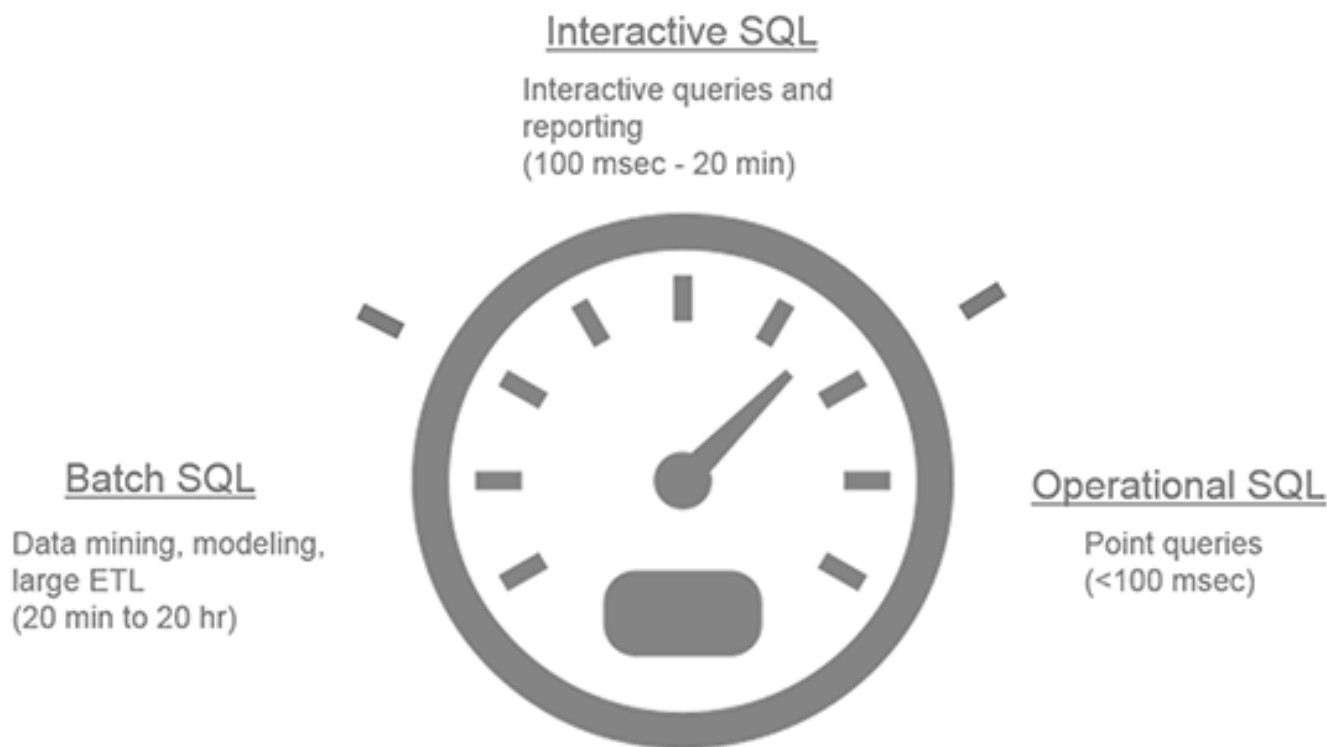


SQL on Hadoop:场景和结论

以下文章是转载自国外网站，介绍了Hadoop生态系统上面的几种SQL：Hive、Drill、Impala、Presto以及SparkWShark等应用场景、对比以及一些结论

Within the big data landscape there are multiple approaches to accessing, analyzing, and manipulating data in Hadoop. Each depends on key considerations such as latency, ANSI SQL completeness (and the ability to tolerate machine-generated SQL), developer and analyst skillsets, and architecture tradeoffs.

Below is a discussion segmented by broad latency characteristics of each approach.



Batch SQL

Technologies such as Hive are designed for batch queries on Hadoop by providing a declarative abstraction layer (HiveQL), which uses the MapReduce processing framework in the background. Hive is used primarily for queries on very large data sets and large ETL jobs. The queries can take anywhere between a few minutes to several hours depending on the complexity of the job. The Apache Tez project aims to provide targeted performance improvements for Hive to deliver interactive query capabilities in future. MapR ships and supports Apache Hive today.

Interactive SQL

Technologies such as Impala and Apache Drill provide interactive query capabilities to enable traditional business intelligence and analytics on Hadoop-scale datasets. The response times vary between milliseconds to minutes depending on the query complexity. Users expect SQL-on-Hadoop technologies to support common BI tools such as Tableau and MicroStrategy (to name a couple) for reporting and ad-hoc queries. MapR supports customers using Impala on the MapR distribution of Hadoop today. Apache Drill will be available Q2 2014.

In-Memory SQL and Streaming

In-memory computing has enabled new ecosystem projects such as Apache Storm and Apache Spark to further accelerate stream and query processing, respectively. Shark is a new project which also uses in-memory computing while retaining full Hive compatibility to provide 100x faster queries than Hive. MapR customers are using Storm and Shark on Spark with the MapR Distribution of Hadoop today.

Operational SQL

Unlike batch and interactive queries that are used by business teams for decision making and operate as read-only operations on large datasets (OLAP), point queries are typically done by OLTP and web applications, operating over smaller datasets and typically include insert, update, and deletes. The expected latency is usually very low (e.g., milliseconds) due to the high volume of requests from these applications. MapR ships and supports operational SQL capabilities with both Apache Hbase and MapR M7 Enterprise Database Edition.

Interactive SQL-on-Hadoop Technology Landscape

SQL technologies complement traditional data warehouse and analytical environments for:

- Interactive and ad-hoc queries on large-scale data
- Data exploration to discover new insights worth modeling into a data warehouse schema
- Interactive queries on more or new types of data
- Queries for online data archives in Hadoop vs. backing up to tape

Technologies and approaches for interactive SQL vary and include (but are not limited to)

1. Querying the data using connectors from Hadoop to analytic platforms (upfront or at query run time with external tables)
2. Running traditional SQL engines side by side on every node of the Hadoop cluster
3. Providing purpose-built SQL engines directly on top of Hadoop (native SQL options)
4. Efforts to improve MapReduce performance to make it suitable for interactive queries

Options (1) and (2) excel at SQL support, performance optimizations, and overall enterprise readiness. However, native SQL-on-Hadoop options (3) are evolving as cost-effective alternatives because they have stronger compatibility with the Hadoop ecosystem (e.g., use Hadoop native file formats and common metadata store through Hive)

Key Considerations for SQL-on-Hadoop Approaches

For organizations with existing skills in SQL and investments in business intelligence (BI) tools, ANSI SQL completeness is key for easy adoption and reuse.

SQL-on-Hadoop Technology	SQL Completeness (L, M, H suggests range of support)			
	SELECT query	DDL/DM L	Package d om	UDFs/Cust om Analytic functions function s
Hive1	M2	M	L	H
Drill1	M	L	-	L3
Impala	M	L	-	L
Presto	M	-	L	-
Spark/Shark	M	M	-	H

1 - Hive 0.13, Drill 1.0 are in development. Drill 1.0 is expected to be available Q2 2014.

2 - Limited subquery options in Hive and HiveQL-based tools (Impala, Shark)

3 - Impala added support for UDFs/UDAFs in 1.2.1 release. UDTFs are not supported but in roadmap

The type of protocols and interfaces for the client to access the SQL engine in each SQL-on-Hadoop approach is worth considering, depending on your preferences and use case.

SQL-on-Hadoop Technology	Client access		
	Shell	JDBC	ODBC
Hive	Yes	Yes	Yes
Drill	Yes	Yes	Yes
Impala	Yes	Yes	Yes
Presto	Yes	Yes1	-
Spark/Shark	Yes	Yes	Yes

1 - Alpha release

Various SQL-on-Hadoop technologies have different methods for metadata handling. This provide a mechanism to capture and maintain information about where data is stored, how it is structured, and more.

	Metadata		
SQL-on-Hadoop Technology	Hive	Proprietary	Optional
Hive	Yes	-	-
Drill	Yes	-	Yes
Impala	Yes	-	-
Presto	Yes	-	-
Spark/Shark	Yes	-	-

Harnessing the value of big data requires the ability to ingest and analyze multiple data types. Also, as companies mature in their use of Hadoop, they use tables (Apache HBase) and files (HDFS) for storing different data types. Consider how each SQL-on-Hadoop approach supports queries on different Hadoop ecosystem sources and data types.

	Data Sources		Data Types		
SQL-on-Hadoop Technology	Files	HBase	Can query on-Hadoop sources?	Relational	Complex
Hive	Yes	Yes	-	Yes	Yes
Drill	Yes	Yes ¹	Yes	Yes	Yes
Impala	Yes	Yes ²	-	Yes	-
Presto	Yes	-	Yes	Yes	Yes ³
Spark/Shark	Yes	Yes	-	Yes	Yes

1 - On MapR platform, Drill supports querying on both HBase and M7

2 - There are known performance, functionality issues with Impala on HBase

3 - Supports JSON functions only. Other complex types from Hive such as Structs, Arrays are returned as JSON and can be manipulated

In addition, each SQL-on-Hadoop technology has differing approaches to file formats supported. This affects interoperability and the ease of switching between approaches as these technologies mature – another key consideration for when to use which.

	Common file formats support									
SQL-on-Hadoop Technology	Text	CSV	Sequence	RC	ORC	Parquet	Avro	JSON	Compression	Hive SerDe
Hive	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Drill	-	Yes	-	-	-	Yes	-	Yes	-	Yes
Impala	Yes	-	Yes ¹	Yes ¹	-	Yes	Yes ¹	-	Yes	-
Presto	Yes	-	Yes	Yes	-	-	-	-	-	-
Spark/Shark	Yes	Yes	Yes	Yes	-	-	Yes	Yes	Yes	Yes

1 - Impala can create tables and query from Text and Parquet, but only query the data from Sequence, Avro, RC. For these formats, data must be loaded through Hive. Parquet is the optimized format for Impala.



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：[iteblog_hadoop](#)

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#) ()