Shay Walker

Box:613 221 - Comparison Project

Professor Sabal

Hadoop is Great for MapReduce data analysis on large amounts of data. More specific cases include data searching, data analysis, data reporting, large-scale indexing of files (log files), and other data processing tasks or “Big Data”. You want to use Hadoop if your data takes up at least terabytes or petabytes of data, if your data is not very large and only takes up gigabytes then there are other tools that would cost much less to implement and maintain than Hadoop. Careful planning is needed because your data may be smaller now but could grow to the size where Hadoop would be useful and better to have. Hadoop also stores and processes sand file data, you at any point can change how you process and analyze your Hadoop data. This provides flexibility with what your data can do and be used for. The MapReduce algorithm requires that you can parallelize your data processing and works very well in situation where variables are processed one at a time. However, if you want to process them jointly the model no longer works. Therefore, graph-based data processing is not a great fit for Hadoop.

There are also many instances where Hadoop is not the right fit. Since Hadoop works by the batch and not everything at once, these jobs take more time to process than a relational database query on some tables. For very large data sets it can take hours to days to complete a process. Thus, Hadoop should not be used for Relational Databases, general network file systems or for non-parallel data processing.

MongoDB is a document -oriented database management system and sores data in collections. These collections, in which data fields data can be queried once, versus multiple queries required by RDBMS. When compared to Hadoop, is greatest strength is that it is a more robust solution and has greater flexibility than Hadoop. MongoDB is also better at handling real-time data analytics. The readily available data also makes it capable of client-side data delivery (in a typical client-server web-based scenario), which is not as common with Hadoop configurations. Another strength of MongoDB is its geospatial indexing abilities, making an ideal use case for real-time geospatial analysis.

A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables. When you have a smaller amount of data it is better to use a relational database. Relational database are also used to store and sort information in tables and relate tables to one another. Also, you want a relational database if you need to normalize your schema. Relational databases use Structured Querying Language (SQL), making them a good choice for applications that involve the management of several transactions. The structure of a relational database allows you to link information from different tables through the use of foreign keys (or indexes), which are used to uniquely identify any atomic piece of data within that table. Other tables may refer to that foreign key, so as to create a link between their data pieces and the piece pointed to by the foreign key. This comes in handy for applications that are heavy into data analysis. If you want your application to handle a lot of complicated querying, database transactions and routine analysis of data, you’ll probably want to stick with a relational database (ACID).

SOURCES:

<https://www.pluralsight.com/blog/software-development/relational-non-relational-databases>

<http://www.aptude.com/blog/entry/hadoop-vs-mongodb-which-platform-is-better-for-handling-big-data>

<http://www.hostingadvice.com/how-to/what-is-hadoop/>