RULE BASED ETL (RETL) APPROACH FOR GEO SPATIAL DATA WAREHOUSE

Norhaira Nordin¹, Azman Yasin² and Mazni Omar³
¹Universiti Utara Malaysia(UUM), norhairanordin@gmail.com
²Universiti Utara Malaysia(UUM), yazman@uum.edu.my
³Universiti Utara Malaysia(UUM), mazni@uum.edu.my

ABSTRACT

This paper presents the use of Service Oriented Architecture (SOA) for integrating multisource heterogeneous geospatial data in order to facilitate geospatial data warehouse. In this study, Real Based ETL (RETL) concept is adapted in order to extract, transform and load data from a variety of heterogeneous data sources. ETL will transform data to schematic format and loading data into the Geospatial data warehouse. By using a rule-based technique, the distribution of parallel ETL pipeline will enhance and perform more efficient in large scale of data and overcome data bottleneck and performance overhead. This can ease the disaster management and enables planners to monitor disaster emergency response in an efficient manner.

Keywords: Geographical Information System, Service Oriented Architecture, Extract Transform Load, Pipeline, Rule Based.

I INTRODUCTION

In recent times, we often see extreme changes in meteorological phenomena such as floods, landslide, and tsunami. This extreme change has contributed to the utilization of information technology to facilitate disaster management and enables planners to work primarily from monitoring form to disaster emergency response more effective. The critical task of disaster management system is the geographical information (GI) itself which consists of large amount of data collected from satellite, airborne, and ground based imaging systems from multi-platform, multi-source, multi-temporal image and data manipulation to facilitate decision making (Song, Fang, Chen, & Yang, 2011; Yvan Bédard & Han, 2009).

In order to deal with the integration of multisource heterogeneous geospatial data, the design for data integration is based on a Service Oriented Architecture (SOA) (Niu, Guo, Lu, & Chen, 2011; Wenfeng, 2010; Zheng, He, Xiong, Liu, & Framework, 2011; Zheng, Huang, & Zhang, 2010). The architecture of SOA mainly composed of data layer, services layer and display layer (Wang, Liu, Wang, & Xu, 2011) which can offer the flexibility to handle different task and provide services and the separation of business logics from the foreground and background (Feng & Lee, 2010; Zheng, Huang, & Zhang, 2010). In Figure 1, it is shown SOA framework and spatial information services in GI (Ying, Yang, Ya-fu, & Li-zhou, 2010).
transform, derived column transform and so on (Tok, Parida, Masson, Ding, & Sivashanmugam, 2012). Thus, to increase the data integration performance of geospatial data warehouse, this study proposed the use of rule-based concept of ETL pipeline (RETL) in Geo Spatial SOA Framework. In this study, RETL is focused on performance in parallel ETL pipelines for extracting data and actions from operational databases and loading them into enterprise geospatial data warehouse. The remainder of this paper is organized as follows. In the section II, we present SOA spatial data integration and following section III presents the proposed RETL concept. Sections IV the expected result and section V will summarize our conclusion and action for future work.

II SERVICE ORIENTED SPATIAL DATA INTEGRATION

Spatial data warehousing for geographic knowledge often have several departmental or application-oriented independent databases which may overlap in content. By using Service Oriented Architecture (SOA), the integrated platform can supports various distributed resource from several geographical server (Wang et al., 2011). SOA is particularly consisting of data layer, service layer, and display layer. In Figure 2, it is shown Proposed SOA framework and Spatial Data Warehouse.

A. Data Layer
The data layer mainly includes the fundamental geographic information, the Aerial photo, satellite image, and the interpretation information of remote sensingsuch as population data, socio-economic data, water data, and weather data and so on.

B. Service Layer
The service layers mainly distribute and manage the various types of services, establish the links between application layer and services, datalayer, and access the distributed multi-source heterogeneous data. The ETL system are used for extracting business data, transform and cleansing it into an analytical format.

C. Display Layer
Display layer is designed to visualized information from multisource spatial data directly use to end user and the corresponding function.

Taking into account on the complexity of spatial data, mass characteristics, data bottleneck, and volume of GI data, the execution efficiency need to be improve. Next, we will focus on SOA service layer for spatial data integration by enhancing ETL pipeline job.

III PROPOSED RULE BASED ETL (RETL)

One of the most difficult processes during the creating any data warehouse is extracting, transforming, cleansing and loading data processes from data source to warehouse destination (Santos, Bernardino, & Vieira, 2011).

A. ETL Technology
The Extract-Transform-Load (ETL) system is the foundation of the data warehouse (Kimball & Caserta, 2004). ETL is typically used to integrate data from a variety of applications, developed and supported by different server or hosted on separate computer hardware. It integrates the scattered, disorder and heterogeneous data into the target data warehouse, which is used by different business systems. The ETL process start with retrieves data from data source then processes the data by enforces data quality and consistency standards, conforms data so that separate sources can be used together and finally loads it into the database. The following steps are process in ETL: In Figure 3, it is shown detail ETL process (Kimball & Caserta, 2004).

B. Rule Based ETL (RETL) Architecture
The current ETL system allows parallel data processing and aggregation which can scale up to the most demanding systems. In many studies, the parallel execution is distributed for different ETL task process (J. Song, Bao, & Shi, 2010). The proposed
RETL will introduce the motivated ideas and the background of ETL Rule Based and Parallel Pipeline.

C. RETL Architecture
The proposed RETL contains five main components:
- Listener
- Triggering Rules
- ETL Job Dispatcher
- ETL Rule Based
- ETL Engine

In figure 4, it is shown proposed RETL architecture.

RETL starts when the triggering rules or messaging mechanism notifies the listener to analyzing new data source to be extract. The triggering rules will identify the ETL jobs defined by users, analyzing the syntax jobs and checking the trigger method rules of ETL job such as BEGIN, CHECK, INIT and END so the RETL system can define the next phase of ETL flow. Next, the job dispatcher will determine which ETL pipeline task is used according to user-deﬁne pipeline process. The Rule Based is responsible for dividing data in a batch and set how many pipelines will be used to run the ETL task in parallel. Our proposed component will be ETL Rule based and Parallel Pipeline.

The monitoring RETL system is based on the analysis of logs generated SQL Integration module, so the status of RETL system can be monitored when deploys it on production environment.

D. ETL RULE BASED
ETL Rule based will optimize the distribute resource in parallel pipeline. There are maximum N thread access data in parallel pipeline. (Basco, Gerardo, Dofitas Jr., & Byun, 2012; Mendes & Falcone Sampaio, 1998)

E. PARALLEL PIPELINE
Parallel processing divides a large task into many smaller tasks, and executes the smaller tasks concurrently on several nodes. As a result, the larger task completes more quickly. There are three points need to be considering while setting the rule base for parallel processing:
- The effectiveness for main memory and CPU usages.
- Optimization problem that relates the volume of data to be propagated from a data source towards the data processing area.
- In several operations might be mutually exclusive and don’t have any dependencies.

Step 1: We have to specify the pipeline task can work parallel or not. It depends on GI data whether it dependent or independent data.

Step 2: A set of rules was used based on the individual core’s capabilities and how many threads, total size of threads. Below in Figure 6 is an illustration of the rules of how a thread is assigned to a processor or core (Basco et al., 2012).

Step 3: Rule based will be sent to the ETL to run the ETL system.

Check if thread size + Processor1.taskload ≤ SafeLoadLimit
If thread size + Processor1.taskload < SafeLoadLimit, Processor 1 will receive the thread, else the thread will be given to Processor 2 to be evaluated.
If thread size + Processor2.taskload < SafeLoadLimit, Processor 2 will receive the thread, else the thread will be given to Processor 3 to be evaluated.
If thread size + Processor3.taskload < SafeLoadLimit, Processor 3 will receive the thread, else the thread will be given to Processor 4 to be evaluated.
If thread size + Processor4.taskload < SafeLoadLimit, Processor 4 will receive the thread, else the thread will be given back to Processor regardless condition # 2.

Figure 5. Parallel Optimization Diagram
Figure 6. Example of rules assign to a processor
among each other. These operations can easily be parallelized.

Loading or querying massive data from into one table will result time costly. Splitting large data file into small data can improve the efficiency in RETL (Sun, 2012).

\[ C = \sum_{i=1}^{n} C_i \times T \Rightarrow \sum_{i=1}^{n} t_{c_i} \]

Where \( C \) is the total counts of the records reside in the data file, and \( C_i \) is the counts of the records reside in one of partitions, and \( T \) is the total time cost of loading the data file, and the \( t_{c_i} \) is the time cost of loading one of partitions. The clause above means that the time cost of loading one large file at a time is significantly larger than the sum of time cost of loading its partitions separately.

IV EXPECTED RESULT

In sequential processing, the query is executed as a single large task. In parallel processing, the query is divided into multiple smaller tasks, and each component task is executed on a separate node.

Figure 4 and Figure 5 contrast sequential processing of a single parallel query with parallel processing of the same query (Oracle, 1997).

Basic performance goals of parallel system can be measure in terms of two important properties (Oracle, 1997):

- **Speedup**

\[ \text{Time}_{\text{Parallel}} = \sum_{i=1}^{n} \text{ETL} \cdot \text{count} + N_q \cdot \delta \]

AET is a relative value to measure the performance and it will reach a stable value when the system is full-load. Basically, the smaller the AET is, the more efficient RETL is.

V CONCLUSION

In this paper, data integration in geo spatial data warehouse is studied to resolve the problem of preparing and handling the spatial data warehouse in the data warehousing industry. The large amount of structured and unstructured GI data will result in data organization and management of GI application bottlenecks. The ETL plays an important role in spatial data warehouse (Chen & Chi, 2004). However the integration of data for each data loading window can cause performance overhead when there are many extra ETL component during transformation data to data warehouse such as aggregation, conditional split routes, lookup, data conversion transform, derived column transform and others. An integrated SOA geo spatial data is introduced that incorporating a rule based technique to distribute parallel ETL pipeline so that the transformation performance can be enhanced. In the future, we still need to do some research works that expand the spatial data dependency, spatial data complexity and ETL rule based algorithm that can integrates with multisource heterogeneous geospatial data.

ACKNOWLEDGMENT

"The authors wish to thank the Ministry of Education, Malaysia for funding this study under the Long Term Research Grant Scheme (LRGS/b-u/2012/UUM/Teknologi Komunikasi dan Infomasi)"
REFERENCES


Yvan Bédard, & Han, J. (2009). Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery, 45–68.
