IJARST

International Journal of Advanced Research in Science and Technology

journal homepage: www.ijarst.com

ISSN 2319 – 1783 (Print)

ISSN 2320 – 1126 (Online)

Solution Architecture for Master Data Management in Hadoop.

Bongu Narayana Rao and G. Rajendra Kumar*

Department of CSE, Sri Sivani College of Engineering, Srikakulam, Andhra Pradesh, India.

****Corresponding Author's Email: rajendragk@rediffmail.com***

A R T I C L E I N F O

Article history:

Received : 03 Jan. 2016

Accepted : 21 Feb. 2016

Available online : 26 Feb. 2016

Keywords:

MDM, Hadoop, Hadoop Distributed File System (HDFS), CRM, Match Matrix, Data Profiling, Workflow Management, Scoring Algorithm, Bucketing, Profiling, Cardinality Measurement, Solution Architecture.

A B S T R A C T

Master Data Management (MDM) is a set of techniques and procedures (shared with Data Integration tools) put in place to collect raw data from several internal and external sources, conduct data quality analysis and profiling. cleanse them, combine/merge to provide "Single Source of Truth" and 360 view of a business entity and redistribute to internal transactional and business intelligence systems. MDM is useful in Business Intelligence world primarily for managing Dimensions in a typical Data Warehouse setup. Majority of the MDM deployments use MDM as the hub to supply Dimensional data to several downstream systems like Data Warehouses and Data Marts within the organization to facilitate reuse of unified/standard set of Dimensions across several reporting systems. Most of the MDM tools are also equipped to suck in data from several types of both external and internal data sources and redistribute the constructed Dimension data to several types of downstream destinations in both batch and real-time. Current MDM solutions are also usually accompanied by Data Quality and Data Profiling tools to assist during Data Preparation process.

PAPER-QR CODE

IJARST Volume 5, Issue 1

Citation: *Narayana Rao. et. al., Solution Architecture for Master Data Management in Hadoop, Int. J. Adv. Res. Sci. Technol. Volume 5, Issue 1, 2016, pp.520-524.*

Introduction:

Master Data Management (MDM) contains procedures, administration, approaches, benchmarks and instruments that reliably characterize and deal with the basic information of an association to give a solitary perspective. The information that is beaten might include: reference information and analytical data. Reference information is used in the business objects for exchanges, and dimensions for analysis. And the analytical data supports in decision making or underpins choice making.

In figure 1, an expert information administration apparatus can be utilized to strengthen expert information administration by uprooting copies, institutionalizing information (mass keeping up), and fusing standards to take out erroneous information from entering the framework so as to make a definitive wellspring of expert information. Expert information are the items, records and gatherings for which the business exchanges are finished.

The underlying driver issue comes from specialty unit and product offering division, in which the same client will be overhauled by various product offerings, with excess information being entered about the client (otherwise known as gathering in the part of client) and record keeping in mind the end goal to handle the transaction. The repetition of gathering and record information is intensified in the front to back office life cycle, where the definitive single hotspot for the gathering, record and item information is required however is frequently at the end of the day repetitively entered or expanded.

Expert information administration has the target of giving procedures to gathering, collecting, coordinating, merging, quality-guaranteeing, persevering and dispersing such information all through an association to guarantee consistency and control in the continuous upkeep and application utilization of this data. The term reviews the idea of an expert document from a prior registering time.

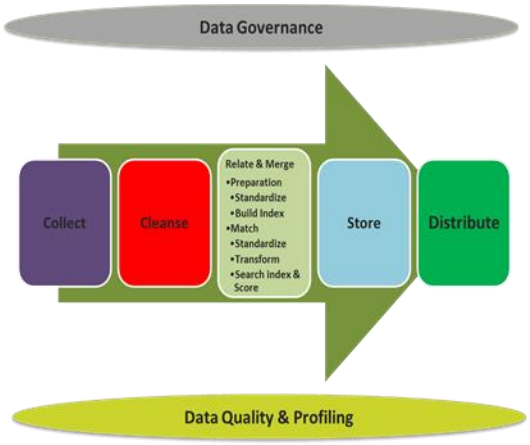

Fig. 1: expert information administration apparatus

Methodology:

Typical MDM process starts from the first phase called Data Collection. The data collection from various systems can be classified as internal or external such as archival databases, files or tapes or any data source using predefined or custom connectors. The data from these sources will be in their own specific formats and may not be in sync.

In the next phase is the data cleansing, which is to make the whole data in an acceptable unique file format and uniquely identifiable. For cleansing the data, need to use the various tools, need to follow the regular expressions (regex) and/or any techniques used in ETL are applicable standard and normalize the input data. Some techniques are explained below.

Match and Merge:

Let's assume "Customer" record from "CRM" and a "Clickstream" is to be matched and merged to form a single source of truth for Customer business entity.

1. ***Data Element Selection for Matching:***

   After successful data analysis, "Name", "Phone Number" and "Address" data elements have been chosen for matching logic implementation.

2. ***Standardization Name & Address (selective first key)Case conversion:***

   Make sure all the selected keys in unique case i.e., either upper or lower case.

   Example: "Smith John" to "SMITH JOHN"

   "John S" to "JOHN S"

   "John Smith" to "JOHN SMITH".

   Order the words

   form a chronological order

   Example: "SMITH JOHN" to "JOHN SMITH"

   Phone Number

   Strip off non-numeric characters

   Ex: 917-667-5972 to 0019176675972

   0019176675972 to 0019176675972

1. ***Transformation:***

Transformation functions can be applied to expand the search net by taking into account alternate spellings, nick names and typing errors.

2. ***Matching***

Deterministic or Probabilistic:

- Build an index for the Standardized form of the selected data elements of Business entity.
- For every record, using the index, calculate the match score and group the records if the score is past a certain threshold.
- Probabilistic approach also uses cardinality measure as weight for the scores representing the strength of element matches.
- In the end, "Match Matrix" is constructed to group related records.

Functional Dependencies (FD)

FD are classified into 3 categories as below

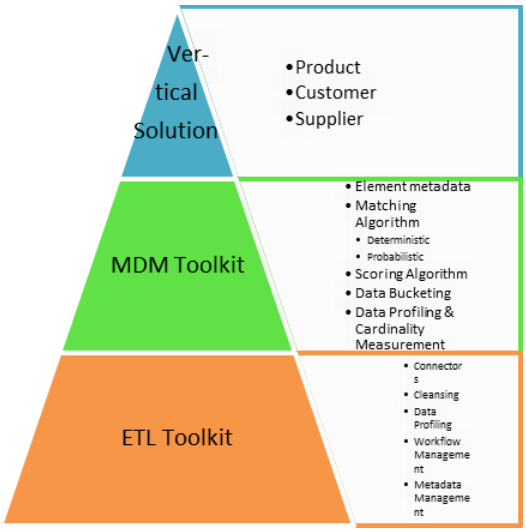

Fig. 2: Functional Dependencies

1. ETL tool kit:

ETL tool kit is the first step in categorizing the functional dependencies. Especially this is applied on connectors, cleansing, data profiling, workflow management, meta data management etc.

2. MDM tool kit:

In this step the main processing logic will applied on the data components for element metadata (EM), matching algorithm (MA), deterministic probabilistic (DetP), scoring algorithm (SA), data bucketing (DB), data profiling (DP) and cardinality measurement (CM)

3. Vertical Solution

The high-end business understands the integrity of various database objects which are used to pull the data

in preparing the data reports. For an example 3 master tables are classified as Product, Customer, Supplier, which are somehow can be related to get the data collectively. Individual tables may have their credibility but on the whole can be used to retrieve the data from these objects.

Related work:

There are couple of approaches identified and described below. Because of the data platform is newer in industry will look like similar approaches but different. Meaningfully both does the same activity. That is solutioning MDM in Hadoop. The first approach is described below.

MDM Solution Approach (1):

As depicted in the colored box, the phases are processed sequentially. Data standardization, Profiling and building hash index are done for data analysis and stored in the system as a file format. Once all these activities performed the data again stored in HBase (It is a kind of NoSQL database used in Hadoop) only.

ROWIDs in the index is an array to accommodate multiple matching rows for the same hash key, which plays vital role in identifying the records as a bunch.

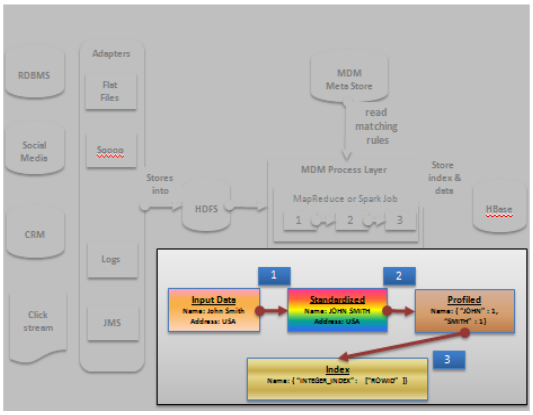

Fig. 3: MDM Solution Approach 1

The following activities will be performed in the next stage. All these activities are processed parallel to identify the matching score, which is shown in below picture.

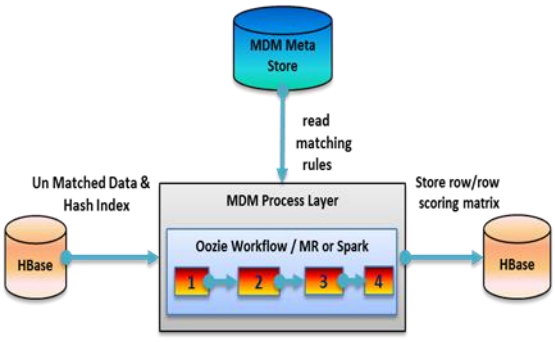

Fig. 4: MDM Approach

Activities

1. Data Standardization
2. Expand search net by applying following transformations
   a) Phonetic Conversion
   b) NickName
3. Generate hash using the transformed data (Optional)
4. Apply weights to the keyword based on the cardinality or probability distribution,
   a) extract the matching row ids and calculate matching score, also factor in the weights based on the probability measure.
   b) Prune the candidate rows based on predefined gating criteria (i.e. exclude rows with score < threshold)
   c) Store the scores in Match matrix. i.e. group all the matching rows from source tables with a surrogate key. means the destination is again HBase only.

MDM Solution Approach (2):

The initial load of historical data load is similar to approach 1. The following picture depicts the same.

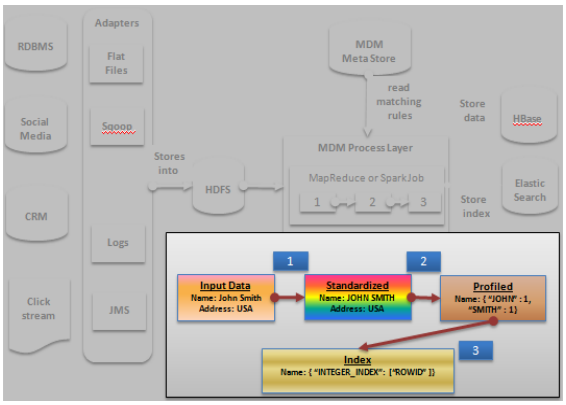

Fig. 5: MDM Solution Approach 2

The next activity is shown pictorially as below, will do the following activities.

Activities:

1. Data Standardization
2. Expand search net by applying following transformations
3. Phonetic Conversion
4. Nick Name
5. Apply weights to the keyword based on the cardinality/probability distribution,

a) Query the index and extract the row ids for matching rows based on the rules (for example, using Levenshtein edit distance based). Apply any cardinality based weight if applicable.

b) Prune the candidate rows based on predefined gating criteria (i.e. exclude rows with score <

threshold)

Store the scores in MATCH matrix. That is group all the matching rows from source tables with a surrogate key. Means the final estimations two. one is HBase and another one is query indexing.

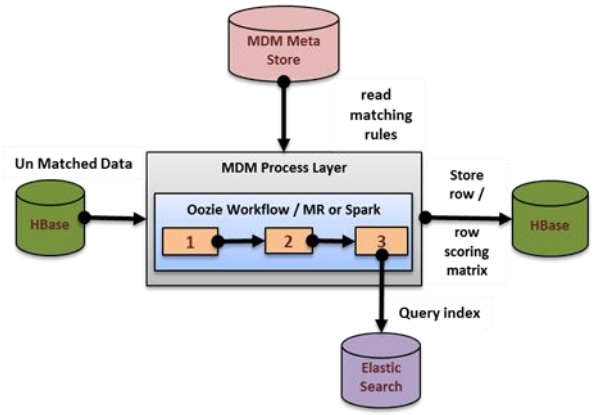

Fig. 6: MDM Solution Approach MATCH matrix

Oozie:

This is a kind of workflow scheduler used to schedule the jobs such as java, python, Mapreduce, Hive and Pig scripts. there are two kinds of scheduling mechanisms existed. One is time driven and another one is Size driven, which means once the file size reaches to certain limit the job will be automatically triggered for execution.

Time driven is widely used for daily activities in banking and finance sectors, retail markets, travel and transportation domain, energy and utility fields for automatic customer or status notifications for their customers in Hadoop Environment. This is a open source software.

HBase:

This is one of the NOSQL (Not Only SQL) databases widely accepted in Hadoop environment. The biggest asset is version controlling for each cell and retrieval capability from the table. There is no database other than the tables. Each table comprises one or more attributes. A new concept has evolved known as column family. Each column family is nothing but a table in RDBMS. In other words, a big table is nothing but all the collection of interrelated tables.

Mapreduce:

This is a framework developed using Java language for with the capability of processing huge sizes of data in parallel process. All the given input data is divided into number of smaller chunks called blocks and each block has its own program to work on it.

In Hadoop environment Mapreduce is widely used as and when any functionality if not available or cannot be performed in the technical world. One biggest drawback is always the intermediate data will be stored on local hard disk. And the programmers should have Java programming knowledge. This is a open source software.

Spark:

This is another kind of hadoop job processing most of the features are common except storage mechanism. If huge data to be processing and intermediate output will try to store in memory, if not feasible to store all the output then after spills that output onto local Disk. That is the reason, Spark process much faster than the MapReduce. This is a open source software.

Transformation Function Toolkit:

Transformation functions are used to expand the search net by deriving related keywords based on the input data element used for matching.

Following are most commonly used transformation functions:

- NickNames

  By using a Nickname database, search can be expanded.

- Soundex

By using Soundex variants of the input word, search can be expanded. means, pronunciation will be common but the actual spellings are different. It is a form of same vocally spelled will be considered same.

Note: The derived set of keywords after applying above mentioned transformation functions are limited in number and is independent of the underlying data.

- Fuzzy Match

using Levenshtein edit distance.

Note: Fuzzy matching cannot be conducted by simply generating permutations based on edit distance as the number of words derived can be explosive. So, a Levenshtein Automaton representing the index data has to be used to minimize the overhead.

Typical Matching Orchestration process:

The Orchestrator executes multiple steps as required where it matches and merges the merged data from the previous step with the new dataset to be matched & merged.

For example, during Step1, orchestrator Matched & Merged DS1 with DS2 (by applying required transformation functions on DS1) and during Step2, Matched & Merged the result from previous step (i.e. by applying transformations on the merged result from Step1) with data set DS3.

This process can be repeated to cover as many datasets need to be matched.

For example, during Step1, orchestrator Matched & Merged DS1 with DS2 (by applying required transformation functions on DS1) and during Step2, Matched & Merged the result from previous step (i.e.

by applying transformations on the merged result from Step1) with data set DS3.

This process can be repeated to cover as many datasets need to be matched.

Storage Structure

HBase is identified to store

- Input data
- merged data

And Elastic Search to store

- Index

Note: While HBase works for storing input & merged data and Index data in simplest form of Key (keyword) → Row Id mapping (suitable for searching for derived keywords using simpler transformations, for example: searching for nick names), more complex matching rules involving Levenshtein edit distance would require custom implementation to generate, store in HBase, update and maintain the automaton representing the index data

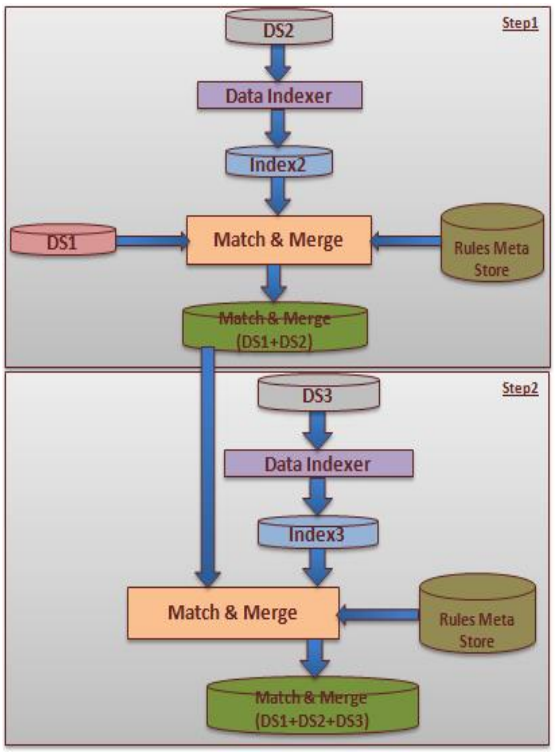

Fig. 6: Storage Structure

Conclusion:

In this paper, I have analyzed the problem of Master Data Management, which can be used in two different areas vitally. One in Business operations and the other in Financial and Analytical area.

For operational activities most of the static data will be synchronized with other integrated systems to run the business as usual. And more of the systems will be on same page in resulting the data as output. Various database objects such as customer details, department details, product details etc will be having common attributes which forms a bigger record for a particular product or set of products of a customer.

In Financial and Analytics, huge data should be collected for analysis such as identifying Anti-Money Laundering (AML) activities (even sometimes layering in AML), to identify fraudulent activities, erroneous data identification, operational data management, credibility of customers, to mitigate risks in case occurred, to meet the compliance and etc.

Futurework:

The current system can be implemented wherever the significant changes and risks to mitigate. The current designed system will be ready to implement. And also can be enhanced for further research. This design can work in any technology and on any platform with minimal changes.

The future research will focus on handling the data allocation mechanism in cluster level without duplicate and designed the distribution wisely. And dynamic data handling is another challenging work observed.

Internet based organizations, telecom, retail business etc holds huge data where the MDM approach can be also be used. social sites uses meaningless words and sentences which will be ignorable.

References:

1. Alex Berson and Larry Dubov, “Master Data Management and Customer Data Integration for a Global Enterprise”
2. Alex Berson and Larry Dubov, "Master Data Management and Data Governance", 2/Edition, 1 Jan 2011
3. Berson, Alex (2011), "Master Data Management and Data Governance".
4. Chisholm, Malcolm (2001), "Managing Reference Data in Enterprise Databases".
5. Dan Mandelstein, Ivan Milman, Erik A O’Neill, Sushain Pandit, Ralph Tamlyn, Fenglian Xu, Whei-Jen Chen, John Baldwin, Thomas Dunn, Mike Grasselt, Shabbar Hussain, "A Practical Guide to Managing Reference Data with IBM InfoSphere Master Data Management Reference Data Management Hub", First Edition (May 2013).
6. David Loshin, "Master Data Management", (The MK/OMG Press) 1st Edition.
7. Mark Rittman, "Introduction to Master Data Management", Director, Rittman Mead Consulting, 9 May 2008
8. Pierre Bonnet, "Enterprise Data Governance Master Data Management and Semantic Modeling: MDM", 1st Edition.
9. Tom White, " Hadoop: The Definitive Guide, Third Edition". Tom White has been an Apache Hadoop committer since February 2007, and is a member of the Apache Software Foundation.
10. Whei-Jen, Chen (2014), "Master Data Management for SaaS Applications".