

Capítulo 4

Hadoop

Al finalizar el capítulo, el alumno podrá:

- Comprenderá y diseñara una arquitectura Hadoop.

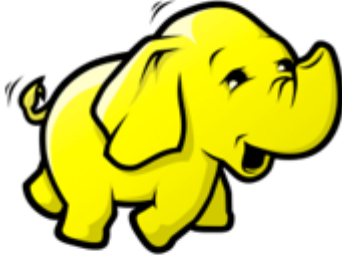
Temas


1. Introducción al Hadoop
2. Ecosistema de Hadoop
3. Big Data y el Cloud

1. Introducción al Hadoop

Introducción a Hadoop

- Hadoop = Big Data
- Entorno Distribuido (Datos y Procesos)
- Es escalable de forma horizontal con commodity hardware.



4 - 4Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.

Resumen de la historia de Hadoop

Creado en 2005 por Mike Cafarella y Doug Cutting (que le puso el nombre del elefante de juguete de su hijo), Hadoop estaba destinado originalmente a datos de búsqueda en Internet. Hoy en día, es un proyecto de código abierto comunitario de Apache Software Foundation que se usa en todo tipo de organizaciones e industrias.

¿Qué es Apache Hadoop?

El proyecto Apache TM Hadoop® desarrolla software de código abierto para una computación distribuida confiable y escalable.

La biblioteca de software Apache Hadoop es un marco que permite el procesamiento distribuido de grandes conjuntos de datos en clústeres de computadoras utilizando modelos de programación simples. Está diseñado para escalar desde servidores únicos a miles de máquinas, cada una de las cuales ofrece cómputo y almacenamiento local. En lugar de confiar en el hardware para ofrecer alta disponibilidad, la biblioteca está diseñada para detectar y manejar fallas en la capa de aplicaciones, por lo que ofrece un servicio altamente disponible sobre un grupo de computadoras, cada una de las cuales puede ser propensa a fallas.

El proyecto incluye estos módulos:

- Hadoop Common: las utilidades comunes que son compatibles con los otros módulos de Hadoop.
- Hadoop Distributed File System (HDFS™): sistema de archivos distribuido que proporciona acceso de alto rendimiento a los datos de las aplicaciones.
- HADOOP YARN: un marco para la programación de trabajos y la administración de recursos de clúster.
- Hadoop MapReduce: un sistema basado en YARN para el procesamiento paralelo de grandes conjuntos de datos.

Otros proyectos relacionados con Hadoop en Apache incluyen:

- Ambari™: una herramienta basada en web para aprovisionar, administrar y monitorear clústeres de Apache Hadoop que incluye soporte para Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig y Sqoop. Ambari también proporciona un panel de control para ver la salud del clúster, como mapas de calor y la capacidad de ver las aplicaciones de MapReduce, Pig y Hive visualmente junto con las características para diagnosticar sus características de rendimiento de una manera fácil de usar.
- Avro™: un sistema de serialización de datos.
- Cassandra™: una base de datos escalable de múltiples maestros sin puntos únicos de falla.
- Chukwa™: un sistema de recopilación de datos para administrar grandes sistemas distribuidos.
- HBase™: una base de datos distribuida y escalable que admite el almacenamiento de datos estructurados para tablas grandes.
- Hive™: una infraestructura de depósito de datos que proporciona un resumen de datos y consultas ad hoc.
- Mahout™: una biblioteca escalable de aprendizaje automático y minería de datos.
- Pig™: un lenguaje de flujo de datos de alto nivel y un marco de ejecución para el cómputo paralelo.
- Spark™: un motor de cálculo rápido y general para datos de Hadoop. Spark proporciona un modelo de programación simple y expresivo que admite una amplia gama de aplicaciones, incluyendo ETL, aprendizaje automático, procesamiento de flujo y cálculo de gráficos.
- Tez™: Un marco de programación de flujo de datos generalizado, construido en Hadoop YARN, que proporciona un motor poderoso y flexible para ejecutar un DAG arbitrario de tareas para procesar datos para casos de uso tanto por lotes como interactivos. Tez está siendo adoptado por Hive™, Pig™ y otros marcos en el ecosistema Hadoop, y también por otro software comercial (por ejemplo, herramientas ETL), para reemplazar a Hadoop™ MapReduce como motor de ejecución subyacente.
- ZooKeeper™: un servicio de coordinación de alto rendimiento para aplicaciones distribuidas.

HDFS Architecture

El Sistema de archivos distribuidos de Hadoop (HDFS) es un sistema de archivos distribuidos diseñado para ejecutarse en hardware básico. Tiene muchas similitudes con los sistemas de archivos distribuidos existentes. Sin embargo, las diferencias con respecto a otros sistemas de archivos distribuidos son significativas. HDFS es altamente tolerante a fallas y está diseñado para implementarse en hardware de bajo costo. HDFS proporciona acceso de alto rendimiento a los datos de la aplicación y es

adecuado para aplicaciones que tienen grandes conjuntos de datos. HDFS relaja algunos requisitos POSIX para permitir el acceso continuo a los datos del sistema de archivos. HDFS fue originalmente construido como infraestructura para el proyecto de motor de búsqueda web Apache Nutch. HDFS es parte del proyecto Apache Hadoop Core. La URL del proyecto es <http://hadoop.apache.org/>

Supuestos y metas

- **Fallo de hardware:** La falla de hardware es la norma más que la excepción. Una instancia de HDFS puede consistir en cientos o miles de máquinas servidor, cada una de las cuales almacena parte de los datos del sistema de archivos. El hecho de que haya una gran cantidad de componentes y de que cada componente tenga una probabilidad de falla no trivial significa que algún componente de HDFS siempre es no funcional. Por lo tanto, la detección de fallas y la recuperación rápida y automática de ellos es un objetivo arquitectónico central de HDFS.
- **Acceso a datos en tiempo real:** Las aplicaciones que se ejecutan en HDFS necesitan acceso de transmisión a sus conjuntos de datos. No son aplicaciones de propósito general que normalmente se ejecutan en sistemas de archivos de propósito general. HDFS está diseñado más para procesamiento por lotes en lugar de uso interactivo por los usuarios. El énfasis está en el alto rendimiento del acceso a los datos en lugar de la baja latencia del acceso a los datos. POSIX impone muchos requisitos difíciles que no son necesarios para las aplicaciones que están destinadas a HDFS. La semántica POSIX en algunas áreas clave se ha comercializado para aumentar las tasas de rendimiento de datos.
- **Grandes conjuntos de datos:** Las aplicaciones que se ejecutan en HDFS tienen grandes conjuntos de datos. Un archivo típico en HDFS es de un tamaño de gigabytes a terabytes. Por lo tanto, HDFS está sintonizado para admitir archivos de gran tamaño. Debería proporcionar un gran ancho de banda de datos agregados y escalar a cientos de nodos en un único clúster. Debe admitir decenas de millones de archivos en una sola instancia.
- **Modelo de Coherencia Simple:** Las aplicaciones HDFS necesitan un modelo de acceso de escritura para lectura de varios para los archivos. Un archivo una vez creado, escrito y cerrado no necesita ser cambiado, excepto para agregar y truncar. Se admite el agregado del contenido al final de los archivos, pero no se puede actualizar en un punto arbitrario. Esta suposición simplifica los problemas de coherencia de datos y permite el acceso a datos de alto rendimiento. Una aplicación MapReduce o una aplicación de rastreo web se ajusta perfectamente con este modelo.
- **"Mover el cálculo es más barato que mover datos":** Un cómputo solicitado por una aplicación es mucho más eficiente si se ejecuta cerca de los datos en los que opera. Esto es especialmente cierto cuando el tamaño del conjunto de datos es enorme. Esto minimiza la congestión de la red y aumenta el rendimiento general del sistema. La suposición es que a menudo es mejor migrar el cálculo más cerca de donde se encuentran los datos en lugar de mover los datos a donde se ejecuta la aplicación. HDFS proporciona interfaces para que las aplicaciones se muevan más cerca del lugar donde se encuentran los datos.

- Portabilidad a través de plataformas de hardware y software heterogéneas: HDFS ha sido diseñado para ser fácilmente portátil de una plataforma a otra. Esto facilita la adopción generalizada de HDFS como plataforma de elección para un gran conjunto de aplicaciones.

Hadoop MapReduce

Hadoop MapReduce es un marco de software para escribir fácilmente aplicaciones que procesan grandes cantidades de datos (conjuntos de datos de varios terabytes) en paralelo en clústeres grandes (miles de nodos) de hardware básico de una manera confiable y tolerante a fallas.

Un trabajo de MapReduce generalmente divide el conjunto de datos de entrada en trozos independientes que son procesados por las tareas del mapa de una manera completamente paralela. El marco ordena los resultados de los mapas, que luego se ingresan a las tareas de reducción. Normalmente, tanto la entrada como la salida del trabajo se almacenan en un sistema de archivos. El marco se encarga de programar las tareas, supervisarlas y volver a ejecutar las tareas fallidas.

Normalmente, los nodos de cálculo y los nodos de almacenamiento son los mismos, es decir, el marco MapReduce y el sistema de archivos distribuidos de Hadoop (consulte la Guía de arquitectura HDFS) se están ejecutando en el mismo conjunto de nodos. Esta configuración permite que el marco planifique efectivamente las tareas en los nodos donde los datos ya están presentes, lo que resulta en un ancho de banda agregado muy alto en todo el clúster.

El marco MapReduce consta de un único ResourceManager maestro, un NodeManager de trabajador por cluster-node y MRAppMaster por aplicación (ver la Guía de Arquitectura de YARN).

Como mínimo, las aplicaciones especifican las ubicaciones de entrada / salida y el mapa de suministro y reducen las funciones a través de implementaciones de interfaces apropiadas y / o clases abstractas. Estos y otros parámetros del trabajo comprenden la configuración del trabajo.

El cliente de trabajo de Hadoop luego envía el trabajo (jar / executable, etc.) y la configuración al ResourceManager, que luego asume la responsabilidad de distribuir el software / configuración a los trabajadores, programar las tareas y monitorearlos, proporcionando información de estado y diagnóstico al trabajo cliente.

Aunque el marco de Hadoop se implementa en Java [™], las aplicaciones de MapReduce no necesitan estar escritas en Java.

Hadoop Streaming es una utilidad que permite a los usuarios crear y ejecutar trabajos con cualquier ejecutable (por ejemplo, utilidades de shell) como el asignador y / o el reductor.

Hadoop Pipes es una API C ++ compatible con SWIG para implementar aplicaciones MapReduce (no basadas en JNI [™]).

2. Ecosistema de Hadoop

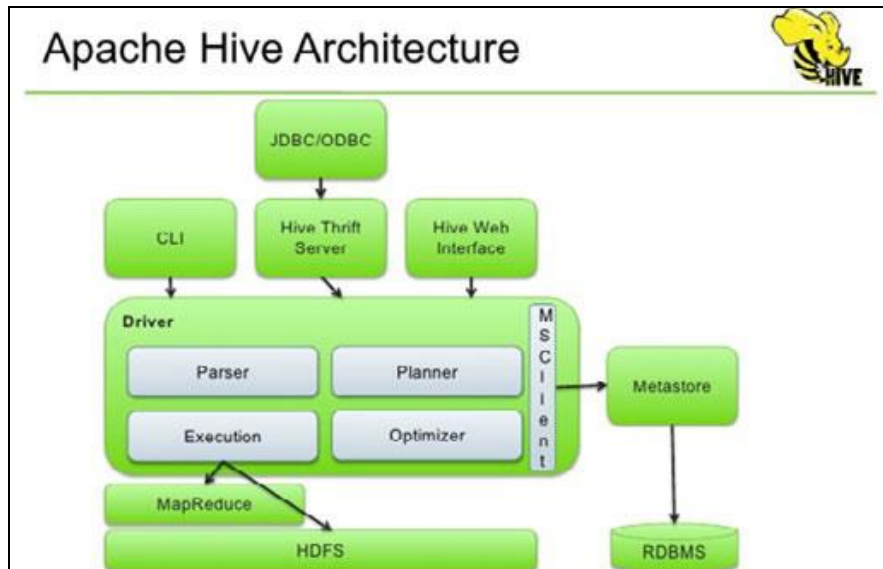


Entre las principales herramientas del ecosistema Hadoop tenemos:

Apache Hive

El software de almacenamiento de datos Apache Hive TM facilita la lectura, escritura y administración de grandes conjuntos de datos que residen en el almacenamiento distribuido y se consultan mediante la sintaxis SQL.

- Herramientas para permitir el acceso fácil a los datos a través de SQL, lo que permite tareas de almacenamiento de datos como extraer / transformar / cargar (ETL), informes y análisis de datos.
- Un mecanismo para imponer estructura en una variedad de formatos de datos
- Acceso a archivos almacenados directamente en Apache HDFS TM o en otros sistemas de almacenamiento de datos como Apache HBase TM

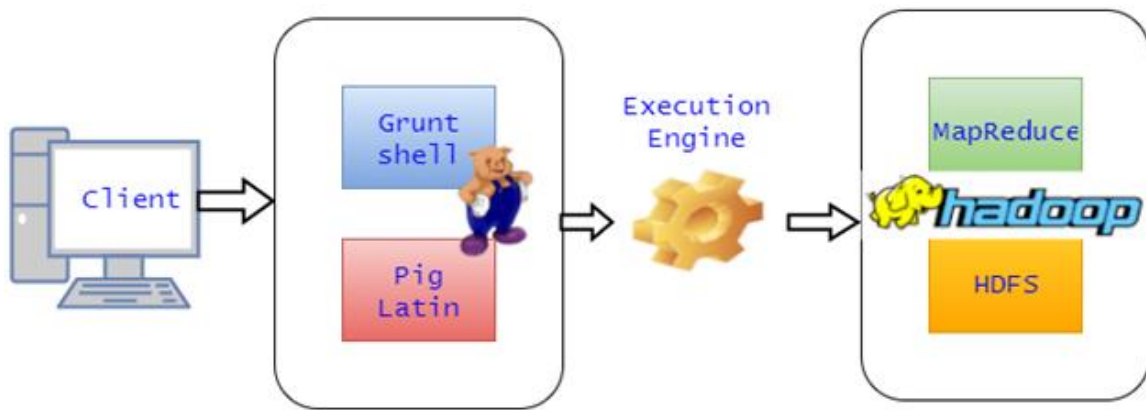


Apache Pig

Apache Pig es una plataforma para analizar grandes conjuntos de datos que consiste en un lenguaje de alto nivel para expresar programas de análisis de datos, junto con infraestructura para evaluar estos programas. La propiedad principal de los programas de Pig es que su estructura es susceptible de una paralelización sustancial, lo que a su vez les permite manejar conjuntos de datos muy grandes.

En la actualidad, la capa de infraestructura de Pig consiste en un compilador que produce secuencias de programas Map-Reduce, para los cuales ya existen implementaciones en paralelo a gran escala (por ejemplo, el subproyecto de Hadoop). La capa de idioma de Pig consiste actualmente en un lenguaje textual llamado Pig Latin, que tiene las siguientes propiedades clave:

- Facilidad de programación. Es trivial lograr la ejecución paralela de tareas de análisis de datos simples, "embarrassingly parallel". Las tareas complejas formadas por múltiples transformaciones de datos interrelacionadas se codifican explícitamente como secuencias de flujo de datos, lo que facilita su escritura, comprensión y mantenimiento.
- Oportunidades de optimización. La forma en que se codifican las tareas permite que el sistema optimice su ejecución de forma automática, lo que permite al usuario centrarse en la semántica en lugar de la eficiencia.
- Extensibilidad. Los usuarios pueden crear sus propias funciones para hacer un procesamiento especial.



Apache Sqoop

Apache Sqoop es una herramienta diseñada para transferir datos de manera eficiente entre fuentes de datos estructurados, semiestructurados y no estructurados. Las bases de datos relacionales son ejemplos de fuentes de datos estructurados con un esquema bien definido para los datos que almacenan. Cassandra, Hbase son ejemplos de fuentes de datos semiestructuradas y HDFS es un ejemplo de fuente de datos no estructurados que Sqoop puede admitir.

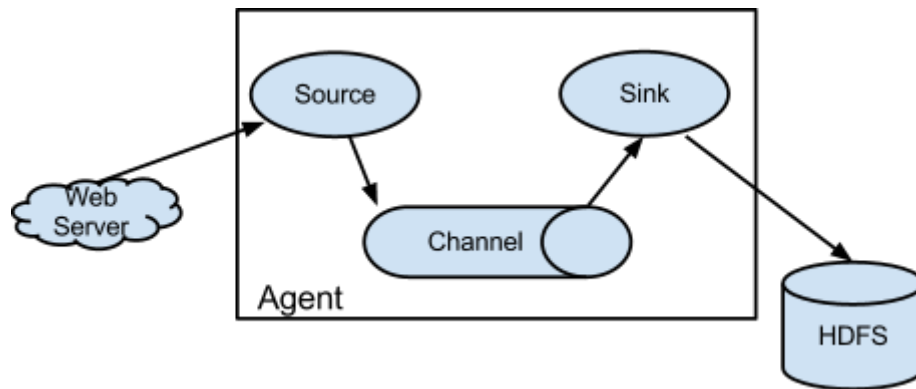
Scoop – Interfaces

- Get data from
 - Relational databases
 - Data warehouses
 - NoSQL databases
- Load to Hive and Hbase
- Integrates with Oozie
 - for scheduling



Apache Flume

Flume es un servicio distribuido, confiable y disponible para recopilar, agregar y mover grandes cantidades de datos de registro de manera eficiente. Tiene una arquitectura simple y flexible basada en flujos de datos de transmisión. Es robusto y tolerante a fallas con mecanismos de confiabilidad ajustables y muchos mecanismos de conmutación por error y recuperación. Utiliza un modelo de datos extensible simple que permite la aplicación analítica en línea.

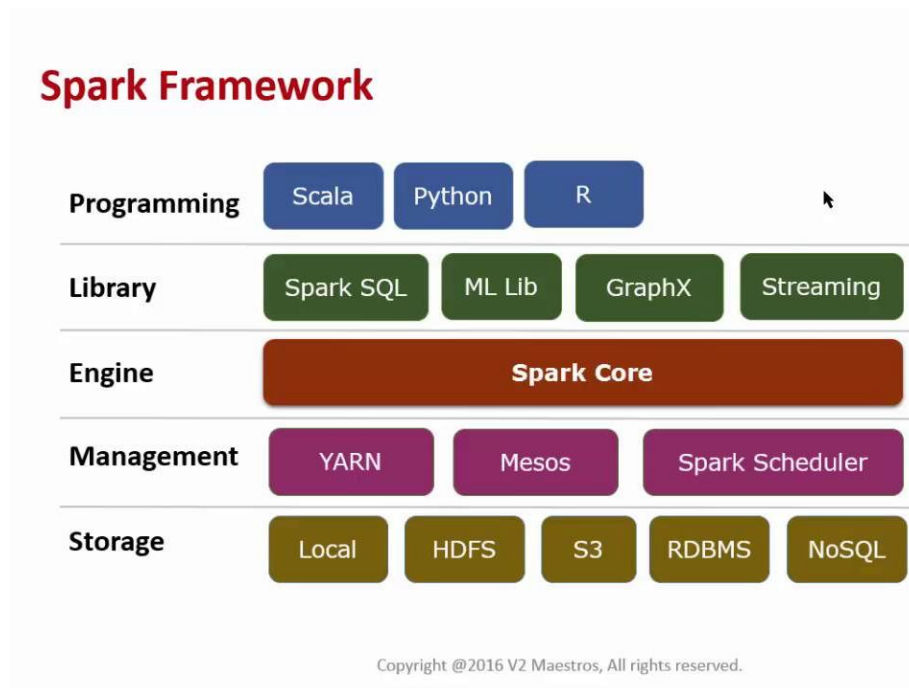


Apache ZooKeeper

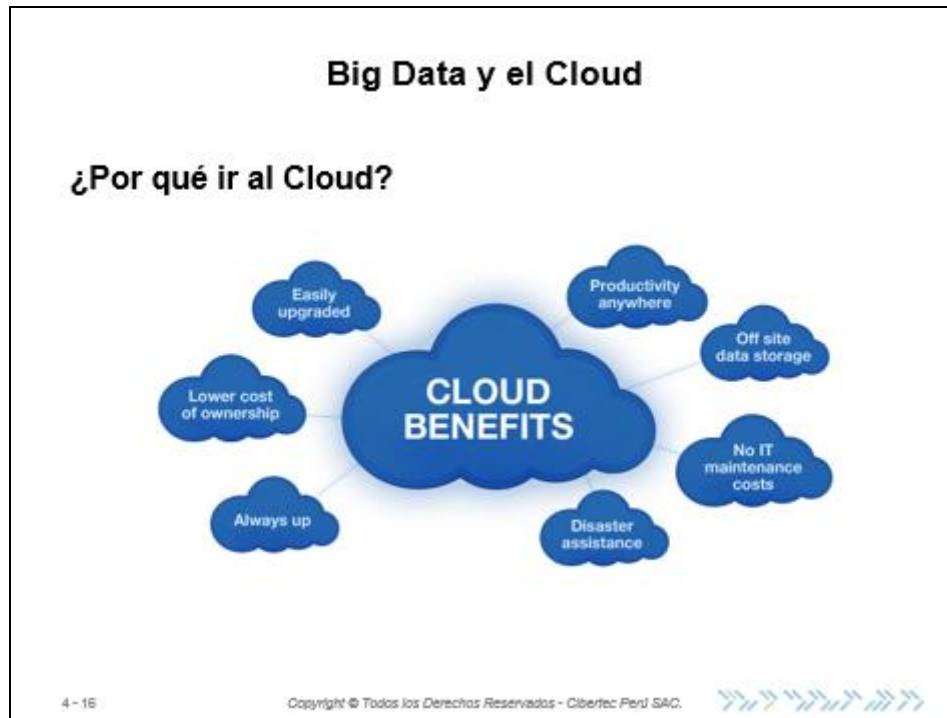
ZooKeeper es un servicio centralizado para mantener la información de configuración, nombrar, proporcionar sincronización distribuida y proporcionar servicios grupales. Todos estos tipos de servicios son utilizados de una forma u otra por aplicaciones distribuidas. Cada vez que se implementan hay mucho trabajo para resolver los errores y las condiciones de carrera que son inevitables. Debido a la dificultad de implementar este tipo de servicios, las aplicaciones inicialmente suelen escatimar en ellos, lo que los hace frágiles en presencia de cambios y difíciles de gestionar. Incluso cuando se realiza correctamente, las diferentes implementaciones de estos servicios conducen a la complejidad de la administración cuando se implementan las aplicaciones.

Apache Spark

Apache Spark es un sistema de computación en clúster rápido y de uso general. Proporciona API de alto nivel en Java, Scala, Python y R, y un motor optimizado que admite gráficos de ejecución general. También es compatible con un amplio conjunto de herramientas de alto nivel que incluyen Spark SQL para SQL y procesamiento de datos estructurados, MLlib para aprendizaje automático, GraphX para procesamiento de gráficos y Spark Streaming.



3. Big Data y el Cloud



¿Porque ir al Cloud?

Por las siguientes razones:

1. **Ahorro de costes.** Pago por la utilización de productos y servicios, eliminando costes adicionales como la compra de licencias, la inversión en infraestructura informática, el mantenimiento de los equipos y sistemas o la adaptación de los mismos a nuevas necesidades. Cabe destacar además, el menor consumo energético derivado del uso de servidores y equipos
2. **Almacenamiento y seguridad.** Existen proveedores que ofrecen servicios de almacenamiento de datos de capacidad prácticamente ilimitada. Además, junto al almacenamiento se incluyen servicios de backup y restauración de la información.
3. **Fácil acceso.** Acceso compartido y en tiempo real a toda la información desde cualquier parte y a través de cualquier dispositivo con conexión a Internet.
4. **Fácil manejo.** Integración de sistemas de forma automática. La integración de sistemas se produce de forma prácticamente automática en la nube, lo que significa que las empresas no necesitan preocuparse por resolver problemas técnicos complejos de interoperabilidad entre las soluciones contratadas. La integración de soluciones en la nube garantiza a las empresas el acceso a información coherente e integrada desde cualquiera de las soluciones, y elimina la necesidad de realizar tareas de registro de datos por duplicado.
5. **Actualizaciones automáticas.** Siempre se dispone de la última versión del software
6. **Personalizado.** Los sistemas en la nube se personalizan según los requerimientos y necesidades del cliente.

Microsoft Azure HDInsight

Azure HDInsight es un servicio de análisis, de código abierto, espectro completo y totalmente administrado para empresas. HDInsight es un servicio en la nube que hace que sea fácil, rápido y rentable procesar grandes cantidades de datos. HDInsight también admite una amplia gama de escenarios, como la extracción, transformación y carga (ETL), el almacenamiento de datos, el aprendizaje automático e IoT.

Escenarios de uso de HDInsight

Azure HDInsight se puede usar para una amplia variedad de escenarios de procesamiento de macrodatos. Pueden ser datos históricos (datos ya recopilados y almacenados) o datos en tiempo real (datos que se transmiten directamente desde el origen). Los escenarios de procesamiento de tales datos se pueden resumir en las siguientes categorías:

- **Procesamiento por lotes (ETL):** El de extracción, transformación y carga (ETL) es un proceso en el que se extraen datos estructurados o no estructurados de orígenes de datos heterogéneos. Estos datos se transforman a un formato estructurado y se cargan en un almacén de datos. Los datos transformados se pueden usar para ciencia de datos o almacenamiento de datos.
- **Internet de las cosas (IoT):** Puede usar HDInsight para procesar los datos de streaming recibidos en tiempo real desde varios tipos de dispositivos. Para más información, lea esta entrada de blog de Azure que anuncia la versión preliminar pública de Apache Kafka en HDInsight con Azure Managed Disks.
- **Ciencia de datos:** Puede usar HDInsight para compilar aplicaciones que extraigan información crítica de los datos. También puede usar Azure Machine Learning para predecir tendencias futuras de la empresa. Para más información, lea este caso de cliente.
- **Almacenamiento de datos:** Puede usar HDInsight para realizar consultas interactivas a escalas de petabytes sobre datos estructurados o no estructurados en cualquier formato. También puede generar modelos conectándolos a herramientas de BI. Para más información, lea este caso de cliente.
- **Híbrido:** Puede usar HDInsight para ampliar la infraestructura local de macrodatos existente en Azure para aprovechar las avanzadas funcionalidades de análisis en la nube.

Tipos de clúster de HDInsight

HDInsight incluye tipos de clúster concretos y funcionalidades de personalización del clúster, tales como la de agregar componentes, utilidades y lenguajes.

Spark, Kafka, Interactive Query, HBase, personalizado y otros tipos de clúster

HDInsight ofrece los siguientes tipos de clúster:

- **Apache Hadoop:** una plataforma que utiliza HDFS, administración de recursos YARN y un modelo de programación de MapReduce simple para procesar y analizar datos por lotes en paralelo.
- **Apache Spark:** una plataforma de procesamiento paralelo que admite el procesamiento en memoria para mejorar el rendimiento de las aplicaciones de análisis de macrodatos. Spark funciona con SQL, datos de streaming y aprendizaje automático. Consulte [¿qué es Apache Spark en HDInsight?](#)

- Apache HBase: base de datos NoSQL en Hadoop que proporciona acceso aleatorio y gran coherencia para grandes cantidades de datos no estructurados y semiestructurados; potencialmente miles de millones de filas multiplicadas por millones de columnas. Consulte ¿qué es HBase en HDInsight?
- Microsoft R Server: un servidor para hospedar y administrar procesos de R distribuidos en paralelo. Proporciona a los científicos de datos, estadísticos y programadores de R acceso a petición a métodos escalables y distribuidos para realizar análisis en HDInsight. Información general de R Server en HDInsight.
- Apache Storm: sistema distribuido de cálculo en tiempo real para el procesamiento rápido de grandes transmisiones de datos. Storm se ofrece como clúster administrado en HDInsight. Consulte Análisis de datos de sensor en tiempo real con Storm y Hadoop.
- Versión preliminar de Apache Interactive Query (también conocido como Live Long and Process): almacenamiento en caché en memoria para consultas de Hive interactivas y más rápidas. Consulte Uso de Interactive Query en HDInsight.
- Apache Kafka: una plataforma de código abierto que se usa para crear canalizaciones y aplicaciones de datos de streaming. Kafka también proporciona funcionalidad de cola de mensajes que le permite publicar flujos de datos y suscribirse a ellos.