Prototype Development of Integrated Export Control Database System

Byoungchan Han[*], Tongkyu Park, Sung-Kyun Zee
FNC Technology, Institute of Future Energy Technology, 46 Tapsil-ro,
Giheung-gu, Yongin-si, Gyeonggi-do, 17084, Republic of Korea
[*]Corresponding author: bchan007@fnctech.com

1. Introduction

The Cold War between the United States and the Soviet Union and their respective allies sparked a nuclear arms race that threatened to wipe out the human race. Since the late 1960's, the United Nations established the first framework for the Nuclear Nonproliferation Treaty (NPT) to prevent the spread of nuclear weapons, and to promote the peaceful use of nuclear energy. Republic of Korea has also joined the international nuclear nonproliferation regime since the NPT was ratified in 1975. Korean government controls the exportation of nuclear-related items through the Nuclear Safety and Security Commission (NSSC) and Korean Institute of Nuclear Nonproliferation and Control (KINAC).

KINAC, which is the professional regulatory organization for managing nuclear proliferation and threats, has several regulatory systems and databases. However, as the regulatory workload increased, there was a need to analyze a variety of data together, and independent systems and databases hampered such a situation. Therefore, we proposed the integration of independent export control databases and developed a prototype to examine its feasibility.

2. System architecture

Integrated export control database system consists of data extraction layer, data transformation layer, data load layer, and data analysis layer. Each layer describes the data flow between repositories.

2.1. Data extraction layer

Data extraction layer serves as the foundation of the overall data pipeline. Since the function of this layer is essentially to replicate data from the source system, efficient modern data handling techniques are used. Change data capture (CDC) is used in particular to track data changes because the big data extraction method doesn't always succeed in copying entire records. It scans the Redo log generated by the source database to determine which events have changed since the last transfer.

We adopted Debezium connector, which is one of the most widely used CDC services. It writes all row-level changes throughout entire database tables to the change event streams [1]. These streams can be read by downstream applications to determine which data should be mapped to the next layer.

In data extraction layer, CDC service accesses various source databases that have different database management systems (DBMS) and reads changed logs, then converts them into the change event streams of the same format. These streams are passed to the next layer via data pipeline.

2.2. Data transformation layer

Data transformation layer is in charge of converting data from its original form to the integrated structure. It reads the change event streams generated by data extraction layer and stores replicated data in the staging database. The data is then cleansed, consolidated, and structured in order to be stored in the data warehouse and analyzed by business intelligence (BI) applications.

The data cleansing process improves data quality by removing corruptions, inconsistencies, duplications, and incomplete items. Because the source systems contain numerous databases, multiple columns describing the same properties, such as time, gender, and country name, can be found. For example, if one database marks the gender item as "male", "female", and another as "M", "F", we can combine them to be displayed as 0 and 1, respectively. Data consolidation process combines two or more tables that describe the same entity into one. Although most of the tables represent unique entities, several tables with identical entity are found such as the denial list and international treaty participation. The data structuring process finally organizes data so that it can be easily mapped to the data warehouse. This process contains setting up relationships, renaming columns, and adjusting column properties.

2.3. Data load layer

The data load layer transfers data from the staging database to the data warehouse and Elasticsearch service. Data warehouse is a single and centralized repository that structures and stores data so that it can be analyzed efficiently [2]. The main feature of the data warehouse loading process is that it consumes data via CSV bulk import rather than row-level insertion. The distinction refers to the difference in intended use between data warehouse and relational database. Since the data warehouse design is optimized for query operations and analyses, it performs exceptionally well in table joins or numerical calculations. On the other hand, relational database design is optimized for data manipulation, such as insert, delete, update, and is

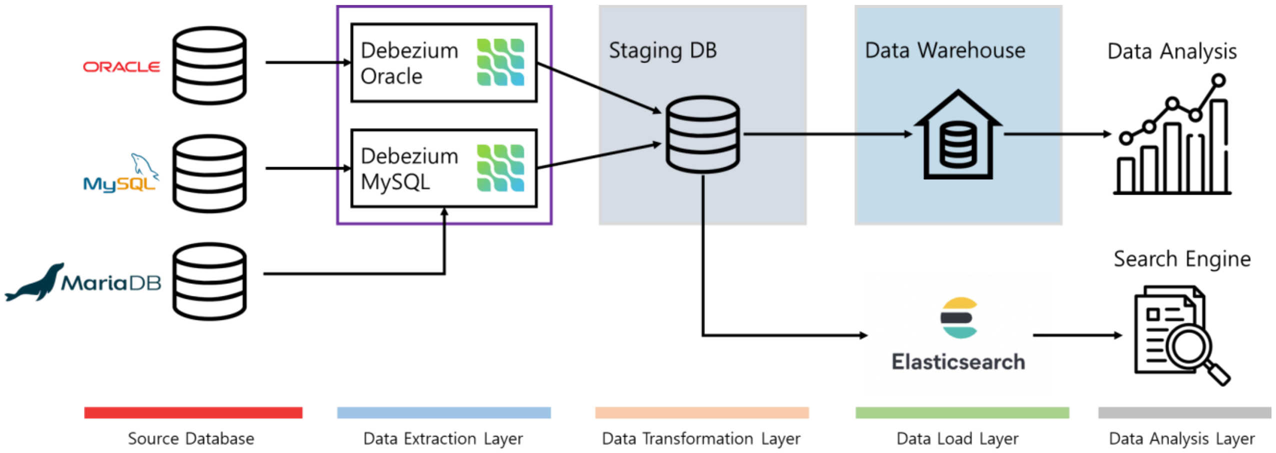

Fig. 1. System architecture of the integrated export control database

primarily used for storing real-time data on a daily basis. Row-level insertion into the data warehouse is thus slow and inefficient.

Elasticsearch service [3] is a search engine based on Apache Lucene. It ingests data from the staging database and converts each data into word tokens and subsequent metadata so that the data may be retrieved. It ingests data from the staging database, then converts each data into word tokens and following metadata in which the data can be retrieved. Such tokenization process is known as indexing. Indexing enables data to be used as NoSQL database by storing them in JSON format.

In addition, a high-dimensional numerical vector describing each indexed data is calculated by language representation model and saved in each JSON object. For its cutting-edge performance in natural language representation, we used the BERT model, which stands for Bidirectional Encoder Representations from Transformers [4]. These vectors are used to evaluate the similarity between word tokens allowing the search engine to find not only the search query but also related or similar data.

2.4. Data analysis layer

The data analysis layer is the uppermost layer of the entire system interacting directly with the clients. It generates queries for previously defined analysis processes and sends them to the data warehouse whenever client demands. Then it outputs the results analyzed by the data warehouse. The data warehouse is composed of independent star schemas each of which is linked to a single data analysis module. Each data analysis module performs BI analysis for specific tasks in the field of nuclear export control. These tasks include data analysis for strategic item judgement, strategic item permission, and international nuclear material transfer. Regulators can directly check the data associated with each BI analysis module at the data analysis layer and perform analysis according to their use.

3. Process orchestration

The overall process is managed by Apache Airflow, a workflow process management tool [5]. Apache Airflow creates a directed acyclic graph by connecting each task with a directed line. Thus, processes can be executed sequentially. The entire system consists of multiple workflow graphs.

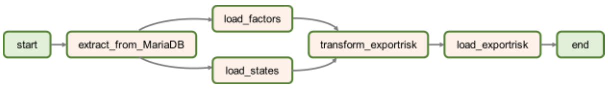

Fig. 2. Example of workflow graph

Each task describes a unit process such as data transformation, or data replication from a specific database. The majority of the tasks consist of operators, which are predefined task templates such as bash shell scripts and arbitrary Python functions. It is possible to modularize each task by dividing the system into independent unit tasks and adjusting the execution order in this way.

Another main feature of process orchestration is process scheduling. Process scheduling is the action taken by the process manager to stop current running process and select other processes to run based on the given algorithm. It schedules processes of different states such as ready, waiting, and running. Apache Airflow manages process scheduling by monitoring all tasks and triggering task instances when their dependencies are complete. Workflows can also be started with their predefined strategies (e.g., midnight every day, once per week) by triggering the start node when the starting conditions are satisfied.

4. Conclusion

Current nuclear export control regulations employ an independent system and database for each regulatory task. As a result, regulators find it difficult to refer to data from other fields. Furthermore, data are often inconsistent with each other because each database is updated within its own system. Such circumstances can lead regulators to make poor regulatory decisions.

In order to close this regulatory gap, we have developed a prototype of an integrated export control database based on data warehouse design patterns. The enhanced and optimized integrated database is currently being deployed.

ACKNOWLEDGMENTS

This work was supported by the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea. (No. 2106048)

REFERENCES

[1] Debezium Documentation, https://debezium.io/documentation/reference/stable/index.html.

[2] R. Kimball, M. Ross, The Data Warehouse Toolkit: the complete guide to dimensional modeling, John Wiley & Sons, 2011.

[3] Elasticsearch, https://github.com/elastic/elasticsearch.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805, 2018.

[5] Apache Airflow, https://airflow.apache.org/docs.