Robust Data Transformations

Alekh Jindal, *CSAIL MIT*

ABSTRACT

Background. Massively parallel data processing systems are ubiquitous in today's big data era. Examples include Hadoop, Spark, Stratosphere, and a number of tools developed on top of them. Users of these systems upload their datasets to a distributed file system and run their analysis in a distributed fashion. However, several analyses require a variety of data preparation steps in order to perform the actual analysis efficiently. Examples include creating a variety of data layouts, indexes, or partitioning for analytical queries; collecting random or stratified sampling for approximate queries; compressing, erasure coding, or flexibly replicating data for tuning the storage space; and cleaning data, i.e. violation detection and repair, to guarantee consistency with respect to a given set of business rules. Traditionally, such data transformations start with the assumption of a query workload, i.e. a representative set of queries provided upfront. Alternative approaches collect the query workload over time and apply transformations based on the observed query workload.

Problem. Unfortunately, the assumption of having a workload is not true in many of the modern data analyses which are ad-hoc and exploratory in nature. For example, an analyst looking for anomalies in a web server log might compute a variety of aggregates in an ad-hoc manner. Such analysis has no predefined query workload. And collecting the workload is tedious since the analyst would typically want to start using the data as soon as it is available, perform his analysis, and move on to the next dataset, possibly before the workload gets collected. The other problem with workload-based data transformation is that they benefit only a small subset of the possible queries. In case the workload changes, the workload-based approaches need to re-transform the data. For example, the data needs to be partitioned all over again when the group-by queries change or cleaned all over again when the business rules change. This is highly undesirable for running ad-hoc data analyses.

Our Idea. We ask ourselves two critical questions: (i) can we transform a dataset right at the beginning and *without* having a query workload, and (ii) can the transformations be *robust* in that they benefit ad-hoc queries in the future. To address these questions, we propose to shift from workload-driven data transformations to *data-driven* data transformations. This means that the data transformations depend only on the data itself. One way to do this is to exhaustively transform the data for all combinations of transformations. Consider data partitioning as a concrete example. To create robust partitioning, imagine partitioning a dataset on all attributes, i.e. ad-hoc group-by or join queries will now not need to re-partition this dataset. Alternatively, we can pick uniformly distributed points in the search space. For example, one could imagine creating all possible equal sized vertical partitions. Although such a transformation may not be perfect for a given query, however it is likely to benefit all queries. Apart from robustness, data-driven transformations offer several other advantages. These include saving the optimization costs to come up with the transformations in the first place, since the data is transformed exhaustively anyways, and avoiding bad decisions due to incorrect cost estimates, which remain elusive in most cases. Furthermore, robust data transformation offers the opportunity to observe the actual costs of different transformation designs and to possibly adapt one or more of them.

Challenges. Data-driven data transformations is a challenging proposition on several accounts. First of all, we need to devise mechanisms to efficiently create the robust data transformations. The transformation cost must not be prohibitively high and defeat the very purpose of ad-hoc analyses. Second, the transformations need to be aware of the skew in the underlying datasets. This is important because real datasets are skewed and thereby have very different transformation needs. And finally, the data transformation engine needs to be tightly coupled with the underlying data storage system. Recent works offer hope that these challenges could be met. For instance, researchers have proposed to piggyback indexing and partitioning on the data upload, along with tight integration with the underlying storage substrate. Similarly, several researchers have developed techniques to handle skew for indexing and partitioning. We need to leverage and extend these techniques for more general and robust data transformations.

Vision. Data transformation is increasingly being viewed as the most critical piece to making effective use of data. However, traditionally, database transformation involves to either create the perfect transformations upfront such that all queries in the provided workload benefit fully and right from the start, or create the transformations adaptively such that the queries in the observed query workload benefit increasingly and over time. However, both these approaches have problems. In the first case, the user needs to provide a query workload upfront which is simply not there in many cases. While in the second case, initial queries do not benefit at all and adaptivity can take quite long before the queries start to benefit.

In contrast, this paper points to a new data transformation paradigm wherein we create robust data transformations upfront for ad-hoc query workloads such that all queries benefit partially right from the start. Later on, the system can adapt and specialize the robust designs to the observed query workload such that a subset of the queries benefit increasingly and over time. Such a data-driven transformation philosophy reduces the barriers to effective data analytics for analysts. This means that instead of being caged to a limited set of queries, analyst should be free to try out ad-hoc queries and discover newer insights. At the same time, the analysts should not be intimated with terrible query performance to start their analysis. Furthermore, data-driven transformations could also be useful for suggesting queries to the analysts, based on the transformations that have already been done on the data. Such queries are likely to have very good performance since data has already been transformed appropriately.

In summary, we envision data transformation as a tool to help and guide analysts to make better and faster use of their data.

7th Biennial Conference on Innovative Data Systems Research (CIDR '15)
January 4-7, 2015, Asilomar, California, USA.