Apache Spark a Unified Engine for Big Data Processing

Al principio eran modelos especializados como lo era MapReduce que soportaba procesamiento por lotes, luego Google desarrolló *Dremel* y *Pregel*, con el propósito de general consultas iterativas SQL y gráficos iterativos, respectivamente. Para el apartado de código abierto, *Apache Hadoop* contiene los sistemas como *Storm* e *Impala* los cuales también están especializados. Sin embargo, la mayoría de las aplicaciones de la actualidad de *"Big Data"* (diversa y desordenada) necesitan combinar muchos tipos de procesamiento distintos.

Apache Spark inició en el 2009 para diseñar un motor unificado para el procesamiento de datos distribuido. Es un sistema de procesamiento de código abierto que está diseñado para realizar análisis de datos a gran escala, utilizando arquitecturas en memoria que le permite procesar los grandes volúmenes de datos con mayor rapidez. Modelo parecido al de MapReduce, pero con abstracción de intercambio de datos con RDD (*Resilient Distributed Dataset*), se pueden realizar procesamiento como SQL, *streaming*, *machine learning* y procesamiento gráfico.

Como beneficios de Spark podemos mencionar: las aplicaciones fáciles de desarrollar por la utilización de API unificada, mas eficiente por la combinación de tareas de procesamiento y habilita nuevas aplicaciones

Modelos de programación

Modela una programación abstracta (RDD) estos se exponen a través de una API de programación funcional en Scala, Java, Python y R, donde los usuarios pueden pasar funciones locales para ejecutarse en el clúster. Con el fin de obtener soporte explícito para el intercambio de datos entre cálculos La tolerancia a fallos es como Spark se recupera automáticamente de los fallos con el enfoque llamado "*lineage*" Cada RDD rastrea el gráfico de transformaciones que se utilizó para construirlo y vuelve a ejecutar estas operaciones en datos base para reconstruir cualquier partición perdida.

Bibliotecas de nivel superior

El modelo RDD mantiene una variedad de bibliotecas de alto nivel los cuales a algunas logran un rendimiento de vanguardia en cada tarea a tiempo que ofrecen beneficios significativos. Entre las bibliotecas se tienen: *SQL and Dataframes, Spark streaming, GraphX, MLlib, Combining processing tasks* y *Performance*

Aplicaciones

Apache Spark utiliza una amplia gama de aplicaciones en las áreas de biotecnología y financiamiento. Así como en el mundo científico. Los usuarios a menudo utilizan combinaciones múltiples de las bibliotecas para crear aplicaciones más eficientes y de aspectos general. Algunas de las aplicaciones que más se visualizan en el mercado son: *Batch Processing, Interactive queries, Streaming Processing, Scientific Applications, Spark Components Used, Deployment environments*

Generalidad del modelo Spark

Si bien Apache Spark demuestra que un modelo de programación de clúster unificado es factible y útil, sería útil entender qué hace que los modelos de programación de clúster sean generales, junto con las limitaciones de Spark.

Los RDD pueden emular cualquier cálculo distribuido, y lo harán de manera eficiente en muchos casos a menos que el cálculo sea sensible a la latencia de la red. En el punto de vista de los sistemas, demostramos que los RDD dan a las aplicaciones control sobre los recursos de cuello de botella más comunes en clústeres-red y almacenamiento E/S-y así permiten expresar las mismas optimizaciones para estos recursos que caracterizan a

los sistemas especializados.

Para estudiar la expresividad de los DDRs, comenzamos comparando los DDRs con el modelo MapReduce, que los DRDs construyen. Para la perspectiva de sistemas es independientemente del enfoque de emulación para caracterizar la generalidad de Spark, podemos adoptar un enfoque de sistemas.