Offload EDW to Big Data Lake

Oracle Exadata, Teradata, Netezza and SQL Server

Ample White Paper
EDW (Enterprise Data Warehouse) Offloads

The EDW (Enterprise Data Warehouse) platform provides an effective way of integrating data from disparate and distributed sources and it serves as the foundation for any enterprise that requires analytic reporting of current and historical data and trends. Originally, the term “data warehouse” was coined by IBM researchers, Barry Devlin and Paul Murphy, to summarize a model for providing decisional support systems with data that has been collected from various operational systems, and centralizing that information so that it can be processed and analyzed.

Data warehouses continue to be used as the basis for making tactical and strategic decisions, and the EDW can provide answers to business questions that would not otherwise be possible. Figure 1 is an example of a data warehouse configuration that shows how data from various operational systems can be integrated and housed in a ‘data vault’ from which new information can be developed.

EDW platforms represent the realization of the ‘hub-and-spoke’ architecture in which systems act as spokes, and hubs are represented as locations from where data is loaded, integrated for analytics, and

---

**Figure 1**

(Source: Wikipedia.org)
Offload EDW to Big Data Lake

optimized to preserve integrity. For example, CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning) solutions generate transaction data sets and act as the ‘spokes’ in this model. Different vendors provide EDW platforms, and each may act as a ‘hub’ in the overall system.

EDW Challenges and Limitations

The support and maintenance activities of EDW processes have become more complex due to the increased growth of data sources, which at the current rate, doubles almost every year. For example, new data sources may be added to the data warehouse monthly, if not weekly. In addition the IoT (Internet of Things) is adding a tremendous amount of data to the Internet and experts estimate that the IoT will consist of almost 50 billion objects by the year 2020. Synchronization is a common problem when data needs to be consistently distributed among several databases. Currently, objects are processed in their databases sequentially, and sometimes the slow process of database replication may be involved as the primary method of transferring data between databases. Problems arise when the time to extract and load the data is limited. A further complication is that existing EDW platforms are generally not scalable; consequently they cannot process incoming data rapidly enough. To summarize, the challenges faced when processing massive amounts of data are as follows:

- Current EDW platforms were not designed for procedural processing.
- Disparate data types cannot be supported (such as semi-structured and unstructured data).
- The cost per terabyte of storage is prohibitively expensive.
- The I/O per gigabyte is not justified for complex SQL databases.
- Linear scalability among these systems is virtually non-existent.

Solution: The Ample Big Data ETL Offload Process

Ample Big Data provides solutions to solve the problems inherent in traditional data warehouse technologies by employing the NoSQL (Not Only SQL) database architecture within a Hadoop software ecosystem.

NoSQL is based on the concept of distributed databases, where unstructured data may be stored across multiple processing channels, and often across multiple servers. Unlike relational databases that are highly structured, NoSQL databases are unstructured, trading stringent consistency requirements for speed and agility. This distributed architecture allows NoSQL databases to be horizontally scalable. As data volume continues to increase, more hardware is added to keep up with processing, without slowing the performance down.
Offload EDW to Big Data Lake

The Hadoop software ecosystem allows for massive parallel computing. It is an enabler of NoSQL distributed databases allowing data to be spread across servers with little reduction in performance. Because Hadoop is schema-free, where a key characteristic of the implementation is ‘no-schema-on-write’, it can manage structured and unstructured data with ease, and aggregate multiple sources to enable deep analysis. Hadoop makes it is unnecessary to pre-define the data schema before loading; it can be loaded as is, regardless of whether incoming data has an explicit or implicit structure. After loading, the data can be integrated and then prepared for business consumption through the access layer.

Due to the economies of the scale that Big Data represents, enterprises no longer need to discard unused data that might otherwise provide value when later analyzed from multiple perspectives. Ample has built multiple connectors to NoSQL platforms such as Hadoop for rapid bulk data ingestion. Parallel processing and unstructured data ingestion with Big Data technology can be leveraged to enhance the existing ETL capabilities for most enterprises.

Ample has built a proven framework called the ‘Ample Data Lake’, to address the problems that arise during ETL processing. The Ample Data Lake can resolve these problems holistically, or it can determine a solution in a step-by-step fashion, depending on the nature of the problem being analyzed. Ample has benchmarked its Data Lake Solutions at every stage of the process; by terabyte utilization, by CPU workload, and by I/O limitations. Multiple channels where customers interact have their own unique data silos with data sets existing in multiple data structures, data formats and types. Ample has developed the models necessary for extracting, loading, and transforming this varied data, making it available for analytic processing.

An illustration of an Ample Data and Analytic Ecosystem can be seen in Figure 2. It shows the dual pathway that data processing may take during Data Gathering, depending on whether the data source is semi-unstructured or structured, followed by Data Integration and Data Accessing.
Offload EDW to Big Data Lake

Figure 2
Figure 3 is an illustration of an Ample Data and Analytical Ecosystem that highlights some of the features, properties and services that are provided using this data processing configuration.

**AMPLE DATA AND ANALYTIC ECOSYSTEM**

- Historical data
- Exploratory data
- Low latency and batch processing
- Mapreduce
- Graph processing

---

**Figure 3**

**Ample Big Data EDW Offload Benefits**

The Ample Big Data framework saves time and money while delivering more optimized EDW solutions. Because of the inherent nature of the Big Data platform and the infrastructure used for the Data Lake process, scaling of these technologies proves to be extremely cost-effective.

- Enterprises no longer need to discard large data sets.
- Multiple websites or cross-sites can be quickly analyzed in one platform.
- The Data Lake Solution performs much faster for high concurrency queries with optimization.
- Data growth can be managed more effectively.
- The cost per terabyte and data processing costs can be decreased.
Conclusion

Focusing on the EDW offload process has never been as prevalent as it is now when data ingestion, data processing, data mining and data visualization are integrated together using Big Data technologies and environments. Needless to say, the exponential growth in data is the main reason why the Big Data platform is appropriate for EDW offloading.

However, it is difficult and often impossible to perform deep levels of analysis on traditional platforms because of their non-scalable aspects, which results in increased costs and processing complexities. The ingestion and processing of semi and unstructured data sources pose other technological challenges for traditional EDW platforms. An added challenge is that any solution to these issues must be accomplished with quality data oversight and a high-degree of data governance.

Why compromise on quality, principles, best practices and utilization when the Ample framework provides a comprehensive solution to these data driven issues and strategic needs?