JONAH GROUP

Agile Business Intelligence
Data Lake Architecture

TABLE OF CONTENTS

INTRODUCTION

Building a central data warehouse is a long and expensive process. At the end of this process we often find that the initial requirements were either not completely met, incorrectly specified or they had changed dramatically. In fact BI requirements are as fluid and volatile as the business itself.

With our foundation in Custom Software development Jonah is always looking for ways to improve the effectiveness and timeliness of BI projects. We have found that data warehouse projects are typically organized in a waterfall project approach that suffers from several issues:

- A very long development cycle for medium-large data warehouses on the order of 2-3 years to release usable data marts and systems.
- Data models are built that do not accommodate business environment and requirements changes. Needs are often identified once data is used leading to expensive changes and rework.
- Highly precise models require expensive data cleansing, exact ETL programs and careful tracking of upstream data sources.

In the Software Development world much progress has been made using Agile processes that allow projects to proceed in short iterations releasing usable systems frequently. Agile processes are more responsible to business changes.

This white paper outlines a new architecture based on the concept of Data Lakes – unstructured data landing areas that allow experimentation, analysis and creating usable data much more quickly than standard data warehouse methods.

DATA LAKE ARCHITECTURE

The Data Lake architecture model is a new BI Architecture that eliminates and/or reduces deficiencies in other models. It is similar in concept to the Centralized Data Warehouse with dimensional Data Marts but with the advantages available now through the use of new technologies such as Hadoop and MapReduce.

In this architecture, Hadoop is used as the central repository and staging area of all data. The following table outlines some of the standard components of Hadoop:

Architecture Component	Description
Apache Hadoop	Hadoop is an open source project from Apache that has evolved rapidly into the mainstream of the technological tools used by many large companies. Originally it emerged as the best way to handle massive amounts of data, both structured corporate data as well as complex unstructured data. The Hadoop environment has become a new paradigm for BI strategy and deployment. Its popularity is due in part to its ability to store, analyze and access large amounts of data quickly and cost effectively across clusters of commodity hardware. Additionally, Hadoop vastly simplifies the BI process from the collection of corporate data through to final consumption by the business community. Apache Hadoop is an information ecosystem comprised of several components that create a coherent and inexpensive platform to run, amongst other things, a robust BI environment.
Hadoop Distributed File System (HDFS)	A reliable and distributed Java-based file system that allows large volumes of data to be stored and rapidly accessed across large clusters of commodity servers.
MapReduce	A framework for writing applications that processes large amounts of structured and unstructured data in parallel across large clusters of machines in a very reliable and fault-tolerant manner.
Pig	A platform for processing and analyzing large data sets. Pig consists on a high-level language (Pig Latin) for expressing data analysis programs paired with the MapReduce framework for processing these programs in parallel.

One of the key benefits of the Data Lake approach is that it takes advantage of inexpensive storage in Hadoop with its inherit simplicity of storing data based on its schema-less write and schema based read modes.

While writing data to the Hadoop File System there is no need to define the schema of the written data. This means that we can have a staging environment that stores all the data without designing and building a centralized Data Warehouse. However, we now have all of the data at our disposal for analysis and reporting. In fact, the Data Lake architecture offers all of the functionality of a traditional Centralized Data Warehouse but without the upfront development costs.

From the central Hadoop repository business-specific models can be constructed and designed to create Data Marts. Each Data Mart implements a dimensional model for a specific subject area.

The architecture in Figure 1 illustrates how the Data Lake components work together:

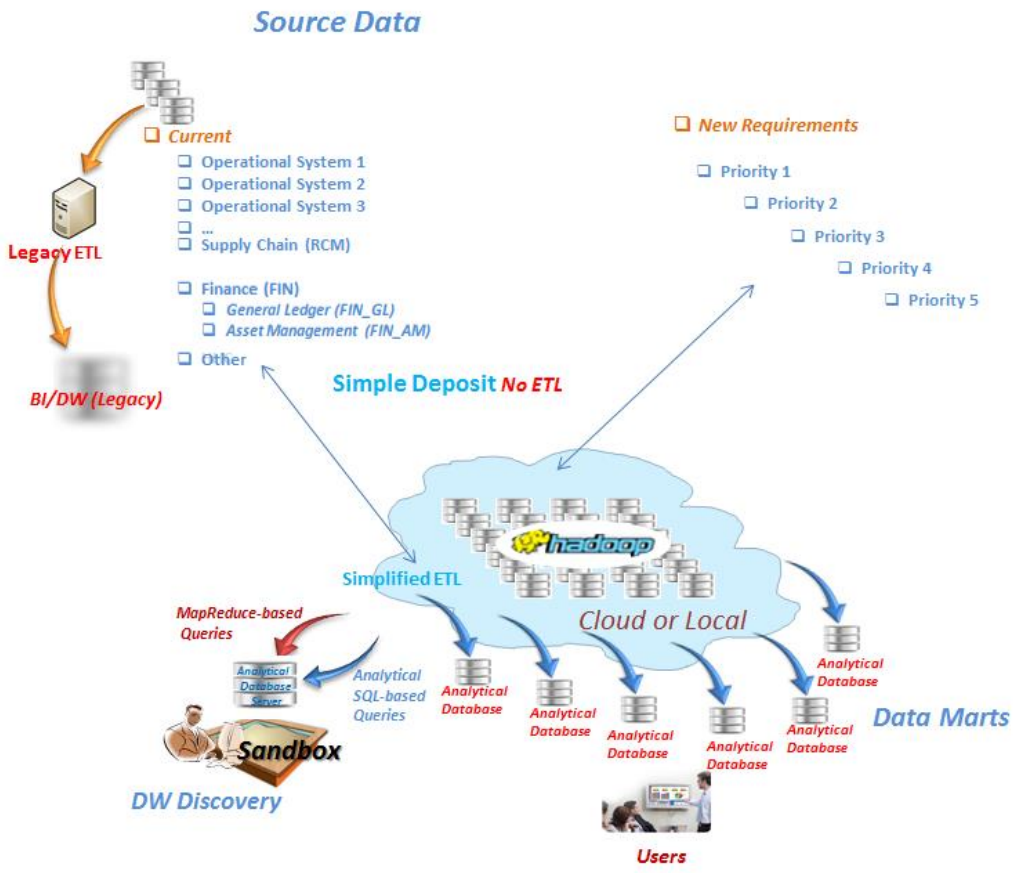

Figure 1 - BI Data Lake Architecture

The second significant simplification and cost reduction is the elimination of upfront ETL processes. In most projects the majority of the effort is spent on analysis, design and development of ETL processes. This is often a multimillion-dollar commitment both in initial license and development costs as well as on-going maintenance costs.

In the Data Lake Architecture the traditional heavy ETL scripts and programs can be implemented using the Pig language in conjunction with MapReduce. This results in very lightweight ETL that will save thousands of lines of traditional ETL scripts and programs. Pig operates at a higher level so developers can focus on data processing instead of ETL programming.

STEP 1 – EXTRACT FROM SOURCE DATA

The periodical extracts from various source data will be deposited directly into a Hadoop Data Lake (HDFS).

- If the file does not exist in the landing area, it will be recorded (metadata managed by HCatalogue will be updated with the name of the file, timestamp and any other necessary info) and passed on for further processing as described in Step 2.
- If the file already exists and needs to be replaced or modified, HCatalogue will be updated and the processing continues in Step 3.

STEP 2 – REGISTER AND CATALOGUE DATA SETS

The Hadoop HCatalogue component contains all of the metadata necessary to maintain HDFS integrity and is an integral part of the Hadoop architecture. The proper registration and audit trail of all objects copied to Hadoop will be kept and maintained an on-going basis. This will allow the system to properly identify and track duplicate and redundant data. The Hadoop HDFS system serves as a single, large and perpetual, staging data area.

STEP 3 –EXTRACT INTO DATA MARTS

The new extract will be processed by Pig Latin scripts to load data into business-domain specific Data Marts.

STEP 4 – BI FRONT-END

The front-end BI tool (e.g. Business Objects) will connect to a Data Mart and provide information for consumption by the business user. The data may be enriched by GIS

systems or by access to a searchable document repository, allowing the user to support data visualization and executive dashboards.

DATA WAREHOUSE DISCOVERY MART

THE PURPOSE

A major challenge in developing worthwhile and relevant business solutions is to accurately capture the underlying or hidden business information requirements. This is a difficult task since end-users may have difficulty articulating what they really need.

The Data Warehouse Discovery Mart framework provides a basis for identifying, evolving and validating business requirements as well as developing a technology plan that satisfies cross-functional business needs. It also provides a test bed of emerging solutions that can be vetted and modified prior to making significant commitments in resources and development.

On the surface, gathering requirements for informational systems may appear to be a somewhat easy task. However, in many cases this exercise is most effective when an individual experienced in decision support environments leads the business community using "peel-the-onion" techniques to effectively draw out and share requirements. This involves much more than merely asking the business community to articulate what they think they need. The best way to conduct such an exercise is to have a flexible data repository with the ability to query data in real time during conversations with the business user. The individual leading the exercise needs to have a detailed understanding of the nature of the end-state solution.

DISCOVERY INFORMATION FLOW

In order to setup the discovery framework all of the source systems will be identified and documented and then a "fast load" of data to the discovery repository will be completed with little or no ETL effort. At that point the repository will be ready to for any ad-hoc queries to:

- Articulate user needs
- Vet and verify the needs interactively with business users
- Communicate the needs to the IT team
- Present the proposed solution to the user

Figure 2 illustrates the information flow:

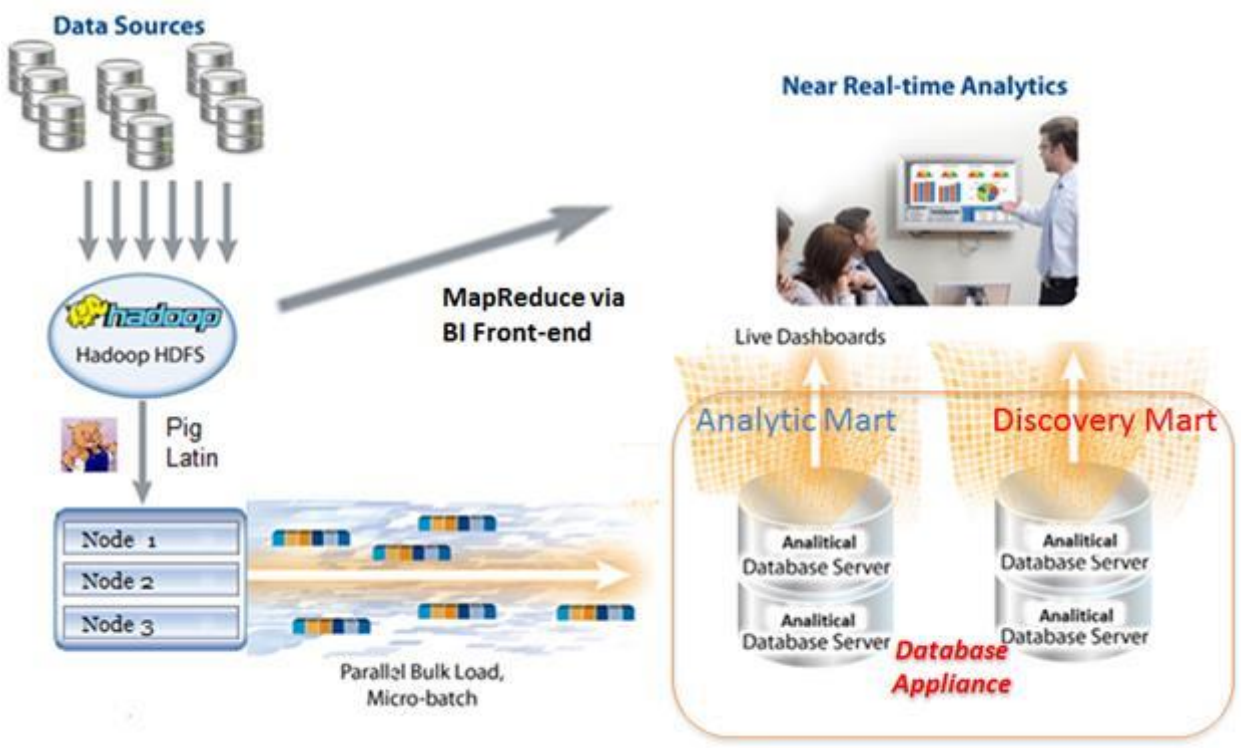

Figure 2 - Discovery Mart Information Flow

A separate database repository is created for the sandbox environment. A portion of the production data is loaded into the sandbox in "as is" format with little or no ETL effort being applied to the data. The sandbox is used to explore the data with business users and identify business requirements.

Furthermore, this repository will not require any physical design, tuning or any significant database administration effort. The source data is loaded to support the initial and ongoing architecture design. Data is loaded into "large and flat" tables with no need for indexes or table spaces. This design provides a low cost sandbox environment with minimal maintenance requirements.

The following features are instrumental in its development and implementation:

- Ability to support wide and de-normalized data sets
- Small footprint
- Fast load
- Ad-hoc query capability across de-normalized tables
- Low maintenance costs

- No need for physical design
- Low technology footprint
- Capable of hosting and/or integrating with and analyzing "Big Data"
    - Social Network data
    - Huge volumes
    - Logs of different origins
    - Unstructured and semi-structured data

SHORT TERM BENEFITS

- Enhanced Data Relationship Knowledge and Business Rule Design – allows identifying additional detailed BI query needs in a low cost environment.
- Mitigate the risk of one shot large builds that do not meet business requirements.
- Validate schema and evolve the design.
- Validate and vet detailed requirements and analyze queries before undertaking a costly and lengthy development process.

LONG TERM BENEFITS

- As business conditions change the Data Lake allows quick testing and analysis of the impact of these changes.
- The rapid design, analysis and build cycle, in a DW Discovery Mart, takes 1-2 weeks versus 6-12 months using traditional data warehousing techniques.
- Low cost and rapid deployment – easy to maintain with less need for ETL programs, integration logic, and minimal data modeling.