online Analytical Processing (OLAP) data. These types of approaches are very helpful for real-time data analysis but does not concern about business trend specific data analysis.

Data virtualization or maintaining the data warehouse virtually [5, 9, 17] could be one of the desired changes over traditional data warehouses. It is in the form of middleware [8] to integrate and supply data between several data sources and data consumers. Data virtualization applied in the business environment for quick data analysis and decision making, which in turn increases business agility [8]. Several architectures [9, 17] are proposed over the time for data virtualization. The concept of particular and expressive virtual data mart [17] proposed for better availability of data. This concept of virtual data mart is further expanded in terms of the virtual data warehouse. Two different architectures [9] proposed for the organizations that maintain enterprise data warehouse and for the other organizations that do not maintain an enterprise data warehouse. This concept is further expanded in terms of product modification [10] by analyzing customers' sentiment. Several modified ETL approaches [12, 15] were proposed in order to mitigate the high time complexity of ETL. In order to maximize thespeedup of data access a new model named as CloverETL [15] was proposed where after the extraction and transformation stage, instead of loading data into the warehouse, it directly loads data into a virtual data warehouse and clients feed on it. This methodology [15] also hosts source information in the cloud [15, 21] that increases data availability from each site and solves the problem of unified information protection. Further, this concept expanded by Artificial Neural Network (ANN) [21] based prediction of execution time. Another dynamic and progressive approach named as TEL [12] proposed for data extraction. According to the users' requirement of data transformation, TEL technique extracts data directly to the data warehouse. The TEL approach creates a virtual layer between the data warehouse and various data sources. This virtual layer formed to bypass excessive I/O overload and for discarding heterogeneous problems. Virtual tables [12] are formed in order to hold relational schemas of the heterogeneous data environment. Based on these virtual tables, data objectification technique is used to extract and load data in the target data sources. This data objectification technique avoids redundant ETL extraction. The TEL technique is effective, but experimental results show that it is limited to some applications only.

The survey work that we carried out here from that we found no significant work on virtual data warehouses according to the changing business trends in real-time. Hence, the objectives like faster and relevant query processing, quick decision making for abnormal business crash down or abnormal hike of business, etc. remain unaddressed for Virtual data warehouse in real time environment. These issues are addressed in the proposed methodology to add new features in the existing virtual data warehouse to enhance its functional capability.

3. Problem Definition

In the warehouse, data may be stored in aggregated form and classified accordingly. For example, a company may grade their business progress in the scale of –le to +le, where le ∈ ℕ. +le means the slope of the business curve is in a positive direction and the value of the rising slant is maximum. Similarly, -le means the slope of the business curve is in a negative direction and the falling slant is maximum. Now if the present slope of the business curve is in a positive direction and value of the slope is +a (where $a \in \mathbb{N}$ and $0 < a <= le$) then the data with a positive slope having the level range between $+(a + \partial)$ to $+(a - \partial)$ (where $\partial \in \mathbb{N}$ and $\partial << le$) is more relevant. But this type of relevant data analysis in real time is not an inherent property of traditional data warehouses. Hence for accurate and relevant data analysis, the data warehouse needs some practical changes. VDW is one of the solutions in order to create a business trend specific warehouse virtually that contains only those relevant data for quick processing. Hence, warehouse need to be sub-divide and store in different VDWs according to classified business trend. Data analysis needs to perform only upon matches relevant data present in VDW. This practice will discard irrelevant data processing that in turn reduces time complexity. In addition, recent data that comes from various sources of OLTP may or may not match between previous classifications of data. For example, the present business trend may not categorize between previously mentioned –le to +le grades. These unmatched data of recent time are treated as abnormal data. Abnormal situations may arise in the business environment for several reasons. If abnormality found above any permissible limit, what would be the necessary action to manage those abnormal situations is another dimension of this research work. Three cases are possible in this regard, the abnormality is either permanent or temporary or periodic. The first aim is to identify the type of abnormality then finding a solution for that specific type of abnormality. After this, the system will decide whether to include this in the data warehouse and to consider this as a normal business case for future computation otherwise not to include this in the warehouse and to consider it as a special case.

4. Methodology Selection

In order to match the present data into previously classified virtual data warehouses, the proposed methodology calculates z-score [1]. It is a measurement of score in terms of its distance from the mean, when measured in standard deviation units. That is, through z-score we can measure how many standard deviations below or above the population mean of any cluster a data is. Here we have chosen z-score as it handles outliers [1], but does not produce normalized data with the exact same scale. For example, a z-score +2 means 2 standard deviations above the mean, similarly a z-score +5 means 5 standard deviations above the mean and a z-score -3.5 means 3.5 standard deviations below the mean. And this research work is interested in finding whether the present business trend is above the normal business trend or below the normal business trend. Another reason for choosing z-score as the sample size is too large. In this research work, we have taken z-score as:

$$\text{z-score} = (x - \mu) / \sigma$$

Where, x = mean of grouped OLTP data.
μ = mean of i^{th} VDW data.
(Where, $0 <= i <=$ number of warehouse)
σ = standard deviation of i^{th} VDW data
(Where $0 <= i <=$ number of warehouse)

Here we have modified 'x' as to mean of grouped OLTP data as the formula demands ‘x’ as a single data element.

5. Proposed Methodology

A two-phase methodology is proposed in this work to perform the data analysis in order to identify different business cases under the virtual data warehouse framework. Thereafter we analyse whether the identified business cases are known cases or new business cases. Analysing these new business cases may generate new business intelligence for the organizations. In the 1st phase, in order to reduce the query-fetching time, it aims to create business trend specific warehouse virtually by subdividing the main warehouse data. Next, the goal is to map the OLTP data into previously created virtual data warehouses and perform a fast relevant query processing. If present OLTP data does not match with the previously created business trend specific virtual warehouses, these are treated as abnormal data. Handling of these abnormal situations of business in real-time is another challenge of this research in the 2nd phase.

5.1. Step-by-Step Description of Phase 1

The organization that keeps there valuable data in warehouse, need to analyse the entire warehouse for future business trend prediction. This practice increases time complexity as well as unnecessary irrelevant data analysis. In this phase, the proposed methodology aims to compute only relevant data for predicting the future business trend that reduces time as well as space complexity. Therefore, it needs to create a business trend specific warehouse virtually, so that we could able to analyse only the relevant data. In order to perform only relevant data analysis in real-time, the methodology proposes a modified architecture for data virtualization, which is shown in Figure 2. It computes the number of transactions per unit of time and groups them accordingly (For example data of 1 week / 10 days or according to the requirement). The methodology also computes the mean and standard deviation of each group. For example, in Figure 1 Time versus Sale graph sliced according to time unit (time slice is 1 unit).

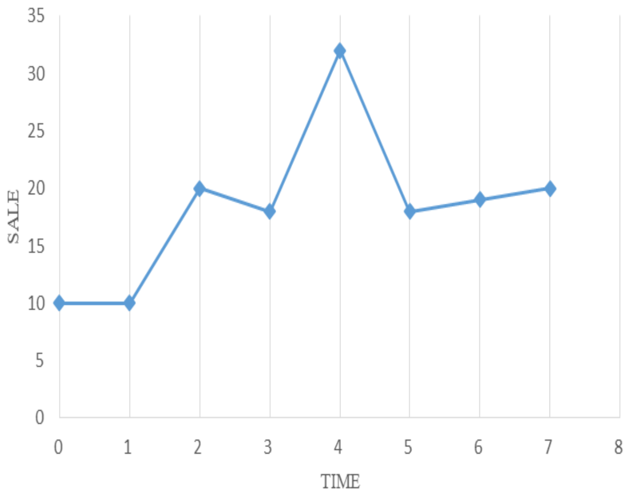


Figure 1. Time vs. sales graph.

According to the calculated mean, the methodology stores data slices virtually within n-number of separate VDWs (shown in Figure 2). One VDW may contain more than one data slices of the data warehouse as two or more data slice may have the same mean value. Data comes directly from the OLTP and a mapping technique introduced here to map the OLTP data into relevant VDW. The proposed methodology calculates z-score [1] of grouped OLTP data with respect to all VDWs. Finally, it maps OLTP data into proper VDWs according to the best match. Data analysis upon relevant data performed on a fly within these VDWs. At the same time, OLTP data mapped within those previously created n-numbers of VDWs need to be stored periodically in the EDW (Enterprise Data Warehouse) also. A batch processing technique is being employed here that will store those OLTP data into EDW after every certain interval.

It may also be possible that data fetches from OLTP do not match with any previously created VDWs. This situation treated as an abnormal situation.

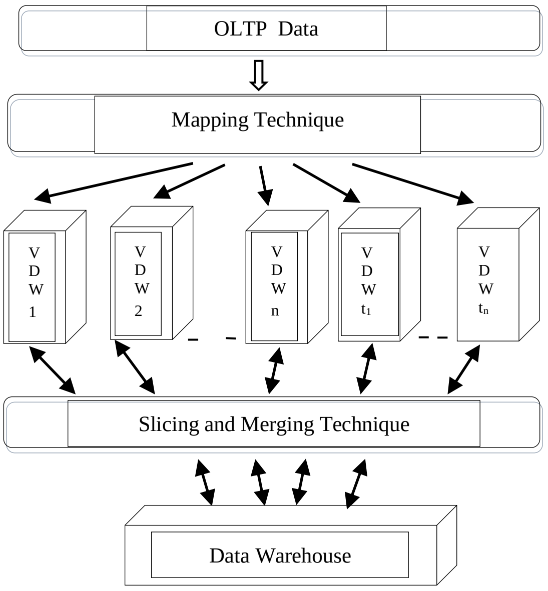


Figure 2. Proposed virtual data warehouse architecture.

Abnormal situations are causes due to abrupt hike or abrupt downfall of the business. In order to handle abrupt hike or abrupt downfall of the business, temporary virtual data warehouses VDW_{t1} to VDW_{tx} (where $x \in \mathbb{N}$, that depends upon the type of abnormality) is being proposed. These temporary virtual data warehouses are created for managing a similar type of abnormality in the future. That is, in future if any similar abnormality occurs, the system could analyse only relevant data and can make a prediction of business changes rapidly.

5.2. Algorithm

In this section we describe the proposed algorithm. The workflow diagram of it is depicted in Figure 3.

The main Algorithm-1: Real-time data Analysis is as follows

/* Input: Number of existing permanent Virtual Data Warehouses that need to create according to the business organizations.

Purpose: Fast real-time query execution with Relevant Virtual Data Warehouses even in crisis situation.

Output**:** 1) Number of increased permanent Virtual Data Warehouses with their specification.
2) Number of temporary Virtual Data Warehouses with their specification. ***/**

```
# n is the number of permanent VDWs
Take "n" as an input
# Slice Warehouse's data according to the time-stamp
and map them within n-number of permanent VDWs.
VDW_Creation( n )
# Feed recent data directly from OLTP and perform only
relevant data analysis in real-time.
match_OLTP(n, Permanent VDW's specification)
return (New permanent VDWs, Temporary VDWs)
```

Sub-Algorithm 1: VDW_Creation(number n)

/* Purpose: Slicing of Warehouse data according to time stamp and putting them into proper permanent VDWs.

Description: Data are arranged with respect to time. Here we create “n” number of VDWs, and the aim is to map k^{th} data slice into proper VDWs. "m" is a variable that is used to identify present number of permanent VDWs (m<=n). “t” is another variable that takes the unit of time slice from the user. “slice” represents the data slice that further maps into proper VDW. “msd” is an array of structure that holds the corresponding mean and standard deviation of each VDW. */

```
#Initialization
 t= take_unit_of_time_slice()
 p=1st element of data-warehouse w.r.t. time
    m=1
K=1

#Putting data slice into proper VDWs
while (p!= End of Warehouse data)
    {
```

```
slice=NULL
        for (i=1 to t)
        {
                slice=slice ∪ p
                p=p(next data)
        }
Calculate mean and Standard Deviation (SD) for this slice.
if (mean matches with any previous case)
        {
                Store this slice into matched VDWs
        }
        else
        {
                Store this slice into $m^{th}$ VDW
                m=m+1
        }
        k=k+1
}
# Storing mean, standard deviation of each VDWs
for (i=1 to n)
  {
  $msd_i$ =Calculate & store mean, standard deviation
(SD) for each $i^{th}$ VDW
   }
return (msd)
```

Sub-Algorithm 2: match_OLTP (number n, Permanent VDWs specification)

/* Purpose: Matching OLTP data into proper VDWs and real-time data analysis.
Description: "d" is an OLTP data slice grouped according to "t" of the previous algorithm. "m" is a variable that is being used for holding mean of data slice "d". "Z-Sc" is a one-dimensional array that is being used for holding calculated z-score of data slice "d" with respect to each previously created VDWs. "SD_i" is the standard deviation of i^{th} VDW. $VDW_{matched}$ is the set of matched VDWs with present OLTP data. In order to handle abnormal situations, this algorithm uses another sub-algorithm "track_abnormality", that is elaborated in phase 2 */

```
while(fetch_OLTP_data() = TRUE)
{
d=group data w.r.t. "t"
m= calculate mean of "d"
for( i=1 to n (number of VDW))
{
        Z-Sc[i]= ($mean_i$- m) / $SD_i$
}
match=false
$VDW_{matched}$ = NULL
for( i=1 to n (number of VDW))
{
        if (| Z-Sc[i]|< = threshold)
        {
              i) match=true
ii) The data in the $i^{th}$ VDW is relevant for predicting
business trend.
        iii)$VDW_{matched}$ =$VDW_{matched}$ ∪ $VDW_i$
        iv) Store this "$d_i$" (data slice) into $i^{th}$ VDW.
   }
}
if (match=true)
{
        # Normal Situation
Perform data analysis upon $DW_{matched}$
}
else
{
        # Abnormal Situation
        i) Create new temporary VDW with the new "$d_i$"
(data slice) mean.
        ii) Store this "$d_i$" (data slice) into this newly
        created temporary VDW.
        iii) track_abnormality (new temporary VDW)
                        # Discussed in phase 2
        iv) Perform data analysis upon nearest
        matched VDW in order to manage crisis
        situation.
  }
}
```

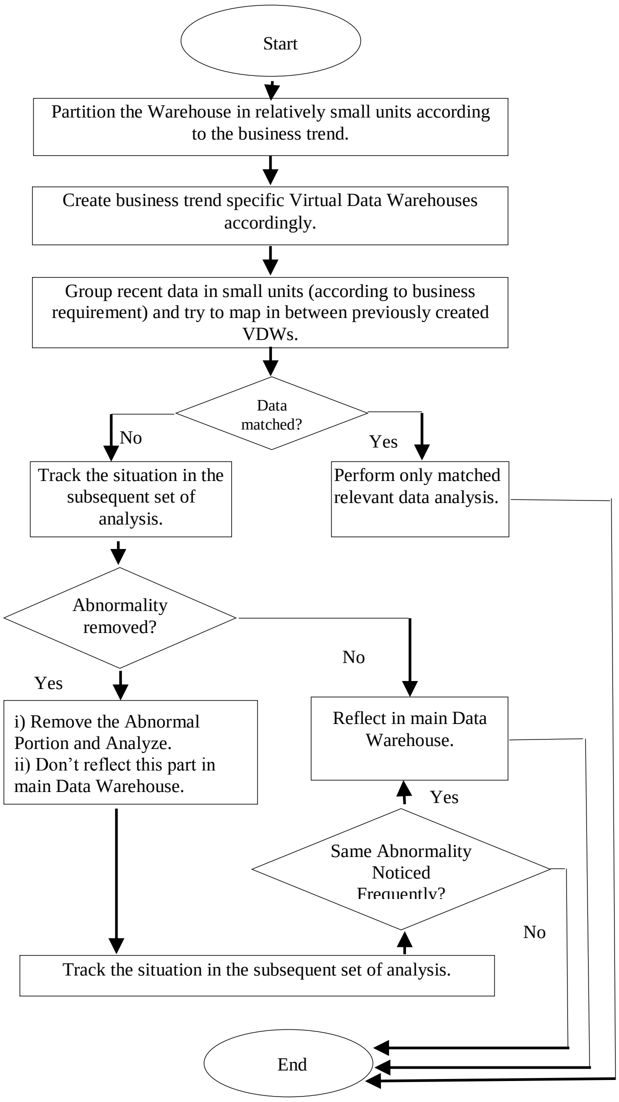

Start
Partition the Warehouse in relatively small units according to the business trend.
Create business trend specific Virtual Data Warehouses accordingly.
Group recent data in small units (according to business requirement) and try to map in between previously created VDWs.
Data matched?
No
Yes
Track the situation in the subsequent set of analysis.
Perform only matched relevant data analysis.
Abnormality removed?
Yes
No
i) Remove the Abnormal Portion and Analyze.
ii) Don't reflect this part in main Data Warehouse.
Reflect in main Data Warehouse.
Yes
Same Abnormality Noticed Frequently?
No
Track the situation in the subsequent set of analysis.
End

Figure 3. Workflow diagram of proposed methodology.

5.3. Step-by-Step Description of Phase 2

This phase deals with managing of abnormal situations. Present OLTP data that does not match with previously classified warehouse data treated as abnormal data. In order to identify the possible varying nature of data, the proposed methodology computes the z-score of those grouped OLTP data.

Due to the large set of sample sizes and known standard deviation, here we choose z-score. z-score more than a threshold value indicates an abnormal situation. The system tracks abnormal situations in a newly created separate VDW, but continue to process with the next data set. In a twinkling, if the system recovers from the abnormal situation, then it needs to analyse reasons for abnormality. This abnormal portion needs to be stored in a separate data cube, and it would not reflect in the Enterprise Data Warehouse (EDW) as it is a special case. In the future, if a similar type of abnormality noticed, then only this separate data cube will be analysed for managing the abnormal situations. That, in turn, provides abnormal situation-specific business analysis. However, if after some time period the system doesn't recover from the abnormal situation or the same abnormality is noticed frequently, it means that the abnormality is permanent, then the system should reflect it in the EDW. If any VDW remains temporary, for that the methodology suggests data analysis upon nearest matched virtual data warehouses. Hence, depending upon the situation, three different cases are possible.

- *Case* 1: Temporary Abnormal Pick

Consider the following situation as shown in Figure 4, where abnormality noticed for a twinkling only.

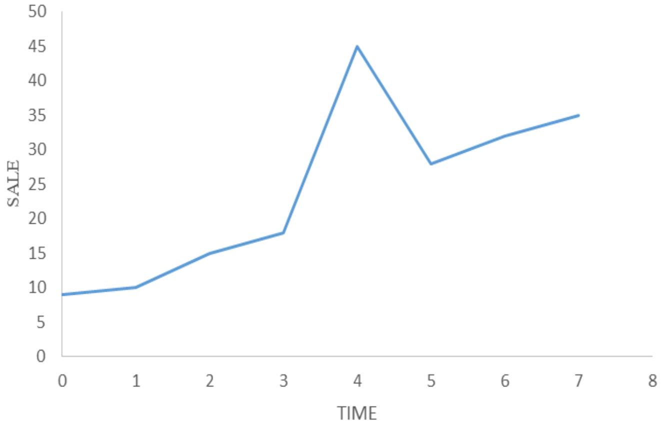


Figure 4. Abnormal pick.

Hence, this should not reflect in the data warehouse. However, it will reflect in the special VDW in order to handle this kind of situation in the future.

- *Case* 2: Abnormality is Permanent

Consider the following situation as shown in Figure 5, where abnormality is noticed and that persists for a long.

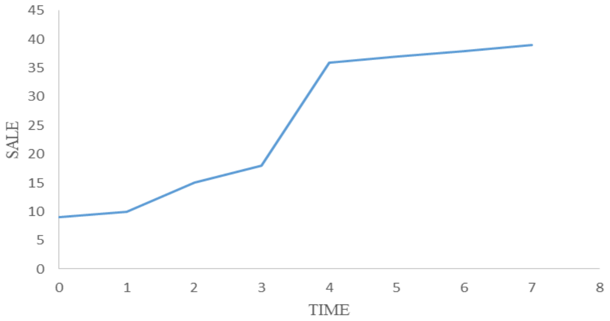


Figure 5. Abnormality is permanent.

Hence, abnormality is permanent and therefore needs to reflect it in VDW as well as in the main warehouse.

- *Case* 3: Temporary Abnormal Pick, but Frequently

Consider the following situation where the time vs. sale graph of a particular product is shown in Figure 6:

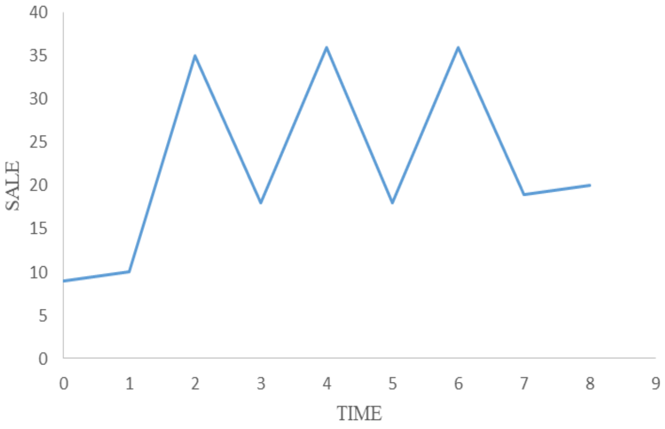


Figure 6. Temporary Abnormal Picks, but Frequently Occurring

Abnormality is noticed temporarily in Figure 6 but occurring frequently. Hence, these parts need to reflect in the VDW as well as in the main warehouse.

Sub-Algorithm 3: Tracking and maintaining abnormal situations

track_abnormality(Temporary VDW)

/* Purpose: It is used for tracking and maintaining abnormal situations. That is to manage abnormality in business trend.

Description: This is a sub-method, called from the method “match_OLTP” of phase 1. */

```
if (Consecutive two hit in temporary VDW)
{
     # The abrupt change is permanent
     Need to reflect this temporary VDW in main
          Warehouse
}
else if (hit in temporary VDW ≥ “Some threshold
          number of hit”)
{
     # Abnormality is frequent
```

```
        Need to reflect this VDW in main Warehouse
 }
else
{
        # The VDW remains temporary
                No operations are required
   }
```

6. Case Study

We have applied our proposed methodology upon Amazon (a pioneer online retailer) review data [2] span of 22 years, available in "https://nijianmo.github.io/amazon/index.html". As these are only historical data set, time period in the range between May 1996 - Oct 2018 (time period vary for data set to data set), we have subdivided the data set as training data (1st 80% of the dataset w.r.t. time) and test data (rest 20%). We have considered training data as warehouse data and create VDWs by classifying this according to the business trend and have considered test data as OLTP data. All experiments are conducted on a Intel Core i5-8250U processors, 16 GB RAM along with 50 GB SSD is used as RAM (Hence total RAM used is 66 GB). The operating system is Linux Mint 19.2 Tina. Different Software required for this experimental environment are Python 3.6.9, scikit-learn 0.21.3 used for machine learning, Matplotlib 3.1.2 used for data visualization, pandas 0.25.3 used for large data manipulation and python-dateutil 2.8.1 used to manipulate date-time related data.

- **Case 1**

We have chosen a relatively large data set "Electronics" as a first case. The "Time" vs. "Sale" graph is shown in Figure 7 (Blue: Training Data, Green: Test data). It contains data span of June 1999 to July 2014. Here, the first dimension represents "Time" and the second dimension represents "Sale". We slice training data with seven days' span and create initially 10 permanent VDWs according to the proposed algorithm. The mean and standard deviation of each VDW is shown in Figure 8.

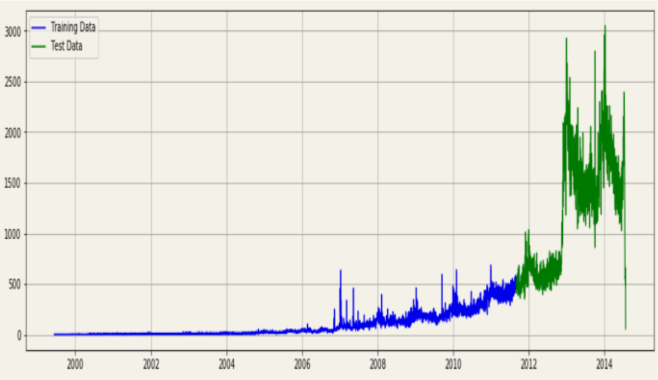
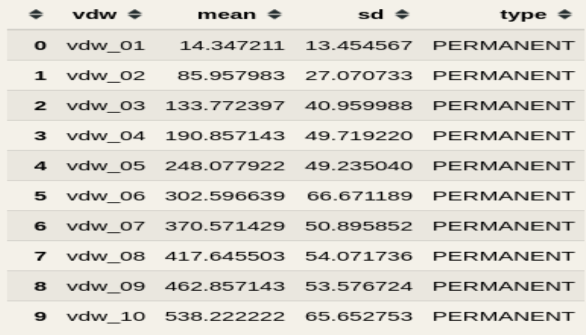


Figure 7. "Time" vs. "Sale" graph (snapshot).



	vdw	mean	sd	type
0	vdw_01	14.347211	13.454567	PERMANENT
1	vdw_02	85.957983	27.070733	PERMANENT
2	vdw_03	133.772397	40.959988	PERMANENT
3	vdw_04	190.857143	49.719220	PERMANENT
4	vdw_05	248.077922	49.235040	PERMANENT
5	vdw_06	302.596639	66.671189	PERMANENT
6	vdw_07	370.571429	50.895852	PERMANENT
7	vdw_08	417.645503	54.071736	PERMANENT
8	vdw_09	462.857143	53.576724	PERMANENT
9	vdw_10	538.222222	65.652753	PERMANENT

Figure 8. Mean and Standard deviation of initially created VDWs.

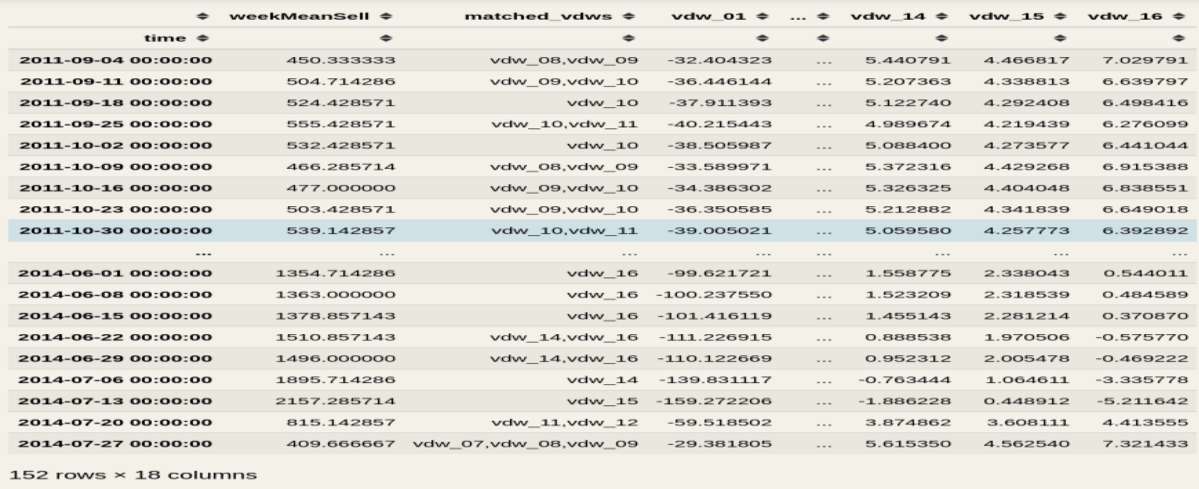
Similarly, we slice test data with seven days' span and try to map those slices with previously created VDWs. If test data matches with previously created VDWs then we analyse only those matched portions for business trend analysis and for other decision-making policy, otherwise, create separate VDW for that part containing those unmatched data. In order to match OLTP data with previously created VDWs, the proposed methodology computes z-score of each OLTP data slice, considering the mean of OLTP data slice as a single data and try to map that in any of the previously created VDWs. If the z-score is less than one, then data of those VDWs are relevant for analysis. A snapshot of intermediate test result is shown in Figure 9 that represents OLTP data slices and their corresponding match to each previously created VDWs. Present OLTP data may match with one or more VDWs or not at all as shown in the Figure 9 (a snapshot).

Next, we have applied the query "Find average customer satisfaction when the number of reviews given is the same as OLTP data". When the warehouse not sliced according to the business trend, the time required to execute this query is 6.55 second as shown in the Figure 10.

When the warehouse sliced virtually according to the proposed methodology, the time required to execute the query is only 4 milliseconds (with matched vdw_07, vdw_08, vdw_09) as shown in the Figure 11.

Hence, it clearly shows that the warehouse sliced virtually according to business trend specific provides more than a thousand times faster processing.

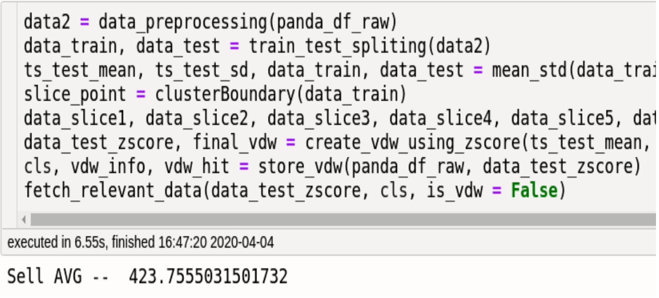
Next thing is to check whether there is any abnormality in test data or not, that is, to check whether all OLTP data matches with previously created VDWs or not. In addition, if any abnormality found then check whether this is permanent or a special case. Among the test data, due to consecutive two hit or frequent hit, the algorithm suggests that there is a need of sixteen permanent VDWs and no temporary VDW as shown in Figure 12. Therefore in this case, six new business cases are identified as shown in Figure 12. Among this six new VDWs, all would be reflected in main data warehouse as all are permanent.



	weekMeanSell	matched_vdws	vdw_01	...	vdw_14	vdw_15	vdw_16
time							
2011-09-04 00:00:00	450.333333	vdw_08,vdw_09	-32.404323	...	5.440791	4.466817	7.029791
2011-09-11 00:00:00	504.714286	vdw_09,vdw_10	-36.446144	...	5.207363	4.338813	6.639797
2011-09-18 00:00:00	524.428571	vdw_10	-37.911393	...	5.122740	4.292408	6.498416
2011-09-25 00:00:00	555.428571	vdw_10,vdw_11	-40.215443	...	4.989674	4.219439	6.276099
2011-10-02 00:00:00	532.428571	vdw_10	-38.505987	...	5.088400	4.273577	6.441044
2011-10-09 00:00:00	466.285714	vdw_08,vdw_09	-33.589971	...	5.372316	4.429268	6.915388
2011-10-16 00:00:00	477.000000	vdw_09,vdw_10	-34.386302	...	5.326325	4.404048	6.838551
2011-10-23 00:00:00	503.428571	vdw_09,vdw_10	-36.350585	...	5.212882	4.341839	6.649018
2011-10-30 00:00:00	539.142857	vdw_10,vdw_11	-39.005021	...	5.059580	4.257773	6.392892
...	...	...	...	...	...	...	...
2014-06-01 00:00:00	1354.714286	vdw_16	-99.621721	...	1.558775	2.338043	0.544011
2014-06-08 00:00:00	1363.000000	vdw_16	-100.237550	...	1.523209	2.318539	0.484589
2014-06-15 00:00:00	1378.857143	vdw_16	-101.416119	...	1.455143	2.281214	0.370870
2014-06-22 00:00:00	1510.857143	vdw_14,vdw_16	-111.226915	...	0.888538	1.970506	-0.575770
2014-06-29 00:00:00	1496.000000	vdw_14,vdw_16	-110.122669	...	0.952312	2.005478	-0.469222
2014-07-06 00:00:00	1895.714286	vdw_14	-139.831117	...	-0.763444	1.064611	-3.335778
2014-07-13 00:00:00	2157.285714	vdw_15	-159.272206	...	-1.886228	0.448912	-5.211642
2014-07-20 00:00:00	815.142857	vdw_11,vdw_12	-59.518502	...	3.874862	3.608111	4.413555
2014-07-27 00:00:00	409.666667	vdw_07,vdw_08,vdw_09	-29.381805	...	5.615350	4.562540	7.321433

152 rows × 18 columns

Figure 9. z-score of each OLTP data slice with VDWs (snapshot).

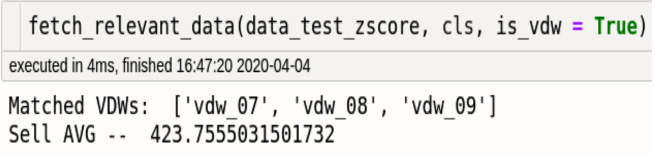


```
data2 = data_preprocessing(panda_df_raw)
data_train, data_test = train_test_spliting(data2)
ts_test_mean, ts_test_sd, data_train, data_test = mean_std(data_trai
slice_point = clusterBoundary(data_train)
data_slice1, data_slice2, data_slice3, data_slice4, data_slice5, dat
data_test_zscore, final_vdw = create_vdw_using_zscore(ts_test_mean,
cls, vdw_info, vdw_hit = store_vdw(panda_df_raw, data_test_zscore)
fetch_relevant_data(data_test_zscore, cls, is_vdw = False)
```

```
executed in 6.55s, finished 16:47:20 2020-04-04
```

```
Sell AVG --  423.7555031501732
```

Figure 10. Time required to execute a query when warehouse not sliced according to the business trend (snapshot).

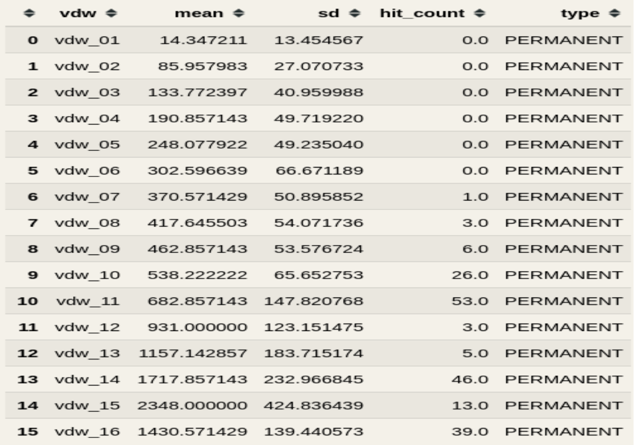


```
fetch_relevant_data(data_test_zscore, cls, is_vdw = True)
```

```
executed in 4ms, finished 16:47:20 2020-04-04
```

```
Matched VDWs:  ['vdw_07', 'vdw_08', 'vdw_09']
Sell AVG --  423.7555031501732
```

Figure 11. Time required to execute a query when warehouse sliced according to the business trend (snapshot).

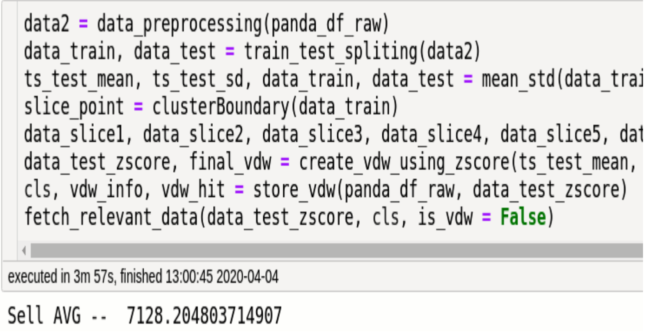


	vdw	mean	sd	hit_count	type
0	vdw_01	14.347211	13.454567	0.0	PERMANENT
1	vdw_02	85.957983	27.070733	0.0	PERMANENT
2	vdw_03	133.772397	40.959988	0.0	PERMANENT
3	vdw_04	190.857143	49.719220	0.0	PERMANENT
4	vdw_05	248.077922	49.235040	0.0	PERMANENT
5	vdw_06	302.596639	66.671189	0.0	PERMANENT
6	vdw_07	370.571429	50.895852	1.0	PERMANENT
7	vdw_08	417.645503	54.071736	3.0	PERMANENT
8	vdw_09	462.857143	53.576724	6.0	PERMANENT
9	vdw_10	538.222222	65.652753	26.0	PERMANENT
10	vdw_11	682.857143	147.820768	53.0	PERMANENT
11	vdw_12	931.000000	123.151475	3.0	PERMANENT
12	vdw_13	1157.142857	183.715174	5.0	PERMANENT
13	vdw_14	1717.857143	232.966845	46.0	PERMANENT
14	vdw_15	2348.000000	424.836439	13.0	PERMANENT
15	vdw_16	1430.571429	139.440573	39.0	PERMANENT

Figure 12. Final result with sixteen permanent VDWs.

- **Case 2**

We have chosen another relatively large data set "Cell_Phone_and_Accessories" as the second case from the same link mentioned in case study 1. It contains data span of October 1999 to October 2018. In the same way, we have created initially 10 permanent VDWs according to the proposed algorithm. Next, we have tested the same query "Find average customer satisfaction when the number of reviews given is the same as OLTP data". Due to the space limitation, we skip the intermediate results and show the final outputs. We found that, when the warehouse not sliced according to the business trend, the time required to run this query is 3 minute 57 seconds (shown in Figure 13). When the warehouse sliced virtually according to proposed methodology, the time required to run the same query is 247 milliseconds (with matched vdw_08 and vdw_09). Shown in Figure 14. Hence, it provides around a thousand times faster query processing.

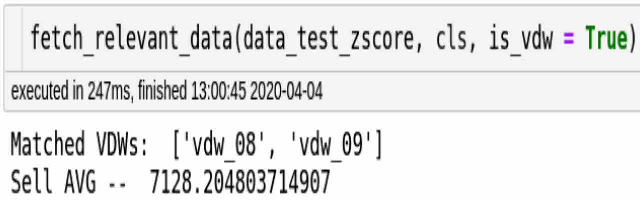


```
data2 = data_preprocessing(panda_df_raw)
data_train, data_test = train_test_spliting(data2)
ts_test_mean, ts_test_sd, data_train, data_test = mean_std(data_trai
slice_point = clusterBoundary(data_train)
data_slice1, data_slice2, data_slice3, data_slice4, data_slice5, dat
data_test_zscore, final_vdw = create_vdw_using_zscore(ts_test_mean,
cls, vdw_info, vdw_hit = store_vdw(panda_df_raw, data_test_zscore)
fetch_relevant_data(data_test_zscore, cls, is_vdw = False)
```

```
executed in 3m 57s, finished 13:00:45 2020-04-04
```

```
Sell AVG --  7128.204803714907
```

Figure 13. Time required to execute a query when warehouse not sliced according to the business trend (snapshot for case 2).



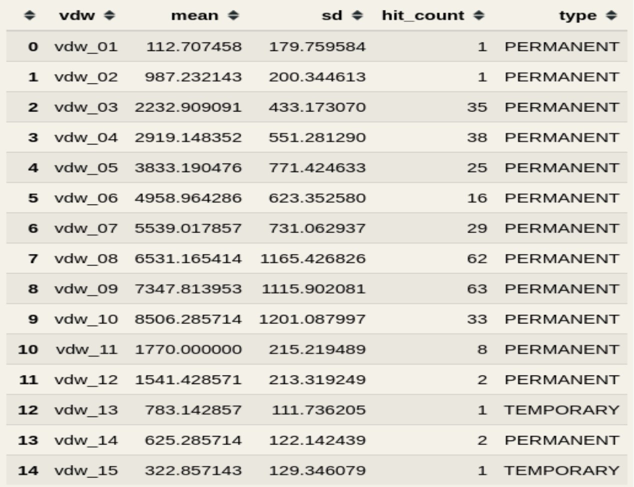
```
fetch_relevant_data(data_test_zscore, cls, is_vdw = True)
```

```
executed in 247ms, finished 13:00:45 2020-04-04
```

```
Matched VDWs:  ['vdw_08', 'vdw_09']
Sell AVG --  7128.204803714907
```

Figure 14. Time required to execute a query when warehouse sliced according to the business trend (snapshot for case 2).

Next, we check whether there is any abnormality in test data or not. The algorithm suggests that there is a need to make fifteen virtual data warehouses (As shown in Figure 15).



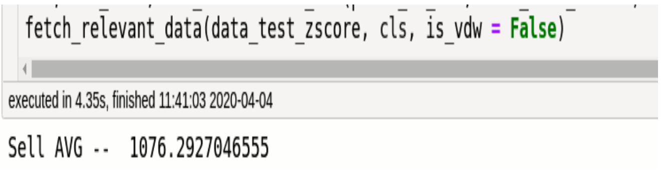
	vdw	mean	sd	hit_count	type
0	vdw_01	112.707458	179.759584	1	PERMANENT
1	vdw_02	987.232143	200.344613	1	PERMANENT
2	vdw_03	2232.909091	433.173070	35	PERMANENT
3	vdw_04	2919.148352	551.281290	38	PERMANENT
4	vdw_05	3833.190476	771.424633	25	PERMANENT
5	vdw_06	4958.964286	623.352580	16	PERMANENT
6	vdw_07	5539.017857	731.062937	29	PERMANENT
7	vdw_08	6531.165414	1165.426826	62	PERMANENT
8	vdw_09	7347.813953	1115.902081	63	PERMANENT
9	vdw_10	8506.285714	1201.087997	33	PERMANENT
10	vdw_11	1770.000000	215.219489	8	PERMANENT
11	vdw_12	1541.428571	213.319249	2	PERMANENT
12	vdw_13	783.142857	111.736205	1	TEMPORARY
13	vdw_14	625.285714	122.142439	2	PERMANENT
14	vdw_15	322.857143	129.346079	1	TEMPORARY

Figure 15. Final result with fifteen VDWs (snapshot for case 2).

Among these VDWs, three (vdw_11, vdw_12, vdw_14) would be reflected in main data warehouse as all are permanent and two (vdw_13, vdw_15) would not be reflected in main data warehouse as these are temporary. It is shown in Figure 15. These temporary virtual data warehouses would be used in future if any similar abnormality occurs, the system could analyse only that relevant data and can make a prediction of business changes rapidly.

- **Case 3**

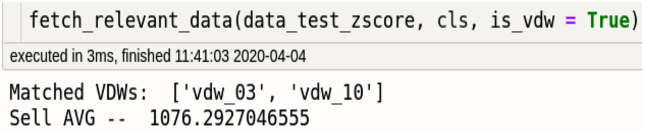
We have chosen another relatively small data set “Amazon_Fashion” as third case in order to check whether the proposed algorithm is also effective in relatively small dataset or not. It contains data span of November 2002 to October 2018. In the same way, we have created initially 10 permanent VDWs. Next, we have tested the same query. We found that, when the warehouse not sliced according to the business trend, the time required to run the same query is 4.35 seconds. It is shown in Figure 16.



```
fetch_relevant_data(data_test_zscore, cls, is_vdw = False)
executed in 4.35s, finished 11:41:03 2020-04-04
Sell AVG --  1076.2927046555
```

Figure 16. Time required to execute a query when warehouse not sliced according to the business trend (for case 3), a snapshoot.

When the warehouse sliced virtually according to proposed methodology, the time required to run the same query is 3 milliseconds only (with matched vdw_03 and vdw_10). It is shown in Figure 17.

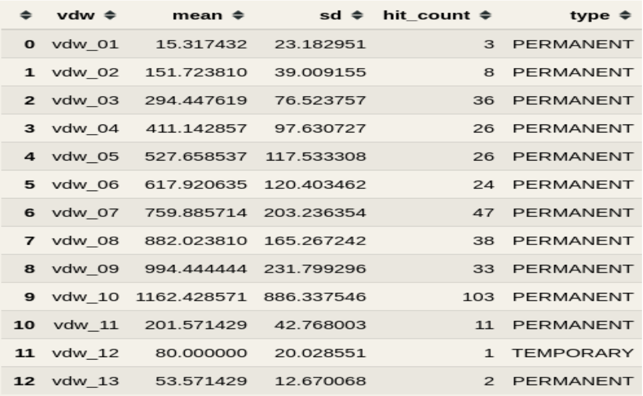


```
fetch_relevant_data(data_test_zscore, cls, is_vdw = True)
executed in 3ms, finished 11:41:03 2020-04-04
Matched VDWs:  ['vdw_03', 'vdw_10']
Sell AVG --  1076.2927046555
```

Figure 17. Time required to execute a query when warehouse sliced according to the business trend (for case 3), a snapshoot.

Hence in this case also, the warehouse sliced virtually according to specific business trend provides more than a thousand times faster processing.

Next, we need to check whether there is any abnormality in test data or not. By using the test data, the algorithm suggests that there is a need to make total thirteen virtual data warehouses as shown in Figure 18. That is three new business cases are identified. Among these, two (vdw_11 and vdw_13) would be reflected in main data warehouse as all are permanent and one (vdw_12) would not be reflected in main data warehouse as this is temporary. It is shown in Figure 18. This temporary virtual data warehouse would be used in future, if any similar abnormality occurs, the system could analyse only that relevant data and can make a prediction of business changes very quickly.



	vdw	mean	sd	hit_count	type
0	vdw_01	15.317432	23.182951	3	PERMANENT
1	vdw_02	151.723810	39.009155	8	PERMANENT
2	vdw_03	294.447619	76.523757	36	PERMANENT
3	vdw_04	411.142857	97.630727	26	PERMANENT
4	vdw_05	527.658537	117.533308	26	PERMANENT
5	vdw_06	617.920635	120.403462	24	PERMANENT
6	vdw_07	759.885714	203.236354	47	PERMANENT
7	vdw_08	882.023810	165.267242	38	PERMANENT
8	vdw_09	994.444444	231.799296	33	PERMANENT
9	vdw_10	1162.428571	886.337546	103	PERMANENT
10	vdw_11	201.571429	42.768003	11	PERMANENT
11	vdw_12	80.000000	20.028551	1	TEMPORARY
12	vdw_13	53.571429	12.670068	2	PERMANENT

Figure 18. Final result with thirteen VDWs (snapshot for case 3).

7. Comparative Study

In this section, we compared the time required to execute the same query in the warehouse where the data are not sliced [3, 10, 14] according to business trend and the same in proposed business trend specific sliced VDW for the above three case studies. Experimental result shows that, when warehouse data are not sliced according to business trend, it requires 6550 milliseconds in the 1st case, and that of 2nd and 3rd case are 237000 milliseconds and 4350 milliseconds respectively. But, when warehouse data are sliced according to business trend, it requires only 4 milliseconds for the 1st case and that of 2nd and 3rd case are 247 milliseconds and 3 milliseconds respectively. Therefore, its improvement in 1st case is about 1637 times better and that of 2nd and 3rd case are around 960 times and 1450 times better respectively. Therefore, the proposed methodology is providing around 10^3 times faster query processing. The timing comparison is shown in Figure 19. At the same time, as the proposed methodology works only with the matched relevant VDWs, therefore the requirement of resources like primary memory, CPU burst time, etc. will be reduced. Hence, overall computational cost will be reduced proportionally.

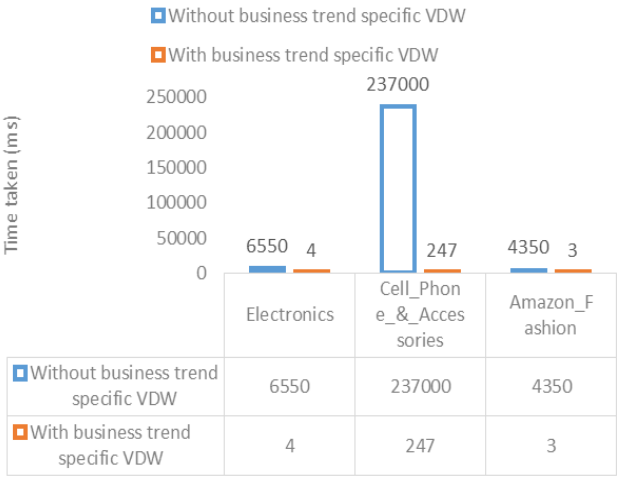


Figure 19. Comparison (time required to execute same query).

8. Conclusions

This research work is aimed to enhance the functionality of Virtual Data Warehouse to work in real time for specific business trends. It includes the functionalities like managing unidentified business cases, identifying the new business cases and also deciding whether these new cases will be included in the main data warehouse or not. This will help to take more decisions directly from virtual data warehouse and therefore the analysis on the same business cases will not be required further. The proposed methodology is applied on different real life data and the experimental results reveal around 10^3 times faster execution over traditional techniques. As the computation speed improves drastically the hardware cost will be reduced to set up the virtual data warehouse environment.

This work can be extended to incorporate the maintenance of VDWs in real-time that will avoid the formation of VDW from scratch. Another extension of this work is to consider data from different big data sources to measure the performance benefits.

References

[1] Aggarwal V., Gupta V., Singh P., Sharma K., and Sharma N., "Detection of Spatial Outlier by Using Improved Z-Score Test," *in Proceedings of International Conference on Trends in Electronics and Informatics*, Tirunelveli, pp. 788-790, 2019.

[2] Amazon Review Data. Link: https://nijianmo.github.io/amazon/index.html., Last Visited, 2020.

[3] Asadullaev S., *Data Warehouse Architectures and Development Strategy*, IBM Companion Guidebook, Open Group, 2015.

[4] Azvine B., Cui Z., Nauck D., and Majeed B., "Real Time Business Intelligence for the Adaptive Enterprise," *in Proceedings of 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, San Francisco, pp. 29-29, 2006.

[5] Chandramouly A., Patil N., Ramamurthy R., Krishnan S., and Story J., *Integrating Data Warehouses with Data Virtualization for BI Agility*, Intel White Paper, 2013.

[6] Conn S., "OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures for Real Time Data Analysis," *in Proceedings of IEEE SoutheastCon*, Ft. Lauderdale, pp. 515-520, 2005.

[7] Dahmani D., Rahal S., and Belalem G., "A New Approach to Improve Association Rules for Big Data in Cloud Environment," *The International Arab Journal of Information Technology*, vol. 16, no. 6, pp. 1013-1020, 2019.

[8] Davis R. and Eve R., *Data Virtualization: Going Beyond Traditional Data Integration to Achieve Business Agility*, Nine Five One Press, United States, 2011.

[9] Ghosh P., Sadhu D., Sen S., and Debnath N., "Service Modelling for Virtual Data Warehouse," *in Proceedings of International Conference on Computer Applications in Industry and Engineering*, San Diego, 2017.

[10] Ghosh P., Som S., and Sen S., "Business Intelligence Development by Analysing Customer Sentiment," *in Proceedings of IEEE International Conference on Reliability, Infocom Technologies and Optimization*, Noida, pp. 287-290, 2018.

[11] Goss R. and Veeramuthu K., "Heading Towards Big Data Building A Better Data Warehouse For More Data, More Speed, And More Users," *in Proceedings of Advanced Semiconductor Manufacturing Conference*, Saratoga, pp. 220-225, 2013.

[12] Guo S., Yuan Z., Sun A., and Yue Q., "A new ETL Approach Based On Data Virtualization," *Journal of Computer Science and Technology*, vol. 30, pp. 311-323, 2015.

[13] Gupta A. and Sahayadhas A., "Proposed Techniques to Optimize the DW and ETL Query for Enhancing Data Warehouse Efficiency," *in Proceedings of International Conference on Computing, Communication and Security*, Patna, pp. 1-5, 2020.

[14] Katkar V., Gangopadhyay S., Rathod S., and Shetty A., "Sales Forecasting Using Data Warehouse and Naïve Bayesian Classifier," *in Proceedings of International Conference on Pervasive Computing*, Pune, pp. 1-6, 2015.

[15] Kholod I., Efimova M., and Kulikov S., "Using ETL Tools for Developing a Virtual Data Warehouse," *in Proceedings of IEEE International Conference on Soft Computing and*

Measurements, St. Petersburg, pp. 351-354, 2016.

[16] Lans R., *Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses*, Morgan Kaufmann, 2012.

[17] Mousa A., Shiratuddin N., and Bakar M., “Virtual Data Mart for Measuring Organizational Achievement Using Data Virtualization Technique (KPIVDM),” *Jurnal Teknologi*, vol. 68, no. 3, pp. 67-70, 2014.

[18] Ni J., Li J., and McAuley J., “Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects,” *in Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, pp. 188-197, 2019.

[19] Sahay B. and Ranjan J., “Real Time Business Intelligence in Supply Chain Analytics,” *Information Management and Computer Security*, vol. 16, no. 1, pp. 28-48, 2008.

[20] Salem R., Salesh S., and AbdulKader H., “Intelligent Replication for Distributed Active Real-Time Databases Systems,” *The International Arab Journal of Information Technology*, vol. 15, no. 3, pp. 505-519, 2018.

[21] Shukla A., Kumar A., and Singh H., “ANN Based Execution Time Prediction Model and Assessment of Input Parameters through ISM,” *The International Arab Journal of Information Technology*, vol. 17, no. 5, pp. 683-691, 2020.

Partha Ghosh is pursuing his Ph.D. from University of Calcutta, Kolkata, India. Also, he is working as an Assistant Professor in the Computer Applications Department at B. P. Poddar Institute of Management & Technology, Saltlake Campus, Kolkata, India. His research areas are Data Warehouse, Big data, Machine Learning and Business intelligence.

Deep Sadhu is working as a Data Scientist at EY (Ernst & Young), dealing with Machine Learning, Deep Learning & Big Data techniques.

Soumya Sen is an Assistant professor in A. K. Choudhury School of Information Technology under University of Calcutta, Kolkata, India. His research areas are Data warehouse & OLAP tools, Distributed database, Big data, Machine learning.