Lazy ETL for Scientific Data

Yağız Kargın
CWI, Amsterdam

Database research is slowly propagating into the domain of science [1]. This is motivated by the need to manage increasingly large amounts of data being generated with advancement of scientific instruments and collected in domain-specific files and repositories (i.e., semi-structured collections of files). This increasing data volume changes the way science works. Today's scientists need to manage this data, run high throughput experiments, generate simulations and then infer interesting knowledge from the results. This became known as eScience and is expressed in Figure 1. Indeed, this process is similar to those which drive business intelligence [2, 3]. Hence, it is natural to consider those processes of business intelligence to help scientists in their daily work. In their work, scientists have to take care of their data management themselves. Thus, they solve their data ingestion problem in a straightforward way by loading data entirely from external data sources (repositories) into a scientific data warehouse before it is analyzed. This process is time and resource intensive and might not be necessary if only a subset of the data is of interest to a particular user. To minimize this burden of data ingestion, we present a query-driven, on-demand Extract, Transform & Load (ETL) system: *Lazy ETL*. Initially, only metadata of the input files are loaded. We extract, transform and load the necessary actual data, only when it is needed by a query. This is different from direct application of the traditional ETL processes where the user has to wait for the lengthy (eager) initial loading of all the input data, even though his queries might only require a subset of the available dataset. Breaking with the traditional paradigm, we consider ETL as part of the query processing and we create a virtual warehouse that is filled with data on demand in a query-driven fashion. We seamlessly integrate lazy ETL with query evaluation. While executing queries, required data that is not yet present in the warehouse is loaded. To select the files to load, the selection predicates on the metadata are evaluated first. Once this part of the query plan is executed, a rewriting operator is executed. This operator modifies the remainder of the query plan to replace all references to data tables with operators that load the required files.

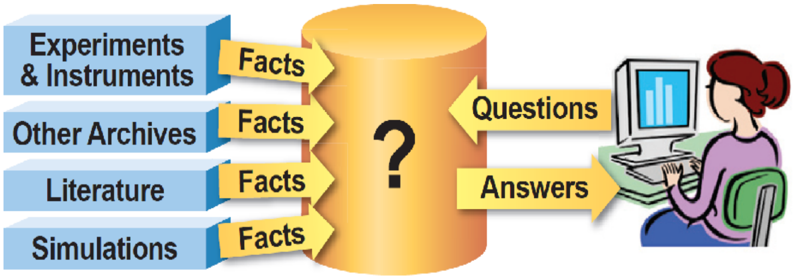

Figure 1: EScience Knowledge Management (taken from [1])

The implementation of our novel Lazy ETL system is done in MonetDB, an open-source column-store Database Management System. In our evaluation, we use seismological sensor data stored in mSEED files in a file repository as our input dataset, and typical seismological queries as our analysis tasks. Extensive experiments show space-efficiency of our approach and the reduction of the overall time from system start to the first query answer. Even though we focus on a particular scientific use case, we believe that other cases, like classical business ETL, can benefit from a similar approach, as also mentioned in [4] and [5].

References

[1] A.J.G. Hey, S. Tansley, and K.M. Tolle. *The fourth paradigm: data-intensive scientific discovery.* Microsoft Research Redmond, WA, 2009.
[2] P. Vassiliadis. *A survey of extract–transform–load technology.* Int. Journal of Data Warehousing and Mining (IJDWM), 5(3):1–27, 2009.
[3] P. Vassiliadis and A. Simitsis. *Extraction, transformation, and loading.* Encyclopedia of Database Systems, 2009.
[4] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson. *Data integration flows for business intelligence.* In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, pages 1–11. ACM, 2009.
[5] L. Haas, M. Hentschel, D. Kossmann, and R. Miller. *Schema and data: A holistic approach to mapping, resolution and fusion in information integration.* Conceptual Modeling-ER 2009, pages 27–40, 2009.