"AUTOMATED QUALITY TESTING CHALLENGES WITH BIG DATA IMPLEMENTATION"

November 2014
Michael Chapiro
MChapiro@solutionsdiscovery.com
Main Topics

1. Introduction to Big Data Paradigm
2. Big Data Solutions
3. Big Data Implementation Using GreenPlum
Introduction to Big Data Paradigm

**CSC**

**BIG DATA**

THROUGH THE EYES OF YOUR IT STAFF

**59%**
OF RESPONDENTS RATE BIG DATA AS A TOP PRIORITY

**10%**
OF ORGANIZATIONS HAVE AN ENTERPRISE-WIDE BIG DATA ARCHITECTURE DEFINED

**46%**
TAKE A DECENTRALIZED APPROACH TO BIG DATA INDI IMPLEMENTATION

TALENT HAS EMERGED AS A KEY ISSUE IN BIG DATA IMPLEMENTATION

**86%**
FINDING THE RIGHT TALENT

**77%**
FINDING THE RIGHT TOOLS

**74%**
TIME

AS BIG DATA TAKES OFF, THE DEMAND FOR DATA SCIENTISTS IS EXPLODING

190,000
PROJECTED SHORTFALL IN DATA SCIENTISTS BY 2018 (McKinsey)

TO GET ALL THE SKILLS REQUIRED, ORGANIZATIONS MAY NEED TO THINK “COMPOSITE DATA SCIENTIST,” COMPOSED OF TEAMS OF INDIVIDUALS

ORGANIZATIONS ARE POISED TO MOVE
What is Big Data?

Total Exabytes of information stored

Source: The Information Explorer
What is Big Data?

Big Data refers to the tools, processes, and procedures used to create, manipulate, and manage very large data sets, on the order of terabytes and petabytes of data.

Big data is changing. In yesterday’s data warehouse and analytic infrastructure, big data is proprietary, expensive, monolithic, loaded once a, and ultimately, slow. The demands of data and business are changing.

This change represents the move to the next generation of Big Data, where queries can be performed in real time against petabytes of data.
One aspect that makes non-relational, or NoSQL, databases unique is the independence from Structured Query Language (SQL) found in relational databases.

NoSQL databases are typically designed to excel in one specific area: speed. No promises for consistency.

Amazon Dynamo. Dynamo is a non-relational databases known as distributed key-value store (DKVS)
Dynamo based DBMS implementations: Project Voldemort, Dynomite, and KAI databases

- Key-value store (KVS) databases are similar in architecture to DKVS, where keys are mapped to values. Instead of being distributed across servers. Redis, an open source database that's currently being funded by VMware, is in the KVS family, as are the Berkeley DB and MemcacheDB databases.

- Databases with embedded tables of data. Column-oriented stores. Google's BigTable is a well-known example of this class of NoSQL database. Hadoop, Cloudera, and Cassandra are also in this class of data storage system.
- Document-oriented store databases, with no rows or columns. **MongoDB and CouchDB** are using this approach. JSON-style objects-as-documents to store information not structured as XML will. Attribute-value to define object.

- Graph-oriented NoSQL database. Data is manipulated in an object-oriented architecture, using graphs to map keys, values, and their relationships to each other, instead of just tables. **Neo4j and HyperGraphDB and Bigdata** are this type of a DBMS.
Non-relational systems must be combined with additional tools in order to process data, and turn it into information you can use.

MapReduce is a programming interface a lot of big data technologies utilize.

The idea of MapReduce: to grab data from a source, using the Map() function and then process that data across multi-core systems, with the Reduce() function. Map() will apply a function to all the members of a dataset, and will post a result set, which Reduce() will then collate and resolve.
The best-known example of a MapReduce-based system is Hadoop, which uses MapReduce in combination with the Hadoop Distributed File System to store data effectively. Cloudera and the Apache Cassandra system use MapReduce.

MapReduce algorithms must be written manually to accommodate distributed data structures.

Other data processing tools: Pig's Pig Latin query language works with Hadoop storage systems. Hive's HiveQL similar to SQL.
Enterprise search products, such as ElasticSearch, Apache Lucene, and Apache Solr, use a concept called facets that enable you to treat data within documents as you would fields within a relational database. Facets are essentially inverted indexes that let you find specific pieces of information in a document, like an address or other customer information.
Greenplum introduced cloud storage as part of the Big Data solution, and enterprise data cloud. This implementation used a private cloud.

In yesterday’s data warehouse and analytic infrastructure, big data is proprietary, expensive, monolithic, loaded once, and ultimately slow.

A data warehouse is a culmination of information gathered about the enterprise. The data normally used to support business decision making.
The source data is moved from one or more source systems or data marts and must be cleaned, verified, and transformed based on the data format of the output, before moving to the warehouse. The industry has multiple tools that support the use of ELT in the data warehousing environment.

Such as Informatica, DataStage (IBM), Oracle Warehouse Builder, Oracle ODI, and many other.
Extract, Load, Transform (ELT) is a data integration process for transferring raw data from a source server to a data warehouse on a target server.

ELT allows raw data to be loaded directly into the target and transformed there.
Data Warehouse Methodologies There are essentially two commonly used methods to model a Data Warehouse.

- Dimensional model – Divides transactional data into facts and dimensions. Facts usually stand for various measures and dimensions give a context information about the facts. This is the most common data warehouse model.

- Normalized model – The level of normalization for this approach is usually not as high as for the typical OLTP data models.
Big Data AQT challenges and solution approach.

- Datawarehouse on GreenPlum
- Informatica ELT
- Size of Target Database:
  Tables 3,500
  Largest Table: 5,362 columns
- Source Data:
  XML document: 3,500 tables
  Total Attributes: 43,568
- Number of records: 25
Tools used for AQT testing:
XML SpyAltova 2014
Oracle Database 11.2.0.3
GreenPlum Database
Java, and Korn shell

Process:
1. Source XML get parsed using XML Spy
2. Parsed XML get converted into relational Data
   DDL/DML Generated via XSLT in XML Spy
3. DDL get loaded into Oracle schema
4. Create a schema in Oracle Database
   Execute SQL in Oracle DB
   Validate the objects.
   Remove not PSQL syntax and replace with PSQL
5. Load DDL/DML into GreenPlum
6. Pre-process metadata xls document:
   (Optional)
   Remove all rows pertaining to translated attributes
   Remove columns not related to SQL generation
   Export into csv format.
7. Run Informatica ETL JOB and note ETL ID.
8. Compare ETL and AQT schemes (Automated)
Generate ETL and AQT SQL.
Run SQL in both ETL and AQT.
Create a difference file and email to QA.

Contact:
Michael Chapiro
MChapiro@solutionsdiscovery.com
Stmas@computer.org