



Universidad Nacional
Autónoma de México



Facultad de Ingeniería

EL MODELO DE COMPUTACIÓN DISTRIBUIDA DE HADOOP

Sistemas Operativos

León Gómez Erick

Martínez Jiménez Israel

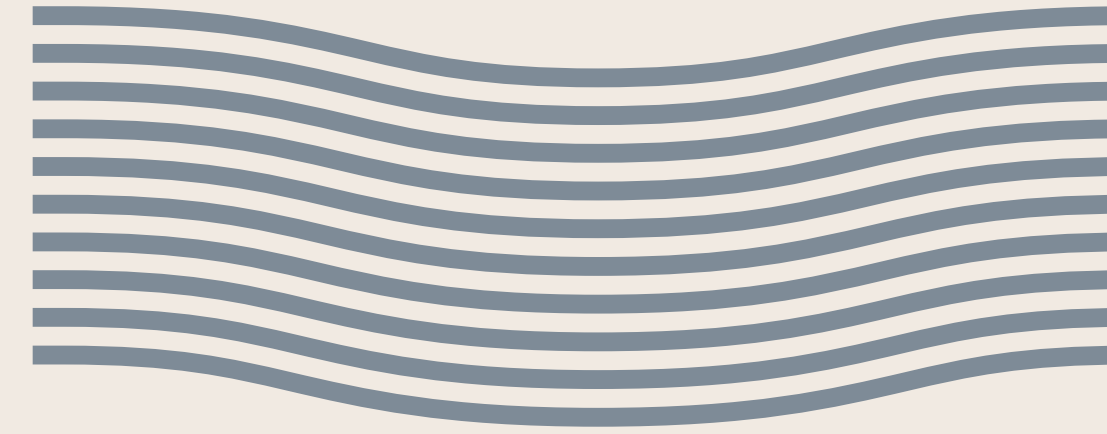
ÍNDICE

1. Introducción
2. Computo Distribuido
3. Arquitectura de Hadoop
4. HDFS (Hadoop Distributed File System)
5. MapReduce
6. YARN (Yet Another Resource Negotiator)
7. Cluster de datos
8. Conclusiones
9. Referencias bibliográficas

INTRODUCCIÓN

- Plataforma para el manejo eficiente de grandes volúmenes de datos mediante clústeres y un modelo de programación simple.
- Escrito en Java, permite desarrollar aplicaciones distribuidas con alta escalabilidad.
- Ideal para programadores no familiarizados con ambientes distribuidos.
- Abstrae complejidades como la paralelización de tareas y la gestión de procesos.
- Ofrece soluciones robustas para balanceo de carga y tolerancia a fallos.





Origen

- Desarrollado por Doug Cutting, creador de Apache Lucene.
- Inspirado en el concepto de MapReduce introducido por Google en 2004.
- Nació de Apache Nutch y se separó como un subproyecto independiente de Lucene en 2006.
- Reconocido como proyecto independiente de alto nivel en Apache en 2008.

Adopción

- Inicialmente respaldado por empresas como Yahoo y Facebook.
- Ampliamente utilizado en finanzas, tecnología, telecomunicaciones, medios de comunicación, entre otros sectores.
- Destacado por su escalabilidad, acceso paralelo a datos distribuidos y robusto sistema de seguridad.






CÓMPUTO DISTRIBUIDO

Definición

- Método que combina varias computadoras para resolver problemas complejos como una única entidad potente.
- Redes de computadoras unidas para ofrecer recursos a gran escala.

Acoplamiento en Sistemas Distribuidos

- Acoplamiento Flexible:
 - Componentes débilmente conectados.
 - Ejemplo: Clientes y servidores con comunicación temporalmente separada.
 - Acoplamiento Ajustado:
 - Redes rápidas conectan computadoras en clústeres.
 - Todas realizan tareas similares bajo un middleware central.
- 



VENTAJAS

Escalabilidad

- Adición dinámica de nodos según la carga de trabajo.

Tolerancia a Fallos

- Continúa operando incluso si una computadora falla.

Consistencia

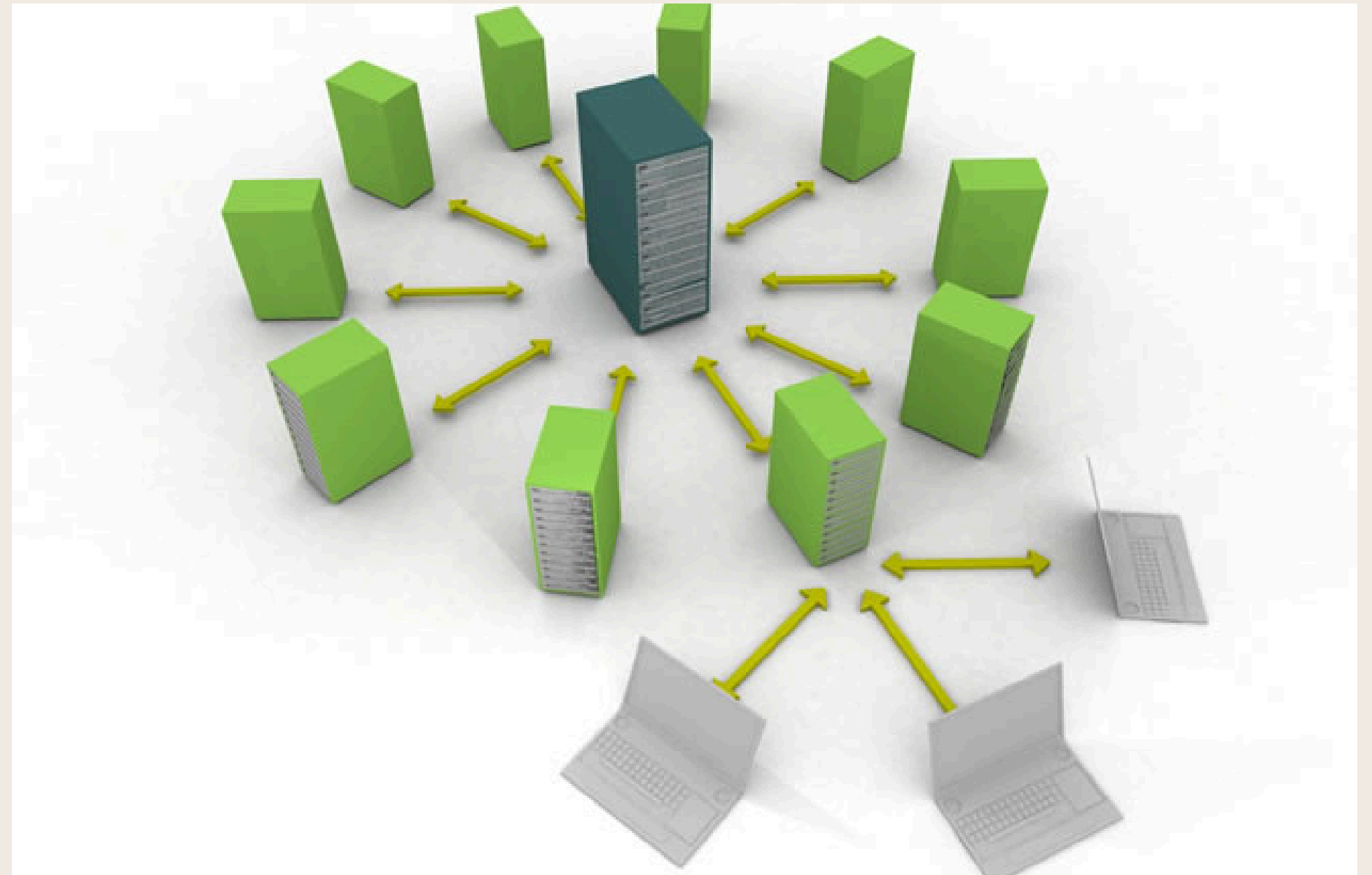
- Gestiona automáticamente la coherencia de datos entre computadoras.

Abstracción

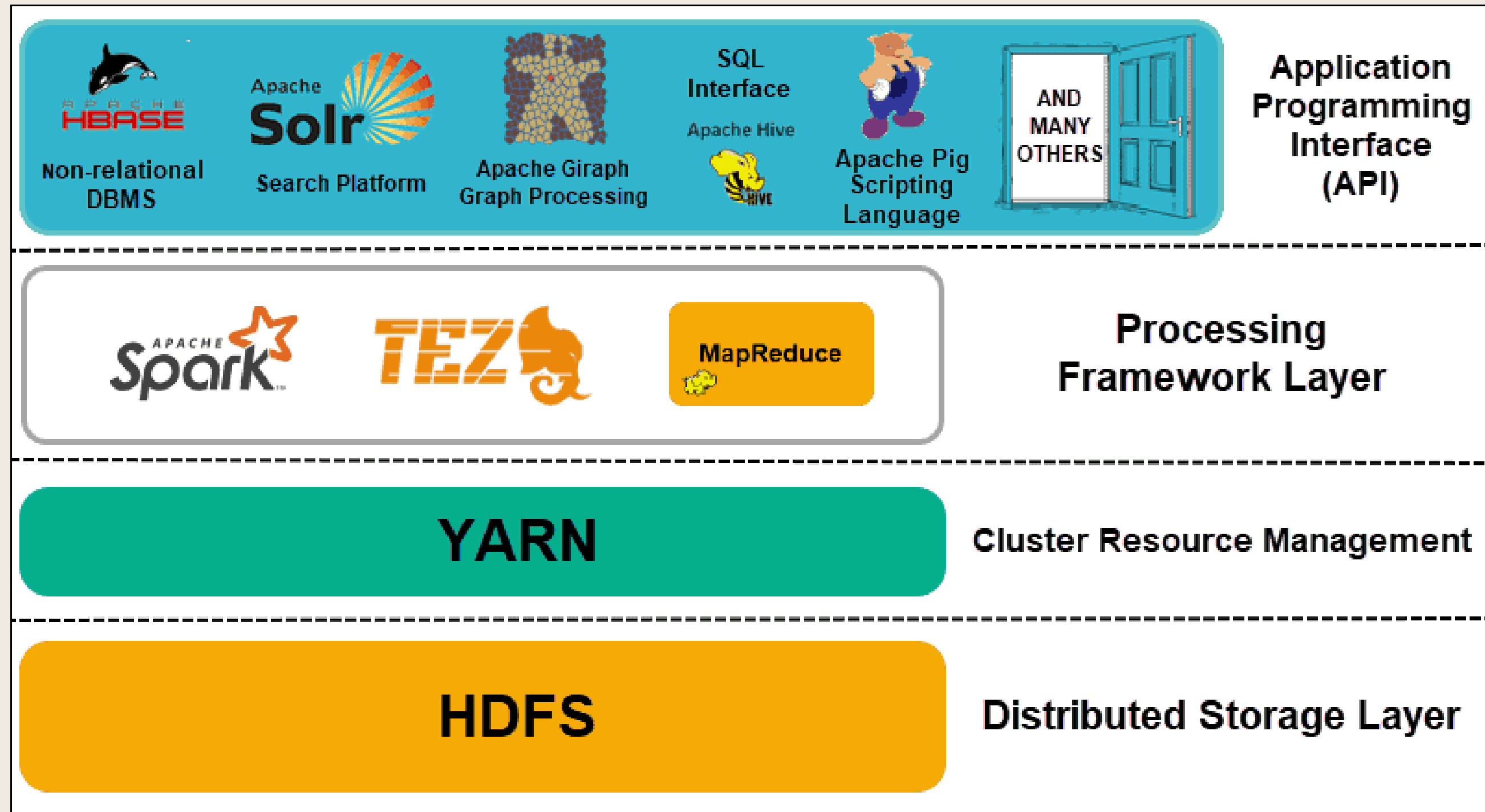
- Interacción como una sola entidad, sin preocuparse por detalles de hardware.

Eficiencia

- Uso óptimo de recursos para un rendimiento más rápido y estable.




ARQUITECTURA HADOOP





ARQUITECTURA HADOOP

- **Capa de Almacenamiento Distribuido:**

- Cada nodo del clúster de Hadoop cuenta con su propio espacio en disco, memoria, ancho de banda y procesamiento.
 - Los datos entrantes se dividen en bloques de datos individuales almacenados en la capa de almacenamiento distribuido HDFS.
 - HDFS, asumiendo la falibilidad de discos y nodos, almacena tres copias de cada conjunto de datos en el clúster.
 - El nodo maestro de HDFS (NameNode) mantiene la metainformación de cada bloque de datos y sus réplicas.
- 

ARQUITECTURA HADOOP

- **Gestión de Recursos del Clúster:**


- YARN, separando las funciones de gestión de recursos y procesamiento de datos, optimiza la coordinación de nodos para compartir recursos entre múltiples aplicaciones y usuarios.
- YARN permite la asignación de recursos a diferentes marcos de trabajo desarrollados para Hadoop, como Apache Pig, Hive, Giraph, y Zookeeper.

- **Capa de frameworkde Procesamiento:**

- Consiste en marcos que analizan y procesan conjuntos de datos estructurados y no estructurados.
- Las operaciones de mapeo, mezcla, ordenamiento, fusión y reducción de datos se distribuyen en múltiples nodos, idealmente cerca de los servidores donde residen los datos.
- Marcos como Spark, Storm y Tez facilitan el procesamiento en tiempo real y la consulta interactiva, mejorando la eficiencia del uso de HDFS.



ARQUITECTURA HADOOP

- **Interfaz de Programación de Aplicaciones (API):**
 - La introducción de YARN ha llevado a la creación de nuevos marcos de procesamiento y APIs, acompañando el crecimiento de big data.
 - Herramientas para plataformas de búsqueda, transmisión de datos, interfaces amigables, lenguajes de programación, mensajería, gestión de fallos y seguridad forman parte integral del ecosistema de Hadoop.
- 



HDFS (HADOOP DISTRIBUTED FILE SYSTEM)


- Diseñado para ejecutarse en hardware de tipo commodity.
- Altamente tolerante a fallos y optimizado para hardware de bajo costo.
- Proporciona acceso de alto rendimiento a grandes conjuntos de datos.

Origen y Evolución

- Desarrollado inicialmente como infraestructura para Apache Nutch.
- Parte esencial del proyecto principal de Apache Hadoop Core.



OBJETIVOS DE HDFS

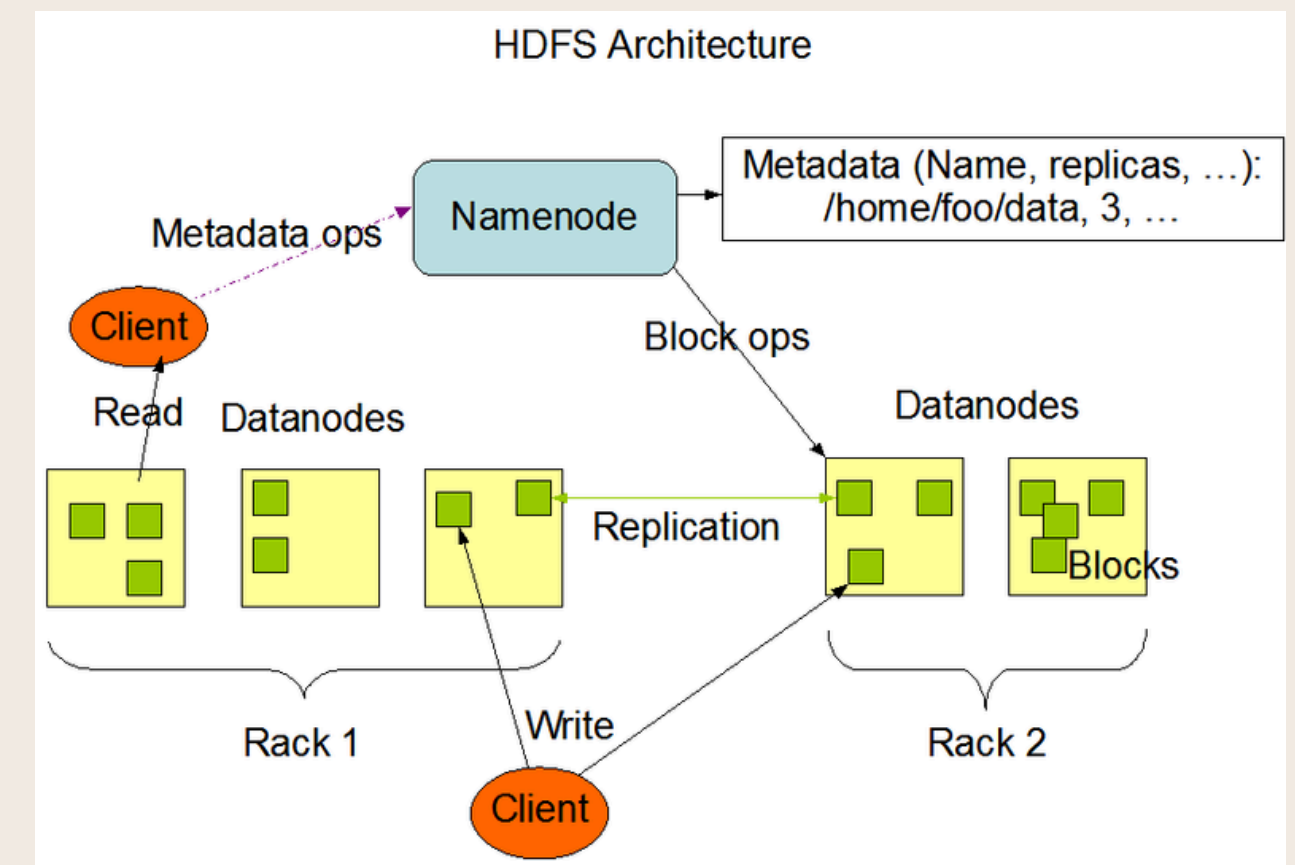
- Tolerancia a Fallos
 - Detectar y recuperarse automáticamente de fallos de hardware.
 - Acceso de Datos en Streaming
 - Priorizar acceso eficiente a grandes conjuntos de datos en lugar de baja latencia.
 - Manejo de Conjuntos de Datos Grandes
 - Optimizado para archivos de gigabytes a terabytes y escalabilidad a cientos de nodos.
 - Modelo de Coherencia Simple
 - Soportar escritura-una-vez-lectura-muchas-veces para simplificar la coherencia de datos.
 - Mover la Computación en Lugar de los Datos
 - Maximizar la eficiencia al ejecutar la computación cerca de los datos en lugar de mover los datos.
 - Portabilidad
 - Diseñado para ser fácilmente portable entre diferentes plataformas, fomentando su adopción generalizada.
- 



NAMENODES Y DATANODES

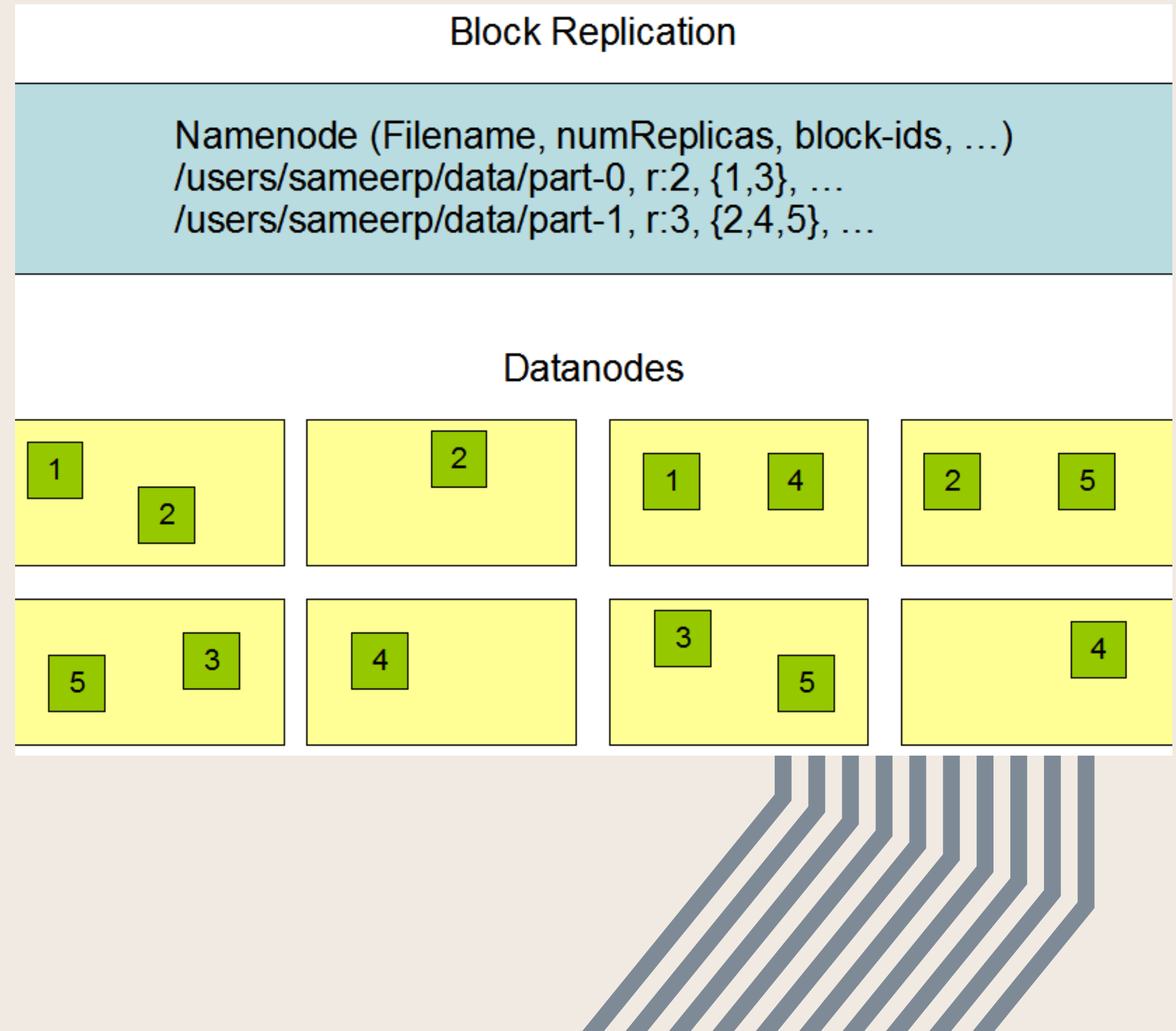
- Maestro/Esclavo
 - Un clúster de HDFS tiene un único NameNode (maestro) y múltiples DataNodes (esclavos).
- Funciones del NameNode
 - Gestiona el espacio de nombres del sistema de archivos.
 - Regula el acceso a los archivos por parte de los clientes.
 - Ejecuta operaciones de espacio de nombres como apertura, cierre y renombrado de archivos.
 - Determina la asignación de bloques a DataNodes.

- Funciones de los DataNodes
 - Gestionan el almacenamiento adjunto a los nodos en el clúster.
 - Sirven solicitudes de lectura y escritura de los clientes.
 - Crean, eliminan y replican bloques bajo la dirección del NameNode.
- Simplificación de la Arquitectura
 - La existencia de un único NameNode simplifica la arquitectura del sistema.
 - El NameNode es el repositorio de todos los metadatos de HDFS.
 - Los datos de usuario nunca pasan a través del NameNode para minimizar la carga de trabajo.



REPLICACIÓN DE DATOS

- Diseñado para Archivos Muy Grandes
 - Almacena archivos como secuencias de bloques en múltiples máquinas dentro de un clúster extenso.
- Replicación y Tolerancia a Fallos
 - Los bloques de un archivo se replican para garantizar la tolerancia a fallos.
 - Tamaño de bloque y factor de replicación configurables por archivo.
- Características de los Bloques
 - Todos los bloques, excepto el último, tienen el mismo tamaño.
 - Usuarios pueden iniciar un nuevo bloque sin llenar el último hasta el tamaño configurado.
- Gestión de Réplicas y Escritura
 - Aplicación puede especificar el número de réplicas de un archivo.
 - Los archivos en HDFS son de escritura única, con un único escritor en cualquier momento.
- Decisión de Replicación por el NameNode
 - NameNode toma decisiones sobre la replicación de bloques.
 - Recibe informes periódicos de estado (Heartbeat) y listas de bloques (Blockreport) de cada DataNode.






ROBUSTICIDAD

HDFS garantiza la fiabilidad mediante la detección y recuperación de fallos, re-replicación de bloques, rebalanceo automático y verificación de integridad de datos para asegurar la confianza en el almacenamiento y acceso distribuido de archivos.

- Tipos de Fallos
 - Fallos del NameNode, fallos del DataNode y particiones de red.

- Heartbeats y Re-Replicación
 - DataNodes envían 'Heartbeats' periódicamente al NameNode.
 - NameNode detecta y marca DataNodes sin 'Heartbeats' como muertos.
 - Inicia la re-replicación de bloques cuando el factor de replicación cae por debajo del valor especificado.
 - Rebalanceo del Clúster
 - HDFS admite esquemas automáticos de rebalanceo para mantener equilibrio de datos entre DataNodes.
 - Integridad de los Datos
 - Implementación de verificación de sumas de comprobación para garantizar la integridad de los bloques.
 - Cliente verifica la coincidencia de los datos recibidos con las sumas de comprobación almacenadas.
 - Fallo del Disco de Metadatos
 - El NameNode puede mantener múltiples copias de FsImage y EditLog para evitar corrupciones.
 - Actualización síncrona de copias para garantizar la consistencia en caso de fallos.
- 



MAPREDUCE

MapReduce ofrece un entorno eficaz y confiable para procesar grandes conjuntos de datos distribuidos en clústeres de hardware estándar, facilitando la implementación de aplicaciones escalables y tolerantes a fallos.

- Flujo de Trabajo
 - Divide el conjunto de datos en fragmentos independientes procesados por tareas de map en paralelo.
 - Ordena y dirige las salidas de mapas hacia tareas de reduce.
 - Almacenamiento de entrada/salida en sistema de archivos; gestión de tareas y reejecución de fallas.
- Configuración y Funcionamiento
 - Nodos de cómputo y almacenamiento integrados en un mismo conjunto.
 - Alta eficiencia al programar tareas en nodos con datos presentes, maximizando el ancho de banda del clúster.
- Componentes del Framework
 - ResourceManager (maestro), NodeManager (por nodo de clúster) y MRAppMaster (por aplicación).

YARN

- Introducción y Evolución
 - Administrador de recursos de código abierto introducido en Hadoop 2.0 para mejorar la eficiencia y flexibilidad en el procesamiento distribuido.
 - Reemplazó el modelo de procesamiento MapReduce de Hadoop 1.x.
- Funcionalidades Principales
 - Divide la gestión de recursos y la programación/seguimiento de trabajos en daemons separados.
 - ResourceManager (RM) global y ApplicationMaster (AM) por aplicación.
- Capacidades de Procesamiento
 - Permite un procesamiento más diversificado y dinámico de datos en el clúster de Hadoop.
 - Evolucionó para convertirse en un sistema operativo distribuido de gran escala para el procesamiento de Big Data.
- Arquitectura de YARN
 - Un ResourceManager a nivel global para gestionar recursos en el clúster.
 - Un ApplicationMaster por aplicación para gestionar la ejecución de tareas específicas.



Componentes de YARN

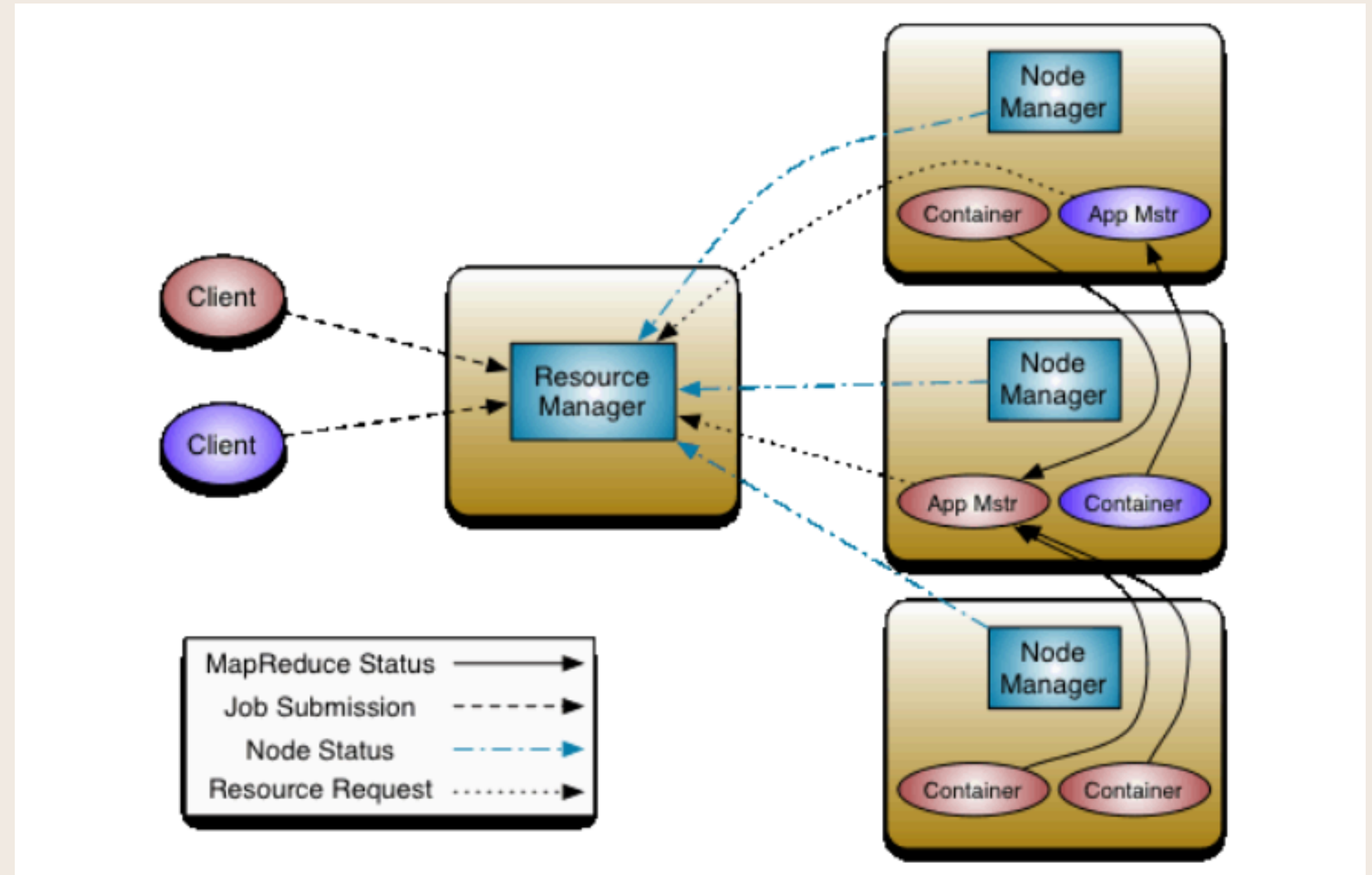
- ResourceManager (RM)
 - Autoridad final que arbitra los recursos entre todas las aplicaciones en el sistema.
- NodeManager (NM)
 - Agente por máquina responsable de los contenedores, monitorizando el uso de recursos y reportándolo al RM.
- ApplicationMaster (AM)
 - Negocia recursos del RM y trabaja con los NM para ejecutar y monitorizar tareas.

Funciones del ResourceManager

- Scheduler
 - Asigna recursos a aplicaciones en ejecución, sujeto a restricciones de capacidades y colas.
- ApplicationsManager
 - Acepta envíos de trabajos, negocia contenedores para ejecutar el AM y proporciona servicios de reinicio en caso de fallo.

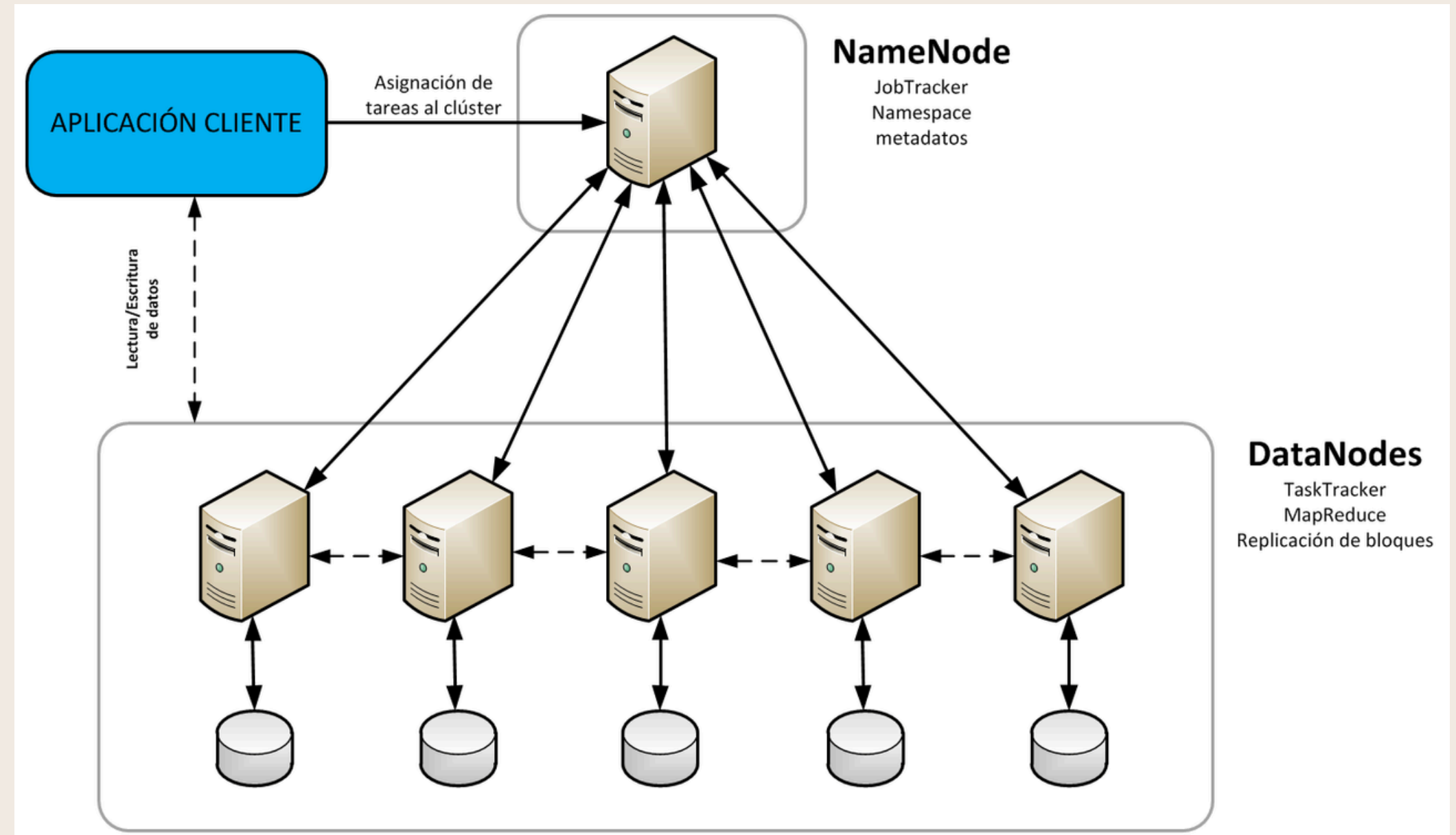
Funcionamiento del Scheduler

- Asigna recursos a aplicaciones según requisitos definidos.
- Utiliza políticas enchufables para particionar los recursos del clúster entre colas y aplicaciones.



CLUSTER DE DATOS

- Definición del Clúster de Hadoop
 - Grupo de unidades no convencionales conectadas con un servidor dedicado.
 - Funciona como unidad centralizada para el procesamiento de datos.
- Características Principales
 - Distribución de la carga de trabajo entre varios nodos.
 - Capacidad de agregar nodos fácilmente para aumentar la capacidad de procesamiento.
- Funciones Clave
 - Procesamiento de datos de manera eficiente y escalable.
 - Recopilación y almacenamiento de grandes volúmenes de datos.
 - Soporte para serialización de datos.






ARQUITECTURA DEL CLÚSTER DE HADOOP

- Componentes del Clúster
 - Nodos Maestros
 - Recopilación y almacenamiento de datos en HDFS.
 - Aplicación de computación en paralelo mediante MapReduce.
 - Nodos Esclavos
 - Responsables de la recopilación de datos y ejecución de cálculos.
 - Nodos Clientes
 - Utilizados para cargar datos en el Clúster de Hadoop.

VENTAJAS

- Solución rentable para almacenar y analizar datos.
 - Procesamiento rápido incluso con grandes volúmenes de datos.
 - Acceso fácil a diversas fuentes de datos, estructurados y no estructurados.
- 

CONCLUSIONES

Hadoop ha revolucionado la gestión y análisis de grandes volúmenes de datos, ofreciendo soluciones escalables y eficientes para empresas de todos los tamaños. Su arquitectura distribuida permite procesar y almacenar enormes cantidades de datos de manera más rápida y coste-efectiva que los sistemas tradicionales.

1. Hadoop puede crecer simplemente añadiendo nodos al cluster, sin necesidad de rediseñar sistemas de datos.
2. Utiliza hardware de propósito general, que es menos costoso y más fácil de obtener que las soluciones de almacenamiento y procesamiento especializadas.
3. Capaz de manejar diferentes tipos de datos, desde estructurados hasta no estructurados, facilitando así la integración y el análisis de datos variados.
4. Diseñado para continuar operando eficientemente ante un fallo, replicando los datos en varios nodos.

Hadoop se ha convertido en una herramienta clave en sectores como finanzas, salud, medios de comunicación, tecnología, y más, proporcionando insights que impulsan la toma de decisiones estratégicas.



FUENTES BIBLIOGRÁFICAS

Dean, J. y Ghemawat, S. (2004). Mapreduce: Simplified data processing on large clusters. In OSDI'04: Sixth Symposium on Operating System Design and Implementation, 137-150. San Francisco, CA

Hadoop, A. (2024b). Hdfs design. <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>.

Hadoop, A. (2024a). Apache hadoop yarn - introduction. <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>

Hadoop, A. (2024c). Mapreduce tutorial. <https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>

PhoenixNAP (2024). Apache hadoop architecture explained. <https://phoenixnap.com/kb/apache-hadoop-architecture-explaine>

Services, A.W. (2024). ¿qué es la computación distribuida? <https://aws.amazon.com/es/what-is/distributed-computin>

GeeksforGeeks (2024). Basics of hadoop cluster. https://www.geeksforgeeks.org/basics-of-hadoop-cluster/?ref=ml_l

The background is a solid light teal color. It features decorative elements: concentric circles in a light pinkish-beige color at the top and bottom centers, and chevron patterns made of multiple parallel lines in a slightly darker teal color on the left and right sides.

MUCHAS
GRACIAS

