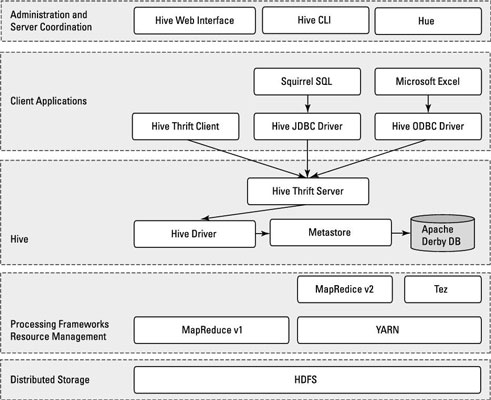
Assgnment 63

As you examine the elements of Apache Hive shown, you can see at the bottom that Hive sits on top of the Hadoop Distributed File System (HDFS) and MapReduce systems.



In the case of MapReduce, the figureshows both the Hadoop 1 and Hadoop 2 components. With Hadoop 1, Hive queries are converted to MapReduce code and executed using the MapReduce v1 (MRv1) infrastructure, like the JobTracker and TaskTracker.

With Hadoop 2, YARN has decoupled resource management and scheduling from the MapReduce framework. Hive queries can still be converted to MapReduce code and executed, now with MapReduce v2 (MRv2) and the YARN infrastructure.

There is a new framework under development called Apache Tez, which is designed to improve Hive performance for batch-style queries and support smaller interactive (also known as *real-time*) queries. At the time of writing, the Apache Tez project is still in incubation, and doesn’t yet have a production-ready release.

If it helps you visualize how all the pieces fit together, think of the HDFS and MapReduce systems as being parts of the Apache Hadoop operating system, with Hive — as well as other components, such as HBase — as higher-level functions or applications. (You can see a common theme emerge: HDFS provides the storage, and MapReduce provides the parallel processing capability for higher-level functions within the Hadoop ecosystem.)

Moving up the diagram, you find the Hive Driver, which compiles, optimizes, and executes the HiveQL. The Hive Driver may choose to execute HiveQL statements and commands locally or spawn a MapReduce job, depending on the task at hand. The Hive Driver stores table metadata in the metastore and its database.

You probably have some familiarity with SQL and the relational database model from the world of RDBMSs. A *table* or *relation* is composed of vertical columns and horizontal rows. *Cells* are stored where the rows and columns intersect. If you’re not familiar with SQL and the relational database model, you can find helpful learning sources using your favorite search engine.

By default, Hive includes the Apache Derby RDBMS configured with the metastore in what’s called embedded mode. *Embedded mode* means that the Hive Driver, the metastore, and Apache Derby are all running in one Java Virtual Machine (JVM).

This configuration is fine for learning purposes, but embedded mode can support only a single Hive session, so it normally isn’t used in multi-user production environments. Two other modes exist — *local* and *remote* — which can better support multiple Hive sessions in production environments. Also, you can configure any RDBMS that’s compliant with the Java Database Connectivity (JDBC) Application Programming Interface (API) suite. (Examples here include MySQL and DB2.)

The key to application support is the Hive Thrift Server, which enables a rich set of clients to access the Hive subsystem. The open source [**SQuirreL SQL client**](http://squirrel-sql.sourceforge.net/) is included as an example. The main point is that any JDBC-compliant application can access Hive via the bundled JDBC driver.

The same statement applies to clients compliant with Open Database Connectivity (ODBC) — for example, unixODBC and the isql utility, which are typically bundled with Linux, enable access to Hive from remote Linux clients.

Additionally, if you use Microsoft Excel, you’ll be pleased to know that you can access Hive after you install the Microsoft ODBC driver on your client system. Finally, if you need to access Hive from programming languages other than Java (PHP or Python, for example), Apache Thrift is the answer. Apache Thrift clients connect to Hive via the Hive Thrift Server, just as the JDBC and ODBC clients do.

To continue with the Hive architecture drawing, note that Hive includes a Command Line Interface (CLI), where you can use a Linux terminal window to issue queries and administrative commands directly to the Hive Driver. If a graphical approach is more your speed, there’s also a handy web interface so that you can access your Hive-managed tables and data via your favorite browser.

There is another web browser technology known as [**Hue**](http://gethue.com/) that provides a graphical user interface (GUI) to Apache Hive. Some Hadoop users like to have a GUI at their disposal instead of just a command line interface (CLI). Along with Hive, Hue supports other key Hadoop technologies as well like HDFS, MapReduce/YARN, HBase, Zookeeper, Oozie, Pig, and Sqoop. You’ll like the name for Hue’s Apache Hive GUI — it’s called Beeswax.