Assignment 01

Summary of ‘MAD Skills: New Analysis Practices for Big Data’

Name: Dean D’souza

H.U. ID: 168424

Summary:

The paper discusses an emerging practice in data analysis, focused on performing data analysis closer to the data store and getting new data to the repository as fast as possible, known as MAD (Magnetic, Agile and Deep) analytics. The paper also gives examples which were implemented for the Fox Interactive Media advertisement network though the Greenplum parallel database system.

Over the years, the cost of storage media has scaled so as to allow high capacities for reasonable prices, such that “the world’s largest data warehouse from just over a decade ago can be stored on less than 20 commodity disks priced at under $100 today” [1]. The value of data analysis for large scale databases has also become more recognized by many companies and organizational units for cost savings and direct revenue. This has also given rise to a need for large-scale data collection through various sources, giving rise to MAD analysis skills. This technique differs from traditional EDW (Enterprise Data Warehouse) techniques, highlighted as follows:

1. Magnetic: Unlike in traditional EDW approaches, which incorporates data only after it has been properly cleaned and integrated to the specifications of the warehouse, thus preventing the addition of new data easily, the MAD approach tries to gather all the data sources in an organization, irrespective of the data quality.
2. Agile: the MAD approach tries to make a data warehouse to be a constantly evolving source which can allow analysts to easily access, analyze and produce results which can be adapted to further improvements of the data warehouse.
3. Deep: the MAD approach also tries to get the data warehouse to become not only a source of data on which many modern statistical analysis techniques can be used, but also as a source on which such techniques can be used completely and in parallel.

The paper also suggests that it is time for the locus of power to shift from database administrators to the analysts, so as to integrate a wide variety of programming styles into a single parallel dataflow engine, due to the varied fields from which analysts come from. [1] It further goes on to explain that while Data Cubes ad OLAP (On-Line Analytic Processing) are useful for obtaining information about the data, through descriptive statistics, there is a need for inferential/inductive statistics which require more computation power in order to help with predictions based of the data. There is also a need to closely integrate statistical computations with large parallel databases in order to prevent the loss of details, which could be lost through sampling techniques, that could be important for answering questions over small subpopulations. It also highlights the importance of MapReduce and parallel programming towards creating tools where statistical analysis can be implemented in parallel on distributed databases.

The paper further goes on to describe the FAN (Fox Audience Network) data warehouse, with hardware specifications, and the data which it collects on a daily basis. It also goes on to highlight the importance of the analysts and the analytics team as the producers of new data products, which could eventually become customer facing features. It then agrees with a basic principle of traditional EDW, that there are benefits in getting an organizations data into a one repository, but differs on how to achieve the same.

The paper gives us the following iterative process for an evolving data warehouse:

1. Perform analysis to identify areas of improvement
2. React to or ignore the analysis
3. If the business reacts, new data sets are created
4. Incorporate the new data sets
5. "How can we improve?"

The paper goes on to give a three-layer approach to complement this process and speed up the pace of evolution of the data warehouse:

1. Produce a Staging Schema for raw fact tables and raw action logs which engineers and a few analysts can manipulate for research purposes.
2. Produce a Production Data Warehouse Schema which holds the aggregates used by most users and those comfortable with a large-scale SQL environment.
3. Produce and maintain a Reporting Schema which holds specialized, static aggregates for the reporting tools and more casual users. This schema should be tuned to provide rapid access.

It should be noted that the layers are not physically separate but present though cross-joins between layers and schemas. In general, aggregates personalized for analysts usually end up being promoted to the production schema from the staging schema as a result of effective communication between researchers and database administrators. The production schema thus serves to answer common questions that were dug up by the analysts, but not so common as to have the need to create reports.

The paper also suggests the introduction of a fourth "sandbox" schema, which the analyst would have full control over for the purpose of experiments.

The paper then goes on to give a semi-detailed approach for the development of mathematical concepts in SQL in order to facilitate the goal of performing high level analysis on the data in parallel. It describes the various layers of abstraction, and provides a bit of explanation for implementation, in order to perform analysis.

The first layer is that provided through traditional SQL databases in the form of data types and functions of simple (or scalar) arithmetic. The next layer is that of vector arithmetic, which facilitates linear algebra, and includes examples of vector and matrix operations by expressing them as relations and then performing basic operations on them. This includes adding matrices, multiplying matrices and transpose of matrices. It also describes a bit about operations for finding the common document similarity metric and the standard distance metric. It also gives details about the matrix based analytical methods such as OLS (Ordinary Least Squares) and Conjugate Gradient.

The next layer is the function level, implemented through operations called "functionals" of which the probability density function forms a base. This level aims at providing tools for modelling and comparative statistics. Some data-parallel methods such as the Mann-Whitney U test (a substitute for the student's t-test for non-parametric data) and Log-Likelihood Ratios are explained.

The paper also discusses some resampling (the process of repeatedly taking samples from the data) techniques, in order to make up for the deficiency of simple SQL aggregates which are not as robust.

The two standard resampling techniques include bootstrap and jackknife methods. In the bootstrap method, a number of members of the population are picked and a desired statistic is calculated. This is done again with another set of members a number of times to get the same summary statistic till we can obtain what is known as a sampling distribution. In the Jackknife method on the other hand, we recompute the summary statistic a number of times by leaving out one or more data items from the full data set, in order to measure the influence of certain subpopulations, with the resulting set of observations used for the sampling distribution. Two important points to keep in mind is that as the implementations use a random generator the researcher needs to verify the scaling of the function to have reproducible research and also a DISTINCT clause is needed to prevent same values for identifiers of the SQL query.

The paper further goes on to describe the MAD Data Base Management System (DBMS). It goes on to describe the needs of the DBMS to be able to allow users to run queries directly against the raw feeds from files or services which form the external tables. It makes a not about how Greenplum implements fully parallel access for both loading and query processing of such tables, through technique known as Scatter/Gather streaming. This technique involves scattering such tables across all nodes of the DBMS, such that each node gathers inputs form multiple distinct sources. These tables can then be added to the database tables or used directly as purely external tables with parallel Input/Output.

The paper also makes a distinction between ETL and ELT, where ELT is the approach of performing transformation scripts at the database level. Again, the advantages of Greenplum, which supports such transformations written in SQL and supports MapReduce scripting (made popular by google through hadoop), are highlighted.

It also discusses the data lifecycle in the MAD data warehouse, which begins as data which analysts iterate over frequently in order to obtain transformations and table definitions. This data is then used for ad-hoc analysis and reporting. The data eventually becomes significantly old enough to be archived in external archives. Again the advantages of Greenplum for supporting heap storage format for data which is frequently updated and the append-only table format, which is highly compressible, are highlighted. Additionally, it also provides a medium compression mode for improved table scan time at slightly lower loads.

The paper also talks about the advantage of Greenplum having interfaces for a number of commonly used data mining languages such as python, R and Perl, which are enabled to run data-parallel on a cluster. However, it is up to the developer to implement them correctly and efficiently.

The paper concludes with a few thoughts about key questions such as:

1. Co-optimizing storage ad queries for linear algebra: This requires a query optimizer which can efficiently choose between redundant layout of matrices and libraries of linear algebra routines for different layouts.

2. Automating physical design for iterative tasks: There is a need for making improvements to automate the tasks of deciding how to perform the iterative tasks of analysts such as deciding how to store the data, place it in the storage formats, perform repeated computations, etc.
3. Online query Processing for MAD analytics: While Online Aggregation and such techniques can help speed up the process of iteration in the MAD approach, more research needs to be done to further improve them and provide more in depth data on runtimes and alternatives.

References:

[1] "MAD Skills: New Analysis Practices for Big Data", authors: Jeffrey Cohen (Greenplum), Brian Dolan (Fox Interactive Media), Mark Dunlap (Evergreen Technologies), Joseph M. Hellerstein (U.C. Berkley), Caleb Welton (Greenplum)