5 Best Practices for Tableau & Hadoop
Tableau was designed to facilitate real-time conversations with data across multiple data platforms. Business users who have felt stymied by traditional tools have flocked to this modus operandi. So what happens when queries return in hours or minutes rather than seconds? Can their ‘flow’ be maintained?

We are in an age where people can analyze millions or even billions of rows of data at their fingertips yet a user’s expectations is that they have near instantaneous results (see study on the 2 second rule for information retrieval). When a user’s interactions and response times take more than 2-3 seconds, they become distracted from being “in the flow of visual analysis.” Thus, it is imperative to provide fast query speeds to keep users engaged so that they can gain more insight from their Big Data deployments.

Users can apply a number of best practices to maximize the performance of their Tableau visualizations and dashboards built on Big Data platforms. The best practices largely fall into the following five activities:

1. Leverage A Fast Interactive Query Engine
2. Strategically Utilize Live Connections Vs. Extracts
3. Curate Your Data From The Data Lake
4. Optimize Your Extracts
5. Customize Your Connection Performance
Leverage A Fast Interactive Query Engine

Hive queries executed on Hadoop using MapReduce are inherently slow due to the overhead associated with mapping the SQL queries into a MapReduce job. Hive on MapReduce is great for performing batch processing such as in ETL applications because it is highly fault tolerant, but it falls short in performance. Advancements in Hive introduce new application frameworks such as Tez (enabling interactive queries) and Spark (enabling in-memory processing) significantly improving query speeds.

Outside of Hive on Hadoop, there are a number of great options for accelerating your queries. Impala is widely known to have the fastest performance on Hadoop according to recent benchmarks. Though it is early in development, Spark SQL has shown great potential as a fast data processing engine. It can process data stored on Hadoop or Spark Schema RDDs that is referenced by a Hive Metastore. Both Impala and Spark SQL are supported as named connectors in Tableau. Pivotal HAWQ, Presto and Apache Drill are also technologies that come up commonly in discussions on performance on Hadoop.

Another option is looking outside of Hadoop altogether. Fast analytical databases such as Actian Vector, HP Vertica, Teradata Aster Data, SAP Hana, ParAccel, Pivotal Greenplum and others can serve as an excellent place to host your data for low latency queries for Tableau business users after it has been processed in Hadoop. Cloud-hosted infrastructure services are also becoming increasingly popular. Google BigQuery leverages Google’s massive infrastructure that excels at both processing data and enabling fast queries especially on large datasets. On the other hand, Amazon Redshift is a fully-managed columnar-storage datawarehouse that focuses on fast data access. Lastly, there are a group of emerging technologies from startups and open source projects using OLAP Cubes (AtScale, eBay Kylin) or Indexing engines (JehtroData) for Hadoop that provide the ability to query one billion rows or more with low latency.
Strategically Utilize Live Connections Vs. Extracts

Tableau’s hybrid data architecture for connecting either live to a data source or via an in-memory extract to Tableau’s Data Engine gives users ultimate flexibility when working with Big Data. Extracts are ideal for situations where fast query engines are not available, datasets are small to medium in size (100s of millions of rows or less), or offline analysis is needed. For larger dataset sizes, Hadoop Hive and other query engines will scale better than Tableau due to its distributed execution. Also, when a fast database engine is available or real-time analysis is required, a live connection is the better choice. The full list of recommended scenarios is shown below in Figure 1.

<table>
<thead>
<tr>
<th>Extracts Recommended</th>
<th>Live Connection Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow database query execution</td>
<td>When a fast database query engine is available</td>
</tr>
<tr>
<td>When smaller dataset sizes are used (e.g. 100s of millions of rows or less)</td>
<td>When larger dataset sizes are required</td>
</tr>
<tr>
<td>When offline analysis is required</td>
<td>When real-time analysis is required</td>
</tr>
<tr>
<td>When using Custom SQL</td>
<td>When a workbook uses pass-through RAWSQL functions</td>
</tr>
<tr>
<td>When additional analytic functionalities are needed (set, rank, distinct, median)</td>
<td>When robust user-level security is required (except for extracts published to Tableau Server)</td>
</tr>
</tbody>
</table>

Figure 1: Recommended scenarios for extracts vs live connections.
Curate Your Data From The Data Lake

One of the many benefits of Hadoop is that its scale, cost efficiency and ability to handle unstructured data make it the ideal Data Lake - a repository to hold all of your data in its native format. Tableau is an effective tool for exploring your data in the Data Lake, but when you want the best performance for your visualization against Hadoop for your knowledge workers to leverage, it is always best to curate your dataset first.

As an IT administrator, there are several techniques that you can apply to improve the efficiency of your Hadoop cluster:

**Partition Design** - Organizing a Hive table into separate files (each with many data blocks) in a distributed system with one or more partitioning fields can greatly accelerate queries over a query that’s filtered by a non-partitioned field.

**Dataset size** - When you know the dimensions and measures you want to look at for a set of analysis and the range of records, limiting the ultimate dataset that’s exposed to your knowledge workers will always improve performance.

**Clustered Fields as Grouping Fields and Join Keys** - Fields that are clustered can dictate how the data in the table is separated on disk. JOINs and GROUP BYs of clustered fields will see improved performance.

**Storage File Format** - File format plays a key role in the efficient execution of queries. Utilize the file format that best matches the query engine you are using. For Hive this is Optimized Row Columnar (ORC) format, and for Impala this is Parquet.

**Data Model Design** -

- Data types - Use numeric types whenever possible as they are much faster than strings.

- Joins - Avoid unnecessary joins as they are imperfectly implemented on many Big Data systems. If you do use joins, first run a COMPUTE STATs statement to help the processing engine automatically optimize the performance for join queries.

- Formulas - Avoid the use of formulas that can’t be efficiently evaluated.
4. Optimize Your Extracts

Tableau’s Data Engine is an in-memory analytics database that leverages the complete memory hierarchy from disk to L1 cache. It can be a powerful tool for accelerating your analysis. Though it is not built for the same scale as Hadoop, the Tableau Data Engine can deliver low latency results against extracts of data with a cardinality of hundreds of millions of rows and a wide number of columns. Although leveraging extracts in Tableau’s Data Engine will usually boost performance out-of-the-box, there are a number of opportunities to accelerate your queries by condensing the size of your data:

**Define Filters** - Create a filter so that you are only focused on the data of interest.

**Hide Unused Fields** - Hide fields that are not required for analysis so that the extract is compact and concise.

**Aggregate Visible Dimensions** - Pre-aggregate data to a more coarse-grained view when the fine-grained data is not needed to yield the same insights with faster queries.

**Roll Up Dates** - Roll up dates to coarser-grained timelines when possible.

**Sampling** - For databases that support it, data sampling can greatly compact the data while still representing the broad trends in the data.

**Top N** - If you are just looking for the highest values in a dataset, this is an efficient means to reduce the dataset size.
5.

Customize Your Connection Performance

As a Tableau user, there are a number of opportunities to optimize your connection performance for live queries:

**Custom SQL** - Custom SQL allows for SQL expressions to be used as the basis for a connection in Tableau. Custom SQL can be especially effective for limiting the data set size (by using the LIMIT clause) so that you can explore or profile a new data set.

**Initial SQL** - Initial SQL provides the ability to set configuration parameters and perform work right when you establish a connection. You can do things such as:

- Increase parallelism for data analysis by reducing the default block size for Maps and Reduces.
- Optimize join performance by turning on clustered fields.
- Adjusting configurations for uneven distribution by turning on settings that informs Hive to take a different approach for MapReduce jobs.

**Summary**

The Big Data era has arrived - data volumes are accelerating and organizations are shifting their data infrastructure to Hadoop, Spark & NoSQL to support the new normal in data. Tableau's ability to empower everyday business users is bringing Big Data visual insights to the masses. Though applying best practices and tailoring them to custom fit your application, you will be able to maximize the value of your Big Data investments.
About Tableau

Tableau Software helps people see and understand data. Offering a revolutionary new approach to business intelligence, Tableau allows you to quickly connect, visualize, and share data with a seamless experience from the PC to the iPad. Create and publish dashboards and share them with colleagues, partners, or customers—no programming skills required. See how Tableau can help your organization by starting your free trial at tableau.com/trial.

Additional Resources

Download Free Trial
Product Demos
Training & Tutorial
Community & Support
Customer Stories
Solutions

Related Whitepapers

Tableau’s Vision on Big Data
7 Tips to Succeed with Big Data
Fostering a Data-Driven Culture: A Special Report from the Economist Intelligence Unit and Tableau
Big Data: The Next Industrial Revolution
Tableau Software and Big Data
Aberdeen Group: Maximizing the Value of Analytics and Big Data
See All Whitepapers