Data Management Environment with Data Warehousing For Big Data Architecture

Mr. Yannam Apparao, Mr. Pilli Sudheer, Mrs. Kadiyala Laxminarayanamma

Abstract—Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, with extended data warehousing Architecture to the result of management reports, graphs, bar charts ,pie charts can provided, paper provides the repots and formats using extended data warehousing in big data.

Keyword-Big data; data warehousing; reports;

I. INTRODUCTION

Big data problems are often complex to analyze and solve. The sheer volume, velocity, and variety of the data make it difficult to extract information and business insight. A good first step is to classify the big data problem according to the format of the data that must be processed, the type of analysis to be applied, the processing techniques at work, and the data sources for the data that the target system is required to acquire, load, process, analyze and store.

A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject. Integrated: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product. Time-Variant: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

On-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered. Ralph Kimball provided a more concise definition of a data warehouse: A data warehouse is a copy of transaction data specifically structured for query and analysis. This is a functional view of a data warehouse. Kimball did not address how the data warehouse is built like Inmon did; rather he focused on the functionality of a data warehouse.

Manuscript received April 15,2015

Mr. Yannam Apparao, Associate Professor, Marri Laxman Reddy Institute of Technology and Management, Dundigal,Quthbullar(M),R.R. District, Telangana-500043.

M Pilli Sudheer, Associate Professor, Marri Laxman Reddy Institute of Technology and Management, Dundigal, Quthbullar (M), R. R. District, Telangana-500043

Mrs. Kadiyala Laxminarayanamma, Associate Professor, Institute of Aeronautical Engineering, Quthbullapur (M), R.R. Distic, Telangana, India

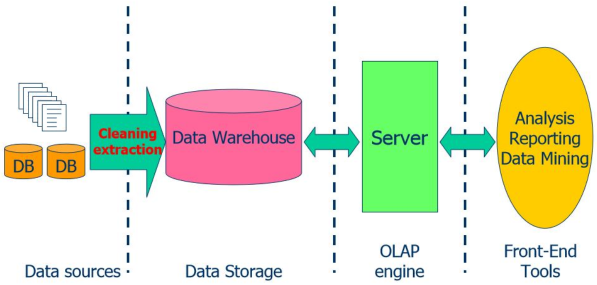

Figure 1: extended data warehousing Architecture to the result of management reports,

A) Overview

Big data can be stored, acquired, processed, and analyzed in many ways. Every big data source has different characteristics, including the frequency, volume, velocity, type, and veracity of the data. When big data is processed and stored, additional dimensions come into play, such as governance, security, and policies. Choosing an architecture and building an appropriate big data solution is challenging because so many factors have to be considered.

This "Big data architecture and patterns" series presents a structured and pattern-based approach to simplify the task of defining an overall big data architecture. Because it is important to assess whether a business scenario is a big data problem, we include pointers to help determine which business problems are good candidates for big data solutions.

B) Classifying big data to choosing a big data solution

To simplify the complexity of big data types, we classify big data according to various parameters and provide a logical architecture for the layers and high-level components involved in any big data solution. Next, we propose a structure for classifying big data business problems by defining atomic and composite classification patterns. These patterns help determine the appropriate solution pattern to apply. We include sample business problems from various industries. And finally, for every component and pattern, we present the products that offer the relevant function.

Part 1 explains how to classify big data. The following topics:

- Defining a logical architecture of the layers and components of a big data solution
- Understanding atomic patterns for big data solutions
- Understanding composite (or mixed) patterns to use for big data solutions

- Choosing a solution pattern for a big data solution
- Determining the viability of a business problem for a big data solution
- Selecting the right products to implement a big data solution

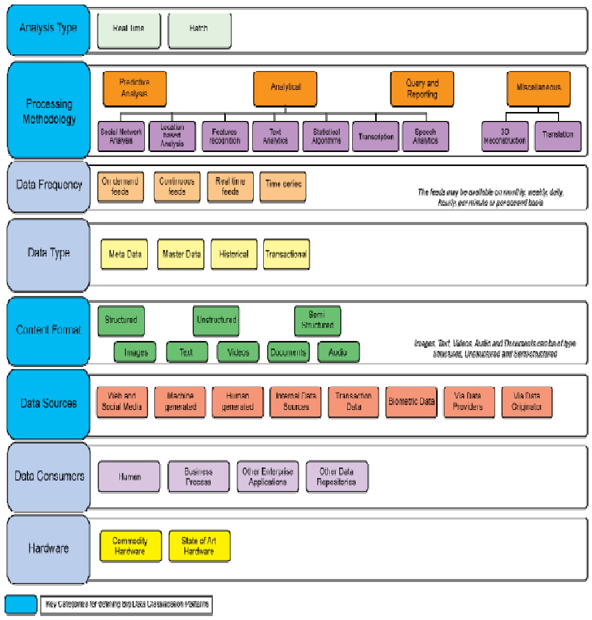

Figure 2: Big data Classification

II. BIG DATA FOR THE ENTERPRISE

With Big Data databases, enterprises can save money, grow revenue, and achieve many other business objectives, in any vertical.

A. Build new application

Big data might allow a company to collect billions of real-time data points on its products, resources, or customers – and then repackage that data instantaneously to optimize customer experience or resource utilization. For example, a major US city is using MongoDB to cut crime and improve municipal services by collecting and analyzing geospatial data in real-time from over 30 different departments.

B. Improve the effectiveness and lower the cost of existing applications

Big data technologies can replace highly-customized, expensive legacy systems with a standard solution that runs on commodity hardware. And because many big data technologies are open source, they can be implemented far more cheaply than proprietary technologies. For example, by migrating its reference data management application to MongoDB, a Tier 1 bank dramatically reduced the license and hardware costs associated with the proprietary relational database it previously ran, while also bringing its application into better compliance with regulatory requirements.

C. Realize new sources of competitive advantage

Big data can help businesses act more nimbly, allowing them to adapt to changes faster than their competitors. For example, MongoDB allowed one of the largest Human Capital Management (HCM) solution providers to rapidly build mobile applications that integrated data from a wide variety of disparate sources.

D. Increase customer loyalty

Increasing the amount of data shared within the organization – and the speed with which it is updated – allows businesses and other organizations to more rapidly and accurately respond to customer demand. For example, a top 5 global insurance provider, MetLife, used MongoDB to quickly consolidate customer information from over 70 different sources and provide it in a single, rapidly-updated view.

III. BIG DATA ARCHITECTURE TAKES LOGICAL TURN IN 'EXTENDED DATA WAREHOUSING FOR LOGICAL REPORTS'

A) Operational Big Data

For operational Big Data workloads, NoSQL Big Data systems such as document databases have emerged to address a broad set of applications, and other architectures, such as key-value stores, column family stores, and graph databases are optimized for more specific applications. NoSQL technologies, which were developed to address the shortcomings of relational databases in the modern computing environment, are faster and scale much more quickly and inexpensively than relational databases.

Critically, NoSQL Big Data systems are designed to take advantage of new cloud computing architectures that have emerged over the past decade to allow massive computations to be run inexpensively and efficiently. This makes operational Big Data workloads much easier to manage, and cheaper and faster to implement.

In addition to user interactions with data, most operational systems need to provide some degree of real-time intelligence about the active data in the system. For example in a multi-user game or financial application, aggregates for user activities or instrument performance are displayed to users to inform their next actions. Some NoSQL systems can provide insights into patterns and trends based on real-time data with minimal coding and without the need for data scientists and additional infrastructure.

B) Analytical Big Data

Analytical Big Data workloads, on the other hand, tend to be addressed by MPP database systems and MapReduce. These technologies are also a reaction to the limitations of traditional relational databases and their lack of ability to scale beyond the resources of a single server. Furthermore, MapReduce provides a new method of analyzing data that is complementary to the capabilities provided by SQL.

As applications gain traction and their users generate increasing volumes of data, there are a number of retrospective analytical workloads that provide real value to the business. Where these workloads involve algorithms that are more sophisticated than simple aggregation, MapReduce has emerged as the first choice for Big Data analytics. Some NoSQL systems provide native MapReduce functionality that allows for analytics to be performed on operational data in place. Alternately, data can be copied from NoSQL systems into analytical systems such as Hadoop for MapReduce.

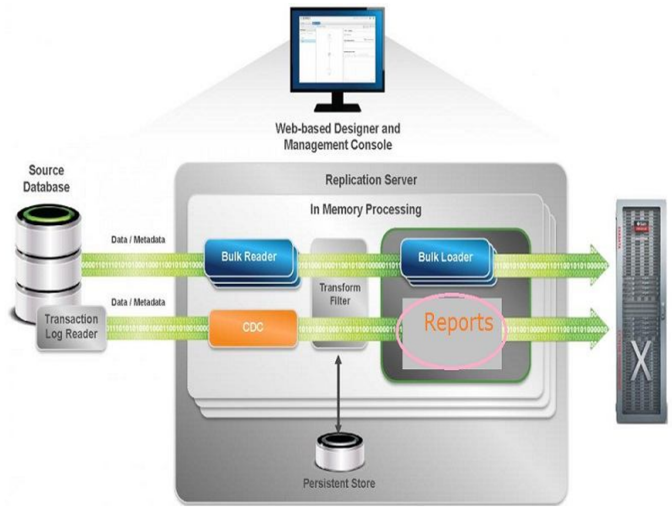

Figure 3: Architecture to the result of reports of big data by using data warehousing analysts.

IV. PROBLEM OR NEEDS ANALYSES

A. Preliminary Reports

A problem or needs-analysis report—a very preliminary piece of writing—examines a particular issue that the client faces. A problem or needs analysis is particularly appropriate when the need or problem is complex or ill defined. Perhaps the client is not convinced of the need for change, or personnel in the client organization have different views about the issue (which is often the case). A problem or needs analysis names the problem that you think shouls be addressed and provides analysis that supports your position. Such analyses are especially appropriate to a problem-solving paradigm.

Professional communication includes a variety of reports common to business and industry. Below we define many of them briefly and then point to further discussions and models available elsewhere in Professional Writing Online.

We have argued that you can identify reports by differences in audience and purpose. Actually, there are even more factors to include. Every discussion of report formats is complicated by the fact that reports within the same classification may vary in

- length
- complexity of audience
- formality
- purposes and functions inside and/or outside the organization.

This section describes the popular report types listed above and points to more detailed descriptions: it is a "nuts and bolts" discussion. We also direct your attention to writing reports in Principles: that section addresses writing reports more generally.

To comfortably grasp the selection of report types, you need to embrace report truth # 1:

Reports may be classified similarly and still function (and even appear) very differently in two different situations.

This will make sense in an immediately accessible context: Imagine that you want to ask a close relative (who gives you money freely) for money to support your business. Your approach would be very different than if you were to ask a venture capitalist for that same money. While you might write **a** proposal in each instance, these two proposals will not be similar in length, tone, support for the argument, financial statements,

V. REPORT SERVER DATABASE IN NATIVE MODE

A report server is a stateless server that uses the SQL Server Database Engine to store metadata and object definitions. A native mode Reporting Services installation uses two databases to separate persistent data storage from temporary storage requirements. The databases are created together and bound by name. By default, the database names are **reportserver** and **reportservertempdb**, respectively.

A SharePoint mode Reporting Services installation will also create a database for the data alerting feature. The three databases in SharePoint mode are associated with Reporting Services service applications. For more information, see Manage a Reporting Services Service Application

The databases can run on a local or remote Database Engine instance. Choosing a local instance is useful if you have sufficient system resources or want to conserve software licenses, but running the databases on a remote computer can improve performance.

You can port or reuse an existing report server database from previous installation or a different instance with another report server instance. The schema of the report server database must be compatible with the report server instance. If the database is in an older format, you will be prompted to upgrade it to the current format. Newer versions cannot be down graded to an older version. If you have a newer report server database, you cannot use it with an earlier version of a report server instances. For more information about how report server databases are upgraded to newer formats, see Upgrade a Report Server Database.

The report server database is a SQL Server database that stores the following content:

- Items managed by a report server (reports and linked reports, shared data sources, report models, folders, resources) and all of the properties and security settings that are associated with those items.
- Subscription and schedule definitions.
- Report snapshots (which include query results) and report history.
- System properties and system-level security settings.
- Report execution log data.
- Symmetric keys and encrypted connection and credentials for report data sources.

VI. ENCRYPTION KEYS IN BIG DATA AND CONFIGURATION MANAGER (MLRITkeymgmt)

To manage symmetric keys, you can use the Reporting Services Configuration tool or the **MLRITkeymgmt** utility. The tools included in Reporting Services are used to manage the symmetric key only (the public and private keys are managed by the operating system). Both the Reporting Services Configuration tool and the **MLRITkeymgmt** utility support the following tasks:

- Back up a copy of the symmetric key so that you can use it to recover a report server installation or as part of a planned migration.
- Restore a previously saved symmetric key to a report server database, allowing a new report server instance to access existing data that it did not originally encrypt.

- Delete the encrypted data in a report server database in the unlikely event that you can no longer access encrypted data.
- Re-create symmetric keys and re-encrypt data in the unlikely event that the symmetric key is compromised. As a security best practice, you should recreate the symmetric key periodically (for example, every few months) to protect the report server database from cyber attacks that attempt to decipher the key.
- Add or remove a report server instance from a report server scale-out deployment where multiple report servers share both a single report server database and the symmetric key that provides reversible encryption for that database.

A) Add and Remove Encryption Keys for Scale-Out Deployment

Use the **MLRITkeymgmt** utility to initialize a report server instance to use a shared report server database. Adding a report server to a scale-out deployment requires that you initialize the report server. Initialization requires administrator permissions. You must have administrator credentials for the remote computer that hosts the report server you are joining to the deployment.

- Run rskeymgmt.exe locally on the computer that hosts a report server that is already a member of the report server scale-out deployment.
- Use the -j argument to join a report server to the report server database. Use the -m and -n arguments to specify the remote report server instance you want to add to the deployment. Use the -u and -v arguments to specify an administrator account on the remote computer. If you are creating a scale-out deployment using multiple report server instances on the same computer, the syntax to use is slightly different. For more information about the syntax you should use, see rskeymgmt Utility (SSRS).

```
rskeymgmt -j -m <remotecomputer> -n <namedreportserverinstance>
-u <administratoraccount> -v <administratorpassword>
```

How to remove a report server from a scale-out deployment (rskeymgmt)

1. Open the rsreportserver.config file of the report server you want to remove and find the installation ID. By default, this file is located at Program Files\Microsoft SQL Server\MSSQL.*n*\Reporting Services\ReportServer).
   If you installed a single instance, there will only be one rsreportserver.config file on the computer. If multiple instances of Reporting Services are installed, use the Server Status page in the Reporting Services Configuration tool to find the instance identifier (for example, MSSQL.2) for the report server that you want to remove. The name of the folder that stores the program files for the report server instance will be based on the instance identifier (for example, Program Files\Microsoft SQL Server\MSSQL.2).
2. Run rskeymgmt.exe. You can run it on any report server that is part of the report server scale-out deployment.
3. Use the -r argument to release the report server instance from the scale-out deployment. The following example illustrates the arguments you must specify:

```
rskeymgmt -r <installation ID>
```

B) Configure and Manage Encryption Keys

Reporting Services uses encryption keys to secure credentials and connection information that is stored in a report server database. In Reporting Services, encryption is supported through a combination of public, private, and symmetric keys that are used to protect sensitive data. The symmetric key is created during report server initialization when you install or configure the report server, and it is used by the report server to encrypt sensitive data that is stored in the report server. Public and private keys are created by the operating system, and they are used to protect the symmetric key. A public and private key pair is created for each report server instance that stores sensitive data in a report server database.

Managing the encryption keys consists of creating a backup copy of the symmetric key, and knowing when and how to restore, delete, or change the keys. If you migrate a report server installation or configure a scale-out deployment, you must have a backup copy of the symmetric key so that you can apply it to the new installation.

VII. EXAMPLE REPORTS ON ENTERPRISE NVIRONMENT

Risk management practices taking place under a different label, such as those implemented by different staff functions or under the auspices of executives other than the CRO, have been excluded from these studies. But our main criticism is that many of the studies rely on simplistic variables to represent complex behavior. For example, the single 0-1 dummy variable of ERM adoption does not capture how ERM is actually implemented. Studies that rely on S&P's ERM ratings must assume that the rating agency's arm's-length assessment of a firm's ERM processes, based on public information, is a valid indicator of the risk management processes actually implemented in the firm. Because of these shortcomings, most empirical studies explain only a small fraction of the variability in the adoption or impact of risk management and have low statistical significance for key explanatory variables. Further, the large-sample cross-sectional studies focus on the adoption of a particular risk management framework (for example, COSO's ERM) but ignore how the framework was implemented by the organization's leadership and employees.

The effectiveness of risk management ultimately depends less on the guiding framework than on the people who set up, coordinate, and contribute to risk management processes. It is people, not frameworks, that identify, analyze, and act on risk information. Their actions often require approval from the CEO and board. So the different organizational and cultural contexts in companies following the same ERM framework can lead to different implementation and use of risk management frameworks.

For example, all Wall Street financial firms had risk management functions and CROs during the expansionary period of 2002-2006. But some of these firms failed in the subsequent crisis while others survived quite well. The existence of a risk management department and an individual with the title of chief risk officer explains very little about the quality, depth, breadth, and impact of a firm's risk management processes. For example, the fact that a company had a risk management department with a CRO does not predict that the department had the backing of the CEO and board to encourage the production and dissemination of risk information, nor that it had the resources, leadership, and support to mitigate the principal risks the risk department identified.

The essence of a contingency theory of ERM (beyond the simple selection / correlation studies) would be to find "fit" between contingent factors and firms' ERM practices, and to establish propositions of fit that will result in desired outcomes (for a review of contingency studies in management accounting, see for example Otley, 1980; Fisher, 1995; Chenhall, 2003, 2006). Moving towards a contingency theory of ERM requires a more sophisticated understanding of not only the nature of relevant contingencies, but also the nature of ERM itself.

Figure 4: Sales charts yearly wise.

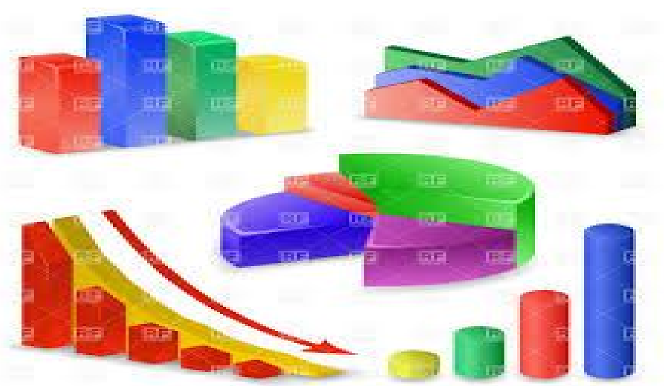

Figure 5: Sales Up and down prices in periodically charts

VIII. FUTURE SCOPE

The scope of the paper will extended up to the time period automatic generation of reports in the form of graphs, charts.

IX. CONCUSSION

This paper supports the reports with respect to the visualization in big data by using extended data warehousing architecture.paper provides the automatic generation of reports with in the peak time period by using Architecture takes logical turn in ‘extended data warehousing for Logical Reports’ and data management environment for unstructured information.

REFERENCES

[1] http://www.hbs.edu/faculty/Publication%20Files/13-063_5e67dffe-aa5e-4fac-a746-7b3c07902520.pdf
[2] http://wps.pearsoncustom.com/pls_1256647969_pwo/217/55689/14256538.cw/content/index.html
[3] https://msdn.microsoft.com/en-IN/library/ms156016.aspx
[4] Considerations for Big Data: Architecture and Approach by Kapil Bakshi – Paper published in IEEE
[5] Big Data Analytics by Sachidanand Singh – Paper published in 2012 International Conference on Communication Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India
[6] McKinsey Global Institute
[7] http://hadoop.apache.org/
[8] Kimmo Palletvuori, Security of Data Warehousing Server, Helsinki University of Technology
[9] John D. Porter and John J. Rome, Lessons from a Successful Data Warehouse Implementation, Arizona State University
[10] Security and the Data Warehouse, An Oracle White Paper, April 2005

Mr. YANNAM APPARAO, Currently working as Associate Professor, Marri Laxman Reddy Institute of Technology and Management, Dundigal, Quthbullar(M),R.R. District, Telangana-500043.

Mr. P.SUDHEER, Currently working as Assistant Professor, MLR Institute of Technology & Management, Dundigal,Quthbullar(M),R.R.Distic, Telangana-500043. India

Mr. P.SUDHEER, Currently working as Assistant Professor, MLR Institute of Technology & Management, Dundigal,Quthbullar(M),R.R.Distic, Telangana-500043. India