UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN

ANÁLISIS DE LA TECNOLOGÍA DE BIG-DATA
Y PRESENTACIÓN DE TRES DE SUS PRINCIPALES SISTEMAS

TESINA

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA

JUAN CARLOS DOMÍNGUEZ RUBIO

Asesor: ANDRÉS HERNÁNDEZ BALDERAS

SANTA CRUZ ACATLÁN, NAUCALPAN, ESTADO DE MÉXICO OCTUBRE 2016

INDICE

INTRODUCCIÓN

Actualmente hablar de tecnología es parte de la vida diaria. Millones de dispositivos están presentes en cada una de las diferentes tareas que se llevan a cabo. Conectarse a Internet para revisar redes sociales, consultar noticias, subir fotografías, revisar el estado del tiempo o consultar la ruta más eficiente del GPS son tareas que se hacen de manera natural y casi automática.

Aparecen en el mercado con una alta frecuencia, dispositivos mejores y más eficientes, que permiten acceder a nuevas funcionalidades basadas en un mejor hardware o software. Estas mismas funcionalidades cambian el uso de la tecnología y permiten desarrollar otras necesidades para el consumidor, creando un círculo en la oferta y demanda.

Facebook™, Twitter™, Instagram™ y muchas otras plataformas, ayudan a estar en contacto con múltiples personas sin importar la localización física de las mismas; escuchar sus opiniones sobre algún tema en particular, generar debates, ver contenido multimedia, organizar eventos y establecer comunicación entre otras acciones que ocurren como parte de la rutina diaria.

Plataformas como Facebook™ o Twitter™ cuentan con un soporte tecnológico que hace que millones de usuarios puedan conectarse de manera simultánea. Analizar el trabajo que implica hacer una búsqueda en Google o encontrar un Trending Topic en Twitter™, no es tarea de un usuario final; estas tareas las realizan expertos en desarrollo de tecnología.

En ambientes empresariales donde los administradores de tecnologías de la información trabajan a diario con sistemas, rara vez se trabaja con computación de alto desempeño debido a que frecuentemente no es necesaria. Los sistemas de cómputo tradicionales que incluyen grandes servidores y almacenamiento del orden de Terabytes son suficientes para muchas de las tareas de operación que se llevan a cabo.

En estos ambientes empresariales se trabaja constantemente con nueva tecnología que ayude a resolver problemas o a identificar y aprovechar ventajas competitivas para el desarrollo del negocio. Obtener información privilegiada y de primera mano permite a los administradores y directores de las grandes compañías definir el rumbo del negocio a través de la correcta toma de decisiones.

Aún en este tipo de ambientes empresariales, al hablar de nuevas tecnologías no es totalmente claro identificar qué ventajas que se obtienen del el uso de las mismas y muchas veces tampoco es evidente que problemas resuelven o que oportunidades de negocio se crean con ellas.

Personas que dirigen departamentos de tecnologías de la información y que trabajan día a día en departamentos de cómputo, usan tecnología de cómputo de alto rendimiento, y deducen que estas tecnologías pueden ser aprovechadas para análisis de datos. Big Data como una de las relativamente nuevas tecnologías promete ayudar en este tipo de tareas y otorgar ventajas competitivas a las empresas o resolver tareas complicadas de análisis en el campo de la

investigación, donde de otra forma los cálculos de datos no serían posibles o el resultado del análisis tomaría tanto tiempo que se perdería la relevancia del mismo.

Acorde a este planteamiento aparecen diversos cuestionamientos: ¿cómo identificar qué ventajas puede proveer esta tecnología?, ¿cómo saber en qué momento la tecnología actual deja de ser suficiente y debe de utilizase Big Data?, ¿quién está apto para tomar estas decisiones? Estos cuestionamientos son naturales cuando se realiza un análisis de Big Data; sin embargo, no es clara aún para algunos departamentos de tecnologías de la información la respuesta a estas preguntas.

El presente trabajo explica qué es la tecnología Big Data y cómo se puede aprovechar; muestra la evolución de la misma e identifica de una manera práctica cuáles son las consideraciones que se deben tener en cuenta para saber cuándo un problema es un candidato a resolverse con esta tecnología. El objetivo es poder entender de una manera amplia y clara que tipo de problemas se pueden resolver con Big Data y cómo identificarlos a través del entendimiento de las diferentes capas lógicas de su arquitectura.

El texto se estructura explicando qué es la tecnología Big Data a través de la evolución de los sistemas de datos hasta llegar a ella; explica la arquitectura de esta tecnología y la naturaleza de los problemas que con ella se resuelven para finalizar profundizando en la estructura de algunos sistemas que existen actualmente en el mercado.

CAPÍTULO I ¿Qué es Big Data?

En la actualidad es común escuchar el término “Big Data” como parte de los nuevos conceptos que la evolución de la computación va introduciendo. El significado no es muy claro debido a que no hay un parámetro adecuado que permita visualizar de una manera correcta e inmediata el alcance de la palabra “Big” dentro del término.

En el universo de los sistemas de datos, el significado de la palabra “Big” es relativo al entorno en el que se desenvuelve el término; esto es debido a la cantidad de información que se maneja en estos sistemas y a las necesidades que cada ambiente requiere. Existen sistemas de datos que son administrados en computadoras de escritorio, mientras que en entornos empresariales o de investigación científica pueden existir complejos sistemas de almacenamiento y procesamiento para una sola plataforma de datos.

La interacción directa o indirecta con sistemas de datos es parte de la vida diaria; el procesamiento de un retiro de efectivo en una cuenta bancaria, la inscripción del nacimiento de un bebé en el registro civil, o el simple hecho de pagar la cuenta de compras en un supermercado son claros ejemplos de las interacciones que ocurren a diario.

Para explicar el significado del término Big en un sistema de datos, se debe tomar en cuenta a la cantidad de datos que el sistema almacena o procesa. Esta cantidad de información varía de un entorno a otro; por ejemplo, mientras que para el procesamiento manual en situaciones como el pasar la lista de asistencia en el salón de una escuela el hablar de Megabytes de información o datos es algo casi impensable, para ambientes empresariales donde el procesamiento de datos es ejecutado por computadoras o servidores grandes, un tamaño superior a Gigabytes es común y aceptable.

Para clarificar la idea de que el entorno del problema en el cual trabaja un sistema de datos afecta y transfiere parte de sí mismo a los sistemas, es útil analizar algunos ejemplos o casos de sistemas.

a) Un sistema bancario cuenta con una gran cantidad de datos de sus cuentahabientes y cuentas asociadas a cada uno. Las principales necesidades del sistema radican en:

    - Soportar grandes volúmenes de datos
    - Manejar atomicidad de operaciones
    - Soportar alta disponibilidad
    - Centralización y distribución.

    Las necesidades del sistema bancario debido a su naturaleza, lo definen como un posible sistema de bases de datos distribuido; teniendo como características una

alta disponibilidad de datos y el aseguramiento de la integridad de operaciones de acceso y escritura. Esto requiere que el sistema de base de datos se encuentre construido bajo estrictas plataformas de alto desempeño tanto en hardware y software.

b) Un sistema de almacenamiento de información para una escuela privada en donde se registran las inscripciones del ciclo escolar requiere operaciones sencillas; y debido al uso por una o dos personas, el problema de acceso concurrente es muy poco probable o quizá no exista. El sistema se compone mínimamente de las siguientes necesidades

   - Volumen de almacenamiento bajo
   - Manejo de concurrencias bajo o innecesario.
   - Nula distribución de datos.

   Las necesidades del sistema de información de la escuela, definen al sistema como una base de datos centralizada donde la concurrencia es baja o nula, este sistema no requiere una plataforma de alto desempeño o distribución de datos.

Con estos casos o ejemplos se observa que para el manejo de información, el sistema de datos está completamente desarrollado en relación al entorno del problema que soluciona.

Para entender de una manera adecuada el término “Big Data” y sabiendo que de primera instancia el significado es relativo a grandes almacenes de datos, es necesario entender el ¿cómo? Y el ¿por qué? de la evolución de los sistemas de información y bases de datos. Responder peguntas como: ¿qué problemas se han resuelto con su evolución? y ¿cuáles problemas han aparecido?; ayudará a identificar en que momento una plataforma tradicional de datos deja de ser suficiente para resolver algunos problemas de muchos datos, ya que es ahí donde el significado de Big Data se clarifica.

Antes de realizar el análisis, se aborda la histórica evolución de los sistemas de bases de datos para encontrar en que momento y bajo qué condiciones aparece el término Big Data. Esto ayudará a entender su función en la actualidad y cómo es que una gran cantidad de personas interactúan con sistemas de Big Data prácticamente sin saberlo.

Evolución de los sistemas de datos

A medida que la computación ha ido evolucionando, muchos de los problemas tradicionales se han resuelto gracias a que las capacidades de cómputo y almacenamiento aumentaron; esta misma evolución ha incentivado el desarrollo de nuevas tecnologías; la creación de nuevas generaciones de dispositivos fueron posibles y con este desarrollo otros paradigmas y lenguajes de programación llegaron al mercado.

La ley de Moore, formulada por Gordon Moore y que es aplicable a los transistores que caben dentro de un procesador es uno de los identificadores más comunes de esa evolución. La ley indica que cada 3 años la potencia del procesador se multiplica por 4 con un costo similar o inferior y manteniendo el mismo espacio.[1]

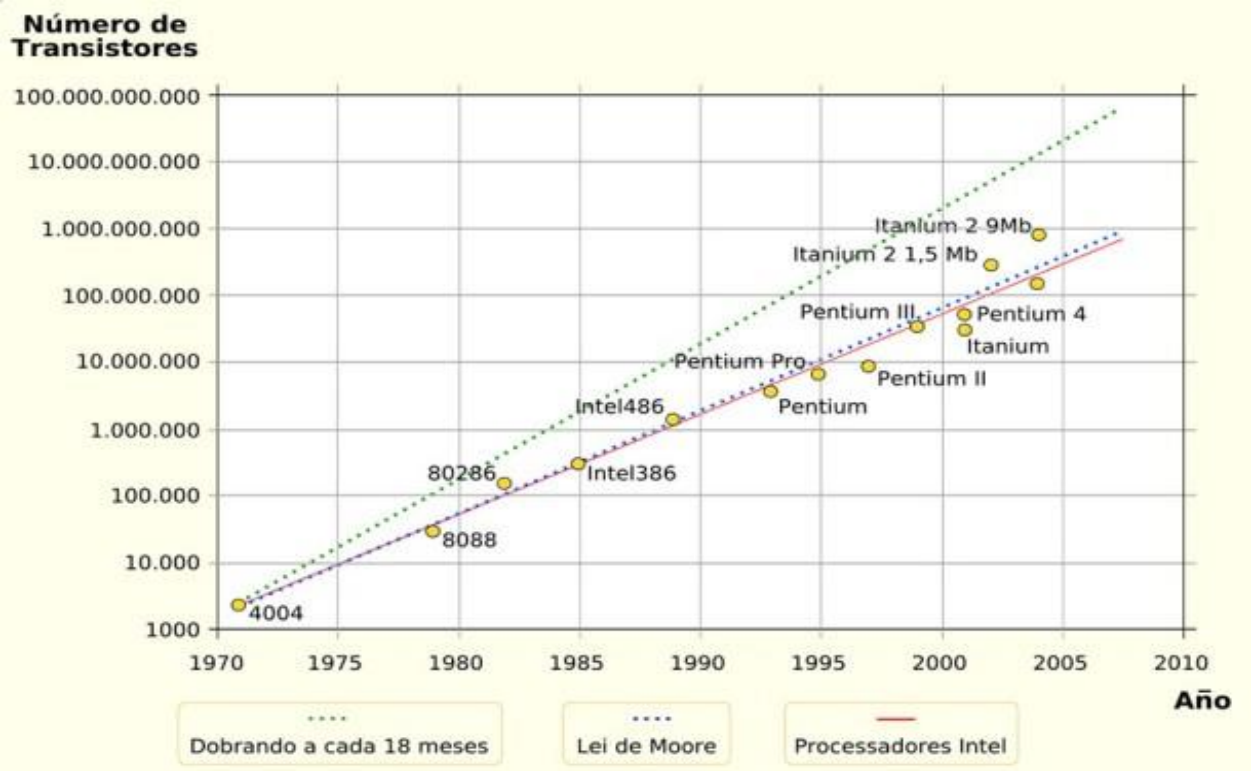

Fig. 1 Conteo de transistores 1970-2010[2]

La miniaturización, el descubrimiento de nuevos materiales y el abaratamiento de tecnología permitieron que los sistemas de almacén de datos cambiaran radicalmente. En 1956 IBM

[1] Tubella I. (2005). Sociedad del conocimiento. Barcelona España: UOC. p. 3.
[2] Hipertextual, (2015), Ley de Moore [Figura]. Recuperado de http://hipertextual.com/2015/04/ley-moore-50-aniversario

comercializaba el IBM 350[3]; un disco de 5MB que pesaba más de una tonelada y era necesario usar montacargas para su traslado. Para ese momento esta era considerada la tecnología de punta. En ella se podía almacenar 1 canción promedio del formato MP3 actual.

Para el año de 2011 se contaban con formatos de almacenamiento que podían soportar GB de datos y que eran considerados portables. En el mercado se podían encontrar discos portables de entre 100 a 500 GB de almacenamiento del tamaño de medio ladrillo; para ese mismo año IBM anuncio la creación del disco duro más grande hasta entonces fabricado. Este disco estaba construido con 200,000 discos duros convencionales con una capacidad total de 120 PB, podía almacenar 24 billones de MP3[4].

Un sistema de datos tradicional almacena datos y ejecuta operaciones sobre los mismos, el evidente cambio en las capacidades de hardware permitió a estos sistemas evolucionar de tareas básicas a tareas de procesamiento complejo algoritmos sumamente elaborados.

Sin embargo, los sistemas de datos son bastante variados. Puede considerarse una hoja de Excel como un sistema que almacena datos, que opera con ellos y obtiene resultados. Este sistema cumple necesidades básicas de almacenamiento y capacidad de procesamiento. Existen otros sistemas más avanzados que sirven para almacenar cantidades de datos significativamente más grandes; atienden necesidades específicas para grandes organizaciones como universidades, empresas públicas y privadas así como organismos gubernamentales. Estos sistemas son llamados Sistemas Gestores de Bases de Datos.

Un sistema de base de datos debe cumplir al menos las siguientes características

- Almacenar datos
- Conservar los datos íntegros
- Permitir la consulta de datos

Estos tres objetivos fundamentales los cumple cualquier sistema actual con diferentes niveles de complejidad. Existen bases de datos locales para tareas de oficina o plataformas de alta disponibilidad que permiten configuración de clústeres para redundancia y tolerancia a fallos.

Almacenar datos tiene un significado diferente si se refiere a Megabytes o Terabytes. El acceso a la información, dependiendo del volumen, puede ser inmediato para archivos o bases pequeñas o generar complejos escenarios para bases de datos que no caben en un disco duro tradicional. Estos escenarios dificultan el acceso a los datos ya que para sistemas que permiten acceso simultáneo, la integridad es un factor fundamental a considerar.

[3] Simonite T. (ND). IBM 350 disk storage unit. Abril 8, 2015, de IBM Sitio web: https://www-03.ibm.com/ibm/history/exhibits/storage/storage_350.html

[4] Simonite T. (2012). IBM Builds Biggest Data Drive Ever. Mayo 4, 2015, de MIT Technology Review Sitio web: https://www.technologyreview.com/s/425237/ibm-builds-biggest-data-drive-ever/

La necesidad de sistemas de bases de datos para resolver problemas es histórica; Herman Hollerith, fundador de Tabulate Machine Company, una de las precursoras de la actual IBM, es uno de los pioneros en almacenar información en un medio físico a través de tarjetas perforadas y procesarlas con una máquina; básicamente guardar información y recuperarla[5].

Para entender como evolucionaron los sistemas de bases de datos, es necesario conocer un poco más en la historia de la evolución de las bases de datos y de los pioneros de las mismas.

En el año de 1959 un grupo de industrias privadas e instituciones gubernamentales de Estados Unidos que trabajaban con procesamiento de datos, se dan a la tarea de generar un estándar en lenguajes de programación, el objetivo fue generar un lenguaje que pudiera ejecutarse en cualquier computadora. La organización se llamó CODASYL (Conference on Data Systems Languages). El resultado de esta tarea fue el lenguaje de programación COBOL cuyo uso continúa aun en la actualidad para algunas grandes compañías como los bancos.

COBOL fue concebido para poder ejecutarse en diferentes computadoras y para trabajar fundamentalmente con ficheros. El uso de sistemas de ficheros para almacenar datos era hasta ese entonces el paradigma de sistemas de información.

Para el año de 1960 la compañía General Electric contrata a un informático proveniente de la Universidad de Michigan de nombre Charles Bachman. Este informático, cuya trayectoria en la computación de datos fue premiada, genera para esta compañía el IDS (Integrated Data Store), el cual es considerado uno de los primeros sistemas reales de base de datos.

Dentro de la CODASYL en el año de1965 se formó el grupo List Procesing Task Force, cuya tarea fue la de generar soporte a acceso de ficheros del lenguaje de programación COBOL. La utilización de punteros es una de las características de esta mejora del lenguaje cuyo uso se mantuvo para acceso a memoria en muchos de los lenguajes de programación procedurales.

En esta etapa de la historia, el almacenamiento se realizaba en cintas magnéticas. El acceso secuencial a memoria trabajaba perfectamente con el uso de punteros; puesto que para ingresar a un determinado contenido o dato, había que navegar a través de ellos.

En el año de 1968 y 1969 IBM introduce un sistema que ayuda al manejo de datos para el programa APOLO de la NASA llamado IMS (Information Management System). Este modelo es considerado el primer sistema de información bajo el paradigma de bases de datos de red. Más tarde en el año de 1969 CODASYL publica las especificaciones de los modelos de base de datos jerárquicos o de árbol. Es por esta razón que a este modelo suele denominarse modelo CODASYL.

El modelo de red o modelo DBTG (Data Base Task Group) como lo denominó la propia CODASYL suponía grandes ventajas sobre el manejo de datos respecto a sus predecesores. Una de ellas

[5] Cruz F. (2011). Herman Hollerith. Febrero 3, 2015, de Columbia University PhD Sitio web: http://www.columbia.edu/cu/computinghistory/hollerith.html

es que estandarizaba operaciones incluyendo ya una semántica de un lenguaje de definición de datos (DDL) y un lenguaje propio de manipulación de datos (DML).

El nombre del modelo de red proviene de la forma en la que se encontraban los datos, los cuales se organizaban en una jerarquía y se recorrían a través de punteros en sus nodos hasta encontrar el dato específico de búsqueda.

En el año de 1970 Edgar Frank Codd, un matemático perteneciente a IBM y proveniente de Oxford, publica las primeras ideas de lo que es el modelo de bases de datos relacional a través del artículo “A Relational Model of Data for Large Shared Data Banks”. La idea era almacenar y presentar de manera lógica los datos a través del concepto de relaciones con el objetivo de poder acceder a los datos de una manera más rápida y óptima que el modelo DBGT permitiendo operaciones de mayor complejidad.

La idea tenía varias características dignas de explotarse; entre ellas la clara separación entre el almacenamiento y la lógica de consulta del usuario. Una premisa era que un usuario no tenía que preocuparse sobre cómo se almacenaban los datos sino sólo como se explotaban; por lo que introducía el concepto de un lenguaje de lectura de datos a alto nivel. Además evitaba la necesidad de un ordenamiento al momento de almacenar datos y dejaba la representación física en un nivel diferente.

Aun cuando la idea de las bases de datos relacionales tenía sin duda un alto potencial, IBM no le dio credibilidad a esta idea ya que en ese momento era dueña de uno de los mejores productos de bases de datos jerárquicos; el IMS. Cambiarlo significaba un gran costo y por ello Edgar F. Codd publica el modelo para bases de datos relacionales en una revista de divulgación científica debido a que dentro de IBM no hubo soporte para su idea.

Para 1971 CODASYL informa sobre la actualización de las especificaciones del modelo de red, en el cual ya existía la posibilidad de definir grupos de usuarios y vistas de datos.

En este mismo año la Universidad de California en Berkley con apoyo económico proveniente de la NSF (National Science Foundation) comienza el desarrollo del primer proyecto sobre bases de datos relacionales. Este proyecto fue conocido comercialmente con el nombre de Ingres.

Hasta ese momento resolver problemas de acceso a datos, gran capacidad de consultas y una superior abstracción habían sido los principales motores del desarrollo sistemas de base de datos; pero el modelo relacional venía a cambiar la forma de concepción de estos sistemas de una manera radical, sugiriendo ventajas como la separación lógica del almacenamiento, la extracción y el ordenamiento lógico en función de las relaciones.

Para el año de 1974 IBM lanza a pruebas su primer sistema de bases de datos relacionales con el nombre de “System R”. Este sistema cuenta con características como ser el primer sistema multiusuario. Además el acceso de datos se da a través de un nuevo lenguaje de alto nivel denominado SEQUEL, el primer precursor del actual SQL (Structured Query Languaje).

En este mismo año aparece la primera versión funcional de Ingres, el producto y el código eran de dominio público y contenían un lenguaje de alto nivel llamado QUEL. El hecho de que el producto fuera libre hizo que muchas organizaciones y universidades lo adoptaran.

El American National Standards Institute, Standards Planning And Requirements Committee ANSI/SPARC publica para 1975 una especificación más adecuada a los niveles de abstracción del modelo relacional; para esta publicación CODASYL comienza a introducir el uso de la lógica relacional al proponer un lenguaje DDL para este tipo de bases de datos.

Para el final de la década de los 70´s IBM es la compañía que presenta los más grandes desarrollos en cuestión de los sistemas de bases de datos con el lanzamiento de dos productos: System R y DB2.

Para finales de la década y principios de los 80 aparecen la mayoría de los sistemas de bases de datos conocidos en la actualidad. Entre estos gestores de base de datos denominados de cuarta generación aparecen Ingres, Sybase y Oracle.

Uno de los avances más significativos es el uso del cálculo relacional para realizar los cálculos de datos; previamente se utilizaba algebra relacional; este cambio da origen a múltiples lenguajes de alto nivel para consultas de datos. Con el tiempo el lenguaje que predominaría sería una mezcla de los lenguajes propuestos por varias compañías y basado tanto en el álgebra relacional como en el cálculo relacional escrito en inglés natural y basado en su mayoría en el SEQUEL de IBM.

Para principios de la década de los 80 el SEQUEL de IBM pasa a manos del ANSI (American National Estandar Institute) quien estandariza el lenguaje y genera el SLQ ANSI el cual sufriría algunas modificaciones menores hacia el futuro. La estandarización del lenguaje de consulta así como el desarrollo de las bases de datos relacionales resolvían el problema de consulta y acceso a datos. Para 1980 Codd gana el “Turing Award” por sus aportaciones para el desarrollo de la normalización de bases de datos.

Aparece en el año de 1988 el sistema ObjectStore; el primer sistema de datos orientado a objetos. La evolución de las bases de datos hacia el almacén de objetos respondía a la posibilidad de limitar el acceso a datos a través de programación, optimizar la forma en la que se almacenan los datos y proporcionar evidentes ventajas de lectura escritura por encima de las bases relacionales.

A pesar de ello, este tipo de sistemas de bases de datos no lograron una gran popularidad. El principal problema de la adopción de las bases de datos orientadas a objetos se deriva del mismo problema que puede presentarse a un programador al momento de querer recuperar un objeto serializado. Si se desconoce el patrón o clase del objeto hace que el programador necesite más información sobre cómo se componen y almacenan los objetos generando un claro retroceso a los problemas de las bases de datos jerárquicas.

Para el año de 1998 aparece en el mercado el paradigma objeto-relacional que trataba de solventar el problema de traducción heredado por los programadores entre las bases de datos relacionales y los lenguajes de cuarta generación orientados a objetos.

Empresas como Sun Microsystems (Actualmente Oracle) aprovecharon esta mezcla de paradigmas para generar productos de gran soporte y amplia aceptación; tal es el caso del lenguaje de programación Java y su conector para bases de datos JDBC (Java Data Base Conectivity) basado en el ODBC de Microsoft. Estas aplicaciones en conjunto con extensiones al modelo relacional permitían almacenar datos complejos e inclusive declarar tipos de datos de usuario y almacenarlos en tablas relacionales.

En los 90´s también se aborda la solución de otros problemas que tenían que ver con el volumen de la base de datos. La conectividad de red no era buena en términos de velocidad y era demasiado cara, además el procesamiento de datos aún estaba bastante limitado por la capacidad de procesamiento del hardware por lo que aparecieron las bases de datos distribuidas; las cuales eran instancias de las mismas bases colocadas en más de una ubicación física y conjuntadas a través de procesos batch en una base de datos central. Esta práctica es común en grandes empresas como los bancos.

IBM lanza para finales de los noventa el gran representante de este tipo de sistemas distribuidos. El System R*.

También en esta década se aborda la problemática de poder almacenar en una base, datos no estructurados como canciones, videos y fotos. Por lo que aparecen las bases de datos multimedia y se ocupan metadatos para poder indexar y almacenar estos tipos de datos.

A partir del año 2000 la evolución de los sistemas de datos camina a pasos agigantados. Se enfoca principalmente en la explotación de los mismos. Conceptos como Bussiness Inteligence hacen su aparición y las bases de datos OLAP (On-Line Analytical Processing) junto con ellos. Estas bases de datos tienen la característica de ser dimensionales típicamente en tiempo; es por eso que comúnmente se les llama cubos. La necesidad que cubren es la de poder hacer análisis histórico de datos con miras en la toma de decisiones empresarial. Aparecen para este paradigma los Data-warehouses, los cuales son arreglos de bases de datos que permiten almacenar de manera centralizada datos que tienen una relación lógica aunque no necesariamente se encuentren en la misma tecnología; y para controlar el acceso a los datos se generan vistas especiales de los Data-warehouses, denominadas DataMarts, los cuales limitan la visibilidad de los datos y soportan permisos de usuario.

Las técnicas de minería de datos permiten realizar un análisis de datos en estos almacenes y aquí aparece un concepto que es uno de los más utilizados en Big Data, el mezclar diferentes fuentes de datos externas para análisis.

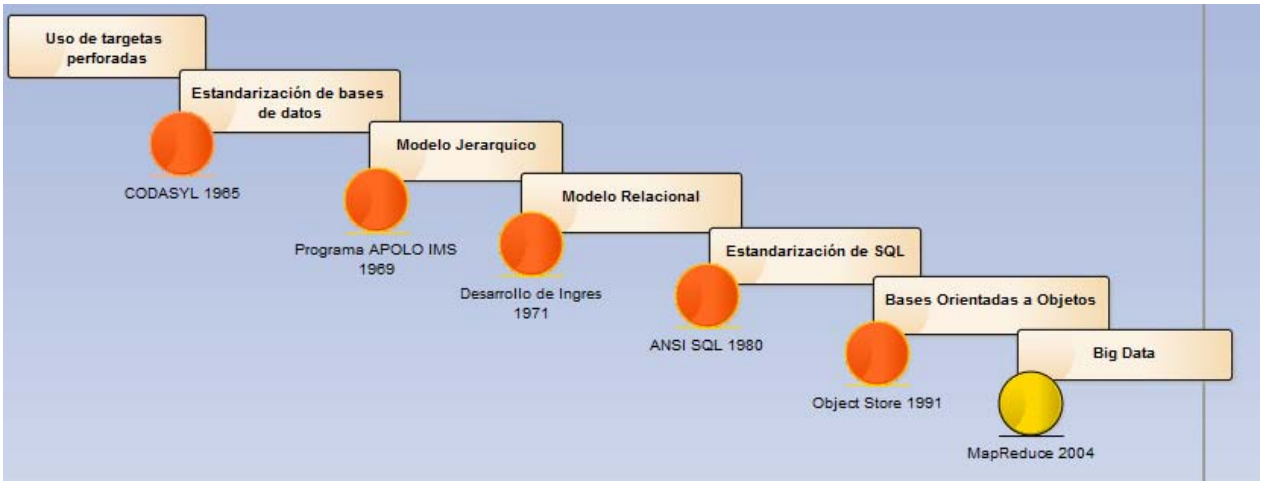

Fig. 2 Evolución de los sistemas de datos

Los sistemas hasta este punto han evolucionado bastante; para finales de la década de los 90 con la popularidad de internet creciendo, y con el aumento significativo en publicación de páginas web, aparece la problemática derivada de la cantidad de información disponible; los retos de los buscadores de páginas son soportar búsquedas rápidas en grandes cantidades de datos; es aquí donde aparece el término de Big Data.

Características de Big Data

A través de la historia, los datos siempre han existido y estado presentes en múltiples partes de la vida diaria; pero habían sido hasta ahora subutilizados o simplemente ignorados. En el mundo empresarial los datos habían tenido que ser estructurados antes de ser consumidos y normalizados para ser explotados muy frecuentemente en volúmenes que van en rangos de megabytes hasta terabytes.

Ahora todo es digital. Para los usuarios de sistemas de información los datos se pueden encontrar estructurados (en bases de datos u hojas de cálculo) o no estructurados (fotografías, video, streaming, redes sociales, blogs, etc) junto con metadatos que permiten interactuar con los mismos datos no estructurados, a estos metadatos se les conoce como datos interactivos.

Acorde a Mike Loukides vicepresidente de estrategias de contenido de O'Reilly Media, Inc. El concepto ***Big*** en el ámbito de Big Data se refiere a ***cuando el volumen de los datos para resolver un problema determinado es un problema en sí mismo.***[6]

Big Data representa un paradigma donde los datos estructurados y no estructurados se mezclan de manera natural a la vista del usuario final; con el soporte tecnológico suficiente para manejar un volumen de datos en constante crecimiento, con la posibilidad de agregar datos interactivos que logren añadir velocidad y variedad a los datos y mantenerlos utilizables sin importar su cantidad o su origen.

Para los autores de Big Data Imperatives; Soumendra Mohanty, Madhu Jagadeesh y Harsha Srivatsa Big Data es más un concepto que un término preciso, indicando que la mayoría de personas asocian el término a un problema relacionado con el volumen de datos usualmente en términos de petabytes o superior.

La mayoría de autores coincide en que para hablar de un problema del ámbito Big Data, se deben considerar que al menos se cuente con características de 3v's Volumen, Velocidad y Variedad, algunos de ellos agregan también Veracidad y Valor, aunque no todos concuerdan en las ultimas 2.

El Volumen se refiere a la escala de los datos con los que se trabaja, volúmenes en términos de Petabytes, Zetabytes, Yotabytes o superiores son comunes a esta escala. Los sistemas tradicionales de base de datos difícilmente podrían manejar estos volúmenes a una velocidad aceptable para su uso.

La Velocidad, viene de la necesidad de almacenar, analizar y leer los datos en un tiempo de espera adecuado para cada tarea que realice el usuario final, por ejemplo, hacer una búsqueda

[6] Loukides M. (2011). Big Data Now. California USA: O'Reilly Media., p.8.

en Internet, consultar Facebook o usar Twitter, deben ser para el usuario final experiencias casi inmediatas, aun cuando el volumen de datos de Tweets por día para el año de 2013 ya alcanzaba los 500, 000,000[7]

Variedad es un término que engloba el hecho que Big Data debe poder manejar datos que vengan prácticamente de cualquier fuente y en cualquier formato. Leer datos estructurados es sumamente más fácil para los sistemas que obtener datos que no contienen estructura. Fotos, videos, RSS's generalmente necesitan mecanismos más elaborados para su lectura, sin pensar aun en su análisis, donde la programación semántica hace su aparición.

Existen otras V´s que otros autores consideran importantes; por ejemplo IBM agrega Veracidad a la tripleta inicial[8]. Algunos autores consideran en ese sentido que en tal volumen de datos no siempre es relevante la veracidad; ya que es prácticamente lo mismo saber si se tiene en Twitter 5, 000,000 de seguidores o se tienen 5, 000,012, puesto que la diferencia no es representativa. Puede dejársele esa decisión al experto en Data Science quien sabrá que tan importante es la Veracidad en el ambiente que se está desarrollando.

El potencial uso de los datos internos y externos lleva a analizar que de primera impresión, no es posible para la mayoría de las empresas establecer y aprovechar sistemas de Big Data asumiendo el TCO (Total Cost of Ownership), ya que si bien es calculable de manera inmediata el costo de implementación; el costo de mantenimiento es algo sumamente más difícil de considerar y soportar.

Partiendo de la premisa de las 3 V's que necesita cumplir cualquier problema del ámbito de Big Data, se clarifica que es lo que Big Data resuelve y algunas de sus características.

Para realizar un análisis de datos es necesario identificar primero la condición de los mismos; es decir, se necesita saber si los datos se pueden utilizar o no. Algunas de las condiciones a considerar son si los datos están limpios, si están completos, si las partes que faltan de los datos se pueden simplemente omitir o se necesitan completar, si el formato en el que se encuentran puede ser leído y que tan difícil es leerlo.

La limpieza de los datos es comúnmente el primer paso. Limpiar formatos (estandarizar formatos) o realizar un análisis y selección automática de imágenes es el primer proceso donde Big Data comienza a trabajar manejando grandes volúmenes de datos y generando metadatos de manera automática que como resultado darán una fuente de datos utilizable para otro proyecto de análisis.

Una vez limpia la fuente de información hay que decidir qué hacer con los datos que están incompletos o erróneos. Una opción es descartarlos de inmediato, pero por otra parte los datos

[7] Twitter. (ND). Twitter Usage Statistics. Marzo 16, 2015, de Twitter Sitio web: http://www.internetlivestats.com/twitter-statistics/

[8] IBM. (ND). The Four V's of Big Data. Marzo 16, 2015, de IBM Sitio web: http://www.ibmbigdatahub.com/infographic/four-vs-big-data

incompletos puede ser lo mejor que se tiene para trabajar en este punto, por tanto descartarlos no parece ser la mejor opción, además con la posibilidad de agregar datos interactivos, se puede agregar la razón de porque están incompletos estos datos y resultar útiles en algunos escenarios y en las etapas de análisis a través de herramientas avanzadas como es el caso de las basadas en procesamiento natural del lenguaje.

Hay veces que la clasificación y el enriquecimiento de datos debe hacerse a través de los propios humanos para poder generar un origen confiable y efectivo. Al resultado final se le conoce como Data Platform la cual es muy parecida a un Data-warehouse pero con una diferencia fundamental: la Data Platform contendrá una API (Aplication Program Interface) que permitirá explorar y entender los datos crudos almacenados a diferencia del análisis tradicional realizado en bases OLAP.

Big Data por tanto abarca desde la limpieza de datos hasta la preparación de los mismos; para ser consumidos por otros productos de datos y generar nuevos productos o para generar nuevas plataformas de datos.

Big Data es una tecnología que ayuda a procesar grandes volúmenes de datos; se encarga de procesar problemas relacionados con volumen, velocidad y variedad de orígenes de datos. Ahora el siguiente punto es determinar la naturaleza de los problemas que se pueden resolver con este tipo de tecnología.

CAPÍTULO II Identificación de un problema en el ámbito de Big-Data.

Hablar de datos crudos y de información se ha vuelto un tema que trae consigo implicaciones del entorno. Saber de dónde vienen los datos, qué hacer con ellos y a dónde van, implica que las características de los mismos sean importantes para el análisis. Aquí se visualiza que los datos aparecen en prácticamente cualquier lugar y en cualquier momento.

Con los sistemas de cómputo y dispositivos actuales, los datos se encuentran en diversas fuentes, formas y tamaños. Videos, música, datos bancarios, la información se presenta al usuario de múltiples formatos y a través de diversos medios.

Se han desarrollado a través de la historia tecnologías para manejar diferentes tipos y volúmenes de datos y atender problemas que la administración de los mismos trae consigo. Originalmente el desarrollo de estas tecnologías se centraba en que los datos fueran confiables, perdurables y accesibles, resolver problemas de acceso simultáneo a datos o de atomicidad en las operaciones, presentaba grandes retos para la comunidad que se encarga del desarrollo tecnológico.

La aparición de grandes herramientas que soportaban complejos modelos de almacenamiento de datos como System R y Oracle permitió al usuario final, centrarse en la lógica de explotación de datos sin preocuparse por el almacenamiento. Junto con ello la aparición de bases de datos distribuidas y el empoderamiento del hardware llevo a los sistemas a soportar millones de operaciones simultaneas, tal es el caso de los sistemas bancarios actuales.

El auge de Internet y el aumento del acceso a la tecnología por parte de la población hicieron que muchos problemas nuevos comenzaron a ser tratados como problemas de datos. Uno de los casos fue la creación de la CDDB (CD Data Base). Sus creadores basaron su diseño y construcción en la premisa de que cada CD contenía una longitud única en samples de cada canción; así podían subir información de los discos de música como nombre de la canción, nombre del disco, autor, disquera y podrían compartirlos basados en esta longitud. Aún se utiliza este servicio cuando se copia un disco a un archivo de audio y la biblioteca de la computadora descarga la información del disco.

Con el Internet 2.0 se produce una explosión en la generación de datos. Aunque los volúmenes manejados hasta el momento ya eran grandes, estos crecen exponencialmente a un ritmo vertiginoso. Las páginas de Internet dejan de ser estáticas para generarse de manera dinámica y el uso de las mismas se expande a muchos ámbitos en todo el mundo.

La aparición de los teléfonos inteligentes y nuevos dispositivos integrados, las redes sociales como Facebook™, Twitter™, Instagram™, etc. y la posibilidad de desarrollar aplicaciones para

sistemas operativos móviles casi por cualquier persona, logra que la generación de datos se multiplique aún más.

Aunque los datos siempre se han encontrado en todas partes, las compañías y los usuarios solo se enfocaban en los datos que era posible capturar y manejar. La aparición de Internet 2.0 logra que cualquier persona pueda generar contenido y ponerlo disponible en la red. La localización y búsqueda del contenido es atendida por programas denominados “Search Engines”, algunos como Altavista™ (ahora Bing™) o Yahoo™ basados en la búsqueda de las palabras dentro del contenido de la página, o en casos como Google™ basados en la generación de índices de páginas asociándolas a palabras clave (metadatos).

Aquí aparece un problema sobre los datos que no existía en la evolución original de los sistemas manejadores de bases de datos, cuya solución dará origen a nuevos paradigmas tecnológicos. Este es la “Búsqueda a escala”.

Cambio en la problemática de datos.

Hasta este momento los Search Engines, o meta Search Engines, realizaban la búsqueda de datos, en los mejores casos indexando las páginas disponibles en Internet; la búsqueda se realizaba sobre metadatos agregados a la página quienes decidían si una determinada página aparecía en una búsqueda o no.

Cuando el análisis a las páginas y la indexación era automático el resultado no era el más adecuado; y en los casos que los buscadores utilizaban un análisis contextual realizado por humanos como el caso de Yahoo™, el contenido se desarrollaba más rápido que lo que se indexaba en el buscador.

En 1990 Google™ desarrolla un algoritmo que le permite generar mejores resultados de búsqueda que sus competidores. El algoritmo denominado MapReduce que en propias palabras de Google™ es *"MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper."* [9] paraleliza la búsqueda de datos y permite entregar búsquedas en conjuntos enormes de páginas de Internet basándose en la división del trabajo.

Google™ se convierte así en el gigante de los buscadores por su capacidad de procesamiento de volumen de datos pudiendo analizar millones de páginas en fracciones de segundo.

Google™ es una de las compañías que comienzan a utilizar datos para generar más datos utilizando las entradas de su buscador para optimizar las búsquedas subsecuentes y analizando la importancia de las páginas a través del algoritmo Google's Page Rank. Este algoritmo se usa para identificar cuantas veces es referenciada determinada página a través los links que hay de ella en Internet y con ello poder considerar una página relevante o no y otorgar mejores resultados de búsqueda. De esta forma Google™ utiliza metadatos para generar más datos.

Al igual que en el caso de "Búsqueda a escala", los problemas y retos principales de los sistemas de datos van apareciendo por el uso que se le desea dar a los mismos datos. Almacenamiento y explotación de contenido multimedia, análisis de sentimientos, enriquecimiento y contextualización de datos, descubrimiento y exploración analítica son algunos de los problemas que van apareciendo y de las oportunidades que se detectan con las grandes capacidades de los sistemas.

[9] Dean J &Ghemawat S. (2004). MapReduce: Simplified Data Processing on Large Clusters. Enero 4, 2015, de Google Sitio web: http://research.google.com/archive/mapreduce.html

Con el Internet 2.0 comienza una gran explosión de datos en la generación de contenido multimedia. Para el usuario final la fotografía o el video contiene más información de la que se pueda pensar o identificar del propio archivo, como puede ser en dónde se tomó la fotografía, quién o quiénes se encuentran en ella, cuándo se tomó, si fue una ocasión especial, etc. Aun cuando muchos de los metadatos son agregados manualmente por el usuario final existen intensos esfuerzos en desarrollar tecnología de procesamiento y reconocimiento de imágenes que no esté basada en estos; tal es el caso de CBIR (Content-Based Image Retrieval) que permite realizar búsquedas automáticas en almacenes multimedia a través de una imagen de ejemplo[10]. Se puede probar esta funcionalidad en el propio buscador de Google a través de la opción buscar por imagen.

Las estrategias de marketing realizan a través de la tecnología de Big Data análisis de sentimientos para poder identificar el posicionamiento de determinada marca en el gusto del consumidor. Para realizar este tipo de análisis es necesario poder manejar una gran cantidad de datos semi estructurados (tal es el caso de las redes sociales) y poder obtener resultados de ellos y así monitorear la percepción, analizar los resultados del uso de un determinado producto o servicio, identificar problemas de calidad o precio y obtener nuevas oportunidades de negocio a través de las nuevas demandas generadas.

El enriquecimiento de datos se realiza principalmente para contextualizar la información y poder hacer mezclas de orígenes de datos. Por ejemplo, puede ser útil para el departamento de seguridad de algún país relacionar el expediente criminal de una persona con sus redes sociales, las redes de sus conocidos y a través de búsqueda semántica identificar potenciales asociaciones delictivas.

Los principales problemas de datos de la actualidad se centran en cómo utilizar la información y en saber cuáles ventajas se obtienen de correlacionarla, sin importar si la información pertenece a la empresa o está disponible a través de servicios de datos externos.

Los servicios de datos externos aparecen en este paradigma de Big Data donde es claramente identificable que poseer datos y proveer datos se convierte en una herramienta poderosa y potencialmente en un negocio.

Para poder generar este contenido y este análisis sobre tecnología de Big Data, es necesario que las personas cuenten con capacidades y habilidades especiales sobre el tratamiento de datos. A la capacidad de generar información a través del análisis y tratamiento de datos se le denomina “Data Science”[11]. Esta disciplina está tomando una gran relevancia en la actualidad dado que existe la tecnología para analizar grandes orígenes de datos simultáneamente.

[10] Datta R, Li J & Z. Wang J. (2005). Content-Based Image Retrieval - Approaches and Trends of the New Age. Diciembre 29, 2014, de Stanford Sitio web: http://infolab.stanford.edu/~wangz/project/imsearch/review/ACM05/datta.pdf

[11] New York University. (2013). What is Data Science?. Enero 3, 2015, de New York University Sitio web: http://datascience.nyu.edu/what-is-data-science/

Utilizar datos para corregir problemas y generar nuevos productos de datos son tareas que han sido desarrolladas por empresas como Google, Apple, Samsung entre otras. Motores de reconocimiento de comandos de voz como Google OK™, Siri™ y SVoice™ han sido refinados y alimentados durante años a través de las propias búsquedas de voz de los usuarios, convirtiendo entradas de datos de los usuarios en datos para un nuevo producto. El mismo caso surge al poder recoger traducciones y sugerencias de mejores traducciones para Google Translate™, Bing Translator™, etc.

Debido al crecimiento de Internet y a la constante evolución del hardware, el flujo de datos en la red fue incrementándose; tantos orígenes de datos disponibles logran que a través de mezclas de estos datos se puedan obtener nuevos productos.

En términos simples **"Data Science", es la extracción de conocimiento de los datos crudos abarcando toda fuente de datos disponible propia o externa**. Un ejemplo de la aplicación de Data Sicence es un análisis de ejecuciones hipotecarias ocurridas en Philadelphia. Para ello se tomó un reporte público emitido de la oficina del Sherif y se cruzó a través del lenguaje de programación "R" con un origen de datos de Yahoo™ para convertir las direcciones en latitudes y longitudes. Luego colocando estas coordenadas en un mapa y agrupando por vecindario se es capaz de graficar y calcular que vecindarios tienen mayores problemas con el pago de hipotecas, el promedio de ejecuciones per cápita por vecindario y fue posible relacionarlo con otros factores socioeconómicos[12].

En la actualidad se puede obtener datos de la visita de un usuario a una página de Internet cuando este teclea los datos en un formulario. Se puede obtener muchos más datos, aun cuando el usuario no proporciona todos ellos de manera consiente. Por ejemplo un usuario de servicios on-line se levanta a hacer ejercicio y establece su rutina de ejercicio a través de una aplicación en su teléfono inteligente. Después de su rutina de ejercicio se pueden recopilar los siguientes datos.

- Hora del día en que hace ejercicio
- Frecuencia de ejercicio en la semana
- Ritmo cardiaco
- Calorías consumidas
- Preferencias musicales
- Localización en un GPS

El usuario generó datos de manera no-consiente ya que a través de sensores también se capturan, almacenan y analizan datos.

El ciclo de vida de los datos tendría 3 etapas.[13]

- ¿De dónde vienen los datos?

[12] Loukides M. (2011). Big Data Now. California USA: O'Reilly Media, p.4.
[13] Loukides M. (2011). Big Data Now. California USA: O'Reilly Media, p.4.

- ¿Cómo se utilizan?
- ¿A dónde van?

La primera pregunta es fácil de resolver; de cualquier parte sería una respuesta adecuada. Aun cuando la mayoría de los datos que se procesan para llegar a la escala actual vienen a consecuencia de Internet 2.0, actualmente los datos vienen de todas partes. A través de sensores, dispositivos inteligentes o sistemas de cómputo, los datos vienen de las visitas a páginas, transacciones bancarias, compras en línea, compras físicas, rutas de taxis, información de los GPS, o del propio cuerpo.

Con el incremento de sensores en los dispositivos que usamos día a día, aparece el término Internet of Things IoT. Sensores que pueden recopilar datos y a través de conectividad en la red pueden reportarlos a sistemas que los almacenan y los analizan generando con ello más datos.

Las respuestas para las otras dos preguntas es lo que Data Science resuelve. El análisis y la generación de nuevos datos dependen de la persona que realiza ese análisis y del uso particular que se le quiera dar. Por ejemplo una fuente de datos que convierte las direcciones en latitudes y longitudes puede ser usada para un servicio de localización de rutas a través de GPS o para localizar puntos de centros de distribución en un mapa. Aun cuando el origen de datos es el mismo la mezcla y el fin son completamente distintos.

La evolución de los problemas relacionados al aprovechamiento y uso de los datos fue un desencadenante para que Big Data apareciera, pero en términos tecnológicos ¿Cuál es la respuesta a estos problemas? La distribución del trabajo y la programación en paralelo atacan esta problemática. Más tarde el algoritmo MapReduce aparece como una respuesta alterna.

El desarrollo tecnológico logró que la capacidad de almacenamiento se incrementara drásticamente; desde sistemas de información y manejos de datos que podían almacenar 8MB hasta llegar a discos duros de bolsillo que soportan capacidades expresadas en Terabytes. Aunque el desarrollo tecnológico no siempre es homogéneo.

Para el año de 1990 un disco duro tenía una capacidad promedio de almacenamiento de 1370 MB y una velocidad de lectura de aproximadamente 5 mb/s[14], el almacenamiento era aún limitado pero el disco podría ser leído en su totalidad en aproximadamente 5 minutos. Para el año 2010 existían en el mercado discos de 1 Terabyte de capacidad y con una velocidad de transferencia de aproximadamente 100 mb/s. La capacidad de almacenamiento había crecido 750 veces mientras que la de transferencia apenas 20 y tomaba aproximadamente un poco más de 2 horas leer el disco completo.

Para poder aprovechar la capacidad de almacenamiento y solucionar el problema de la velocidad de lectura del disco, guardar el contenido en 2 discos dividiendo el mismo por la mitad reduce el

[14] Tomado de las especificaciones de Seagate ST-41600n SEAGATE TECHNOLOGY, INC. (1991). “ST-41600N (97501-16G) Elite-1 FH SCSI-2”. Extraída el 27/XI/2014 desde ftp://ftp.seagate.com/techsuppt/scsi/st41600n.txt

tiempo de lectura; llevar este tipo de almacenamiento a una gran cantidad de discos reduce el tiempo de lectura prácticamente a nada. Esta idea fundamental es en la que se basa el modelo de programación paralela MapReduce.

Google™ crea, utiliza y refina MapReduce para realizar búsquedas en largos almacenes de datos y potencializar su buscador. De esta manera, una búsqueda es recibida por un procesador central, el cual divide la búsqueda en segmentos más pequeños y delega las tareas de búsqueda a otros procesadores, así cada procesador realiza una búsqueda muy pequeña y regresa los resultados de manera inmediata, logrando con ello que “Google Search Engine™” encuentre millones de registros en fracciones de segundo.

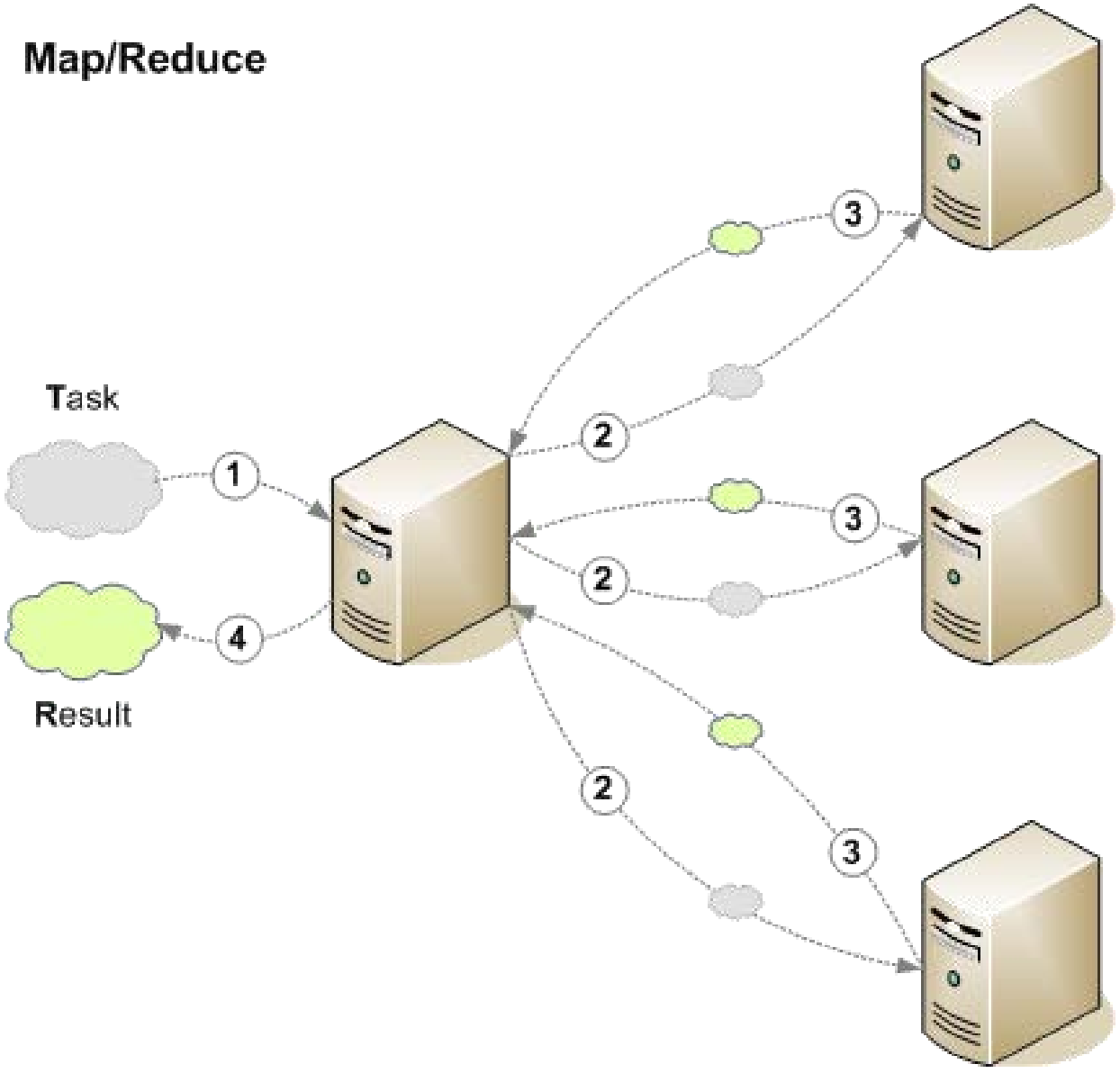

Fig. 3 Distribución del trabajo por MapReduce™

En la figura 3 se representa la distribución del trabajo a través de MapReduce de la siguiente manera

- El servidor maestro recibe la petición de la tarea (1)
- El servidor maestro realiza la división de la tarea y envía un bloque diferente con una porción de tarea a cada servidor secundario (2)
- Los servidores secundarios procesan el bloque que les corresponde y regresan el resultado al servidor maestro (3)
- El servidor maestro reduce el conjunto de resultados mezclando cada resultado que recibe de los servidores secundarios y envía una respuesta única.

De manera simultánea, estos sistemas de búsqueda de datos soportados por MapReduce pierden integridad momentáneamente, pero reducen drásticamente tareas de lectura e inclusive de escritura ya que se almacena el contenido para escribir en los discos de la misma manera en la que se lee. Debido a esto, es de esperarse que Big Data no remplace los sistemas tradicionales de bases de datos; por el contrario viene a ser un complemento de esta tecnología, que resuelve problemas de volumen, velocidad y variedad de datos y permite hacer análisis sobre los mismos de manera personalizada.

La siguiente tabla muestra una comparación entre sistemas tradicionales actuales de bases de datos relacionales y MapReduce

	RDBMS	*MapReduce*
Tamaño	Gigabytes	> Terabytes
Acceso	Interactivo y Batch	Batch
Actualización	Lectura y escritura múltiple	Lectura múltiple escritura única
Estructura	Estática	Dinámica
Integridad	Alta	Baja
Escalabilidad	Lineal	No lineal

Tabla 1 Comparación de características entre un sistema RDBMS y un sistema MapReduce

MapReduce realiza la división de la tarea en 2 fases. En la fase denominada Map la tarea es procesada en intervalos más pequeños de datos generando un conjunto de resultados por cada división. En la fase denominada Reduce los resultados son unificados y se obtienen únicos resultados.

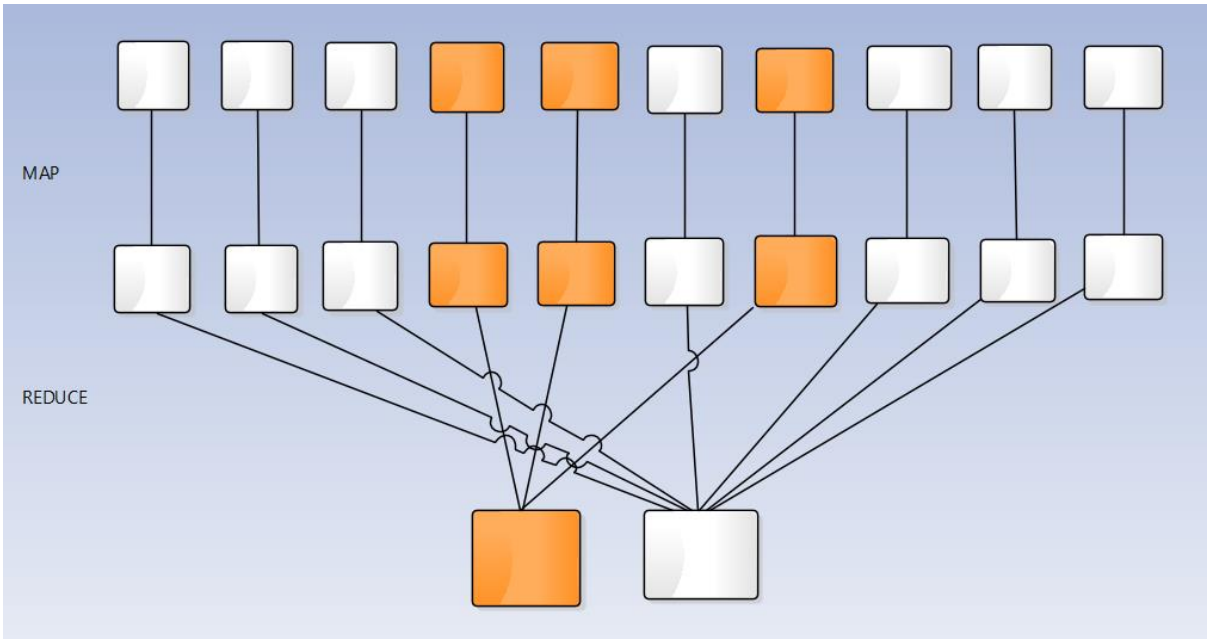

Fig. 4 Fases de MapReduce™

A través de HPC (High Performance Computing) ya se había trabajado con grandes volúmenes de datos y procesamiento en paralelo. Utilizando clústeres se dividía y procesaba el trabajo en cada nodo del clúster, accediendo a la lectura de datos en una SAN (Storage Area Network) común. Al enfrentarse a grandes almacenes de datos, el tiempo de lectura es grande. MapReduce por su parte guarda en cada nodo de procesamiento parte de la información del almacén de datos, lo que hace que el acceso sea inmediato y se diferencie del HPC en cuanto a tiempos de respuesta se refiere.

MapReduce es la primera implementación de un sistema de Big Data. Esta implementación creada por Google™ no fue del dominio público. Yahoo™ por su parte realiza una implementación abierta de MapReduce denominada Hadoop™ que más tarde pasaría a ser del dominio de Apache Software Foundation™, y se convertiría en la base para muchas de las implementaciones empresariales actuales pero es en MapReduce donde comenzó la idea.

Jugadores dentro de Big Data

A MapReduce lo siguió la idea de Yahoo™ de generar Hadoop™, sin embargo Big Data abarca múltiples tecnologías y empresas que han dedicado una considerable cantidad de recursos a este desarrollo.

Actualmente hay una enorme cantidad de empresas que se dedican a desarrollar tecnología sobre Big Data; como es de esperarse, hay una gran variedad de áreas a explotar para las nuevas tendencias tecnologías. Algunas de ellas son las siguientes:

- Big Data Frameworks:

  Dentro de las tecnologías base que soportan Big Data, la más extendida al momento es Hadoop™ la cual es desarrollada por Yahoo™ y Apache Software Fundation™. Es un Framework que permite el manejo de grandes almacenes de datos, pero, Hadoop™ no es la única implementación de MapReduce; existen otras implementaciones que, aunque son menos usadas, están cobrando fuerza poco a poco.

  En el área de almacenamiento eficiente Cassandra™ también desarrollada como proyecto por Apache Software Fundation™ se vislumbra como un gran competidor y HBase™ una implementación de código abierto de BigTable™ de Google. La cual se enfoca a mantener un almacenamiento de datos óptimo y eficaz.

- Analytics Infraestructure

  Para temas de infraestructura que soporte análisis de datos existen empresas que se dedican a ofrecer esta infraestructura como servicios a través de implementaciones de Hadoop™. Hortonworks™, Cloudera™, EMC²™ e Infobright™ ofrecen servicios de análisis sobre su infraestructura para tareas como análisis de datos de Internet of things.

- Operational Infraestructure

  En lo que se refiere a infraestructura operacional, almacenamiento y soporte a grandes volúmenes de datos, empresas como Teradata™, Terracotta™ y 10gen™ ofrecen estos servicios de administración de datos en memoria. Usualmente los sistemas de hardware utilizados para estas tareas son muy robustos, siendo un claro ejemplo SAP HANA™.

- Infraestructure As A Service

  La infraestructura como servicio es un poco más común, ya que muchas de las compañías

actuales que han decidido migrar sus aplicaciones a la nube. Amazon™ web services, Windows Azure™ de Microsoft™, Infochimps™ y Google BigQuery™ provén diversos servicios que van desde bases de datos, hasta Frameworks de integración de negocios como el caso de BizTalk™. Este tipo de infraestructura permite establecer sistemas en cloud o híbridos que aprovechan muchas veces las capacidades de Big Data

- Structured Databases

  ORACLE™, MySQL™, SQL Server™, Sysbase™, PostgreSQL™ son tecnologías de manejadores de bases de datos relacionales actuales. Estas, como se mencionó anteriormente, vienen a ser complementos de Big Data para tareas sobre datos estructurados.

- Analytics and Visualization

  La toma de decisiones en ambientes empresariales ha sido potencializada y en gran parte soportada por el análisis de datos. En este ámbito empresas como Tableau™, Opera™, Teradata™ o centrifuge™ permiten conexión a las Data Platforms para realizar análisis de datos y ayudan en la visualización de resultados.

- Business Intelligence

  El concepto de Business Intelligence aparece en los sistemas de datos al tiempo que las bases de datos OLAP hacen explosión. Análisis, cubos, dimensiones y minería de datos permiten explotar los Data-warehouses y proveer de información que genera inteligencia accionable para los negocios. Su utilidad es tan grande que al aparecer Big Data, grandes compañías generan productos de análisis de inteligencia de negocio que son capaces de explotar Data Platforms. Entre ellos están ORACLE Hyperion™, SAP™, BusinesObjects RJMetrics™, Microsoft Bussiness Intelligence™, IBM COGNOS™, etc.

- Data As A Service

  Generar Data Platforms se ha convertido en un negocio muy rentable. Poner servicios de datos disponibles para ser consumidos y mezclados por otras empresas es algo que no ha pasado desapercibido. Factual™, GNIP™, INRIX™, LEXISNEXIS™ y Kagle™ proporcionan este tipo de servicios que los Data Scientist utilizan en el análisis y la búsqueda de conocimiento.

Como lo muestra esta información, las compañías han descubierto en el Big Data un nuevo paradigma de negocio que ha puesto en marcha la carrera empresarial por generar los mejores servicios y seguir investigando sobre Big Data. Tal es así que Larry Page y Sergey Brin

fundadores de Google Inc. lo describe en su misión empresarial: “La misión de Google es organizar la información del mundo y lograr que sea útil y accesible para todo el mundo”[15]

[15] Tomado de la descripción general de la empresa Google “Descripción general de la empresa” (n. d.). Extraída el 4/VIII/2014 desde http://www.google.com/intl/es-419_mx/about/company/

CAPÍTULO III Arquitectura de los sistemas de Big-Data.

En capítulos anteriores se menciona acerca de los problemas que pueden ser resueltos con Big Data y se establece una premisa fundamental la cual indica que para considerar que un problema de datos puede ser resuelto con Big Data debe de tener al menos características de las 3 V's.

Analizar e identificar cuando un problema debe ser resuelto a través de Big Data, estando este problema está en la vida real y no en algún caso de estudio de un libro, es un poco más difícil. Especialistas de datos con preparación profesional se encargan de este tipo de tareas.

Cada aspecto del problema en sí mismo es resuelto por diferentes tecnologías, todas son parte de las capas arquitectónicas de Big Data. Problemas como la selección de datos, la selección del sistema de almacenamiento, la tecnología de análisis y reportes, la publicación de datos resultantes son condicionales para la selección de estas tecnologías.

Si se parte de la idea de que la selección de la tecnología correcta proviene de la naturaleza del problema, es importante revisar cuales son los tipos de problemas comunes que se encuentran en el ámbito de resolución para las tecnologías Big Data, analizando sus características y las necesidades a resolver, y sin necesariamente tomar en cuenta el entorno empresarial o de investigación en sí mismo.

En este capítulo se mencionan los componentes generales de la tecnología de Big Data y de los componentes lógicos que posee, con el objetivo de conjuntar los términos y darle claridad a lo que esta tecnología abarca.

Arquitectura de Big Data

La clasificación de un problema como candidato a ser resuelto por Big Data debería ser fácil, pero en la práctica no lo es. IBM™ proporciona una guía para ayudar en la clasificación de problemas y en la identificación de las soluciones adecuadas. Es claro que existen muchas empresas, servicios y productos que resuelven varios elementos de Big Data. Aquí se muestran cuáles son esos elementos para clarificar el entorno completo.

Un problema de Big Data típicamente tiene una o varias entradas de datos, una etapa de análisis y una salida. Representado a este nivel, se puede ver como cualquier proceso del área de TI. Sin embargo, las diferencias son enormes. Para un problema de datos que cumplan con características de las 3 V's o 4 como IBM lo sugiere, es posible clasificarlo por las características de sus datos de entrada.

1. Realizar un análisis (sentimental) de la aceptación de una marca con el objetivo de identificar:
    - Lealtad a la marca
    - Oportunidades o problemas con los productos de la marca
    - Nuevas oportunidades de negocio
    - Análisis de competidores
2. Generar un análisis de los periodos de sueño de un país para correlacionarlo con la aparición de enfermedades reportadas por el sistema de salud para identificar:
    - Patrones de sueño
    - Correlación con enfermedades cardiacas o mentales
    - Correlación con datos relativos al peso corporal del individuo.
3. Generar un análisis de las entradas generadas en el Sistema de Transporte Colectivo Metro, la densidad de personas por metro cuadrado reportadas por las cámaras dentro del metro, las horas del día para identificar:
    - Correlación de horarios y densidad
    - Toma de decisiones inmediata sobre número de trenes por línea
    - Predicción del uso del sistema durante eventos o imprevistos
    - Reportes de crecimiento y necesidades para la expansión y mejoramiento del servicio.

Los problemas descritos pueden clasificarse de diferentes formas a través de los datos que se utilizaran. Las clasificaciones que se pueden obtener de los datos son:

Datos	Clasificación
Provenientes de sistemas web como archivos logs	Generados por máquina
Provenientes de interacciones por los usuarios	Transaccionales
Provenientes de Facebook, Twitter, Instagram, etc	Web y Sociales
Provenientes de llamadas telefónicas	Generados por Humanos
Generados por sensores unidos al cuerpo.	Biométricos

Tabla 2 Clasificación de problemas por los datos que utilizan.

La clasificación de los datos proporciona más información de la que se muestra en la tabla. Es importante considerar la siguiente información para el ingreso de datos al sistema:

- Tipo de datos
- Tipo de formato
- Tipo de fuente
- Frecuencia de llegada de los datos
- Necesidades de almacenamiento de los datos
- Destino de los datos (Consumidores)
- Hardware disponible o necesario
- Tipo de análisis necesario
- Metodología de análisis.

Estos factores clave y el tipo de problema en específico, ayudan a determinar cuál es la arquitectura necesaria para un sistema de Big Data. Considérese acorde a IBM que la arquitectura lógica de un sistema de Big Data incluye las siguientes capas[16]:

[16] Mysore D, Khupat S & Jain S. (2013). Big data architecture and patterns, Part 3: Understanding the architectural layers of a big data solution. Agosto 3, 2014, de IBM developerWorks Sitio web: http://www.ibm.com/developerworks/library/bd-archpatterns3/index.html

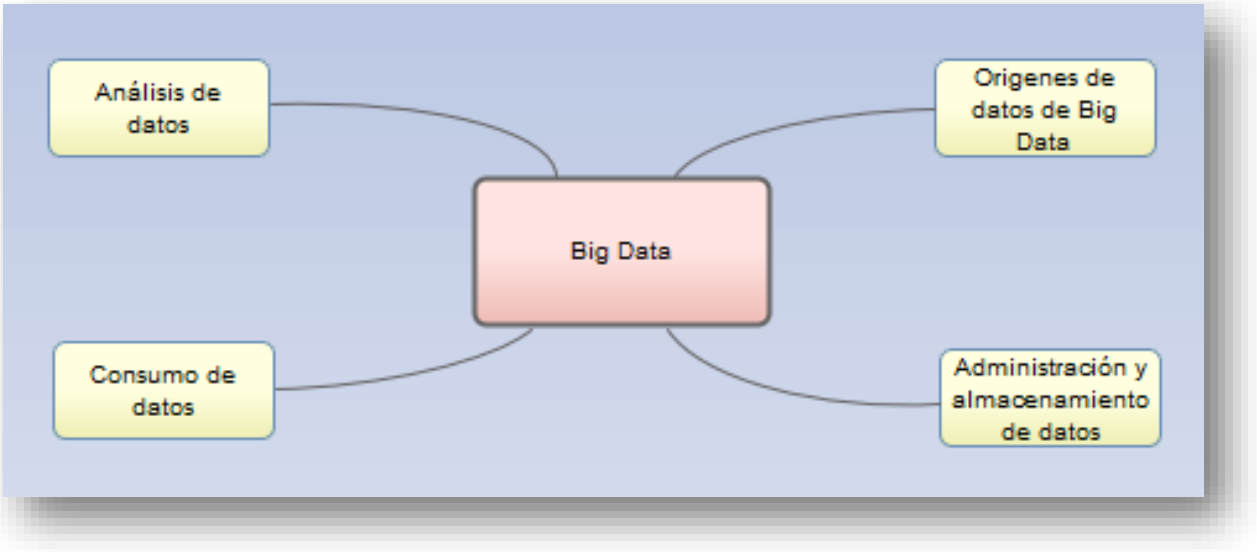

Fig. 5 Arquitectura lógica de Big Data

Orígenes de datos:

Esta capa engloba todas las entradas disponibles (propias o ajenas) para resolver el problema determinado. En este caso es importante considerar dentro de estas entradas aquellas que se ofrecen como servicio y que sean indispensables para la solución, teniendo en cuenta las clasificaciones anteriores.

Tipos de datos, frecuencia, fuente y formato son elementos considerados para clasificar la gran variedad de datos que se encuentran en este entorno: documentos de texto, imágenes, bases de datos relacionales, documentos de Microsoft Office™, páginas de Internet, blogs, redes sociales, mapas, archivos logs, etc. son solo algunos de los tipos de datos a considerar, adicionando los orígenes propios.

Administración y almacenamiento de datos:

El almacenamiento es un tema muy poco trivial en cuanto a Big Data refiere. El poder leer y escribir cantidades enormes de datos en tiempos muy cortos hace que una parte importante de la tecnología dedicada a Big Data se encargue de resolver este problema. Se han desarrollado para ello algoritmos muy elaborados, sistemas de archivos especiales e inclusive la distancia del Hardware que hay entre el almacenamiento y el procesador en cada nodo es algo a considerar.

La adquisición de los datos, el consumo de los mismos y la correcta selección del sistema de archivos a usar tienen un impacto significativo en el desempeño del sistema.

Análisis de datos:

MapReduce hace su aparición en el análisis de datos. Los problemas se descomponen en tareas que son interpretadas en funciones de mapeo y reducción, lo que logra que el análisis de los datos sea efectivo y rápido.

Se requieren analizar e identificar las entidades de datos con las que se manejarán y luego seleccionar la técnica adecuada de análisis. La selección correcta del patrón de análisis influirá definitivamente en el resultado, y es guiada por la propia naturaleza del problema. Machine Learning, NLP (Natural Lenguaje Processing), Predictive Model, Decision Management, Real Time Scored results, Statistical Model, Model execution y Model verification son técnicas que se pueden emplear en esta capa.[17]

Consumo de datos:

Esta capa se considera el resultado final del modelo arquitectónico. Un análisis de datos tiene diversas salidas como dashboards o informes para toma de decisiones, otro origen de datos para un nuevo análisis de Big Data o una nueva Data Platform. En la capa de consumo y en el análisis de problemas referentes a Big Data es considerado para quien está dirigido el resultado final; puede ser un solo usuario, un administrador u otro Data Science.

Las salidas también incluyen patrones necesarios que son parte del propio proceso de análisis. KPI Operacionales (Key Performance Indicator), Bussines Alerts o Self Service Análisis son salidas comunes del análisis ejecutado.

Tomando en cuenta el procesamiento y almacenamiento como punto central de Big Data y su arquitectura, es pertinente la explicación del modelo de programación MapReduce y el modelo de almacenamiento BigTable que sirvieron como base para la generación de las actuales herramientas.

[17] Mysore D, Khupat S & Jain S. (2013). Big data architecture and patterns, Part 3: Understanding the architectural layers of a big data solution. Agosto 3, 2014, de IBM developerWorks Sitio web: http://www.ibm.com/developerworks/library/bd-archpatterns3/index.html

Map Reduce™

En diciembre del 2004 Google publica un White Paper explicando un nuevo modelo de programación, diseñado como respuesta a un problema común experimentado por la compañía para procesar grandes volúmenes de datos el cual denomina MapReduce™.

Google buscaba una solución al problema, que hasta ese momento solventaba con procesamiento en paralelo, que fuera sencilla de entender y separada de la complejidad de la programación paralela; además que pudiera manejar tolerancia a fallos, balanceo de cargas y distribución de datos.

La idea de MapReduce está basada en 2 algoritmos de la programación del lenguaje LISP: Map y Reduce. Map genera una serie de resultados en pares tipo Key-Value a través de un par Key-Value original; “Reduce” reduce los valores juntando todos aquellos que se relacionan con la misma llave combinando los datos de manera apropiada. Este modelo de programación tiene diversas ventajas; entre ellas la de poder paralelizar una gran cantidad de trabajo, y la de reprocesar el trabajo de un solo nodo que utilizaba Map en caso de alguna falla detectada sin la necesidad de reprocesar todo el trabajo.

Por ejemplo, si se quisiera contar cuantas veces ocurre la palabra “manzana” dentro de un conjunto de páginas web. El algoritmo Map contará por cada página generando llaves intermedias de procesamiento de la siguiente forma (key, value) propiciando una salida intermedia generando una entrada para cada vez que encuentre una palabra:

(manzana,1) (manzana,1) (manzana,1)...

El algoritmo reduce tomará esta lista y sumará las veces que se haya encontrado la palabra manzana, obteniendo así el conteo total de las mismas en un determinado conjunto de páginas web.

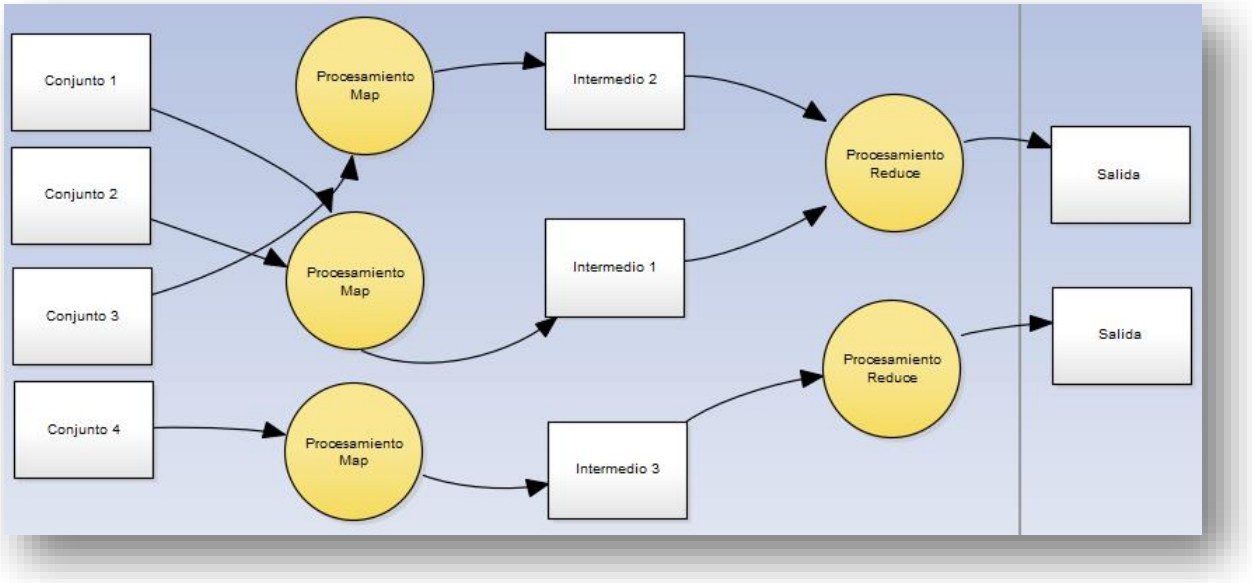

Fig. 6 Flujo de ejecución MapReduce™

Google explica en su White Paper[18] la forma de actuar del modelo de programación de la siguiente manera:

Al ejecutar el programa el modelo MapReduce™ ejecuta las siguientes tareas:

1. La entrada de datos se divide en splits típicamente de 16 o 64 MB controlados por el usuario programador. Después de la división, copias del programa son incluidas en cada uno de los nodos de procesamiento.
2. Una de las copias del programa es nombrada programa maestro, el cual es el programa que controlará la operación global y decidirá quién de los nodos será de tipo Map y quien de tipo Reduce. El resto de nodos se denominan trabajadores Map o trabajadores Reduce según su función. Los trabajadores Map, procesarán la tarea de selección definida, los trabajadores Reduce unificarán los resultados que los nodos Map entregan.
3. El trabajador que es asignado a una tarea Map lee el contenido que tiene asignado en su memoria procesando cada par Key-Value e introduciéndolo a través de la función Map del usuario. Esto genera un procesamiento de valores Key-Value intermedio que se almacena en memoria.
4. Periódicamente, cuando el buffer con el procesamiento intermedio está lleno, es escrito en disco, en la región determinada para ello. Existe una región específica de escritura para cada trabajador que se encargue de tareas Map. La localización es enviada al programa maestro quien la utiliza para enviarla a los trabajadores que se encargan de tareas Reduce.

[18] Dean J &Ghemawat S. (2004). MapReduce: Simplified Data Processing on Large Clusters. Enero 4, 2015, de Google Sitio web: http://research.google.com/archive/mapreduce.html

5. Cuando un trabajador Reduce es notificado por el programa maestro de una nueva escritura de datos, lee los datos intermedios y los ordena para que todas las llaves con un mismo valor aparezcan juntas. Esto reduce el tiempo de procesamiento.
6. El trabajador asignado a una tarea Reduce itera sobre las llaves ordenadas y pasa los valores a la función “reduce” del usuario. Esta función genera una salida útil. Puede ser posible y necesario reducir aún más las salidas de las primeras funciones.
7. Cuando las tareas de reducción acaban el programa maestro regresa el control al programa del usuario para continuar con el flujo estándar.

Cuando una trabajador de tipo Map o Reduce no responde por alguna falla derivada del propio equipo o de la red, el programa central puede reasignar su trabajo a otro nodo elegible; esto hace que MapReduce sea altamente tolerable a fallos y que el principal método de eliminación de fallos sea el reprocesamiento. Google indica que es posible hacer tolerable a fallos al programa maestro a través de checkpoints de ejecución; aunque en la práctica si falla el programa maestro, Google declara como fallido todo el procesamiento.

Existen a partir de aquí varias consideraciones de cómo trabaja MapReduce™ que sugieren interesantes discusiones como qué hacer con fallos parciales de hardware o cómo asegurar que el balanceo de cargas se utiliza de manera adecuada. En cualquier de los dos casos, un mal balanceo o una escritura lenta en disco por falla de hardware puede provocar que el tiempo de procesamiento aumente considerablemente. Para ello, MapReduce desarrolló un mecanismo de respaldo, en el cual, cuando ya no hay splits de datos por procesar y algunos trabajadores están tomando más tiempo del necesario, se hacen un procesamiento de respaldo en los trabajadores ociosos de manera que si falla o tarda un trabajador, se toma el resultado del trabajador de respaldo.

Como ejemplo, supóngase que se tiene flotando en el mar alrededor del mundo, boyas con sensores capases de identificar datos como la temperatura del agua, el valor de la velocidad del viento, la intensidad de lluvia, y que registran estos datos en ficheros con una estructura de la siguiente forma.

[no de boya] [fecha] [temperatura] [velocidad del viento] [volumen de lluvia]

Se han registrado los datos de las boyas alrededor del mundo por un periodo de 10 años y con una diferencia de 10 minutos de entre cada medición de cada bolla, la estructura de datos dentro de los archivos se encuentra de la siguiente manera:

34556 12-12-2007 4° 17km/h 44CM3
34009 12-12-2007 5° 12km/h 0CM3
34134 12-12-2007 3° 15km/h 0CM3
34297 12-12-2007 2° 4km/h 12CM3
34009 12-12-2007 2° 45km/h 17CM3
34308 12-12-2007 2° 12km/h 18CM3
34009 12-12-2007 4° 26km/h 0CM3

34987	12-12-2007	4°	12km/h	17CM3
34009	12-12-2007	5°	46km/h	0CM3
05987	13-12-2007	7°	12km/h	4CM3
08473	13-12-2007	1°	35km/h	66CM3
05987	13-12-2007	2°	12km/h	14CM3

Para calcular el valor promedio de temperatura de cada boya y determinar todas las lecturas obtenidas en el planeta durante el periodo descrito, se requiere una lectura de todos los archivos y un ordenamiento y selección de los datos importantes.

Al resolver el problema del cálculo de promedios utilizando el algoritmo MapReduce, cada nodo destinado a tareas Map, recibirá como entrada una porción de los archivos que almacenan esta información, y a partir de ella podrá generar una salida de pares Key-Value

Dado que el número de la boya es único en el mundo, se puede utilizar como parámetro key, el algoritmo Map identificará a la temperatura como el parámetro Value ignorando todos los demás datos del archivo y generando salidas como la siguiente lista.

(34556,4°)
(34009,5°)
(34134,3°)
(34297,2°)
(34009,2°)
(34308,2°)
(34009,4°)
(34987,4°)
(34009,5°)
(05987,7°)
(08473,1°)
(05987,2°)

Antes de pasar la salida a disco y permitir el comienzo de la tarea Reduce, suele requerirse procesamiento adicional para el ordenamiento de valores, este procesamiento conocido como Shuffle and Short, ordena la salida antes de pasarla al algoritmo Reduce para un procesamiento más eficiente. La salida después de ser ordenada tendrá el siguiente aspecto.

(05987, [7°,2°])
(08473, [1°])
(34009, [5°,2°,4°, 5°])
(34134, [3°])
(34297, [2°])
(34308, [2°])
(34556, [4°])

(34987, [4°])

De esta manera la función reduce obtiene como entrada un Valor key y un arreglos de parámetros Value para cada boya y genera el promedio de temperatura de cada una.

(05987, 4,5°)
(08473, 1°)
(34009, 4°)
(34134, 3°)
(34297, 2°)
(34308, 2°)
(34556, 4°)
(34987, 4°)

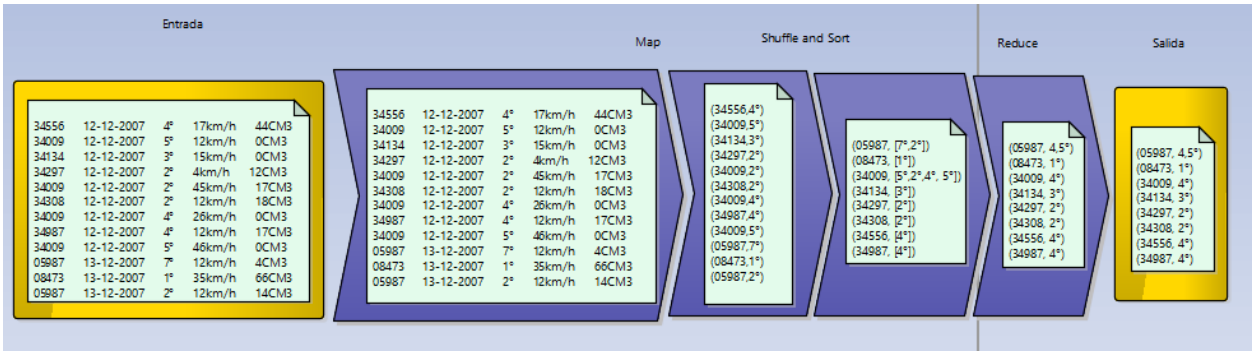

Fig. 7 Cálculo de promedios con MapReduce™

MaReduce cumple así con las tareas de procesamiento y/o análisis de datos dentro de Big Data, y deja al usuario programador la libertad de agregar código para que él sea el que defina cuál es la tarea a realizar.

BigTable™ como modelo de almacenamiento.

El problema de procesamiento de grandes volúmenes de datos fue resuelto a través de MapReduce™; pero aún se necesitaba una forma de almacenamiento que soportara la escritura y lectura. Para ello se generaron diferentes sistemas de almacenamiento basados en modelos hasta entonces desarrollados pero tomando solo las mejores características.

Uno de estos modelos es BigTable™. Este es un sistema de almacenamiento estructurado[19] y distribuido; diseñado para almacenar datos a gran escala. El objetivo de su desarrollo es poder almacenar datos del orden de Petabytes en clústeres o arreglos de computadora con la escalabilidad necesaria para soportar cualquier tamaño en almacenamiento.

Su creación y diseño, de la autoría de Google, fue basado en cumplir con las siguientes características:

- Ampliamente aplicable
- Altamente escalable
- Administración de alta disponibilidad (HA)
- Alto desempeño (HP)

El objetivo era contar con un sistema de almacenamiento que pudiera ser aplicado en múltiples proyectos. BigTable toma ideas del modelo binario relacional, sin embargo su modelo de almacenamiento se basa en un mapa multidimensional ordenado el cual se expresa de la siguiente manera:

El mapa consta de un índice único con una llave renglón, un contenido columna y un dato timestamp, expresados en un arreglo:

(renglón; string, columna string, tiemstamp; int)

El renglón permite almacenar el contenido de una llave única. Todas las operaciones de lectura y escritura se ejecutan dentro del mismo renglón. Para la lectura y búsqueda efectiva dentro de estos renglones, la llave se almacena en BigTable™ de manera ordenada lexicográficamente. El tamaño del renglón se le denomina Tablet.

Para las columnas se maneja un agrupamiento a través de familias. Normalmente todos los datos almacenados en un grupo de familias son del mismo tipo y no hay un número limitado de

[19] Es importante clarificar que el sistema de almacenamiento es estructurado, esto no quiere decir que los datos que almacena lo sean.

columnas por renglón; aunque sí es necesario saber cuántas y cuales familias se necesitan para guardar los datos.

Por ejemplo: para guardar los datos de una página web (la cual no es estructurada) se utiliza como renglón la URL, y como familias de columnas pueden existir las siguientes:

- Lenguaje:
- Contenido:
- Palabras clave:

Las familias permiten leer y hacer búsquedas sobre las tablas de manera casi inmediata, el ordenamiento lexicográfico hace que en el hardware, las familias de columnas o los renglones subsecuentes se encuentren almacenados en muy pocos discos y estos siempre van a estar unos cerca de otros.

La columna timestamp permite realizar escritura con diversas versiones del mismo contenido. Por ejemplo, si una página de Internet existe en más de una versión a través del tiempo; es posible obtener sus diversas versiones gracias a esta columna.

A diferencia del modelo binario relacional o modelo columnar, BigTable permite almacenar datos no estructurados, y mantener el ordenamiento y la cercanía de los datos para ejecutar consultas rápidas, mientras que en el binario relacional, el ordenamiento permite leer rápidamente solo datos estructurados, basados en la orientación a columnas en lugar de renglones como el modelo relacional simple.

Google utilizó este sistema de almacenamiento para potencializar sus búsquedas; y fueron ellos mismos quienes indicaron que es utilizado para almacenar cualquier tipo de datos no estructurados.

Un motor de búsqueda tiene dentro de sus tareas la de almacenar los sitios web, indexarlos y guardar contenido o metadatos de ellos. BigTable al ser un mapa multidimensional resuelve el almacenamiento de la siguiente manera.

Para cada página que almacene, la WebTable contendrá varias entradas y una llave para cada entrada. Las llaves lucen de la siguiente forma:

- com.yahoo
- com.microsoft.www
- com.google.www

Como cada sitio tiene diferente información que almacenar, en la tablet o renglón, se acomoda la información clasificándola en familias de columnas de la siguiente manera

com.yahoo {

links {

cine

finanzas

noticias

correo

La verdad de porque salvo su vida Fernando Alonzo

…

}

}

com.microsoft.www{

links {

tienda

soporte

…

}

}

com.google.www{

links {

Images

Gmail

…

}

}

Y dado que BigTable es un mapa multidimensional, se ordenan los links en pares de llave y valor de la siguiente manera

com.yahoo {

links {

com.yahoo : cine

com.yahoo : finanzas

com.yahoo : noticias

com.yahoo : correo

ar.com.tn : La verdad de porque salvo su vida Fernando Alonzo

…

}

}

com.microsoft.www{

links {

com.microsoftstore :

tienda

com.microsoft : soporte

…

}

}

com.google.www{

links {

mx.com.google : Images

mx.com.google: Gmail

…

}

}

Almacenando de esta manera la información, se visualiza que se agregarán mapas dentro de mapas tantos como el problema a resolver lo requiera.

En el ejemplo anterior se identifica que con ese ordenamiento, es posible hacer una búsqueda rápida de los links de una página a través del sitio al que apuntan.

BigTable™ cuenta con una API para poder acceder a operaciones de lectura y escritura tanto de los datos dentro de la Tablet como para la creación y eliminación de renglones y columnas. Actualmente es posible almacenar por usuarios finales contenido en BigTable™ a través de una subscripción a un servicio.

En el desarrollo de Big Data, BigTable es la idea base para desarrollar el HDFS (Hadoop™ Data File System) el cual es el componente de almacenamiento de los sistemas basados en Hadoop™

Hardware de alto rendimiento.

Cuando se diseñó MapReduce, se hizo pensando en el uso de computadoras que para ese tiempo eran consideradas robustas. Procesadores dual core y computadoras de 2 y 4 Mb en RAM fueron utilizadas para generar las pruebas de MapReduce. En la actualidad se encuentran supercomputadoras dedicadas a este tipo de tareas que manejan memoria RAM con base en Terabytes.

El almacenamiento y la lectura siempre ha sido un tema a tratar para computación de alto rendimiento. Aun cuando las velocidades de transferencia y escritura a un disco duro tradicional han aumentado, para computación en tiempo real o procesamiento de datos del tipo que hace Big Data, la lectura en discos tradicionales para algunos escenarios no es la solución.

Adicional a la velocidad; la durabilidad de los dispositivos de almacenamiento no volátil ha generado que se cambie la forma de almacenamiento de datos; los discos de estado sólido tratan de resolver este problema permitiendo acceso a múltiples ubicaciones de memoria en una misma operación y ha mejorado la velocidad de lectura hasta 100 veces más rápido.

La evolución de los sistemas de bases de datos generó un nuevo paradigma en el cual, las bases de datos de alta disponibilidad y tiempo real requerían de lectura y acceso superior a los discos de estado sólido. Así aparecen las “bases de datos in-memory” las cuales están cargadas completamente en memoria RAM, lo que hace la velocidad de lectura 1,000 veces más rápido que utilizando discos de estado sólido.

Han aparecido en el mercado nuevos productos que manejan parcial o totalmente bases de datos en memoria permitiendo que el tiempo real sea una realidad y manteniendo tablas en memoria para poder actualizar de bases de datos OLTP a OLAP en el momento que se hace una transacción.

Para procesamientos complejos donde se registran millones de transacciones y solo se requiere registrar variaciones significativas, las bases de datos en memoria permiten recuperar todos estos movimientos y registrar a memoria no volátil los movimientos considerados significativos.

La diferencia fundamental sobre usar un sistema in-memory o uno de Big Data radica en las necesidades de la problemática a resolver. Si el sistema tiene requerimientos fuertes sobre coherencia de datos y ocupa procesamiento avanzado en transacciones las soluciones in-memory o parcialmente in-memory son más adecuadas, seleccionado una u otra dependiendo de la demanda de datos y los escenarios de tolerancia a fallos requerido.

Oracle Big Data Appliance™ y SAP HANA™ entre otros han generado soluciones de Big Data poniendo un énfasis significativo en el Hardware de alto rendimiento. Estas dos soluciones ofrecen computación “in-memory” por lo que soportan procesamiento de múltiples procesos relacionales y por supuesto tienen su versión de Big Data basada alguna de las versiones de

Hadoop ya sea Apache Hadoop™ o versiones de Cloudera™. Ahondaremos en estas versiones en los siguientes capítulos.

Retos de Big Data.

La evolución de las tecnologías de Bases de Datos, y la aparición del Big Data revolucionaron también la manera en que las empresas ven a los datos. Han aparecido nuevos modelos de negocio que permiten ofrecer servicios de orígenes de datos o de procesamiento para grandes volúmenes de información.

En la actualidad, empresas como Facebook, Twitter, Instagram, etc almacenan una gran cantidad de contenido. Este contenido es de importancia para múltiples sectores, ya que es contenido que es generado por los usuarios finales; es decir, el mercado meta de muchas empresas.

La computación en la nube, también es una tendencia que los usuarios de servicios están adoptando. Ya sea para la disminución de costos, para aumentar la movilidad o para incrementar la seguridad y controlar el acceso a la información. Empresas como Windows o Google han llevado incluso las aplicaciones de escritorio a un entono cloud en donde la creación, edición y almacenamiento de contenido queda completamente del lado del servidor y permiten movilidad y real portabilidad entre dispositivos.

Con estas tendencias, la cantidad de información que se genera, almacena y administra se incrementa a un ritmo acelerado, por lo que los usuarios tienen o han tenido relación con tecnologías de Big Data.

Sin embargo los propios usuarios de Big Data perciben o identifican el concepto acorde a las necesidades que para ellos resuelven[20]. Es esta misma percepción la que cambia cuando se habla de los objetivos que debería tener Big Data en el futuro.

De acuerdo al estudio “Analitycs: The real-world use of big data” realizado por IBM en colaboración con la Universidad de Oxford, los principales objetivos que debe tener la tecnología Big Data son:

- Generar resultados orientados al cliente
- Llevar a cabo una optimización operativa
- Hacer una efectiva gestión financiera o gestión de riesgos.

Estos objetivos se fundamentan en las necesidades reales de los usuarios finales de Big Data. Aunque por otra parte suelen ser retos que se cumplen en la actualidad, y el correcto cumplimiento depende del análisis de la problemática a resolver.

[20] Schroeck M, Shockley R, Smart J, Romero D & Tufano P. (ND). Analytics: The real-world use of big data. Febrero 3, 2016, de IBM Global Business Services & University Of Oxford Sitio web: https://www.ibm.com/smarterplanet/global/files/se__sv_se__intelligence__Analytics_-_The_real-world_use_of_big_data.pdf

Cada usuario visualiza o entiende la tecnología Big Data acorde a las necesidades de resolución de problemas que vive. La más común definición de Big Data es la de una tecnología que maneja una gran cantidad de información[21], aunque no es la única percepción; entre estas percepciones están.

- Nuevos tipos de análisis de datos
- Información en Tiempo Real,
- Flujos de datos provenientes de nuevas tecnologías
- Tipos de medios no tradicionales
- Grandes almacenes de datos
- Solo un nuevo término tecnológico popular
- Datos sociales.

La gran diversidad de puntos de vista refleja solo necesidades que ya cumple Big Data al momento, por lo que los retos de Big Data deben encontrarse en una visión diferente.

Para los profesionales de TI que manejan sistemas de alto desempeño, grandes volúmenes de datos, sistemas en tiempo real, o que enfrentan problemas de la naturaleza de Big Data, la adopción de la tecnología y el entendimiento de las capacidades de la misma no es un problema trivial. Hay muchas nuevas tecnologías que atacan múltiples problemas, y una correcta selección de las mismas se convierte en una decisión que requiere mucho análisis.

Para Nicolas Demassieux vicepresidente de Orange Labs Research la tecnología de Big Data tiene 6 grandes retos a considerar para los próximos años[22]. Los retos se basan en mejorar la usabilidad y potenciar las capacidades del uso de esta tecnología, los retos que Nicolas describe son los siguientes:

1. Big Algebra

   La idea se basa en la problemática que enfrenta un Data Scientist para la resolución de un problema. El experto en datos recibe el problema, lo analiza, limpia y elige de manera adecuada los datos a ocupar para la resolución del mismo, genera avanzados algoritmos de análisis, y modelos de interpretación de datos.

   Es una tarea muy compleja y no existe un lenguaje o una algebra definida para poder simplificarla. En este momento se depende mucho de la expertise del Data Scientist, y de las selecciones que él haga sobre la tecnología.

   La pregunta que Nicolas deja sobre la mesa es “In order to note, share or reason about Big Data operations, should we not invent a suitable algebra notation? In order to program

[21] Schroeck M, Shockley R, Smart J, Romero D & Tufano P. (ND). Analytics: The real-world use of big data. Febrero 3, 2016, de IBM Global Business Services & University Of Oxford Sitio web: https://www.ibm.com/smarterplanet/global/files/se__sv_se__intelligence__Analytics_-_The_real-world_use_of_big_data.pdf

[22] Demassieux N. (2015). Six challenges for Big Data. Marzo 3, 2016, de Orange Research Sitio web: https://recherche.orange.com/en/six-challenges-for-big-data/

a Big Data processing chain, should we not develop a suitable programming language?[23]"

2. Big Noise

El ruido es la referencia a todo aquello que no necesitamos, o que está como no lo necesitamos. Cuando se habla de información en los volúmenes que se manejan para Big Data, la cantidad relevante de información para la resolución del problema puede estar en conjunto con una cantidad más grande de información, pero irrelevante para la solución.

Adicional a esto, la generación de datos dinámicos o metadatos puede ser alterada o ser irrelevante si se hace de manera incorrecta, así, más datos no siempre significa más información y en lugar de mejorar el escenario de resolución de problema puede estarlo empeorando.

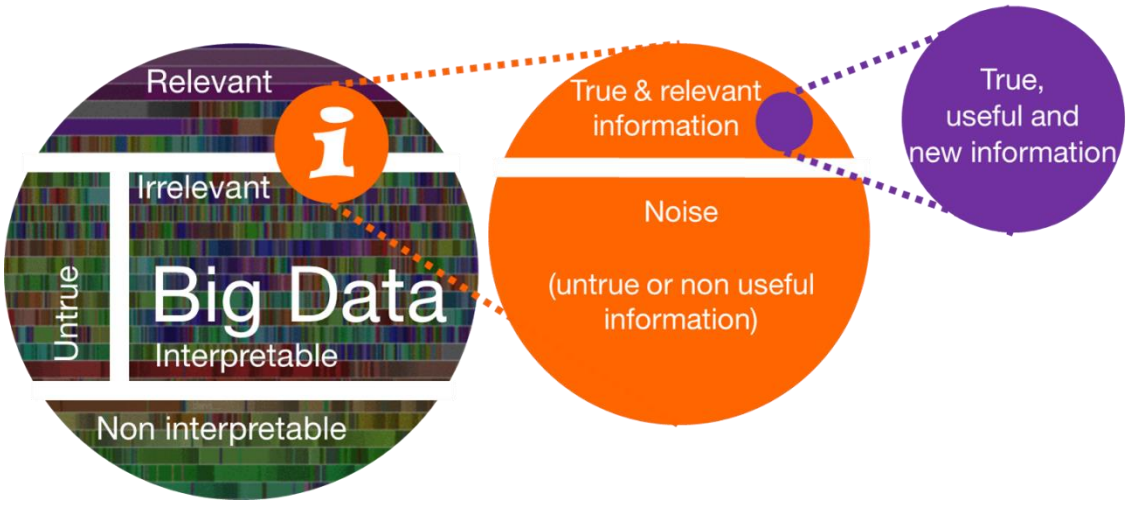

Fig. 8 Ruido en la información[24]

La eliminación de ruido de datos, o del ruido en los procesos de generación de datos o metadatos es un reto que Big Data debe afrontar.

3. Big Time

Cuando se habla de correlación de datos, aun en entornos de mucho volumen y alto poder de procesamiento, la correlación a través del tiempo no es explotada como debiera.

Correlacionar eventos actuales con eventos que ocurrieron hace años e identificar consecuencias, o predecir eventos futuros a partir de la información actual es un reto que aun en Big Data existe.

[23] Demassieux N. (2015). Six challenges for Big Data. Marzo 3, 2016, de Orange Research Sitio web: https://recherche.orange.com/en/six-challenges-for-big-data/
[24] Demassieux N. (2015). Six challenges for Big Data. Marzo 3, 2016, de Orange Research Sitio web: https://recherche.orange.com/en/six-challenges-for-big-data/

Aumentar la variable tiempo al volumen de datos actual para sistemas de Big Data hará que el volumen crezca desmesuradamente, por lo que para agregar datos correlaciónales a través del tiempo es necesario considerar que datos son correlaciónales y que datos deben de ser descartados.

4. Big Structures

   Al comenzar a adoptar la tecnología de Big Data, se utilizan sus capacidades para procesar largos volúmenes de datos estructurados. Conforme se va avanzando, se comienzan a aprovechar las técnicas de explotación para datos no estructurados como pueden ser datos sociales, imágenes, audio, video etc.

   Estos datos se encuentran presentes y han aparecido con forme la evolución tecnológica ha resuelto nuevas necesidades humanas.

   Dado la reciente incorporación de IoT al mercado, la aparición de nuevos orígenes de datos y el incremento en volumen trae consigo nuevas necesidades de almacenamiento y estructuración. Esto en un futuro traerá la generación de nuevas estructuras complejas de información que deberán ser incorporadas y manejadas por las tecnologías de Big Data para aprovechar la explotación de esta información.

   En la actualidad, aun cuando ya se procesan estructuras complejas como sonidos e imágenes o flujos no estructurados, hay limitantes para el análisis y la búsqueda o generación de información, entre otros el procesamiento de imágenes aún no está completamente desarrollado. Así es posible pensar que para las nuevas estructuras de datos que vayan apareciendo Big Data tendrá el reto de procesarlas y poderlas integrar en el mar de información actual.

5. Big Reality

   Para este reto de Big Data, la premisa que Nicolas expone es, que dado que podemos colectar una gran cantidad de datos, procesarlos e interpretar resultados a partir de ellos, el entendimiento de fenómenos físicos o sociales, o la descripción de la realidad puede quedar en términos de los datos que colectamos.

   Si esto es cierto, se corre el riesgo de definir la realidad a partir de los datos que leemos, sin embargo es necesario considerar que pasa con esa parte de la realidad que genera datos los cuales no es posible medir, colectar o analizar, obviarlos mostraría una realidad incompleta o limitada.

6. Big Human

   Si las capacidades de análisis y almacenamiento de datos mejoran como parte de la evolución tecnológica, la capacidad de interpretación de datos masivos por parte de los humanos será insuficiente, y se estaría subutilizando la tecnología. Nicolas indica como uno se los retos la necesidad de cambiar nuestra forma de aprender y de interpretar como un proceso normal de evolución.

Si bien la variedad de retos que propone Nicolas son la idea de la evolución tanto tecnológica como humana, la capacidad de incrementar el procesamiento de datos traerá consigo retos de

múltiples tipos, incluyendo aquellos que se deriven de los descubrimientos o soluciones de los problemas que antes no podían resolverse, y que no sean necesariamente técnicos.

Para profundizar en el entendimiento de las tecnologías involucradas en Big Data, y en las diferencias entre ellas, es necesario explorar más a fondo alguno de los representantes de las propias tecnologías. Dado el gran número de tecnologías existentes en el mercado es pertinente tener una selección de ellas que permitan explorar diferentes ángulos de Big Data.

Acorde al cuadrante mágico de Garthner para soluciones de almacenamiento y administración de datos y para soluciones de análisis, los líderes de la industria durante el 2015 fueron Oracle, Teradata, Microsoft, IBM y SAP.

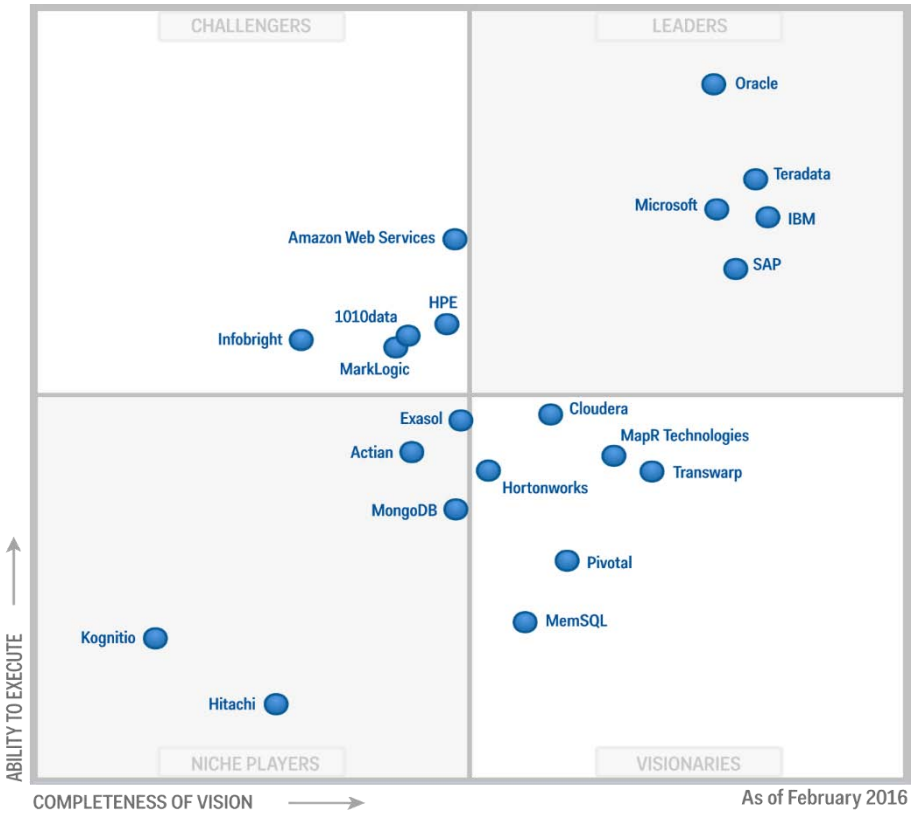

Fig. 9 Cuadrante mágico de Garthner para soluciones de almacenamiento y administración de datos y para soluciones de análisis año 2015 [25]

Oracle provee una serie de productos de manejo de datos. Oracle Big Data Appliance es su representante para manejo de escenarios de Big Data, el cual incluye componentes para bases de datos tradicionales, bases de datos en cloud y administradores de bases y conectores.

[25] Edjlali R & A. Beyer M . (2016). Magic Quadrant for Data Warehouse and Data Management Solutions for Analytics. Marzo 7 2016, de Gartner Sitio web: https://www.gartner.com/doc/reprints?id=1-2SGJMI2&ct=151118&st=sb

Al ser un appliance, los clientes de Oracle aplauden la velocidad en que se puede implementar o adoptar esta tecnología. Hablan también muy bien de una excelente estabilidad y desempeño como herramienta

Una de las características importantes de la herramienta, es su capacidad de integrar bases de datos nativas Oracle y poder trabajar en combinación con cualquier versión de Hadoop.

La razón de la selección de Oracle, es su posición dentro del cuadrante como tecnología líder en Big Data, y sus características de integración.

SAP por su parte ofrece su propia base orientada a columnas y una plataforma sumamente interesante de cómputo de alto rendimiento denominada SAP Hana.

SAP se encuentra también dentro de los líderes de Big Data, pero debajo de Teradata, Microsoft e IBM.

En este caso es conveniente seleccionar SAP Hana por la histórica participación que tiene en el desarrollo de cómputo de alto rendimiento y las bases de datos in-memory. SAP Hana integra poderosas capacidades de procesamiento potenciando aún más las posibilidades de Big Data.

Dado que Big Data es una tecnología que no solo incluye almacenamiento y análisis, sino también soluciones de Bussiness Inteligence, analizar el cuadrante Mágico de Gartner para BI y plataformas de almacenamiento

Este cuadrante muestra entre los líderes de la industria a empresas como Tableau, Microsoft, SAP, Oracle, IBM, SAS y MicroStrategy; la selección anterior, comparada en este cuadrante sigue mostrando a SAP y a Oracle como líderes.

Tanto SAP como Oracle son productos provenientes de la iniciativa privada, para incluir un producto que sea de código abierto y accesible para un mayor número de usuarios, se puede centrar la atención en la base sobre la que se construyó la tecnología Big Data, e indicar que es una correcta selección.

Es por esta razón que el estudio directo de Hadoop es conveniente para el entendimiento de los elementos sobre los cuales se desarrolla la tecnología.

Fig. 10 Cuadrante mágico de Garthner para soluciones de BI y plataformas de almacenamiento año 2015[26]

En los capítulos siguientes entraremos en profundidad en las características de esta tecnología, representada por estos 3 grandes competidores.

[26] Edjlali R & A. Beyer M . (2016). Magic Quadrant for Business Intelligence and Analytics Platforms Septiembre 08 2016, de Gartner Sitio web: http://www.iconresources.com/wp-content/uploads/2013/05/Magic-Quadrant-for-Business-Intelligence-and-Analytics-Platforms-2015.pdf

CAPÍTULO IV SAP HANA™.

Derivado de la competencia tecnológica y por la innovación que grandes empresas llevan a cabo; múltiples productos relacionados a Big Data aparecieron en el mercado. Se ha mencionado en capítulos previos la gran diversidad de empresas que trabaja sobre la no menos diversa arquitectura tecnológica; generando soluciones para cada una de las capas lógicas de Big Data.

No todas las soluciones de Big Data aparecen en el mercado por empresas que inicialmente se dediquen al procesamiento de grandes volúmenes de datos. Algunos como en el caso de la plataforma que se explica en este capítulo, vienen diseñadas por la necesidad de resolver problemas de sistemas existentes. Este es el caso de SAP HANA™.

SAP es una empresa multinacional que ha invertido millones de dólares en el desarrollo de soluciones empresariales. SAP HANA™ es una solución que aparece en el mercado como una plataforma inicialmente no diseñada para Big Data; pero que gracias a sus poderosas características de hardware tiene la capacidad de desarrollar con muy buen desempeño las tareas de procesamiento de este tipo.

SAP (Systeme, Anwendungen und Produkte in der Datenverarbeitung)[27] es una empresa fundada en el año de 1971 dedicada a generar productos de procesamiento de datos; centra su mercado en industrias de proceso, discretas, de consumo, empresas de servicios, servicios financieros y servicios públicos entre otras.

Dentro de los productos desarrollados por SAP a través del tiempo, existe uno muy popular, y que ha sido la base para darse a conocer como una empresa importante para el mundo empresarial. HCM (human capital management) fue durante mucho tiempo su producto insignia, el cual le permitió ayudar a grandes corporativos con la administración de los procesos de recursos humanos y nómina; sus altos estándares de desarrollo permitieron que SAP se presentara como un líder de la industria.

La propia popularidad de sus productos y su fama, hizo que empresas a nivel mundial utilizaran los productos, bases de datos de miles de personas se almacenaban en servidores que manejaban HCM, y la demanda de nuevas funcionalidades provocó la evolución del producto.

Es debido a este tipo de evoluciones que SAP entra al mercado de cómputo de alto rendimiento, genera proyectos que le ayuden a resolver problemas fundamentales de sus productos y con la evolución en paralelo de Big Data aparece en el mercado SAP HANA.

[27] Sistemas Aplicaciones y Productos en Procesamiento de Datos

SAP HANA™ es actualmente un appliance; pero es importante remarcar que su desarrollo fue significativo para los paradigmas de bases de datos “in-memory” y que constituye un gran avance para el cómputo de alto desempeño. .

SAP HANA™

Para SAP es importante la modificación de las reglas de juego en cuestiones tecnológicas. A través del tiempo esta empresa ha trabajado y desarrollado múltiples productos que ayudan a la administración empresarial en tareas de inteligencia de negocio.

HCM SAP™ (Human Capital Management) es uno de sus más conocidos productos y por el que muchas empresas o profesionales de TI reconocen esta compañía, sin embargo cuando no es el único producto sobre el que SAP trabaja.

Como la mayoría de los nuevos productos tecnológicos, SAP ha creado nuevas plataformas a partir de la resolución de problemas comunes; uno de estos productos es SAP HANA™. SAP ha trabajado constantemente con bases de datos de gran tamaño, que crecen cuando sus productos ofrecen nuevas características. Esto aunado a la gran popularidad que han tenido sus productos, ha incrementado sustancialmente su interés en un óptimo desempeño de las transacciones de sus sistemas de datos.

Un sistema de bases de datos tradicional está limitado en su operación por varios factores lógicos y físicos, entre los factores físicos están:

- El número de operaciones por segundo realizadas por el procesador.
- El tamaño físico del disco
- La velocidad de escritura del disco
- El tamaño de la base de datos.

Sin tomar en cuenta factores lógicos y de programación el incremento de operaciones de los usuarios en los productos de SAP hizo que se alcanzara el límite de operaciones permitidas por el Hardware y que la experiencia del usuario no fuese adecuada debido a la lentitud de las operaciones; a este problema SAP lo denominó “El problema del cuello de botella de las bases de datos”

Derivado del abaratamiento del hardware, y a la demanda de uso de los usuarios de aplicaciones SAP; ambientes que se ejecutaban en modo cliente servidor o web, contaban con múltiples servidores de aplicación para la atención del alto número de usuarios; esta característica hacía que a mayor número de transacciones fueran generadas mayor número de conexiones a la base de datos, sin embargo la arquitectura no permitía tener más de un servidor de base de datos, lo cual provocaba lentitud en el sistema. Esta lentitud se incrementaba cuando el sistema trataba de generar reportes bajo demanda, en cuyo caso provocaba una interrupción en los servicios derivado de la cantidad de memoria necesaria para la ejecución de estos reportes, ya que disminuía la cantidad de memoria disponible para las conexiones.

El problema radicaba en el acceso al servidor de bases de datos ya sea limitado por el ancho de banda entre los servidores, por la velocidad de escritura en disco o por la cantidad de memoria

RAM disponible; y para resolverlo SAP debió trabajar en una fórmula que disminuyera los tiempos para cada uno de estos factores.

Para el año 2004 SAP lanza varios proyectos con el fin de resolver este problema a través de la modificación de la arquitectura de Hardware de las plataformas. Como el problema del cuello de botella radicaba en el acceso al disco principalmente, se plantea el objetivo de obtener sistemas que trabajen con la base de datos completamente en memoria RAM “in-memory”.

El primer producto de SAP que trabaja con bases de datos en memoria es LiveCache™ el cual generaba una copia de parte de la base de datos y la mantenía en memoria; más adelante a sus productos les agregó un Search Engine basado en la misma filosofía que el Search Engine de Google al cual denominó TREX (Text Retrieval and information EXtraction) y con el cual lograría una mayor velocidad de consulta.

En el año 2005 SAP comienza “The traker project”. Este proyecto tenía la intención de tomar lo que se había ya validado en LiveCache™ y probar si era posible mantener una copia completa de la base de datos en memoria RAM; lo que eliminaba por completo el uso de discos duros al momento de la operación y por tanto resolvía el problema del cuello de botella.

Aun cuando el proyecto es un éxito, SAP adquiere las compañías Callixa y P*time y forma el proyecto NewDB, el cual era el resultado del proyecto anterior reformulado de manera que ya aprobaba estándares de calidad comerciales. SAP no se conformó con tener una base de datos en memoria por lo que trabajo para obtener un sistema de bases de datos de misión crítica y en 2010 hace públicos sus planes de pasar a bases de datos in-memory” su portafolio de aplicaciones. Más adelante en 2011 lanza la primera versión de SAP HANA™.

Las investigaciones de SAP tenían un objetivo claro, el cual inicialmente no estaba basado en considerar a Big Data como un producto que debieran incluir. SAP HANA™ en sí, puede ser utilizado como un producto que no utilice el módulo de Big Data incluido, sin embargo, actualmente varias empresas han aprovechado en conjunto las capacidades de hardware y el potencial de la solución de datos para resolver problemas de procesamiento inmediato y ofrecer experiencia de usuario que en otros términos no podría ofrecerse.

Arquitectura

SAP HANA™ como plataforma de alto rendimiento tiene múltiples aplicaciones y múltiples configuraciones diseñadas para poder adaptarse a diferentes escenarios. SAP ofrece a través del producto SAP Data Services™ la posibilidad de administrar entornos de Big Data basados en versiones de Hadoop™ utilizando la capacidad de procesamiento del hardware de alto rendimiento de SAP HANA™

Toda plataforma de SAP HANA™ cuenta con al menos el componente base del producto denominado SAP HANA platform™. Esta base tiene los servicios sobre los que se construirá el requerimiento final del usuario, y contendrá productos adicionales que dependen de este requerimiento. Este concreto, el producto base cuenta con los siguientes servicios:

- HANA Database

  Database es la capa del producto que maneja la base de datos en sí. Esta capa es la que tiene la capacidad de administrar los accesos a los discos de estado sólido y manejar las operaciones y la copia de la base de datos en memoria. Maneja también el almacenamiento extendido para bases de datos parcialmente en memoria.

- HANA Client

  HANA Client, es un conjunto de librerías que permite generar conexiones desde las aplicaciones hacia la capa de HANA Database a través del estándar ODBC (Open Data Base Connectivity). Es un componente necesario para desarrollar cualquier aplicación en servidores HANA, y permite integrar cualquier aplicación de consulta que soporte ODBC, como una plataforma de analitycs o una aplicación de escritorio.

- HANA Studio

  HANA Studio es un IDE (Integrated Development Enviroment) basado en eclipse que es usado para programación en servidores SAP HANA. Permite acceso a las bases de datos y ayuda al manejo de autorizaciones y modelos tanto nuevos como existentes

- HANA XS Engine:

  Un componente interesante de este grupo de la base de HANA es “SAP HANA Extended Application Services” la cual es comúnmente nombrada como XS Engine. Este componente que conjunta las funciones de servidor de aplicación, servidor web, y un entorno de desarrollo. La diferencia con cualquier servidor de aplicación que fuera posible instalar en un sistema Linux, es que XS Engine tiene acceso directo a funcionalidad

profunda de la base de datos.[28]

Fundamentalmente este componente permite realizar desarrollo sobre plataformas HANA disminuyendo el TCO y aprovechando el desempeño de la plataforma.

Sobre esta arquitectura de productos se monta el producto SAP HANA Dynamic Tiering™ el cual contiene una implementación de Hadoop pre-configurada y un sistema de almacenamiento HDFS.

Las principales características de HANA Dynamic Tiering™ es que puede soportar un gran volumen de datos como es necesario para los sistemas HDFS en sistemas denominados host, los cuales denominados Warm Store o Hot Store dependiendo de su función, permiten que el almacenamiento se comporte con un desempeño de alto rendimiento durante el proceso de trabajos Map Reduce.

Para generar una implementación de Big Data en HANA Dynamic Tiering ™ se necesitan al menos 2 tipos de host. Hot Store y Warm Store. El primero permite guardar datos que requieren computación en tiempo real utilizando las características in-memory de HANA™, el Hot Store son datos que siempre están cargados en memoria RAM, mientras que los Warm Store permiten almacenar información que no requiere este nivel de latencia y que su uso no es necesariamente frecuente. El almacenamiento Warm se encuentra en discos de estado sólido. Esta flexibilidad permite que HANA™ reduzca el TCO para los usuarios de HANA™.

[28] Jung T. (2012). SAP HANA Extended Application Services. Abril 3 2016, de SAP Sitio web: http://scn.sap.com/community/developer-center/hana/blog/2012/11/29/sap-hana-extended-application-services.

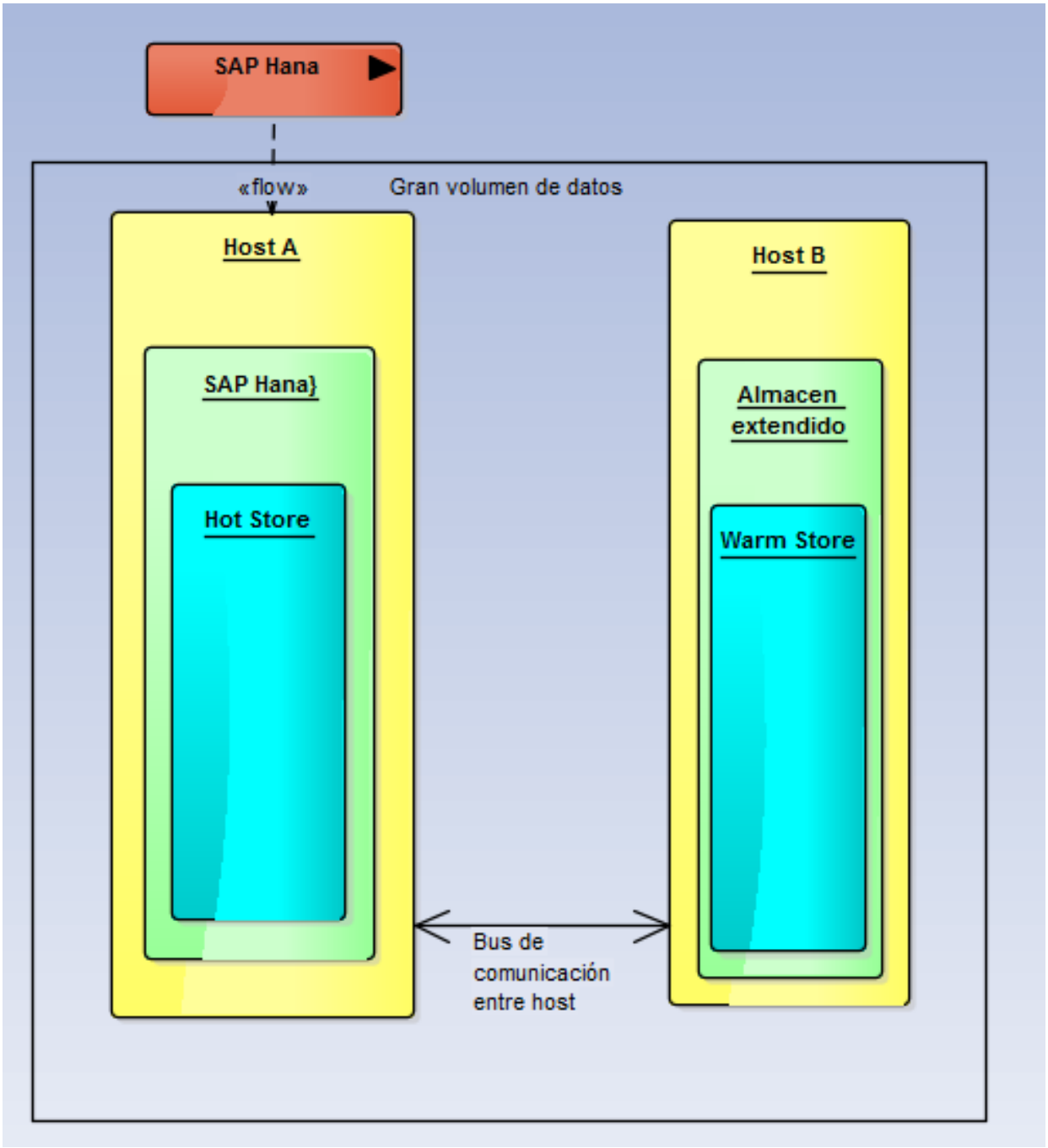

Fig. 11 Implementación de almacenamiento de Big Data en SAP HANA ™

Como puede verse en la figura, existe una comunicación entre los Store de manera que los datos se almacenan en uno u otro dependiendo de la relevancia y el uso de los mismos. Sin embargo cabe aclarar que a través de HANA XS Engine es posible que una aplicación o un usuario consulten datos directamente de cualquiera de los Store.

Hardware

Cuando un proveedor de soluciones de software ofrece productos que se ejecutan sobre plataformas de hardware específicamente diseñadas para el producto se dice que ofrece appliances.

SAP HANA™ fue originalmente creado para soportar transacciones de bases de datos en memoria. Con la finalidad de resolver el problema de cuello de botella. Para ello, durante el desarrollo de los proyectos de SAP se forma una alianza con Intel™ en la que Intel™ proveería a SAP™ los procesadores optimizados para bases de datos “in-memory”. Durante esta alianza Intel desarrollara procesadores optimizados para productos SAP y para operaciones con una arquitectura totalmente en memoria. La generación de procesadores Intel® Xenon® E7 es el resultado directo de esta innovación.

Tradicionalmente los productos de SAP no tenían restricciones respecto al hardware que se utilizaba con ellos. Siempre se montaron sobre sistemas UNIX, y estaban construidos sobre arquitectura java.

SAP, en la generación esfuerzos para construir un appliance que trabaje adecuadamente con sus productos, decide estandarizar además del hardware, el software sobre el que trabajará; y para ello hace algunas modificaciones en su visión de negocio. El soporte para el appliance lo maneja exclusivamente para sistemas operativos Linux SUSE; dado que sus productos corrían en sistemas operativos UNIX el cambio no sería un gran problema.

HANA™ también fue optimizada para soportar memoria RAM Samsung™ con el objetivo de que esta memoria corriera y soportara las operaciones de lectura de los procesadores y los algoritmos de los productos SAP. Basándose en estos tres cambios, el almacenamiento y el adaptador de red de los sistemas pueden ser elegidos por el usuario final sin que ello represente un problema de performance en los productos.

Un servidor típico de HANA™ puede ser empleado en cualquiera de los diversos productos de SAP; la configuración de cada uno depende de lo que se desee instalar y/o configurar. Particularmente existe una denominada HANA Dynamic Tiering™ que contiene software para Big Data, la cual está implementada con Hadoop™ pero utiliza bases de datos Sybase. Esta versión ya viene pre-configurada y precargada en este tipo de servidores.

El hardware de HANA™ está construido con una arquitectura que es fácilmente escalable. Una configuración básica de un servidor estándar consta de 512 GB de memoria RAM y 4 procesadores E7-4870. La configuración escalabre admite tener múltiples servidores físicos que pueden ser administrados como un solo servidor lógico; lo que permite que agregar un nuevo servidor signifique tener más GB de memoria RAM disponible de manera lógica para ejecutar

operaciones distribuidas. 6 Servidores estándar otorgan la capacidad de procesamiento de una supercomputadora[29].

En el año de 2012 SAP completa un Benchmark[30] que fue ejecutado para una implementación de 100 TB en memoria RAM a través de un arreglo de 16 servidores HANA™, demostrando con ello su capacidad de escalabilidad sin que el hardware sea un problema. Cada nodo manejaba 8 procesadores Intel®E7 con 10 núcleos a 2.4ghz.

Con este arreglo SAP fue capaz de escanear en base de datos 100 billones de registros por segundo y cargar en memoria 16 millones de registros por minuto. Ningún sistema estaba cerca de obtener esos resultados. Acorde a SAP, el más próximo sistema era 1000 veces más lento.

Las investigaciones de SAP en cómputo de alto rendimiento generaron grandes avances. Las bases de datos in-memory aparecen como un nuevo paradigma y con HANA se genera una gran opción para correr aplicaciones de Big Data en cómputo de alto desempeño.

[29] Word J. (2012). SAP HANA Essentials. USA: Epistemy Press LLC., p.97.
[30] Word J. (2012). SAP HANA Essentials. USA: Epistemy Press LLC., p.89

Casos de éxito

Dentro del universo de empresas que consideran a SAP como una empresa confiable para sus datos, existen algunas que después de la implementación de HANA como plataforma de Big Data, dan testimonios de la ayuda que obtuvieron implementando este sistema.

NFL

Una de estas empresas es la National Football League (NFL). Para ellos, era importante crear un concepto el cual se centrara completamente en el aficionado. La NFL crea el concepto de NFL Fantasy Football[31] que permite a los aficionados obtener información real de cualquiera de los equipos y jugadores de la liga.

Con información de la historia de los jugadores, estadísticas de juego, noticias, videos, y la posibilidad de generar equipos de fantasía, la NFL tiene el reto de poner una gran cantidad de datos en las manos de los fans para ser consumidos en tiempo real y con la mayor simplicidad.

Utilizando SAP HANA, la NFL permite a los usuarios generar equipos de fantasía, actualizar las estadísticas de sus equipos, comparar jugadores, dar seguimiento y actualizar sus datos al mismo tiempo que van ocurriendo los juegos de la temporada. La plataforma ayuda a la NFL a mantener el interés de los aficionados al juego y con ello incrementar el valor del negocio.[32]

Ciudad de Boston USA

Administrar servicios para una ciudad de más de 700,000 habitantes no es una tarea sencilla. Desde pago de impuestos, servicios públicos, servicios de emergencia, seguridad y atención a incidentes es un mar de información. Si a esto se suma que Boston es un destino turístico dentro de los Estados Unidos, proveer información de lugares para visitar, cosas para hacer, sitios de interés, servicios para visitantes, servicios para estudiantes y una amplia gama de más servicios, convierte el escenario en un mundo de información.

El director de operaciones de la ciudad de Boston David Curt, muestra la forma en la que Boston implementa Analitycs a con Big Data a través de SAP HANA para mejorar la experiencia de lo que significa vivir o visitar la ciudad.[33]

A través de la plataforma de SAP HANA, la ciudad de Boston pone a disposición de la gente la información en tiempo real de lo que sucede en la ciudad. Además de permitirle a la administración de la ciudad en cuanto a servicios e incidentes se refiere.

---

[31] Accesible a través de http://www.nfl.com/fantasyfootball

[32] SAP. (2015). Casos de Éxito. Abril 3 2016, de SAP Sitio web: http://go.sap.com/latinamerica/about/customer-testimonials/finder.html

[33] SAP. (2015). Casos de Éxito. Abril 3 2016, de SAP Sitio web: http://go.sap.com/latinamerica/about/customer-testimonials/finder.html

Ericsson

Dentro de sus proyectos de innovación, Ericsson en conjunto con SAP, están desarrollando un nuevo paradigma en la forma de vivir el día a día. Ericsson siendo una de las empresas líderes a nivel mundial en la generación de dispositivos y software de comunicación ha trabajado constante mente en la forma en la que se comunican las personas.

El concepto que se desarrolla trata también sobre comunicación, sin embargo este concepto trata de la comunicación con nuestros dispositivos. Arum Bhikshesvaran jefe de la oficina de marketing, explica la innovación a través de un ejemplo simple.

Una sombrilla que tenga la comunicación con el sistema meteorológico regional, y a través de sensores, se conozca la localización de la sombrilla. En un día en donde es muy probable que llueva, la sombrilla sería capaz de enviar un mensaje al teléfono inteligente indicando que es recomendable que la lleve y que se encuentra en el closet.

El concepto de Ericsson y SAP es un concepto en el cual, la participación de Internet of Things está completamente integrado a la vida diaria. Aun cuando esto se está desarrollando, es una idea que tiene potencial, y que requerirá de plataformas de alta disponibilidad, en este caso SAP HANA, para soportar el uso diario.

CAPÍTULO V HADOOP™.

En tecnologías de Big Data, Hadoop™ es sin duda uno de los principales representantes. Cuando hablamos de Software de código abierto; las tecnologías LAMP (Linux, Apache, MySql y PHP) aparecen como los principales representantes de las mismas. Para Big Data también aparecen tecnologías de código abierto que resuelven los problemas de procesamiento de grandes volúmenes de datos.

Hadoop™ es este representante que permite conocer a profundidad el uso de esta tecnología y que al igual que otras tecnologías, fue adoptado por múltiples empresas como un estándar; contribuyendo esto al propio desarrollo del producto.

Grandes empresas han invertido en tecnología y desarrollado sistemas que explotan el mar de datos en el que vivimos. En ramos como investigación, educación, servicios y desarrollo tecnológico, el contar con acceso a esta tecnología permite resolver problemas que no estarían en el alcance de la mayoría de pequeñas empresas o investigadores.

Acceder a productos como Hadoop, es posible inclusive para universidades públicas o estudiantes, donde el cómputo de alto rendimiento existe en varios niveles, desde clusters creados por los propios estudiantes hasta supercomputadoras que mantienen la investigación.

Utilizando Hadoop™, una pequeña empresa puede montar su propia infraestructura de Big Data para tareas de análisis o para alimentar Analitycs. Dado su flexibilidad y la arquitectura sobre la que se encuentra diseñado, es posible implementar Hadoop sobre clusters y sistemas operativos Linux, bajando drásticamente el costo de implementación y eliminando el costo de licenciamiento.

Pero no sólo pequeñas empresas han accedido a Hadoop™, grandes y reconocidas marcas han realizado procesamiento de datos en esta tecnología. New York Times™, Ancestry™, Facebook™, Twitter™, Yahoo™ y otras basan su operación en plataformas generadas a partir de Hadoop™. Es por esta razón que este capítulo explica a detalle cómo se desarrolló esta tecnología en el entorno de Big Data, mostrando Características de la plataforma, su sistema de almacenamiento y la forma de escritura y lectura.

Hadoop

La construcción de un Search Engine no fue un problema trivial para la industria de la computación de los años 90's. Apache Lucene™ fue una plataforma creada por Apache Software Foundation con el objetivo de proveer un Text Search Engine.

Doug Cutting quien fuera uno de los creadores de Lucene™, desarrolla el proyecto Apache Nutch™ el cual estaba concebido como un potente y escalable Search Engine con capacidad de análisis del contenido web.

Apache Nutch™ comienza a desarrollarse en el año 2002; con estimaciones iniciales de un billón de páginas, los desarrolladores notaron de manera inmediata que este sistema no sería escalable para las dimensiones actuales de Internet. Uno de los principales problemas a los que se enfrentaron durante el desarrollo fue el almacenamiento de datos. Para ello, Doug desarrolla un sistema de ficheros para resolver este problema, basados en una implementación del modelo de Big Table de Google, el cual denominan NDFS (Nutch Distributed Filesystem).

Para el año 2004 Apache Nutch™ adapta MapReduce™ para poder aprovechar las ventajas de este modelo de programación convirtiéndose en un proyecto independiente llamado Hadoop. Doug Cutting comienza en este mismo año a trabajar en Yahoo™ en donde desarrolla Hadoop para trabajar a escala WEB lanzando la primera versión del producto en el año 2008 en el propio buscador Yahoo™

A partir del éxito de la implementación de Hadoop™ en el buscador Yahoo™, grandes compañías comienzan a utilizar Hadoop™ como plataforma de Big Data. Amazon™ genera una implementación y renta el uso de la plataforma en pago por hora, abriendo nuevos paradigmas de negocio. Facebook™ basa el funcionamiento de su red social sobre una implementación de Hadoop™; particularmente la misma que se utiliza en Yahoo Search Engine.

Apache Software Foundation adopta oficialmente Hadoop™ como uno de sus proyectos más importantes. El NDFS se convierte entonces en HDFS (Hadoop Distributed Filesystem) y comienza una serie de proyectos basados en la tecnología de Big Data; estos proyectos son complementos de Hadoop™ y muchas veces son referenciados por su nombre. Entre ellos están:

- Common: Conjunto de utilerías que permiten el manejo de los diferentes módulos de Hadoop, entre otras cosas administran la comunicación con el HDFS.
- Avro: Provee a Hadoop la funcionalidad de socialización y los servicios de intercambio de datos.
- MapReduce: Framework para la escritura de programas para Hadoop.
- HDFS: Sistema de archivos de Hadoop
- Pig: Plataforma de alto nivel para crear programas para Hadoop
- Hive: Sistema de Datawarehouse que facilita la consulta de datos para sistemas basados

en Hadoop
- HBase: Es el almacén de datos escalable de los sistemas Hadoop.
- ZooKeeper: Servicio que maneja y administra la información del sistema Hadoop para soportar sincronización y administración de sus componentes
- Sqoop: Herramienta que permite administrar transferencias de datos entre Hadoop y almacenes de tipo estructurado como las bases de datos relacionales

Es importante aclarar que aun cuando en esencia Hadoop™ es definido como un Framework de Big Data y funciona para resolver este tipo de problemas, la elección de una determinada versión de la herramienta depende de las necesidades a resolver o de los ambientes a generar. Componentes como:

- Secure authentication
  Disponible en las 3 versiones actuales
- Old configuration names
  Disponible en la versión 2.5.5; en las versiones 2.6.x y superior ya se encuentran marcadas como obsoletas
- New configuration names
  Disponible en las 3 versiones actuales
- Old MapReduce API
  Disponible en la verson 2.5.5 y con compatibilidad para las versiones 2.6.x y superiores
- New MapReduce API
  Disponible en las 3 versiones actuales
- MapReduce 1
  Disponible en las 3 versiones actuales
- MapReduce 2
  Disponible en las 3 versiones actuales
- HDFS Federation
  Disponible en las 3 versiones actuales
- HDFS HA (high.availability)
  Disponible en las 3 versiones actuales

Varían entre versiones, por lo que es importante seleccionar la versión adecuada.

Al momento de escribir este trabajo las versiones disponibles son 2.5.5, 2.6.0 y 2.7.0.[34]

[34] The Apache Software Foundation. . (2014). Apache Hadoop Releases. Agosto 3 2015, de The Apache Software Foundation. Sitio web: https://hadoop.apache.org/releases.html

Hadoop Distributed Filesystem

Para explicar el modelo de almacenamiento de Hadoop™, el cual proviene del modelo de almacenamiento desarrollado por Google, es necesario hablar de algunos términos que ayudarán a entenderlo.

Un sistema de archivos es un programa que se encarga de administrar y asignar el espacio de almacenamiento de una memoria; esta puede ser un disco duro, una memoria flash, una cinta magnética, etc. Se encarga además de estructurar la forma en que se guardan físicamente los archivos y de manejar las restricciones de lectura escritura dentro de un sistema operativo.

Cuando un archivo es suficientemente grande para que no pueda ser almacenado en un solo disco o en un solo nodo (computadora o arreglo de discos), es necesario partir el archivo en varios pedazos y almacenarlo en diferentes nodos comunicados a través de la red. Cuando un sistema de archivos maneja esta estructura, se dice que es un sistema de archivos distribuido.

El HDFS es un sistema de archivos que almacena “grandes” archivos de datos y puede correr en cualquier hardware comercial[35]. Acorde a la definición de Apache Software Foundation; HDFS™ fue diseñado para cumplir con los siguientes objetivos:

- Alta tolerancia a fallos de hardware
- Acceso de datos por streaming
- Soporte para grandes conjuntos de datos
- Soporte a modelo de coherencia simple
- Soporte al principio de MapReduce “Acercar el programa a los datos es más barato que acercar los datos al programa”[36]
- Portabilidad a través de hardware heterogéneo y diferentes plataformas de software.

Clarificando un poco más el alcance de HDFS indicaremos que cuando nos referimos a “Soporte para grandes conjuntos de datos” hablamos de conjuntos de datos del orden de Petabytes o superiores.

La arquitectura del HDFS se define de la siguiente manera:

La arquitectura está construida en una relación maestro/esclavo. Consta de un servidor maestro denominado NameNode el cual será encargado de administrar el espacio y gestionar las peticiones de escritura y lectura por parte de los usuarios. Se agregan además DataNodes que

[35] Se refiere a Hardware no especializado.
[36] El principio explica que en un conjunto grande de datos, es más barato en espacio de almacenamiento replicar el programa que el conjunto de datos.

son catalogados como esclavos. Estos esclavos se encargarán de almacenar el contenido real del conjunto de datos.

El nodo maestro permite al usuario almacenar los datos sin la necesidad de que sepa cómo se almacenan en el hardware. Internamente el nodo maestro divide el archivo guardado por el usuario en splits y lo almacena en un conjunto definido de DataNodes.

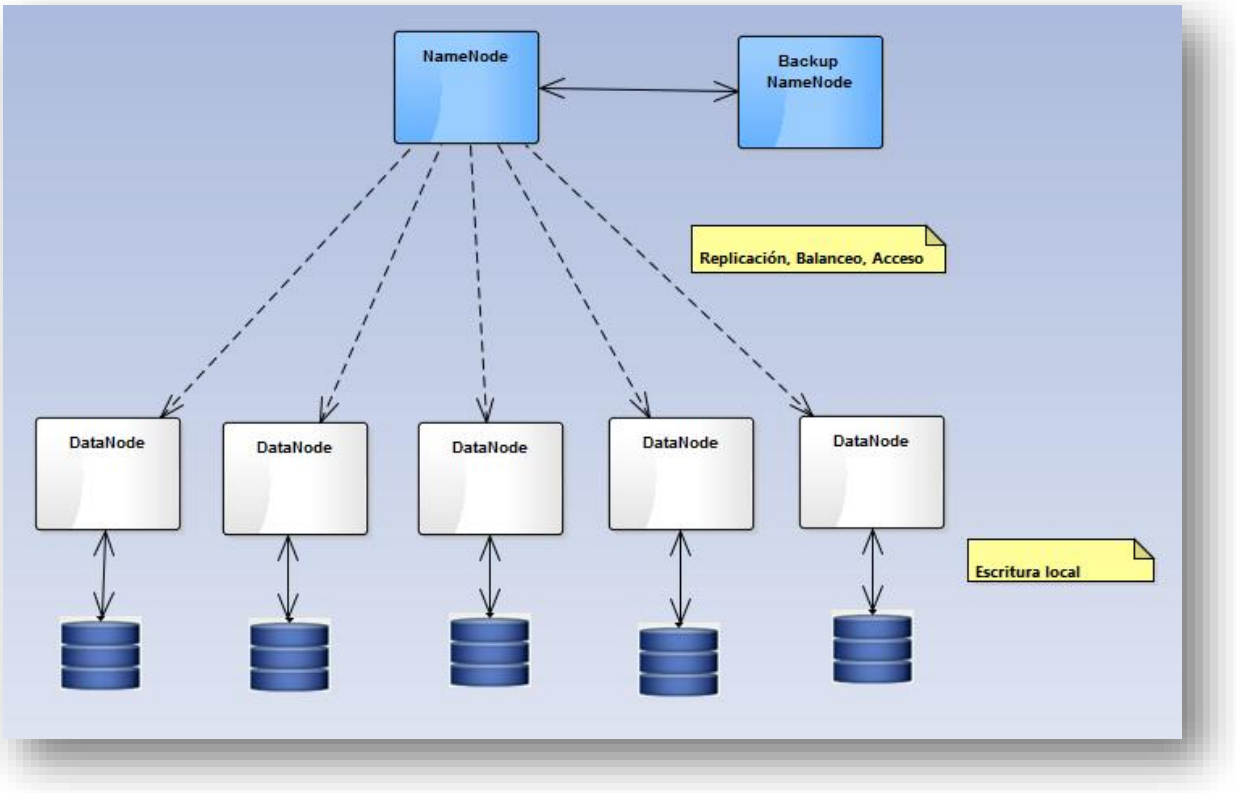

Fig. 12. Arquitectura de Hadoop Data File System

Tanto el servidor maestro como el servidor esclavo corren típicamente sobre sistemas Linux pero pueden ser ejecutados sobre cualquier sistema operativo que sea capaz de interpretar programas Java.

La arquitectura soporta que múltiples DataNodes puedan correr en una misma maquina física, aunque en la práctica esto es extremadamente raro. Se designa una máquina para cada DataNode y una máquina dedicada al NameNode.

El almacenamiento lógico de archivos soporta un sistema ordenado jerárquicamente con directorios y nombres de archivos; esto es análogo al sistema de archivos definido para el usuario final de un sistema Linux o Windows.

Para soportar la pérdida de DataNodes sin la pérdida de datos, HDFS cuenta con un mecanismo de replicación de datos el cual es configurable. El archivo o set de archivos guardado se divide en splits del mismo tamaño y se almacena en un conjunto definido de DataNodes. Después se replica el contenido de cada bloque a otro servidor (1 como mínimo). Las veces que se debe replicar un determinado nodo de datos a otro DataNode es totalmente configurable por el usuario; después se almacena en el NameNode la dirección de cada bloque y sus réplicas. En caso de una falla en el hardware de un DataNode el NameNode sabe en dónde puede buscar las copias de los bloques que se perdieron.

Para la comunicación entre nodos se utiliza el protocolo TCP/IP; manteniendo la comunicación con los clientes en un formato estándar. Para la comunicación entre los servidores esclavos y maestros se utiliza un protocolo propio llamado DataNodeProtocol.

El DataNodeProtocol define interfaces de comunicación en las que cualquier mensaje siempre es comenzado por el DataNode. Las interfaces de comunicación son las siguientes:

1. DataNode Registration: El DataNode envía un mensaje de existencia y registro al NameNode el cual regresa un identificador de registro único que es necesario para cualquier tarea de comunicación, inicio o reinicio del DataNode
2. DataNode Sends Heartbeat: El DataNode envía un mensaje al NameNode cada determinado tiempo (un intervalo en segundos típicamente) el cual incluye información sobre la actividad y capacidad del DataNode. El NameNode regresa comandos de ejecución al DataNode, los cuales pueden ser instrucciones de borrado o replicación.
3. DataNode Sends Block Report: El DataNode envía reportes de los bloques de información que contiene. Típicamente este reporte se envía en periodos superiores a una hora.
4. DataNode Notifies Block Received. El DataNode confirma que ha recibido bloques de información provenientes de un cliente u otro DataNode

Cada operación realizada al HDFS, se genera un registro en un archivo log llamado EditLog el cual está almacenado en el disco duro local del servidor maestro NameNode. El Namespace del sistema por completo está cargado en memoria y está almacenado en un archivo FSImage (FileSystem Image). Para lograr persistencia del estado del HDFS en caso de pérdida de energía o daño del servidor maestro, se carga el archivo FSImage y luego se aplican las operaciones almacenadas en el EditLog, obteniendo con ello el sistema de archivos en su estado anterior a la perdida de energía. .

Es importante aclarar que aún con las ventajas de espacio y replicación que muestra el almacenamiento del HDFS, existen tareas para las que no es adecuado. Basado en las propias características de HDFS no debe ser usado para tareas con las siguientes necesidades:

Baja latencia: Cuando tenemos aplicaciones que requieren una respuesta de lectura de datos del orden de milisegundos, HDFS no es la respuesta. Para esto la computación in-memory, Bases de Datos Relacionales o sistemas de ficheros controlados por HBase™ son mejores opciones que HDFS en estado puro.

Muchos archivos pequeños: Dado que la dirección de almacenamiento es mantenida en el NameNode, Los archivos demasiado pequeños generarían registros que ocuparían espacio. El tamaño total de almacenamiento está limitado por la cantidad de memoria del NameNode; el almacenamiento de múltiples archivos pequeños saturaría esta memoria sin verdaderamente guardar una cantidad grande de información. Para almacenar múltiples archivos pequeños, son recomendable los sistemas de ficheros tradicionales como NTFS o FAT

Escritura múltiple o modificación arbitraria de archivos: La estructura de escritura en HDFS está diseñada para escribir una vez y hacer múltiples lecturas. Usualmente HDFS escribe al final del archivo, por lo tanto no está diseñado para soportar múltiple escrituras o múltiples procesos de escritura simultánea, para ello sistemas tradicionales como NTFS o FAT con arreglos lógicos de discos funcionan mejor en desempeño.

La lectura de datos Hive y HBase

Para explotar todas las características de la nueva tecnología de Big Data se han creado diversos mecanismos que permiten aprovechar de una manera sencilla la explotación de datos. Hive™ es un software de Data-warehouse que permite realizar consultas sobre grandes conjuntos de datos almacenados en sistemas de archivos distribuidos.

El proyecto es administrado por Apache Software Foundation, pero fue originalmente creado por Facebook™ para generar consultas sobre el sistema de archivos de Hadoop™ HDFS. Permite la traducción de sentencias SQL a HiveQL. Técnicamente traduce las sentencias de consulta hechas en SQL a tareas de tipo MapReduce y regresa resultados.

Hive™ no es un gestor de base de datos. Solo permite generar consultas sobre contenido almacenado en HDFS; aun así es muy útil ya que evita la tarea de programación de funciones MapReduce para tareas de búsqueda y permite identificar y optimizar funciones Map y Reduce.

Una consulta en HiveQL luce se la siguiente manera

```
SELECT Name, Code
FROM EmployeeData Join EmployeCodes
ON (EmployeeDataId = EmployeeCodesId)
WHERE Code = ‘Q1’
GROUP BY Code
```

Esta consulta tiene la misma nomenclatura que una en SQL, de esta manera, cualquier administrador de base de datos puede acceder a información almacenada en HDFS usando HiveQL.

Hive™ permite realizar operaciones como si el sistema HDFS se tratara de una base de datos. Entre otras operaciones permite la creación de tablas, carga de datos en tablas, consultas, joins y el almacenamiento de resultados de consultas en tablas de HDFS.

Aun cuando Hive™ ayuda a las tareas de consulta el tiempo de traducción de SQL a HiveQL es alto; es importante considerar este factor cuando se esté trabajando con Big Data si se necesitan resultados en tiempo real.

Para completar la tecnología y poder ejecutar computación de alto rendimiento en Hadoop™ fue necesario construir una tecnología que no se basara exclusivamente en HDFS. Después de la

publicación de Google Big Table™, Hadoop trabajo en el proyecto de una base de datos que soportara consultas a largos conjuntos de datos con una latencia baja[37].

HBase aparece como esta base de datos la cual al igual que BigTable™ es un mapa ordenado multidimensional que soporta cualquier tipo de datos. HBase™ no es una substitución de BigTable™ ni es una substitución de los tradicionales sistemas de bases de datos relacionales; pero sobre todo es importante aclarar que no substituye a HDFS ya que es un complemento del mismo.

HBase genera una estructura de mapa ordenado usualmente de los datos almacenados en HDFS con el objetivo de proveer una menor latencia para realizar búsquedas. Para computación en tiempo real HBase es la tecnología que puede soportar grandes búsquedas en tiempos muy cortos comparativamente con el uso de HiveQL.

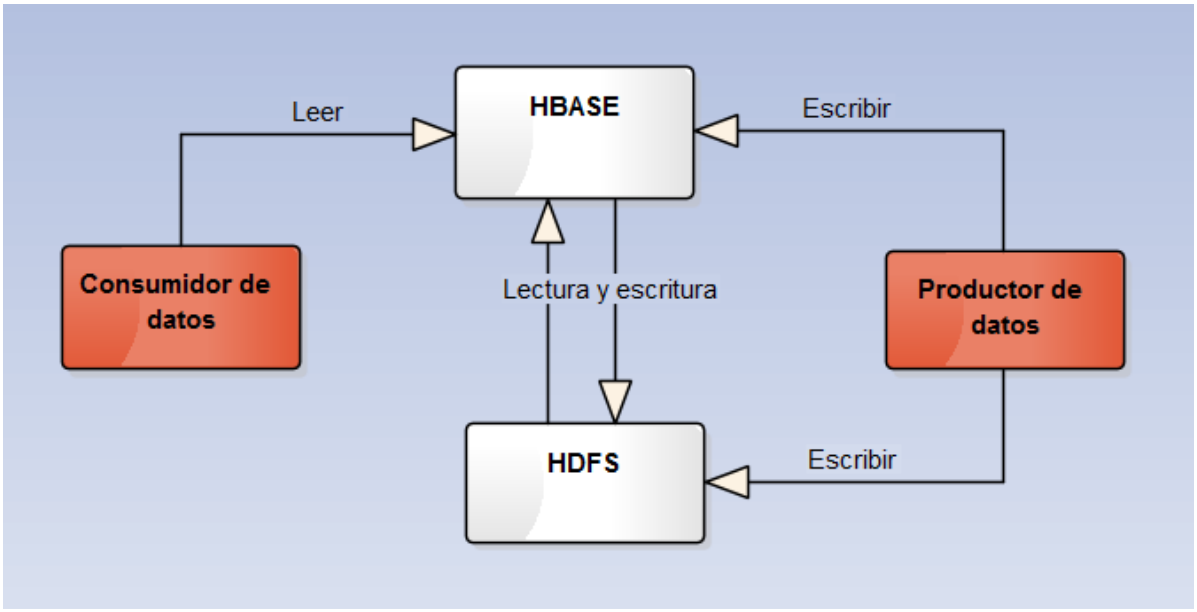

Fig. 13 Esquema de interacción HBASE y HDFS

Una de las características importantes de HBase es que los resultados de los datos obtenidos a través de consultas pueden ser fácilmente integrados a funciones Map y Reduce. Cuenta con una API completamente construida en Java y eso le permite que los resultados de las funciones Map y Reduce sean fácilmente integradas a HBase.

[37] The Apache Software Foundation. (ND). Appendix H. HBase History. Junio 3 2015, de Cloudera Sitio web: http://archive.cloudera.com/cdh5/cdh/5/hbase-0.98.6-cdh5.3.2/book/hbase.history.html

Alternativas a Hadoop

Aunque Hadoop es una herramienta de libre acceso, y disponible para quien la quiera utilizar, al igual que ha pasado con otro software de licencia libre, se crearon empresas alrededor de esta tecnología y surgieron servicios y soporte para versiones modificadas de Hadoop. Algunos de estos productos son Cloudera, MapR y Hortonworks.

Cloudera

En el año 2008 tres ingenieros de empresas de Silicon Valey se unieron para generar esta empresa. .Christophe Bisciglia Ingeniero de Google, Amr Awadallah proveniente de Yahoo, Mike Olson de Oracle y Jeff Hammerbacher de Facebook se re unen y crean Cloudera[38]

Más tarde en el año 2009 Doug Cutting, quien fuera uno de los co-creadores de Hadoop se une como jefe de arquitectos. El objetivo era crear una empresa que contara con uno de las más poderosas implementaciones de Hadoop. Acorde a la propia empresa, ellos proveen la más rápida, segura y poderosa plataforma de Big Data construida sobre Hadoop.

Con una plataforma de Big Data y soporte por el propio fabricante, Cloudera se ofrece como una alternativa a Hadoop, sin embargo, a diferencia del producto que se puede descargar de Apache, Software Foundation, Cloudera tiene un costo y es necesario contratar servicios de implementación para esta plataforma.

Cloudera entrega una gama de servicios y de Data Platforms a través de su producto Enterpise Data Hub, ya sea externos o ayudando a las empresas a subir sus datos a la infraestructura de nube, EDH permite a las empresas opciones de disponibilidad y brinda la expertise y seguridad sin invertir en infraestructura.

MapR

MapR se funda en San José California. Su objetivo fundamental era proveer soluciones de Big Data basadas en Apache Hadoop. En asociación con EMC generan su propia distribución de Hadoop.

La empresa llega a los reflectores cuando Amazon la selecciona para un proyecto de actualización del servicio Elastic Map Reduce.

MapR Ofrece varias versiones de su plataforma como M3, M5 y M7 A diferencia de Cloudera, MapR si ofrece una versión de M5 la cual es libre de costo, y se puede descargar desde su página. Esta versión es limitada en algunas de sus funciones, pero la licencia tiene autorización de usarse comercialmente en producción. Esta es la lista comparativa de las características de la versión abierta contra la Enterprise.

[38] Cloudera. (ND). About Cloudera. Abril 13 2015, de Cloudera Sitio web: http://www.cloudera.com/about-cloudera.html

	Converged Community Edition	Converged Enterprise Edition
Modulos		
MapR-FS	✓	✓
Apache Hadoop and Open Source Projects	✓	✓
MapR-DB	✓	✓
MapR Streams	✓	✓
Características y Capacidades		
Performance	✓	✓
Scalability	✓	✓
Standards-Based APIs and Tools	✓	✓
Direct Access NFS™	✓	✓
Manageability	✓	✓
Integrated Security	✓	✓
Multi-tenancy	✓	✓
Advanced Multi-tenancy		✓
Consistent Snapshots		✓
High Availability		✓
Disaster Recovery		✓
Global Table Replication for MapR-DB		✓
Global Replication for MapR Streams		✓
Real-Time Reliable Transport for MapR-DB		✓

Tabla 3 Comparativo de plataformas de Big Data de MapR. [39]

Hortonworks

Hortonworks se funda en el año de 2011 en Santa Clara california. Ofreciendo soluciones y productos basados en Hadoop.

Siendo creada entre otros por 24 ingenieros del proyecto original de Hadoop, Hortonworks se convierte rápidamente en una empresa que llama la atención a la industria. Las distribuciones de las plataformas de Hortonworks son de código abierto y están disponibles a través de la página de la empresa, los servicios que la empresa ofrece son subscripciones de soporte, capacitación y servicios profesionales.[40]

La plataforma distribuida por Hortonworks es Horton Data Platform HDP, y dado que se puede acceder a ella sin la necesidad de pagar, es un claro competidor de Hadoop para una

[39] MapR. (ND). MapR Converged Data Platform: MapR Editions. Mayo 13 2016, de MapR Sitio web: https://www.mapr.com/products/mapr-distribution-editions

[40] Hortonworks . (2016). Horton Works Quick Facts. Mayo 13 2016, de Hortonworks Sitio web: http://hortonworks.com/about-us/quick-facts/

implementación empresarial, en donde el soporte es una necesidad indispensable para proyectos de gran envergadura.

Entre sus productos adicionales que ofrece se encuentra HDF Horton Data Flow, el cual es un servicio de análisis para IoT.

CAPÍTULO VI Oracle Big Data Appliance.

En la década de los 70´s durante la aparición del modelo relacional de base de datos Larry Ellison, Bob Miner y Ed Oates identificaron el potencial de invertir en comercializar un modelo relacional, esta oportunidad la aprovechan creando una compañía y aparece Software Development Laboratories.

Entre los años de 1978 y 1979 producen las dos primeras versiones de Oracle; la segunda versión, Oracle 2 es la primera versión comercial del producto, en este año cambian el nombre de la compañía a Relational Software Inc y aparece uno de los grandes competidores dentro del mercado de software para datos.[41]

Para el año de 1982, la compañía cambia nuevamente de nombre a Oracle Systems, desarrollando nuevas versiones de Oracle como base de datos relacional, para el año de 1989 ya soporta OLTP (On Line Transaction Prossesing).

Como parte de su visión del crecimiento, Oracle comienza en años posteriores la adquisición de compañías que robustezcan y apoyen el desarrollo de sus productos. Para el año 2005, Oracle adquiere People Soft y Siebel, las cuales eran dos grandes competidoras de sus productos empresariales de bases de datos con transacción en línea.

Para el año 2009 Oracle alcanza un acuerdo para comprar Sun Microsystems.[42], con este acuerdo, Oracle prepara su escenario para competir contra IBM en la creación de productos que conviven con Hardware propietario. Esto es muy importante para que en años posteriores genere appliances

Oracle diseña un appliance para poder soportar de manera optimizada la ejecución de su manejador relacional, aparece en el mercado Oracle Exadata Database Machine. Un appliance que trabaja sobre memoria flash y un bus de comunicación InfinityBand, el cual promete un alto desempeño para productos Oracle[43]

[41] Oracle. (ND). Defying Conventional Wisdom. Mayo 10 2016, de Oracle Sitio web: http://www.oracle.com/us/corporate/profit/p27anniv-timeline-151918.pdf
[42] El PaíS. (2009). Oracle adquiere Sun Microsystems por 5.710 millones. Mayo 08 2016, de El PaíS Sitio web: http://tecnologia.elpais.com/tecnologia/2009/04/20/actualidad/1240216080_850215.html
[43] Oracle. (2016). Oracle Exadata Database Machine. Mayo 13 2016, de Oracle Sitio web: https://www.oracle.com/es/engineered-systems/exadata/index.html

> Más tarde en el año 2011 Oracle anuncia su incursión en el mundo de Big Data con la creación del Big Data Appliance, incluyendo una implementación de Hadoop, una versión de NoSQL Database de Oracle y el software de análisis estático R[44]

Acorde a Oracle, Big Data Appliance "es un sistema optimizado para adquirir, organizar y cargar datos no estructurados en Oracle"[45]. Se puede considerar un competidor directo de HANA™ pero con diferencias fundamentales de Software.

Ahora se profundizará en los detalles de este appliance para conocer sus características y su uso en Big Data.

[44] Darrow B. (2011). Oracle Big Data Appliance stakes big claim. Mayo 09 2016, de GIGAOM Sitio web: https://gigaom.com/2011/10/03/oracle-big-data-appliance-stakes-big-claim/

[45] Oracle. (2016). Oracle Big Data Appliance. Mayo 13 2016, de Oracle Sitio web: https://www.oracle.com/es/engineered-systems/big-data-appliance/index.html

Big Data Appliance

Big Data appliance aparece en el mercado como una solución integrada de sotware y hardware que promete la capacidad de analizar datos estructurados y no estructurados en un mismo entorno, y un análisis bajo una implementación de Hadoop, específicamente Cloudera[46]

Acorde a Oracle, Oracle Big Data Appliance fue creado para soportar un gran número de casos de negocio, basado en Hadoop y Oracle NoSQL, su propósito es trabajar en conjunto con otras plataformas de datos, y no en escenarios standalone, es por eso se ofrece en conjunto con otras plataformas. [47]

El producto se ofrece como Big Data Platform, el cual tiene los siguientes componentes:

- Big Data Appliance: appliance de procesamiento de datos no estructurados a través de una implementación de Hadoop.
- Exadata Database Machine: appliance que ejecuta bases de datos Oracle con alto desempeño
- Endeca Information Discovery: Plataforma de descubrimiento de datos[48]
- Exalitics: Plataforma de Business Inteligence con capacidades de Base de Datos In-Memory
- Oracle Business Inteligence: Suite de Analitycs y Business Inteligence

La justificación de Oracle para la generación de un appliance se basa en la evaluación del costo de construir contra comprar. Siendo que Hadoop™ es un software de código abierto, es obvio que cualquiera puede obtener una copia del producto desde la página de Apache Software Foundation. Además las características de esta tecnología permiten instalar una instancia en hardware no especializado y sistemas operativos comunes de usuario o servidor. El único requisito que presenta Hadoop es que el sistema operativo soporte la ejecución del JRE (Java Runtime Enviroment).

Instalar Hadoop en un clúster creado por una empresa, o en un arreglo de máquinas virtuales es algo posible y muy frecuente; Si se puede obtener Hadoop gratis e instalar sobre infraestructura que ya existe en las compañías (en esencia se puede construir la plataforma de Big Data), la pregunta a esto es: ¿Por qué comprar un appliance completamente nuevo?

[46] Plunkett T, Macdonald B, Nelson B, Hornick M, Sun H, Mohiuddin K, Harding D, Mishra G, Stackowiak R, Laker K & Segleau D. (2013). Oracle Big Data Handbook. . USA: McGRAW-HILL. P 7
[47] Plunkett T, Macdonald B, Nelson B, Hornick M, Sun H, Mohiuddin K, Harding D, Mishra G, Stackowiak R, Laker K & Segleau D. (2013). Oracle Big Data Handbook. . USA: McGRAW-HILL. P 54
[48] Se refiere a una plataforma que permite analizar información proveniente de diversas fuentes a través de un panel de control

La respuesta que Oracle da a esta pregunta se basa en 4 factores que aparecen cuando una compañía aprueba un proyecto de Big Data y comienza a escalar su proyecto. Estos factores son:

- Optimización
- Proveer valor en un corto tiempo
- Reducción de riesgos al ser un producto ya configurado y probado
- Costo beneficio que se puede obtener de la compra del appliance contra la construcción de un ambiente[49]

En una construcción propia de Big Data con Hadoop se deben considerar diversos componentes que pueden marcar la diferencia en cuestión de precios: Procesadores, Discos duros y conectividad de red deben ser consideradas; además para ambientes de alto desempeño, estos componentes pueden llegar a causar problemas inesperados, como el facho por uso continuo, ya que no todos los componentes se diseñan de la misma manera; algunos son de uso común como una computadora de escritorio, y otros están diseñados para trabajar 24X7.

La experiencia sobre la implementación de Hadoop también es un elemento a considerar cuando se está construyendo un sistema de Big Data. Pensar en este sistema indica que las necesidades del mismo se basan en la premura de obtener ventajas competitivas de la tecnología; por lo tanto sin esta experiencia se corre el riesgo de una lenta implementación que al futuro puede tener más costos que beneficios.

Oracle Big Data Appliance está pre configurado y listo para ponerse en producción.

Las diferencias fundamentales entre la construcción contra la compra de un sistema de Big Data acorde a Oracle[50] son las siguientes:

1. La diferencia inicial del precio de hardware considera la instalación pero no la optimización de los componentes
2. Para un periodo de 3 años el precio inicial es inferior para un appliance que para un sistema construido.
3. El precio considerado para el soporte de los sistemas incluidos, a diferencia de Big Data Appliance debe considerar soporte a los proveedores de Sistemas Operativos, Hadoop, encripción de datos y Base de datos,
4. El tiempo de implementación de un appliance es sumamente inferior.
5. El hardware de un appliance está optimizado para obtener un mayor desempeño.

[49] Plunkett T, Macdonald B, Nelson B, Hornick M, Sun H, Mohiuddin K, Harding D, Mishra G, Stackowiak R, Laker K & Segleau D. (2013). Oracle Big Data Handbook. . USA: McGRAW-HILL. P 53

[50] Oracle. (2014). The Data Warehouse Insider. Agosto 5 2015, de Oracle Sitio web: https://blogs.oracle.com/datawarehousing/entry/updated_price_comparison_for_big

Arquitectura

Oracle Big Data Apliance tiene componentes de arquitectura de software y hardware que componen al appliance. La plataforma está instalada sobre un sistema operativo Oracle Linux, y utiliza Cloudera como distribución de Apache Hadoop.

Como sistema de almacenamiento utiliza Hadoop Distributed File System, un engine de MapReduce para programación en java, y Cloudera Manager como administrador de la plataforma.[51]

Adicional a los componentes de comunicación, la plataforma utiliza una base de datos Oracle NoSQL, y un componente Oracle Advanced Analytics para procesar, acorde a Oracle, las tres tareas principales del sistema, Adquirir, Organizar y Analizar datos.

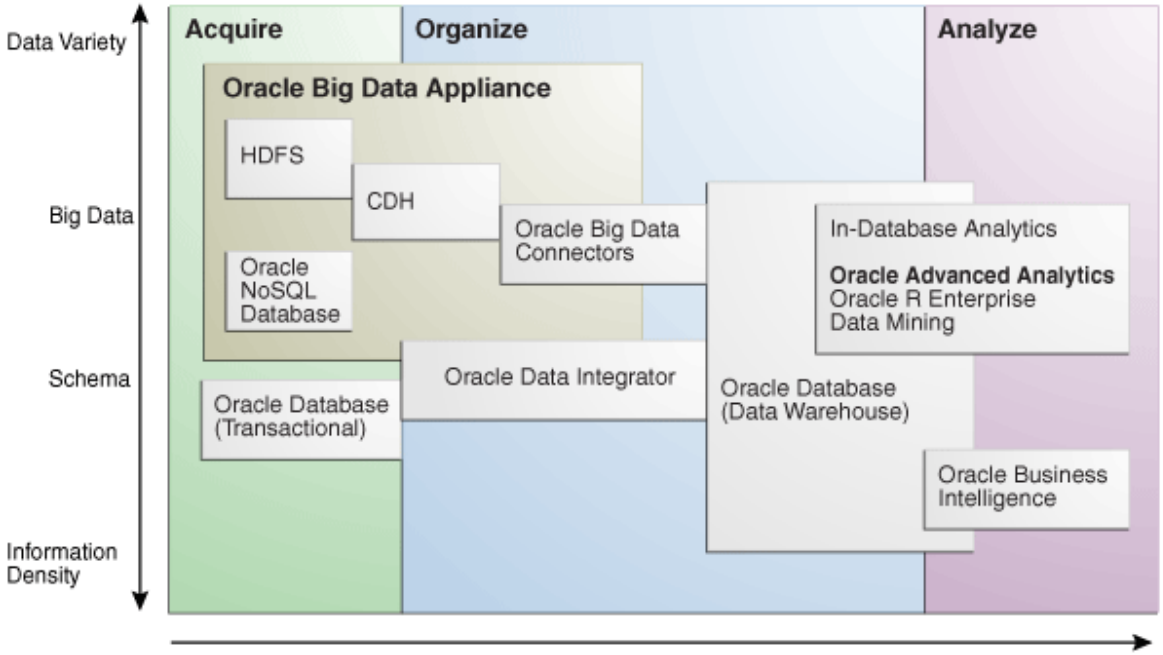

Fig. 14 Arquitectura de Software Oracle Big Data Appliance [52]

[51] Oracle. (ND). Big Data Appliance Software User's Guide. Mayo 17 2016, de Oracle Sitio web: https://docs.oracle.com/cd/E40622_01/doc.21/e40656/concepts.htm#BIGUG107

[52] Oracle (2016). "Oracle Big Data Appliance Software Overview [figura]". Mayo 17 Sitio web: https://docs.oracle.com/cd/E40622_01/doc.21/e40656/concepts.htm#BIGUG108

La plataforma tiene capacidad para procesar tanto las bases de datos relacionales a través de su componente Oracle Database, como datos no estructurados a través del componente Oracle NoSql Database Community Edition™.

Para programación y procesamiento de datos, cuenta con una distribución del lenguaje R modificada, denominada Oracle R. el cual proporciona capacidades de explotación de datos.

La arquitectura de hardware se basa en un arreglo de varios servidores Solaris y un bus de comunicaciones InfiniBand que soporta velocidades de transferencia de hasta 8 Gbps; el cual comunica sus componentes Oracle Big Data Appliance, Oracle Exadata y Oracle Exalitics.

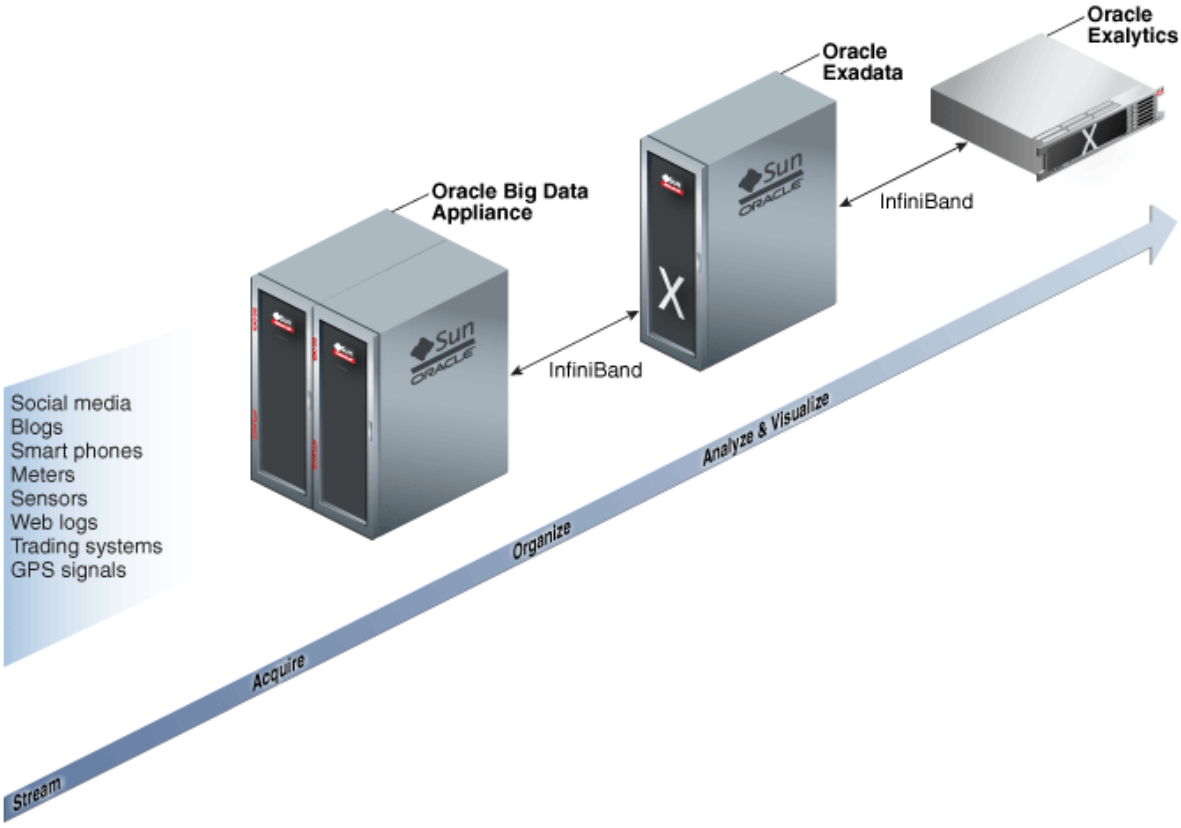

Fig. 15 Arquitectura de Hardware Oracle Big Data Appliance[53]

Una de las características de Oracle Big Data Appliance es la escalabilidad para poder solventar necesidades de procesamiento de los clientes. Por ello, es posible conectar múltiples componentes de procesamiento para extender el hardware.

[53] Oracle (2016). “Oracle Engineered Systems for Big Data [figura]”. Extraída el 17/05/2016 Sitio Web: https://docs.oracle.com/cd/E40622_01/doc.21/e40656/concepts.htm#BIGUG108

Para realizar esta escalabilidad de hardware, se pueden agregar nuevos servidores, los cuales pueden ser controlados por el servidor maestro como computadoras adicionales, en los que Cloudera podrá almacenar, y procesar información de la misma forma que en los servidores base.

Las necesidades de cada caso de negocio que puede resolver Oracle Big Data Appliance varían mucho, por eso existen diversas configuraciones de hardware siempre necesitando un nodo maestro y un nodo esclavo como mínimo, esto es, al menos 2 servidores Solaris.

Cada nodo cuenta con 16 cores de procesamiento y memoria entre 64 y 512 GB.

Las consideraciones para calcular el número de servidores son:

- Carga de trabajo
- Capacidad de procesamiento de cada servidor
- Necesidades de alta disponibilidad
- Necesidades de escalabilidad
- Necesidades de confiabilidad[54].

Debido a estas distintas necesidades Oracle Big Data Appliance ofrece 3 tipos de versiones diferentes

- Big Data Appliance X3-2 Started Rack con capacidad de 6 nodos
- Big Data Appliance X3-2 In Rack Expansion con capacidad de 6 nodos y expandible
- Big Data Appliance X3-2 Full Rack con capacidad de 18 nodos

Este modelo permite seleccionar de un inicio una configuración adecuada y poderla expandir cuando sea necesario.

[54] Plunkett T, Macdonald B, Nelson B, Hornick M, Sun H, Mohiuddin K, Harding D, Mishra G, Stackowiak R, Laker K & Segleau D. (2013). Oracle Big Data Handbook. . USA: McGRAW-HILL. P 82

Bases de datos Oracle NoSQL

Una base de datos NoSQL (Not Only SQL) es una base de datos cuya clasificación difiere de las bases de datos Relacionales. Existen varias diferencias entre las bases de datos relacionales y las bases NoSQL. Una de las más importantes es que soportan datos no estructurados. A diferencia de las bases de datos relacionales, no es necesario que los datos se encuentren normalizados o estructurados para ser consultados.

Una base de datos relacional tiene como principal objetivo cumplir con las características ACID (Atomicity, Consistency, Isolation, Durability); para estas bases es sumamente importante la consistencia de los datos y a través de años de desarrollo soportan complejas operaciones de consulta. En contraste, las bases de datos NoSQL no tienen el soporte operacional que tienen las bases de datos Relacionales; su objetivo principal es soportar la ejecución de consultas a través de grandes conjuntos de datos distribuidos horizontalmente en múltiples nodos de datos proporcionando una latencia mínima.

Las bases de datos NoSQL no tienen el desarrollo que las bases de datos relacionales han alcanzado por años. Dado que su propósito de estas bases es poder consultar datos no estructurados en grandes cantidades, la integridad de datos no es un objetivo primordial en este tipo de bases. Usualmente este tipo de bases han sido optimizadas para poder realizar este tipo de consultas

Como parte de la plataforma Oracle Big Data Apliance se incluye la Oracle NoSQL Database, la cual accede directamente al HDFS para poder hacer consultas sobre los datos almacenados para Big Data

Las bases de datos NoSQL están diseñadas para cumplir con las características BASE (Basically Available, Soft State, Eventually Consistent)

Basically Available: A través de la replicación de datos, una base de datos NoSQL debe soportar una alta disponibilidad y tolerancia a fallos de software y hardware como cualquier componente de Big Data. La distribución de los bloques de archivos a través del arreglo de nodos permite soportar inclusive la falla simultánea de varios nodos de datos sin perder información.

Soft State: Una base de datos Hard State implica, que un dato almacenado en ella tendrá siempre el mismo valor desde que fue almacenado, hasta que un usuario lo altere de manera consiente. Para las bases NoSQL, los datos tenderán a cambiar incluso sin la intervención del usuario, esto es provocado por el modelo de consistencia que manejan.

Eventually Consistent: Aun cuando las bases de datos NoSQL tienden a la inconsistencia debido al número de copias de datos que hay en sus diferentes nodos, En algún punto en el

futuro, durante su uso, este tipo de bases tendrán consistencia nuevamente. Esto es asegurado y realizado por el propio sistema a través de la ordenación de bloques de datos.

Las bases NoSQL fueron diseñadas para soportar aplicaciones con un volumen de datos en gran escala. Estas aplicaciones típicamente requieren operaciones de escritura y la lectura se basa en una llave primaria de un mapa multidimensional; tal es el caso de almacenar y leer páginas web. Proveen para ello escalamiento horizontal sobre un gran número de servidores su diseño debe soportar:

- Manejo de datos horizontalmente distribuido
- Alta tolerancia a fallos
- Procesamiento de lectura y escritura con un bajo tiempo de respuesta
- Esquema de almacenamiento flexible y soporte a varios tipos de datos no estructurados
- Centradas en el desarrollo de aplicaciones a través de un API
- Administración y desarrollo de fácil uso.

Las aplicaciones de tipo NoSQL tienen requerimientos simples y utilizan consultas simples. Debe de tenerse en cuenta para el uso de este tipo de bases, que dado que con cuentan con el desarrollo que tiene una base de datos relacional no soportan procedimientos almacenados u operaciones JOIN

Oracle NoSQL no substituye a las bases de datos tradicionales, dentro de Big Data Appliance, Oracle NoSQL es un componente complemento de las bases de datos relacionales de Oracle, que ayuda a cubrir casos de negocio diferentes que incluyen la lectura y almacenamiento de datos proveniente de diferentes fuentes como pueden ser sensores o componentes de IoT, y la consulta unificada[55]

Actualmente las bases de datos NoSQL están cobrando relevancia y son comunes de encontrar, sin embargo Oracle Big Data Appliance, fue el primer appliance en incluir soporte para este tipo de bases de datos[56]

El usar este tipo de base de datos tiene ventajas y desventajas. Dave Sagleau, quien fuera director de administración de producto de Oracle comenta algunas de ellas en una entrevista para la revista Forbes: “Las ventajas incluyen la capacidad de escalar la capacidad de cómputo y capacidad de almacenamiento horizontalmente sobre una amplia gama de recursos de hardware, provee consultas simples y rápidas, y provee un enfoque flexible y simple de administración de esquemas. Las desventajas son la falta de apoyo a las consultas complejas, la falta de apoyo para las combinaciones de varias tablas, soporte de transacciones limitado, y tener que aprender un nuevo enfoque de la tecnología de base de datos".

[55] Wolf A. (2014). Do You Know NoSQL?. Mayo 17 2016, de Forbes Sitio web: http://www.forbes.com/sites/oracle/2013/08/22/do-you-know-nosql/#6d1cb48872f7

[56] Wolf A. (2014). Do You Know NoSQL?. Mayo 17 2016, de Forbes Sitio web: http://www.forbes.com/sites/oracle/2013/08/22/do-you-know-nosql/#6d1cb48872f7

Conclusiones

Después de la revisión hecha a la tecnología Big Data, es importante recalcar los siguientes puntos sobre que es Big Data, qué alcances y aplicaciones tiene.

- Big Data es una tecnología derivada de la necesidad de analizar grandes conjuntos de datos en tiempos de respuesta cortos y con una escalabilidad no limitada.
- MapReduce es un modelo de programación en paralelo, en el que se basó la evolución de esta tecnología
- Yahoo genera una implementación de MapReduce en código abierto denominada proyecto Hadoop para adaptarla a la problemática de su buscador.
- Una solución de Big Data debe aparecer cuando hay problemas a resolver con características de las 3 V's Velocidad, Volumen y Variedad de datos.
- El verdadero valor que se puede obtener de los datos está basado en el conocimiento que se puede extraer de ellos.
- El conocimiento que se puede obtener de un análisis de Big Data es propuesto y construido por un Data Scientist.
- Existen aplicaciones que ayudan a disminuir la latencia para operaciones de consulta de datos, como son HBASE y Big Table que son complementos de los sistemas de ficheros GFS y HDFS.
- HDFS es el sistema de archivos que utiliza Hadoop y fue desarrollado basándose en GFS.

Existen elementos que ayudan a tomar una elección sobre cuál es la plataforma o appliance que se debe elegir en un proyecto de Big Data. Para ello considerar la situación de la empresa o departamento que va a implementar el proyecto, y las necesidades del problema a resolver, es una base adecuada para la selección.

Considérese el hecho de que Hadoop, es libre de pago para descargar, y que de manera nativa soporta correr sobre hardware que no sea especializado, debe considerarse como una buena opción la selección de Hadoop sin ningún appliance cuando se cumplen al menos uno de los siguientes supuestos:

1. Se está aprendiendo el uso de Big Data con propósitos educativos.

2. Se está evaluando la tecnología como solución a problemas de negocio actuales, pero aún no hay una decisión de implementarla.

3. Se cuenta con hardware en el que se desea instalar la solución de Big Data

4. Se tiene delimitada desde un inicio el tamaño de la solución y esta no tenderá a crecer frecuentemente en el futuro.

5. No se desea comprometerse con un proveedor en específico para la estrategia de crecimiento de la solución.

Si se desea plantear una solución grande, que incluya un appliance, es posible seleccionar cualquiera de las opciones entre SAP HANA y Oracle Big Data Appliance, ya que las dos se pueden utilizar para múltiples escenarios de casos de negocio; tomando en cuenta esto, las siguiente lista es de consideraciones para seleccionar Oracle Big Data Appliance sobre SAP HANA.

1. Si ya se cuenta con algún appliance de Oracle como Oracle Exadata u Oracle Exalitycs, ya que el componente de red InfinitiBand entre estos productos le brinda ventajas de desempeño en red sobre una mezcla de productos.

2. Si se cuenta con bases de datos Oracle que desea incluir en la solución de Big data, ya que Big Data Appliance está optimizada para la extracción de datos de la manera más rápida para este tipo de bases de datos.

3. Existe una preferencia de utilizar Cloudera sobre Hadoop como motor de Big Data

4. Se cuenta con personal que tiene experiencia sobre productos Oracle

5. Se tiene productos Oracle que no sean appliances y que desea incluir en la solución.

Para seleccionar a HANA sobre Oracle Big Data Appliance también hay una serie de consideraciones básicas que hay que tomar en cuenta, y que ayudan a tomar esta decisión

1. Se cuenta con productos SAP que se desean incluir en la solución de Big Data.
2. La solución a implementar tiene alta demanda sobre computación in-memory
3. Se tiene preferencia de bases de datos SyBase sobre otro tipo,
4. Se tiene personal que conoce los productos SAP

La decisión de implementar Hadoop sobre un arreglo de hardware que ya se tiene o implementar un appliance, no es una decisión de un solo elemento a considerar. Implementar Hadoop en un arreglo o clúster de computadoras tiene más implicaciones que solo tener el hardware, y descargar el software; para un proyecto serio, es necesario invertir en soporte o tener el personal capacitado en estas tecnologías, invertir en infraestructura de red y las configuraciones necesarias para conectar a Hadoop con otras tecnologías, sin embargo es posible hacerlo de esta manera para grandes proyectos, tal es el caso de Facebook o Goolge, quienes usan implementaciones de este tipo.[57]

Entonces para poder hacer una selección correcta de la tecnología a aplicar, es importante y mandatorio tener una visión completa de lo que queremos resolver con Big Data, de los recursos con los que se cuenta, tanto económicos como humanos, y de la estrategia que se quiere seguir.

[57] Sheldon R. (ND). The Oracle big data strategy: Weighing appliance vs. commodity. Mayo 22 2016, de TechTarget Sitio web: http://searchoracle.techtarget.com/feature/The-Oracle-big-data-strategy-Weighing-appliance-vs-commodity

Esto ayudará con esta decisión, y la experiencia termina siendo un importante factor para esta selección.

Glosario

Terabyte: Unidad de medida de 1024 Gigabytes

Petabyte: Unidad de medida de 1024 Terabytes

Exabyte: Unidad de medida de 1024 Petabyte

Zettabyte: Unidad de medida de 1024 Exabytes

Yotabyte: Unidad de medida de 1024 Zettabytes

GPS: Sistema de posicionamiento global

Trending Topic: Tendencia o tema del momento.

Atomicidad de operaciones: Se define una operación atómica cuando en la operación se completa en su totalidad.

MP3: Formato de compresión digital para la transmisión de archivos de audio.

Cluster: Conglomerado de computadoras unidas a través de una red de alta velocidad.

CODASYL: Acrónimo para Conference on Data Systems Languages. Conferencia que fue encargada de estandarizar los sistemas de bases de datos.

COBOL: Acrónimo de COmon Business Oriented Languaje. Fue un lenguaje de programación para manejo de ficheros.

IMS: Sistema de manejo de información para la NASA durante el programa APOLO.

ANSI / SPARC: Acrónimo para American National Standards Institute, Standards Planning and Requirements Committee

JDBC: Acrónimo para Java Data Base Conectivity. Conector de base de datos de java desarrollado inicialmente para bases de datos orientadas a objetos.

ODBC: Acrónimo para Open Data Base Conectivity. Conector de base de datos estándar desarrollado por Microsoft.

OLAP: Acrónimo para On-Line Analytical Processing

Data-warehouses: Término anglosajón para referirse a un almacén de datos, el cual es un conjunto de múltiples bases de datos relacionadas entre sí.

DataMart: Porción o vista parcial de un Data-warehouse

Streaming: Término anglosajón para referirse a la transmisión de datos continua.

Data Platform: Término usado para almacenes de datos de Big Data con una API de interfaz.

API: Acrónimo para Application Programming Interface

CDDB: Acrónimo para Compact Disc Database

Samples: Porción de un sonido grabado.

Search Engines: Motores de búsqueda de páginas de Internet.

MapReduce: Modelo de programación generado por Google para poder procesar grandes almacenes de datos.

CBIR: Acrónimo de Content-based image retrieval

Data Science: Se refiere a la extracción del conocimiento a partir del análisis de datos.

Internet of Things: Concepto que se aplica a la interconexión de objetos a Internet para reportar datos.

Batch: Término para el procesamiento por lotes.

HPC: Acrónimo para high performance computing

SAN: Acrónimo para Storage Area Network.

Dashboards: Interfaz gráfica de usuario para controlar o leer información

KPI: Acrónimo para Key Performance Indicator.

Timestamp: Marca de tiempo que denotan la fecha y la hora de algún registro.

HDFS: Acrónimo para Hadoop Data File System.

RAM: Acrónimo para Random Access Memory.

OLTP: Acrónimo para OnLine Transaction Processing.

In-Memory: Término utilizado para computación que corre 100% en memoria RAM sin utilizar almacenamiento persistente.

Core: Núcleo dentro de un procesador.

Benchmark: Técnica utilizada para medir el rendimiento de un sistema o componente del mismo

LAMP: Clasificación para las tecnologías basadas en Linux, Apache, My SQL, PHP

TCP/IP: Protocolos de comunicación para una red.

PLSQL: Acrónimo para Procedural Language/Structured Query Language. Lenguaje de consulta para las bases de datos de Oracle.

NoSQL: Acrónimo para Not Only SQL. Se refiere a la clasificación de bases de datos no estructuradas.

ACID: Acrónimo para Atomicity, Consistency, Isolation, Durability

BASE: Acrónimo para Basically Available, Soft State, Eventually Consistent

Bibliografía

Tubella I. (2005). Sociedad del conocimiento. Barcelona España: UOC.

Loukides M. (2011). Big Data Now. California USA: O'Reilly Media.

Barlow M. (2013). Real Time Big Data Analytics: Emerging Architecture. California USA: O'Reilly Media.

Word J. (2012). SAP HANA Essentials. USA: Epistemy Press LLC.

Plunkett T, Macdonald B, Nelson B, Hornick M, Sun H, Mohiuddin K, Harding D, Mishra G, Stackowiak R, Laker K & Segleau D. (2013). Oracle Big Data Handbook. . USA: McGRAW-HILL.

White T. (2012). Hadoop®. The Definitive Guide. California USA: O'Reilly Media.

Silberschatz A, F. Korth H. (2013). Fundamentos de bases de datos. España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA.

Mohanty S, Jagadeesh M, Srivatsa H. (2013). Big Data imperatives. USA: Apress.

Merino I. (2015). Leyes y paradigmas de la tecnología (II): Ley de Moore. Agosto 14, 2015, de Hipertextual Sitio web: http://hipertextual.com/2015/04/ley-moore-50-aniversario

Simonite T. (2012). IBM Builds Biggest Data Drive Ever. Mayo 4, 2015, de MIT Technology Review Sitio web: https://www.technologyreview.com/s/425237/ibm-builds-biggest-data-drive-ever/

Simonite T. (ND). IBM 350 disk storage unit. Abril 8, 2015, de IBM Sitio web: https://www-03.ibm.com/ibm/history/exhibits/storage/storage_350.html

Cruz F. (2011). Herman Hollerith. Febrero 3, 2015, de Columbia University PhD Sitio web: http://www.columbia.edu/cu/computinghistory/hollerith.html

Twitter. (ND). Twitter Usage Statistics. Marzo 16, 2015, de Twitter Sitio web: http://www.internetlivestats.com/twitter-statistics/

IBM. (ND). The Four V's of Big Data. Marzo 16, 2015, de IBM Sitio web: http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Dean J &Ghemawat S. (2004). MapReduce: Simplified Data Processing on Large Clusters. Enero 4, 2015, de Google Sitio web: http://research.google.com/archive/mapreduce.html

Datta R, Li J & Z. Wang J. (2005). Content-Based Image Retrieval - Approaches and Trends of the New Age. Diciembre 29, 2014, de Stanford Sitio web: http://infolab.stanford.edu/~wangz/project/imsearch/review/ACM05/datta.pdf

New York University. (2013). What is Data Science?. Enero 3, 2015, de New York University Sitio web: http://datascience.nyu.edu/what-is-data-science/

Schroeck M, Shockley R, Smart J, Romero D & Tufano P. (ND). Analytics: The real-world use of big data. Febrero 3, 2016, de IBM Global Business Services & University Of Oxford Sitio web: https://www.ibm.com/smarterplanet/global/files/se__sv_se__intelligence__Analytics_-_The_real-world_use_of_big_data.pdf

Hortonworks . (2016). Horton Works Quick Facts. Mayo 13 2016, de Hortonworks Sitio web: http://hortonworks.com/about-us/quick-facts/

SEAGATE TECHNOLOGY, INC. (1991). ST-41600N (97501-16G) Elite-1 FH SCSI-2 . Noviembre 27, 2014, de SEAGATE TECHNOLOGY, INC Sitio web: ftp://ftp.seagate.com/techsuppt/scsi/st41600n.txt

Demassieux N. (2015). Six challenges for Big Data. Marzo 3, 2016, de Orange Research Sitio web: https://recherche.orange.com/en/six-challenges-for-big-data/

Edjlali R & A. Beyer M . (2016). Magic Quadrant for Data Warehouse and Data Management Solutions for Analytics. Marzo 7 2016, de Gartner Sitio web: https://www.gartner.com/doc/reprints?id=1-2SGJMI2&ct=151118&st=sb

The Apache Software Foundation. . (2014). Apache Hadoop Releases. Agosto 3 2015, de The Apache Software Foundation. Sitio web: https://hadoop.apache.org/releases.html

Jung T. (2012). SAP HANA Extended Application Services. Abril 3 2016, de SAP Sitio web: http://scn.sap.com/community/developer-center/hana/blog/2012/11/29/sap-hana-extended-application-services

The Apache Software Foundation. (ND). Appendix H. HBase History. Junio 3 2015, de Cloudera Sitio web: http://archive.cloudera.com/cdh5/cdh/5/hbase-0.98.6-cdh5.3.2/book/hbase.history.html

Oracle. (2016). Oracle Big Data Appliance. Mayo 13 2016, de Oracle Sitio web: https://www.oracle.com/es/engineered-systems/big-data-appliance/index.html

Cloudera. (ND). About Cloudera. Abril 13 2015, de Cloudera Sitio web: http://www.cloudera.com/about-cloudera.html

MapR. (ND). MapR Converged Data Platform: MapR Editions. Mayo 13 2016, de MapR Sitio web: https://www.mapr.com/products/mapr-distribution-editions

Hortonworks. (2016). Hortonworks : Quick Facts. Mayo 13 2016, de Hortonworks Sitio web: http://hortonworks.com/about-us/quick-facts/

Oracle. (ND). Defying Conventional Wisdom. Mayo 10 2016, de Oracle Sitio web: http://www.oracle.com/us/corporate/profit/p27anniv-timeline-151918.pdf

El PaíS. (2009). Oracle adquiere Sun Microsystems por 5.710 millones. Mayo 08 2016, de El PaíS Sitio web: http://tecnologia.elpais.com/tecnologia/2009/04/20/actualidad/1240216080_850215.html

Oracle. (2014). The Data Warehouse Insider. Agosto 5 2015, de Oracle Sitio web: https://blogs.oracle.com/datawarehousing/entry/updated_price_comparison_for_big

Oracle. (ND). Oracle Exadata Base Machine. Mayo 09 2016, de Oracle Sitio web: https://www.oracle.com/es/engineered-systems/exadata/index.html

Oracle. (ND). Oracle Big Data Appliance. Mayo 01 2016, de Oracle Sitio web: https://www.oracle.com/es/engineered-systems/big-data-appliance/index.html

Darrow B. (2011). Oracle Big Data Appliance stakes big claim. Mayo 09 2016, de GIGAOM Sitio web: https://gigaom.com/2011/10/03/oracle-big-data-appliance-stakes-big-claim/

Pierre J. (2014). The Data Warehouse Insider. Mayo 08 2015, de ORACLE Sitio web: https://blogs.oracle.com/datawarehousing/entry/updated_price_comparison_for_big

Oracle. (ND). Big Data Appliance Software User's Guide. Mayo 17 2016, de Oracle Sitio web: https://docs.oracle.com/cd/E40622_01/doc.21/e40656/concepts.htm#BIGUG107

Wolf A. (2014). Do You Know NoSQL?. Mayo 17 2016, de Forbes Sitio web: http://www.forbes.com/sites/oracle/2013/08/22/do-you-know-nosql/#6d1cb48872f7

MIT (2011). IBM Builds Biggest Data Drive Ever. De MIT TECHNOLOGY REVIEW Sitio Web: http://www.technologyreview.com/news/425237/ibm-builds-biggest-data-drive-ever/page/2/

Montoro S. (2013). El ecosistema de productos Big Data. Abrill 30 2014, de La Pastilla Roja Sitio web: http://lapastillaroja.net/2013/03/eco-bigdata/

Mysore D, Khupat S & Jain S. (2013). Big data architecture and patterns, Part 1: Introduction to big data classification and architecture. Agosto 3, 2014, de IBM developerWorks Sitio web: http://www.ibm.com/developerworks/library/bd-archpatterns1/index.html

Mysore D, Khupat S & Jain S. (2013). Big data architecture and patterns, Part 2: How to know if a big data solution is right for your organization. Agosto 3, 2014, de IBM developerWorks Sitio web: http://www.ibm.com/developerworks/library/bd-archpatterns2/index.html

Mysore D, Khupat S & Jain S. (2013). Big data architecture and patterns, Part 3: Understanding the architectural layers of a big data solution. Agosto 3, 2014, de IBM developerWorks Sitio web: http://www.ibm.com/developerworks/library/bd-archpatterns3/index.html

Mysore D, Khupat S & Jain S. (2013). Big data architecture and patterns, Part 4: Understanding atomic and composite patterns for big data solutions. Agosto 3, 2014, de IBM developerWorks Sitio web: http://www.ibm.com/developerworks/library/bd-archpatterns4/index.html

Mysore D, Khupat S & Jain S. (2013). Big data architecture and patterns, Part 5: Apply a solution pattern to your big data problem and choose the products to implement it. Agosto 3, 2014, de IBM developerWorks Sitio web: http://www.ibm.com/developerworks/library/bd-archpatterns5/index.html

Chang F, Dean J, Ghemawat S, C. Hsieh W, A. Wallach D, Burrows M, Chandra T, Fikes A & E. Gruber R. (2006). Bigtable: A Distributed Storage System for Structured Data. Mayo 01 2015, de Google Sitio web: http://research.google.com/archive/bigtable.html

Soto L. (2013). Bases de Datos En Memoria. Mayo 101 2015, de Software Guru Sitio web: http://sg.com.mx/revista/38/bases-datos-memoria#.V0UhiY-cHIV

The Apache Software Foundation. (2013). PDF HDFS Architecture Guide. Marzo 30 2014, de PDF HDFS Architecture Guide Sitio web: http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

SAP. (2016). Administration Guide. Mayo 13 2015, de SAP Sitio web: http://help.sap.com/businessobject/product_guides/sbods42/en/ds_42_admin_en.pdf

Rouda N. (2014). Getting Real About Big Data: Build Versus Buy. Abril 27 2015, de Oracle Sitio web: http://www.oracle.com/us/corporate/analystreports/esg-getting-real-bigdata-2228170.pdf

Page L & Brin S. (2014). Descripción General de la Empresa. Agosto 4 2014, de Google Sitio web: http://www.google.com/intl/es-419_mx/about/company/

SAP. (2015). Casos de Éxito. Abril 3 2016, de SAP Sitio web: http://go.sap.com/latinamerica/about/customer-testimonials/finder.html

Sheldon R. (ND). The Oracle big data strategy: Weighing appliance vs. commodity. Mayo 22 2016, de TechTarget Sitio web: http://searchoracle.techtarget.com/feature/The-Oracle-big-data-strategy-Weighing-appliance-vs-commodity