Exposé
Master Thesis

at
Humboldt-Universität zu Berlin
Faculty of Mathematics and Natural Sciences II
Department of Computer Science
Knowledge Management in Bioinformatics

working title

Implementation and Evaluation of the TPC-DI Benchmark for Data Integration Systems

Maurice Bleuel, B.Sc.
Fritz-Erler-Allee 160
12353 Berlin
maurice.bleuel@student.hu-berlin.de
Matrikelnummer: 563342

Revision 2
12/26/2015

Expose Master Thesis

Inhaltsverzeichnis

1. Motivation and Background

In the past years, the amount of data needed to be processed by companies - or any organisation working with data - showed a significant increase. In an IDC iView article, the authors looked at only a small fraction of all systems available and estimated the data therein alone to double about every two years (see Gantz et al., 2012, p. 1). This can be attributed to multiple factors. For one, there is more storage space available at much lower cost than any time before: The price per megabyte dropped from roughly 9200 US dollars in 1956 to merely 0.0000317 US dollars in 2015 (McCallum, 2015). This enables companies to accumulate everything that can be stored about their products, their products' use, as well as their customers, contractors, and partners. Once they realised this, many developments to accumulate as much information as possible were set in motion, for example the tracking of license plates (ACLU, 2013) or the gathering of personal information on smartphones via apps installed by the user herself (Wired, 2010).
In parallel, many frameworks in the fields of big data analytics have been created to allow the analysis of the huge amounts of data - for example the software framework Apache Hadoop, which “allows for the distributed processing of large data sets across clusters of computers using simple programming models” (The Apache Software Foundation, 2014).

The most prominent problem regarding the gathering of data to enable integrative data analyses presents itself in the number of sources necessary for obtaining comprehensive information. For example, one can imagine some company holding different chains of supermarkets. One important type of data could list customers shopping in different stores, but there may well be lots of other sources of information. There may be external data regarding how many and exactly which type of returnable bottles were turned in using machines provided by a third party. There may be information on customer relations, ad campaigns and their success provided by different advertisers, as well as data obtained from different social media channels: When people contact the company, what they want, how many of those people got satisfying answers, and so on.
While it may be possible for such a holding to impose one defined data format for their own supermarkets to make the integration of their data easier, other parties will most likely not be willing to adjust their whole systems to some other format - especially if they work with different partners, themselves.

To work around such problems, specialised software for *data integration* (Passionned Group, 2015) exists to extract data from various different sources and transform it to conform to a specified target data scheme. This process, also called the “ETL Process”, is responsible for the extraction of data from any defined source, transforming and cleansing the data while possibly rejecting erroneous or incompatible information, and loading the cleansed datasets into a data warehouse (Lehner, 2003). The steps necessary to complete an ETL process vary significantly in their complexity, depending on the number of source systems to integrate, the data format they provide, and the requirements for the extracted information.
The complexity of the **extraction** step is mainly defined by two factors. For one, many source systems use their own data format which results in the need to integrate data from a number of heterogeneous systems (e.g. ASCII files or spreadsheets, mainframe applications, or relational databases) (Lehner, 2003, p. 139). The second factor are requirements on the performance of the data extraction, as analysis tools depending on the data warehouse may need to be provided with relevant information at a certain point in time to function correctly.
The **transformation** step is the most complex part of the ETL process requiring the most time and resources to complete (Lehner, 2003, p. 124). What kind of transformation is necessary for which part of the provided data is dependent on the quality requirements which have been defined for the data warehouse. There may be the necessity for duplicate checks, data corrections

(e.g. misspelled words or incorrectly formatted numbers) or the resolution of semantic (the same object may be described in different ways by different systems), structural (different databases may name fields differently or use more or less fields to describe objects), as well as data conflicts (different databases may provide different values for the same attribute of the same object).
Once all data has been transformed to adhere to the requirements of the data warehouse, it can be **loaded** into its database. It may be loaded explicitly by using SQL statements or bulk load tools provided by most modern database systems, or by means of automatic replication schemes (Lehner, 2003, p. 151).

As more and more companies recognize the need for the integration of data from various sources, the market for specialised applications for the ETL process became more and more relevant. As of today, there are many different data integration systems. Some of them are provided by big players like Oracle or Microsoft, others are freely available and customisable as open source software - and all of them claim to be the best and fastest solution for data integration purposes. For example, the Microsoft SQL Server Performance Team reported an "ETL world record" (Microsoft, 2008) in an MSDN blog article. They report to have imported one terabyte of TPC-H (see TPC, 2014b for details on TPC-H) data in "under 30 minutes" (Micorosft, 2008, first paragraph). All of those are, of course, biased by design: If the vendor of a business-critical software product measures its own product, the whole test will be conducted with the goal of making the system look good.
Microsoft used TPC-H data for their measurement, although this benchmark is designed for a completely different purpose, namely to measure "business oriented ad-hoc queries and concurrent data modifications" (TPC, 2014b, p. 8). They themselves state in the blog post that, at the time, "there is no commonly accepted benchmark for ETL tools" (Microsoft, 2008).

As of 2014, the TPC published its standardised benchmark suite for ETL systems called *TPC Benchmark DI (Data Integration)*, see (TPC, 2014a). So now, rougly six years after Microsoft conducted its own benchmark on their data integration tool, there exists a specification to create comparable benchmark results for all data integration tools in the market. A year after, however, the big players have provided neither TPC-DI benchmark results nor have they created an implementation of said benchmark for their particular system. Furthermore, the author is not aware of any known open source implementation of the benchmark for any system.

2. Goal

The goal of this thesis is to implement and run the TPC-DI benchmark for at least one data integration system. To achieve this, a system to test has to be assembled following the requirements set by the benchmark. This system will consist of a target database to hold the integrated information, a data integration tool as well as a program to generate input for the benchmark. Such a data generation tool already exists as a Java application called "DIGen", provided by the TPC on their website [1].

In the installed systems, the database schema described in the benchmark specification needs to be implemented and the defined transformations for the integration of the generated information have to be implemented.

As a final result, a "Full Disclosure Report" (FDR) will be created as required by the TPC-DI benchmark specification as a separate document while parts of it may well be integrated in the

[1] http://www.tpc.org/information/current_specifications.asp

final thesis text. This allows the produced benchmark results not only to be used for this thesis, but also to be submitted to the TPC for publication.

3. Approach

3.1. Specification analysis

As a first step the TPC-DI benchmark specification document (TPC, 2014a) has to be analysed in order to extract information regarding the different requirements for an officially accepted benchmark implementation, such as:

Source data

There are several requirements for the source data files to be used for a benchmark run. Although the TPC provides the DIGen tool to create said data files adhering to their specifications, it would still be prudent to get an overview over the expected contents of these files so any errors during data generation can be detected.

System under test

The benchmark specification provides many information about possible configurations and components used in the system under test (SUT). Before the assembly of such a system for a benchmark run, the requirements set for database servers, data integration tools and possible networks connecting them needs to be analysed in detail. This will then allow a reasonable choice as to which specific software and hardware may and will be used for the benchmark implementation.

Transformations

The transformations the benchmark requires need to be analysed as well. This will result in a deeper understanding of what exactly the benchmark operations do and which tools may be necessary in the data integration system to execute them. After this step, there will also be knowledge about whether some transformations need to be coded by hand if, for example, the data integration system does not support them by default.

Metrics

The metrics and execution rules defined in the specification are another important piece of information as they provide the exact way the execution performance will be measured. The data warehouse and the data integration system need to be able to adhere to these rules and provide accurate timings, otherwise no benchmark run will ever yield reproducible or reliable results.

3.2. Assembly of the SUT

Once all requirements are known, specific software to implement the benchmark with can be researched. This includes a database system functioning as the data warehouse as well as a data integration system.

Of special importance is the class of data warehouse, which defines in which class the benchmark results will be published. TPC-DI results can be in the *OPEN* oder *ACID* classes. In order to

have results listed in the latter, the database system functioning as the data warehouse needs to be tested for *ACID* compliance via exactly specified methods (TPC, 2014a, p. 48). If an open source system not being certified is used as data warehouse, the benchmark results be published in the *OPEN* class.

During the assembly of the SUT, every step for installation and configuration of the system will be noted to be included in the final FDR. This also includes pricing information for the components used following the TPC pricing specification in it's current version 1.7.0 (TPC, 2011).

As a result of this phase, there will be a directory structure to hold the source data files as well as any necessary tools (at least DIGen). All tables in the data warehouse will have been created following the defined schema, with no data in them. Also, all transformations specified by the TPC-DI benchmark specification will have been implemented in the data integration tool. The system under test is thus in a state where all parts of the system can be tested to be compliant with the requirements and a benchmark test run can be executed to validate the transformations.

3.3. Validity of the SUT

The validation of the system under test will follow the requirements defined in the benchmark specification (TPC, 2014a, pp. 91) using the data comparison tool provided by the TPC. The final benchmark runs will only be executed once the system under test passes this validation step.

3.4. Benchmark run and FDR

Once the system is validated to be compliant to the TPC-DI benchmark requirements, a real benchmark run is executed. This includes the creation of the source data using the DIGen tool and running all benchmark phases:
The "initialization", "historical load", two "incremental update" phases and the "automated audit phase" (TPC, 2014a, p. 86).

After the phases have completed, the benchmark run is officially finished. Now the metrics can be extracted from the data warehouse tables they were written to and a final result be calculated.

All the information obtained and created thus far will be formalised in the full disclosure report. Once the report data has been generated, the TPC-DI benchmark for this system is complete.

4. Optional additions

Provided there is enough time left after assembling a first SUT, running the benchmark on it and creating the corresponding FDR documents, the following additional tasks may be included in the work of the thesis.

4.1. A different SUT

A different data integration tool will be examined while keeping the rest of the SUT as already defined and implemented. This minimises the work necessary to obtain another system's benchmark metrics for comparison to the installation of another software tool and the re-creation of all transformations for this particular system.

All other configuration data as well as the data warehouse can be re-used - only the database must be emptied again.

4.2. Test for ACID compliance

If a database system with no published results of an *ACID* compliance test as the TPC requires them, this *ACID* compliance test may be implemented and run in addition to the TPC-DI benchmark. This step would allow the benchmark results to be published in the TPC-DI *ACID* class.

5. Related Work

Since the TPC-DI benchmark has been made publicly available only last year, there is not much to go on when it comes to literature about or implementations of it. Most of the documents mentioning ETL or data integration benchmarks pre-date the release of the TPC-DI benchmark specification, many of them even being research that was eventually used when crafting the benchmark itself later. For example, Simitsis et al. (2009) wrote a paper on “Benchmarking ETL Workflows”, where they proposed test suites for ETL workflows based on TPC-H. In the same year, an article titled “A Survey of Extract-Transform-Load Technology” (Vassiliadis, 2009) took a closer look at all steps involved in the ETL process, their properties and problems arising when working with them. In a specialsed part on benchmarks, Vassiliadis states that neither TPC-H nor the more sophisticated TPC-DS benchmarks cover all aspects of ETL since they lack “the notion of large workflows of activities with schema and value transformations, row routing and other typical ETL features” (Vassiliadis, 2009, p. 17).

The first academic mentions of ETL benchmarking are found in Darmont, Bentayeb & Boussaïd (2005), though it is declared as future work there. In Vassiliadis et al. (2007), the authors write about specific ETL benchmarking in more detail, introducing the term *butterfly structure* for common ETL templates: Source data is combined from the left wing to a central storage, the *body* of the butterfly. The right wing, finally, symbolises the aggregation of the tuples as, for example, materialised views.

An early comparison of major tools and frameworks for running ETL processes was given in a survey by Barateiro & Galhardas back in 2005. A more recent comparison, especially looking at open source software in contrast to prorietary ETL frameworks is conducted in Qualitz, 2014.

Another development to be considered is a technically novel approach on data integration which was published as late as early this year in Guo et al., 2015. They describe a way to increase data integration performance by pushing the transformation phase into the source databases using virtual tables, in order to process data transformations even before submission to the target or staging system, which, in theory, eliminates the need for temporary storing of data in a staging area. The call their approach the “TEL process” (Guo et al., 2015, p. 1) for *Tranform, Extract, Load*. It stands to question whether or not the newly devised TPC-DI benchmark would be applicable to this kind of ETL system as well, or not.

When benchmarking ETL systems, or any system connected to data warehousing, big data, or something related, there is normally a huge amount of data involved. In order to get the most accurate results, this data has to be generated for a benchmark while allowing for specific constraints to be defined: In essence, the generated, artificial data shall resemble real datasets as closely as possible. The TPC benchmarks heavily rely on the *Parallel Data Generation*

Framework, or *PDGF* for short, as a means of generating arbitrarily complex and big amounts of source data in different formats. This framework has been developed at the University of Passau and it's use for generating data integration benchmark data is described in closer detail in Rabl & Jacobsen, 2014.

6. Bibliography

Microsoft SQL Server Performance Team (2008). *ETL World Record!*. MSDN Blogs. http://blogs.msdn.com/b/sqlperf/archive/2008/02/27/etl-world-record.aspx. Last checked: October 15, 2015.

Transaction Processing Performance Council (TPC) (2011). *TPC Pricing Specification*. Version 1.7.0, November 18, 2011

Transaction Processing Performance Council (TPC) (2014a). *TPC Benchmark DI (Data Integration) Standard Specification*. Version 1.1.0, November 11, 2014

Transaction Processing Performance Council (TPC) (2014b). *TPC Benchmark H (Decision Support) Standard Specification*. Revision 2.17.1, November 13, 2014

Qualitz, S. (2014). *Vergleich von Open-Source und kommerziellen Programmen zur Durchführung eines ETL-Prozesses*. Diploma thesis. Humboldt-Universität zu Berlin

Simitsis, A., Vassiliadis, P., Dayal, U., Karagiannis, A., Tziovara, V. (2009). *Benchmarking ETL Workflows*. Performance Evaluation and Benchmarking. First TPC Technology Conference, TPCTC 2009, Lyon, France, August 24-28, 2009, Revised Selected Papers. 199-220

Vassiliadis, P. (2009). *A Survey of Extract-Transform-Load Technology*. International Journal of Data Warehousing & Data Mining, 5(3), 1-27, July-September 2009

Darmont, J., Bentayeb, F., & Boussaïd, O. (2005). *DWEB: A Data Warehouse Engineering Benchmark*. Proceedings 7th International Conference Data Warehousing and Knowledge Discovery (DaWaK 2005), pp. 85–94, Copenhagen, Denmark, August 22-26 2005

Vassiliadis, P., Karagiannis, A., Tziovara, V., & Simitsis, A. (2007). *Towards a Benchmark for ETL Workflows*. 5th International Workshop on Quality in Databases (QDB 2007), held in conjunction with VLDB 2007, Vienna, Austria, 23 September 2007

Barateiro, J., & Galhardas, H. (2005). *A Survey of Data Quality Tools*. Datenbank-Spektrum 14, 15-21

Guo, S., Zi-Mu, Y., Ao-Bing, S., Qjang, Y. (2015). *A New ETL Approach Base on Data Virtualization*. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(2): 311-323 Mar. 2015

Rabl, T., Jacobsen, H.-A. (2014). *Big Data Generation*. WBDB 2012, LNCS 8163, pp. 20-27

Gantz, J, Reinsel, D. (2012). *THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East*. IDC iView "Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East", sponsored by EMC. December 2012

McCallum, J. (2015). Disk Drive Prices (1955-2015). Retrieved November 26, 2015 from http://www.jcmit.com/diskprice.htm

ACLU “American Civil Liberties Union” (2013). *You Are Being Tracked: How License Plate Readers Are Being Used To Record Americans' Movements*. July 2013

The Apache Software Foundation (2015). Apache Hadoop Website. Retrieved November 26, 2015 from http://hadoop.apache.org

Lehner, W. (2003). *Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden*. dpunkt Verlag

Passionned Group (2015). *Complete List of ETL tools - comparison included*. Retrieved December 24, 2015 from https://www.etltool.com/list-of-etl-tools/

