CHAPTER 6

Integrating the Data Warehouse and Analytics Infrastructure to Big Data
Big Data initiatives often start as experiments or research and development projects in IT or the lines of business to determine potential business value in analyzing such data.

Once the initial value of the solution is proven, organizations plan full-scale deployment and commonly include the existing data warehouse infrastructure as part of the Big Data infrastructure for several important reasons.

The data warehouse often serves as the home of the historic database of record within an organization. Existing, understood, and well-utilized business intelligence tools for reporting, querying, and analysis serve as the means for extracting business value from this database of record. The data warehouse is typically designed to facilitate navigation of the data by business analysts and to meet service level agreements (SLAs) for performance and availability. In addition, data warehouses based on a relational database provide levels of access and data security not achievable in a Hadoop cluster today. It should therefore come as no surprise that in organizations that have mature Hadoop clusters, data that is deemed valuable in Hadoop usually makes its way to the data warehouse.

We’ll begin by describing the role of data warehouse and business intelligence infrastructure. We’ll also look at how related Oracle products help complete the Big Data infrastructure. Finally, we’ll revisit building out the infrastructure.

The Data Warehouse as a Historic Database of Record

Many organizations have established the need for a historic database of record, a place where data that dates back many years is available for historical reporting and for comparative analysis with current data. The timeframe the data represents and the granularity of the data available are often driven by legal compliance regulations and business analysis requirements. Historically, sources of such data have primarily come from relational databases, though other data sources are now becoming quite common. The kinds of reporting and analysis that occur in this historic database of record also lent themselves to deployment of a relational database.

When data is stored at its most granular level in the data warehouse, it is usually stored in what is called a third normal form (3NF) schema, designed to reduce data anomalies and redundancies. Such a 3NF schema generally serves as the basis of what is frequently described as an enterprise data warehouse. Of course, data in source systems that is to be loaded in the data warehouse is very often not consistently defined or validated. In such cases, the data must be cleansed or transformed when it is extracted from the sources and loaded into the target data warehouse. The need for this extraction, transformation, and loading (ETL) process gave rise to tools and utilities to define and interactively design the process, and generate code for the ETL scripts.
The data transformations can introduce time delays, also known as latency, during the ETL process. For near-real-time data feeds, replication solutions are sometimes selected as alternatives to ETL. However, these are intended to work primarily as store-and-forward mechanisms, so the key to such an approach is usually to ensure that data is clean and consistent at the source.

Data in 3NF is fine when business analysts are using reports that have been generated by an IT person familiar with this type of schema. However, for ad-hoc queries and drill-downs initiated by business analysts, the data should be presented in a more intuitive fashion for exploration. Initially, IT organizations built views on top of 3NF schema, but the maintenance of views proved to be problematic and difficult to modify in timely response to business requests. So the star schema emerged as an approach in data warehouses to enable business analysts to more easily navigate through the data themselves. A star schema stores the data in large transaction or fact tables and is surrounded by lookup or dimension tables. Today, most large-scale Oracle-based data warehouses are deployed as a hybrid mixture of 3NF for base data and star schema that include summary-level hierarchies.

The Oracle Database as a Data Warehouse

Oracle relational databases have served as “decision support systems,” later called data warehouses, almost from their inception. In the mid-1990s, Oracle Parallel Server and Oracle7 became engines for terabyte-sized data warehouses. Today Oracle-based data warehouses have grown into the petabytes thanks to many further improvements in the Oracle Database and the introduction of engineered systems such as the Oracle Exadata Database Machine.

As organizations often rely on their historic database of record to make critical “just-in-time” business decisions, the data warehouse must exceed both high service levels for performance and demanding availability requirements. Oracle has addressed performance needs in the Oracle Database by providing increasing sophistication in its optimizer and improved self-tuning and management over the years. The Database feature list only continued to grow for supporting data warehousing workloads. Some of the key innovations introduced in the Oracle Database along the way to improve query performance include

- Bitmap indexes for optimally storing and retrieving low-cardinality data
- Parallel bitmap star joins for higher-speed joins
- Materialized views and summary tables for facts and dimensions

Smart scans and other performance optimizations unique to Exadata Storage Servers Analytics performance and capabilities have improved in the Oracle Database through support of SQL analytic functions, embedded online analytical processing
(OLAP), and embedded data mining and R support in the Advanced Analytics Option. ETL performance improved through introduction of embedded ETL functions.

As the data warehouses scaled into the petabytes in size, manageability and availability improvements were made through added functionality in Oracle Database releases, often also benefiting performance of queries. Some of these include the Partitioning Option, Automatic Storage Management providing striping and mirroring, Real Application Clusters (RAC) for scaling multiple nodes and instance failover, and Data Guard to propagate data from primary to secondary sites in case of failure.

Making sure data is available to the right business analysts at the right level of detail also leads to data security considerations. Key security functionality in the Oracle Database relevant in data warehousing includes:

- Virtual Private Database (VPD) data access based on user roles
- Advanced security supporting multiple levels of access on a need-to-know basis
- Audit Vault providing the ability to understand who accessed data and what they did

We discuss some of these capabilities elsewhere in this book in more detail. However, as entire books have been written on this subject, you will want to explore those as well.

**Why the Data Warehouse and Hadoop Are Deployed Together**

Many of the features mentioned in the previous section provide capabilities not offered by Hadoop today. This is partly due to the maturity of relational database technology since it has a much longer history than Hadoop and partly due to the very different roles for each as originally intended. Remember, relational databases are optimized for structured high-density data that lends itself to storage in rows and columns. Hadoop is optimized for storing and processing large volumes of high-granularity data in a distributed file system regardless of data relationships. Each Oracle-engineered system (for example, the Oracle Exadata Database Machine and Oracle Big Data Appliance) is an optimized combination of hardware and software for the workloads to be run on the platform.

Of course, when a Hadoop programmer finds valuable data in their analysis, the data is often propagated to the data warehouse for the following reasons:

- The output from Hadoop is used to augment other data in the warehouse for analysis.
- Data in the data warehouse can be more easily navigated by business analysts.
Data in the data warehouse is better secured for broader user communities.

- The data warehouse has well-defined business SLAs for performance and availability.
- Archival policies are well-tested, proven, and enforced for data in the data warehouse.

When data flows from the Hadoop cluster to the Oracle Database, valuable information is exposed to a wider set of consumers. Oracle offers a high-speed loader to facilitate data movement in this direction, Oracle Loader for Hadoop (OLH), which is part of the Oracle Big Data Connectors offering. Given that ETL tools are commonly used for defining loading processes into Oracle data warehouses, OLH is supported as an object in Oracle’s Data Integrator tool (ODI). The Oracle Data Integrator Application Adapter for Hadoop enables ODI to generate optimized HiveQL code. The Oracle SQL Connector for Hadoop Distributed File System (OSCH) provides SQL access to HDFS for querying Hive tables and delimited text files and is supported in ODI.

If the destination for data is a non-Oracle data warehouse, the data is typically loaded into a generic data warehouse using standard Hadoop utilities (for example, Sqoop) or by using high-performance loaders offered by third parties. In some situations, Hadoop itself is made part of custom ETL processes to speed transformations, given its highly parallel nature and its support for advanced statistical analysis useful in building out transformations.

There are also situations where it makes sense for data to flow from the data warehouse into the Hadoop cluster. If most of the data to be analyzed lends itself better to analysis in Hadoop, it can make better sense to load reference data from the data warehouse into the Hadoop cluster. For such situations, you would likely use standard Hadoop Sqoop import capabilities.

The direction in which you choose to move data should match business requirements and functionality provided by the relational database engine and Hadoop cluster. Over time, the functionality mix will increasingly overlap in each as more Hadoop capabilities appear in relational databases and more relational database capabilities appear in Hadoop. For example, the Oracle Database has featured documented support of MapReduce-like functionality through the use of table functions going back many releases, and Oracle Database 12c features improved MapReduce-like functionality. That is not to say that you would use the Oracle Database as a substitute for Hadoop when you investigate the data structures in play and optimizations, the relative costs of the platforms, and other aspects, but you might use this capability where a blended approach is required and heavily tilted toward analysis of data residing in the Oracle Database. Similarly, if the blended approach is heavily tilted toward analysis of data in Hadoop, you might lean toward also loading data from the Oracle Database into Hadoop for analysis.
It is also worth noting in this section that you might want to build applications that can filter your data in an event-based fashion before it arrives in Hadoop or your data warehouse. Oracle Event Processing (OEP) includes an event visualization console useful in setting up ongoing queries of live data streams for such filtering.

Completing the Footprint: Business Analyst Tools

Most business analysts access their Big Data and data warehousing infrastructure using a variety of tools. Data discovery tools can provide an initial starting point where data can be gathered from structured and/or semistructured sources, and the path needed to gain value from the data is initially uncertain. Oracle’s Endeca Information Discovery provides such a tool and includes the Endeca Server, which provides multiple drill paths through these types of data.

When building a schema makes sense, structured data typically is stored as previously mentioned in relational database schema, while semistructured data might be logically defined through Hive in Hadoop. Access to such data is commonly available through business intelligence tools that provide dashboards, reporting, and ad-hoc query capabilities. Oracle’s Business Intelligence Foundation Suite is a typical example of a suite of tools providing access to a variety of data. At the time this book was being published, most major business intelligence tools vendors for relational databases were either supporting Hive or had plans to do so in the future via Open Database Connectivity (ODBC).

There are a growing number of statisticians who are now using the R statistical packages to analyze data residing in relational databases and in Hadoop clusters. Oracle provides support to run R programs without moving data out of an Oracle database as part of its Advanced Analytics Option. The Oracle Big Data Connectors enable R programs to be run against data residing in a Hadoop cluster using Oracle R Connector for Hadoop.

Data mining provides a means to determine and fine-tune mathematical algorithms that will predict the outcome of an event where there are a large number of contributing factors. Oracle provides in-database data mining algorithms as part of its Advanced Analytics Option and also offers a Data Mining Workbench for building out the data mining process flow. Hadoop distributions include Mahout, a programming library for data mining.

Some organizations incorporate a real-time recommendation engine as part of their deployment solution. For example, when someone is navigating a Web site, based on their profile, it might make sense to present them with a special sales offer or point them toward other products they might want to look at or other people they might want to connect with. Real-time recommendation engines are often custom built for this purpose. They are periodically updated using data and mathematical
models in the Hadoop cluster and/or data warehouse. Though the back-end infrastructure operates in batch mode, the recommendations appear to be made in real time. Oracle offers a recommendation engine that is called Oracle Real-time Decisions (RTD) for enabling such business rules.

Figure 6-1 illustrates an all-Oracle version of this complete footprint. Obviously, many organizations mix technology footprints from a variety of vendors in building out a footprint like this. Some are finding management, cost, and time-to-market benefits by consolidating the number of vendors involved as they decrease the number of support points and gain better out-of-the-box integration.

**Building Out the Infrastructure**

There is no single right answer as to how you should go about building a complete infrastructure that includes Big Data. Organizations start with unique analytics footprints they’ve established and have differing business requirements they need to solve. Most organizations taking on a Big Data project have a data warehouse (or at least departmental data warehouses called data marts) and some sort of business intelligence tools.

It should be apparent that Hadoop will not replace a relational database data warehouse. However, many organizations previously were not able to analyze data from sensors, social media, and similar sources because of technical limitations in their relational databases. Hence, establishing and proving a new business use case...
for analyzing this type of data is usually the first step an organization will take when embarking on a Big Data project. A prototype is typically built on a small Hadoop cluster using a representative sample of data. In some situations, using a data discovery tool like Endeca might speed the process in establishing the potential value of analyzing semistructured data and deploying a Hadoop cluster.

Building a Hadoop cluster prototype can also be beneficial in helping to build Big Data related skills in your organization. The combination of programming skills (including MapReduce, Pig, HiveQL, or other scripting languages like Python), data analysis and distributed data optimization, data integration, and business skills is often difficult to find. Many organizations use a prototype effort as a learning experience for how to manage such projects and develop skills among their junior-level programmers and recent graduates.

Once the value is demonstrated, the next step is to move beyond a Big Data prototype and to size the Hadoop production environment. As we mentioned earlier, the Hadoop cluster is not seen as being particularly resilient when compared to the data warehouse, but you should take advantage of Hadoop availability characteristics that do exist, such as the triple replication of data and configuration of multiple highly available name nodes. By the very nature of Big Data, the storage required will grow quickly and incrementally and likely will quickly exceed the size of your data warehouse.

At this point, you should also have determined who will get business value from this data and how it should be accessed and further analyzed. You should then consider how you’ll integrate the Hadoop cluster into your existing analytics footprint and any integration software you might need in addition to what is provided by Hadoop and your existing software. You’ll also need to evaluate the impact on data warehouse availability and management procedures that may be of little interest to your prototype developers, but which likely will be critical to the overall success of the infrastructure in your business.

Further data and Hadoop cluster growth will likely be driven by business needs that mandate that longer periods of data will be kept in the cluster and new demand for storing data from additional sources. At each step of the way, you should be prepared to build more business cases justifying additional expansion of the overall infrastructure. Of course, it won’t hurt to have proof points that validate earlier business cases that were presented as well.