Creado por:

Isabel Maniega

Spark_Vs_Hadoop

Hadoop:

Framework basado en JAVA. Contiene almacenamiento de datos en Hadoop Distributed File System (HDFS) y el procesamiento MapReduce.

Diferencias

- **Propósito:** Pincipalmente Hadoop almacena datos en disco, mientras que Spark puedo hacerlo en memoria.
- **Implementación:** Hadoop administra gran cantidad de archivos mientras que Spark NO, puede integrar Hadoop HDFS, MongoDB, etc
- **Velocidad de Procesamiento:** Hadoop trabaja en disco, mientras que Spark lo hae en memoria lo que le permiter una mayor velocidad de procesamiento.
- **Recuperación de de seguridad:** Los dos permiten recuperación segura.

Comparativa Hadoop y MongoDB:

MongoDB como almacén de datos operativos en tiempo real y Hadoop para el procesamiento y análisis de datos. Algunos diferencias son:

- **Agregación de lotes:** cuando se requiere una agregación de datos compleja MongoDB se queda corto con su funcionalidad de agregación, que no es suficiente para llevar a cabo el análisis de datos. En cambio Hadoop proporciona un potente marco de trabajo que resuelve la situación gracias a su alcance. Para llevar a cabo esta asociación, es necesario extraer los datos de MongoDB (u otras fuentes de datos, si se quiere desarrollar una solución multi-datasource) para procesarlos dentro de Hadoop a través de MapReduce. El resultado puede enviarse de nuevo a MongoDB, asegurando su disponibilidad para posteriores consultas y análisis.
- **Data Warehouse:** en producción, los datos procedentes de una aplicación pueden vivir en múltiples almacenes de datos. Para reducir la complejidad en estos escenarios, Hadoop puede ser utilizado como un almacén de datos y actuar como un depósito centralizado para los datos de las diversas fuentes. En esta situación, podrían llevarse a cabo trabajos MapReduce periódicos para la carga de datos de MongoDB en Hadoop. Una vez que los datos de MongoDB, así como los de otras fuentes, están disponible desde dentro de Hadoop, los analistas de datos tienen la opción de utilizar MapReduce o Pig para lanzar consultas a las bases de datos más grandes que incorporan datos de MongoDB.

- **Procesos ETL:** si bien MongoDB puede ser el almacén de datos operativos para una aplicación, puede suceder que tenga que coexistir con otros almacenes. En este escenario, es útil alcanzar la capacidad de mover datos de un almacén de datos a otro, ya sea desde la propia aplicación a otra base de datos o viceversa. La complejidad de un proceso ETL excede la de la simple copia o transferencia de datos, por lo que se puede utilizar Hadoop como un mecanismo complejo ETL para migrar los datos en diversas formas a través de uno o más trabajos MapReduce para extraer, transformar y cargar datos en destino. Este enfoque se puede utilizar para mover los datos desde o hacia MongoDB, dependiendo del resultado deseado.

Creado por:

Isabel Maniega