DATOS MASIVOS I

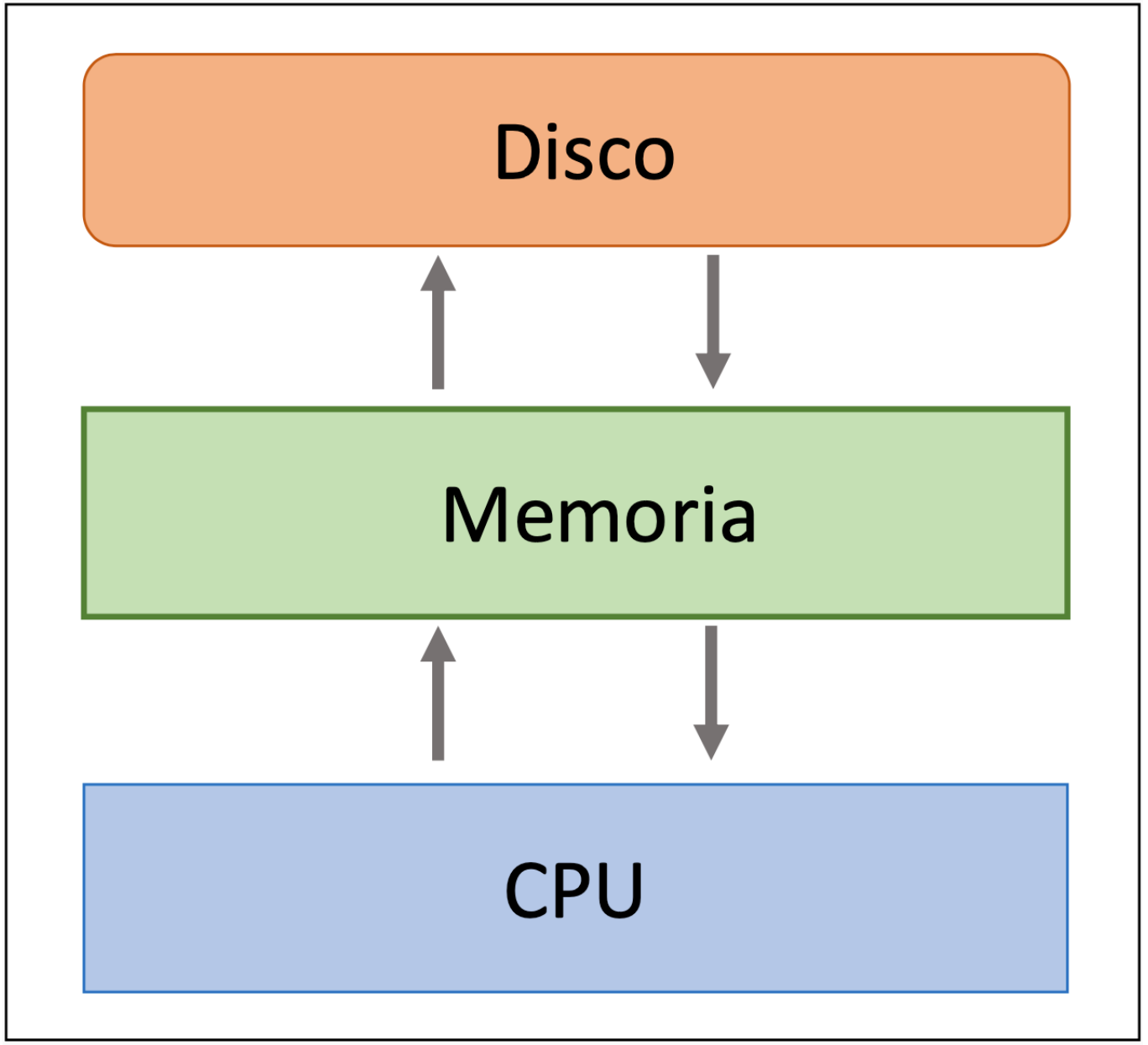
UNIDAD II MODELO DE MAPEO Y REDUCCIÓN

SISTEMA DE ALMACENAMIENTO DISTRIBUIDO

Raúl Galindo Hernández

3 de Febrero de 2023

Cómputo con un Solo Procesador



Flujo de Datos: Ejemplo

- Total de páginas web: ≈ 20 millones
- Tamaño promedio (por página): ≈ 20 KB
- Conteo básico: 20 millones * 20 $KB = \approx 400\ TB$
- Ancho de banda (lectura): 30-35 MB/sec desde disco
- Tiempo de lectura: +11 millones de segundos (≈ 4.4 meses)

- Número de discos: 10,000
- Tiempo de lectura: 1,100 segundos ≈ 19 min

Hilos

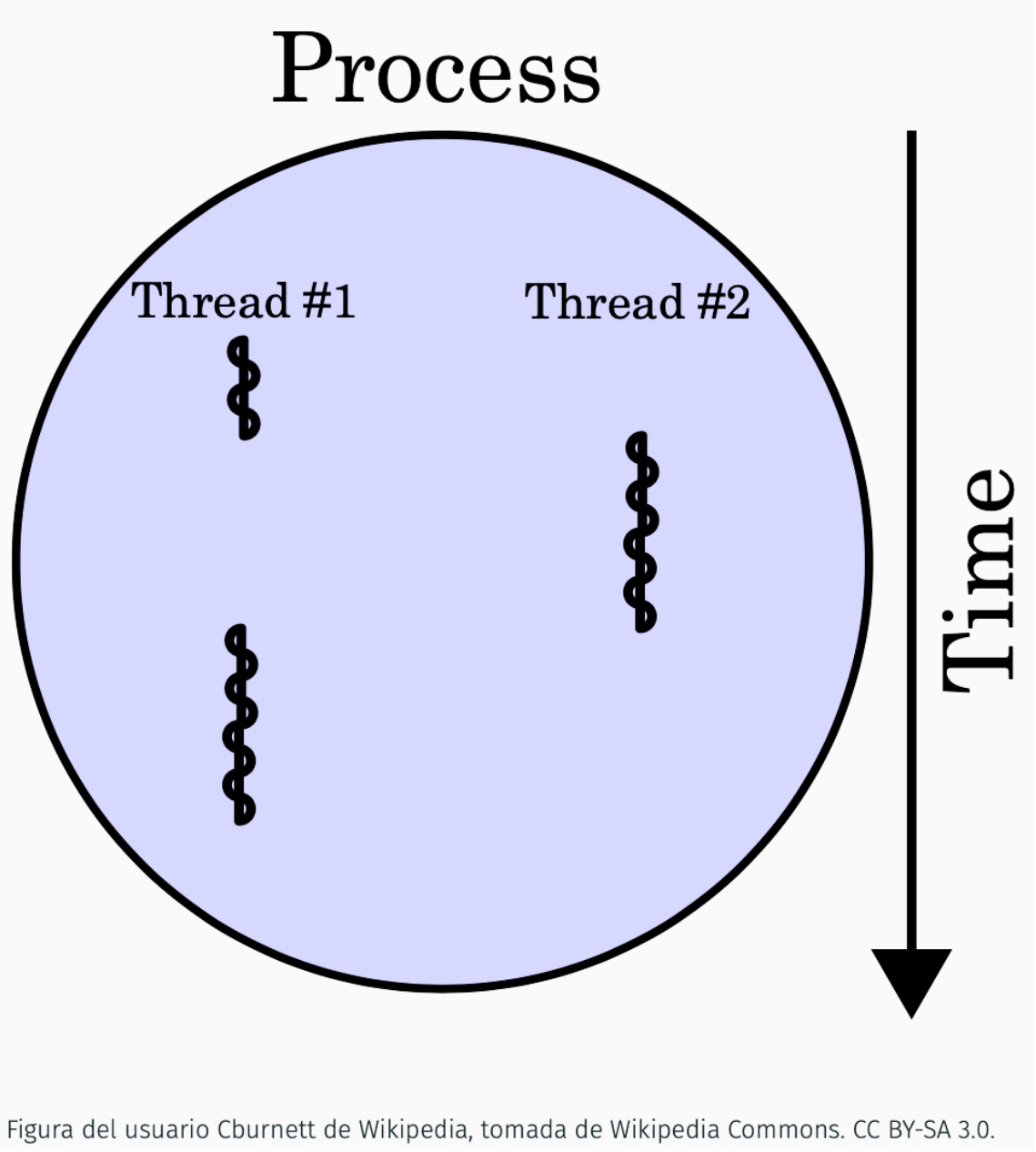


Figura del usuario Cburnett de Wikipedia, tomada de Wikipedia Commons. CC BY-SA 3.0.

Distribución de Almacenamiento

- Acceso a los datos.
  - en cualquier momento,
  - desde cualquier lugar,
  - y solo a aquellas personas que queramos que accedan.

- Gestiona volúmenes lógicos diseñados para procesar el escalado y el acceso a los datos en un entorno de alta disponibilidad.

Distribución de Almacenamiento

- Se compone de datos almacenados en clústeres de nodos de almacenamiento distribuidos geográficamente.

- El sistema de almacenamiento incluye funcionalidades que sincronizan y coordinan los datos en los nodos del clúster.

Distribución de Almacenamiento

- Replicación: los datos se copian en varios nodos y se actualizan consistentemente cada vez que se modifican.

- Escalado: Se puede aumentar o disminuir la capacidad de almacenamiento según sea necesario, agregando o quitando nodos en el clúster.

Recordando…

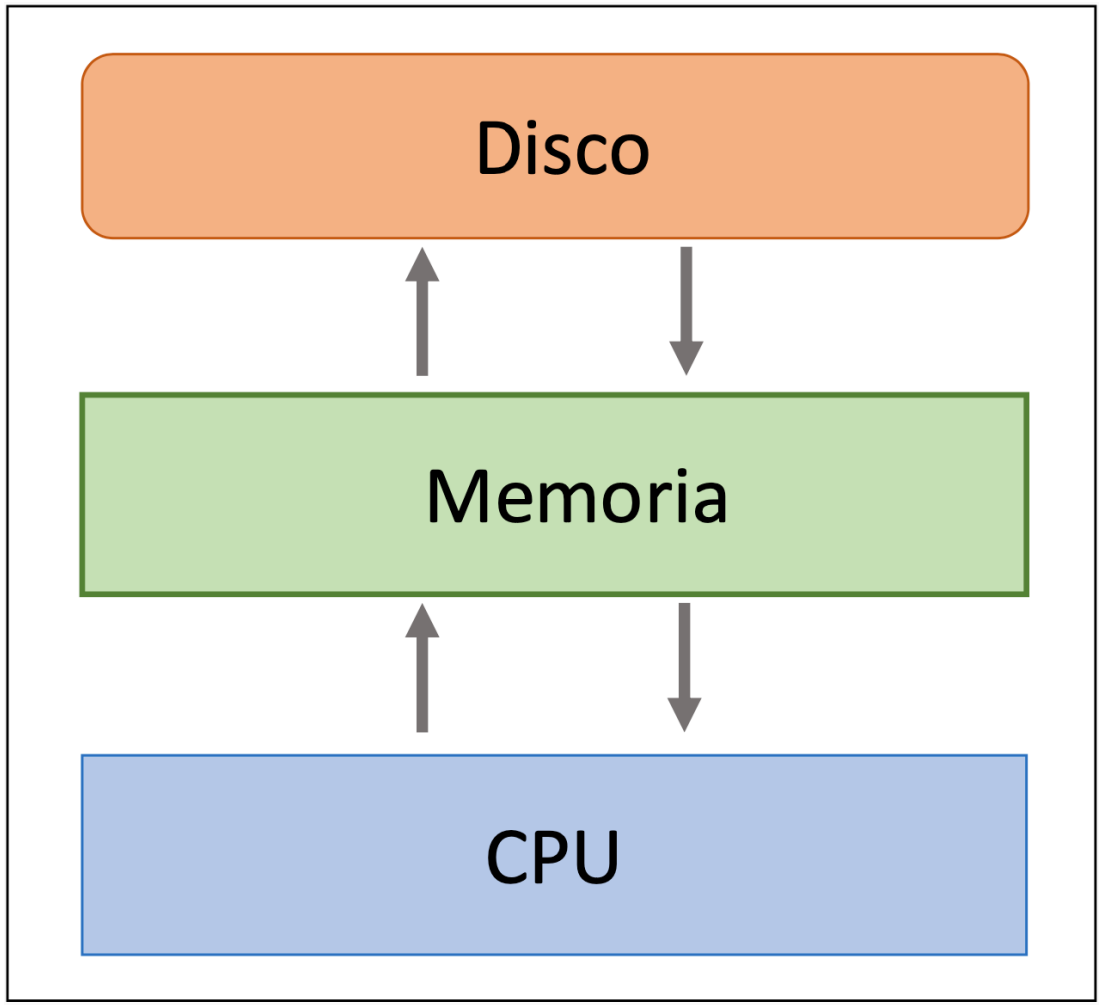
- Conteo básico: 20 millones * 20 $KB \approx 400\ TB$
- Ancho de banda (lectura): 30-35 MB/sec desde disco
- Tiempo de lectura: +11 millones de segundos (≈ 4.4 meses)

Recordando...

- **Número de discos: 10,000**

- **Tiempo de lectura: 1,100 segundos ≈ 19 min**

Recordando...



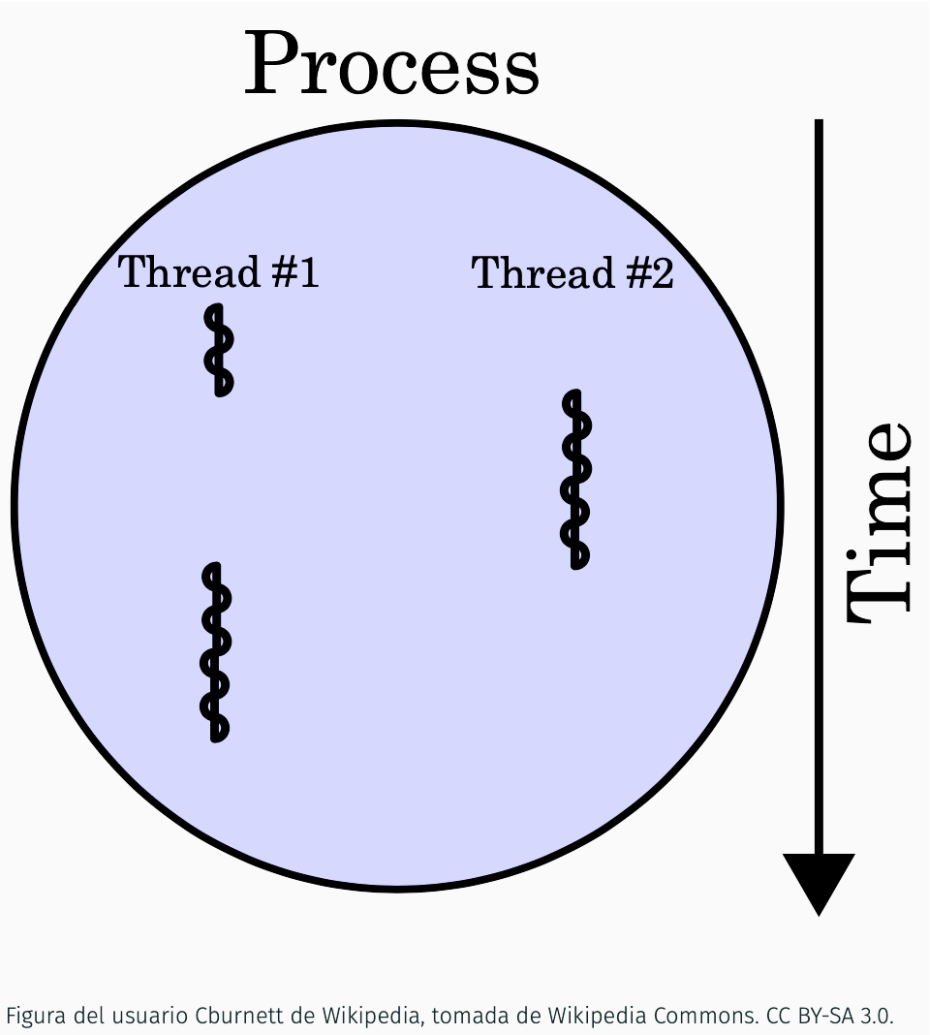


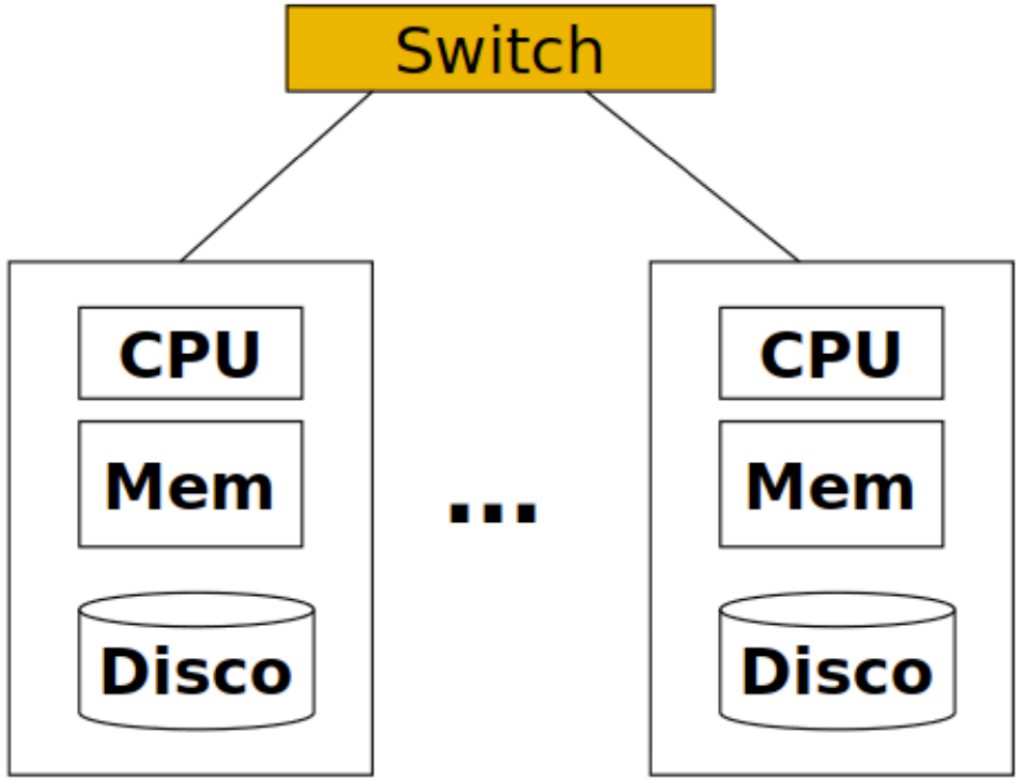
Figura del usuario Cburnett de Wikipedia, tomada de Wikipedia Commons. CC BY-SA 3.0.

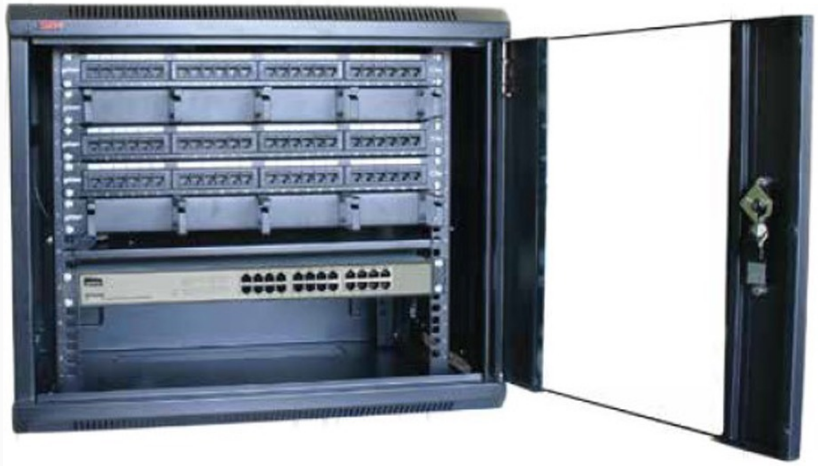
Recordando...

- Acceso a datos de alta disponibilidad.
- Replicación. Los datos se copian en varios nodos y se actualizan consistentemente.
- Escalado. Se puede aumentar o disminuir la capacidad de almacenamiento.

Cómputo en Clústers

1 Gbps entre cualquier
par de nodos en un rack





Cada rack tiene de 16 - 64 nodos

Imagen tomada de J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, *http://www.mmds.org*

Cómputo en Clústers

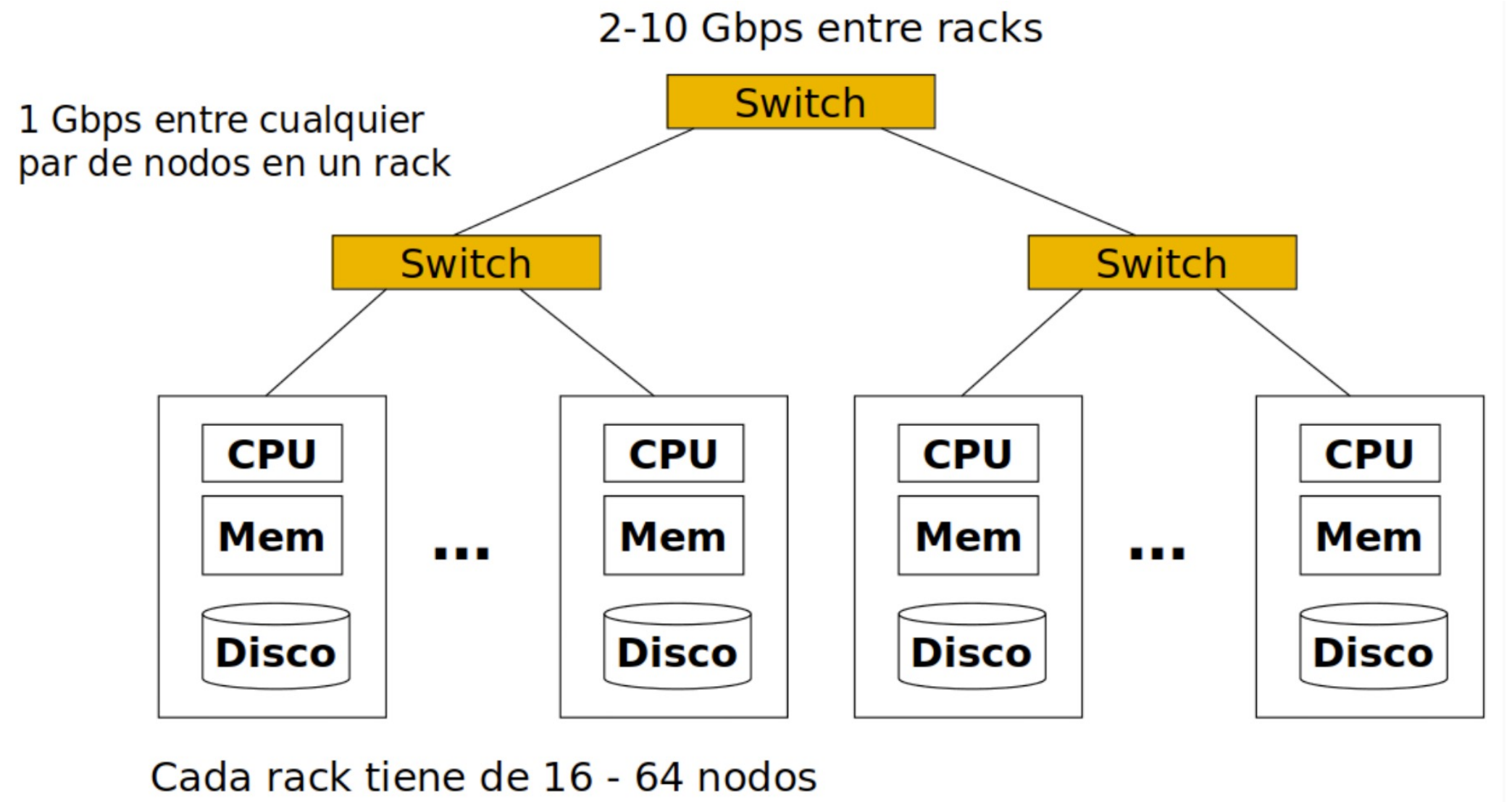


Imagen tomada de J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, *http://www.mmds.org*

Cómputo en Clústers

Imagen tomada de J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, *http://www.mmds.org*

Retos Generales del Cómputo en Clúster

1. Fallo en los nodos.

2. Cuellos de botella en la red.

3. La programación distribuida es difícil.

Retos: Fallos en los Nodos

Contexto.

- Un nodo puede estar activo hasta por 3 años (1,000 días).
- 1,000 nodos en un clúster >> 1 fallo por día.
- 1 Millón de nodos en un clúster >> 1000 fallas por día.

¿Cómo almacenar los datos persistentemente y mantenerlos disponibles si los nodos fallan?

Retos: Fallos en los Nodos

Objetivos.

- Almacenar los datos persistentemente y mantenerlos disponibles aún si los nodos fallan.
- Lidiar con fallas de nodos en cómputo de larga duración.

Retos: Tránsito Lento (Cuellos de botella en la red)

Contexto.

- Transferir demasiado datos a través de la red puede ser muy lento.
- Con un ancho de banda de 1 *Gbps*, tomaría aproximadamente 1 día transferir 10 *TB* de datos de un nodo a otro.

Retos: Tránsito Lento (Cuellos de botella en la red)

Objetivo.

- Minimizar transferencia de datos a través de la red.

Retos: Programación Distribuida es difícil

Contexto.

- La programación distribuida requiere considerar sincronización, carga de trabajo, comunicación, etcétera.

Retos: Programación Distribuida es difícil

Objetivo.

- Es necesario un modelo que oculte la complejidad posible de la programación distribuida.

Retos Generales del Cómputo en Clúster

1. Fallo en los nodos.

2. Cuellos de botella en la red.

3. La programación distribuida es difícil.

Solución: MAP – REDUCE

Map – Reduce

Map – Reduce es un sistema de programación que permite atender los tres retos del cómputo en clúster.

- ✓ Almacenamiento redundante en múltiples nodos para garantizar persistencia y disponibilidad.

- ✓ Minimiza los problemas de cuello de botella.

Map – Reduce

- ✓ Proporciona un modelo simple de programación, ocultando las cuestiones complejas inherentes.

Infraestructura de Almacenamiento Redundante

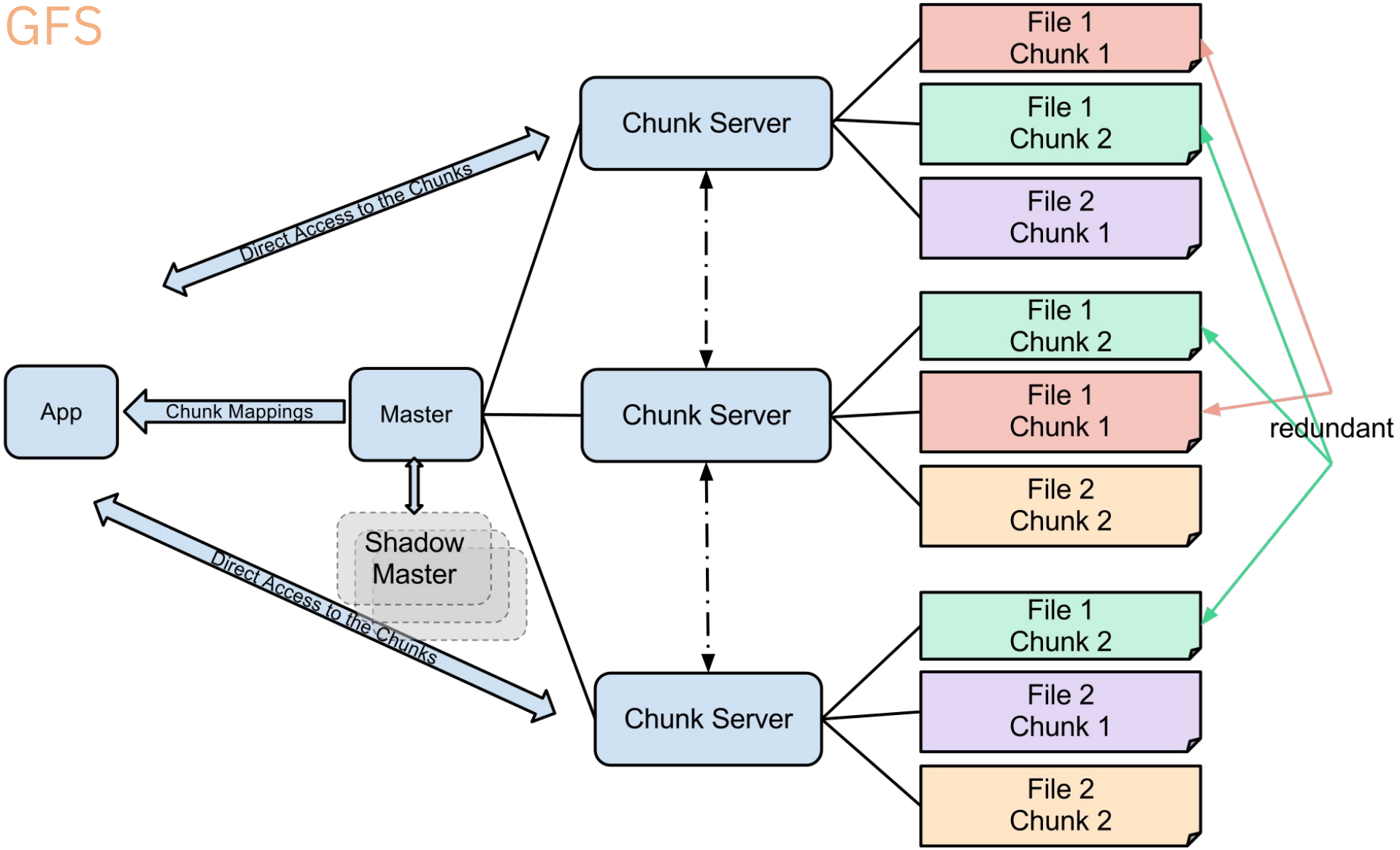
Sistema de archivo distribuido.

- Proporciona un archivo global, persistencia y disponibilidad.

- Ejemplos.
  - Google GFS.
  - Hadoop HDFS.

Infraestructura de Almacenamiento Redundante

Google GFS



Infraestructura de Almacenamiento Redundante

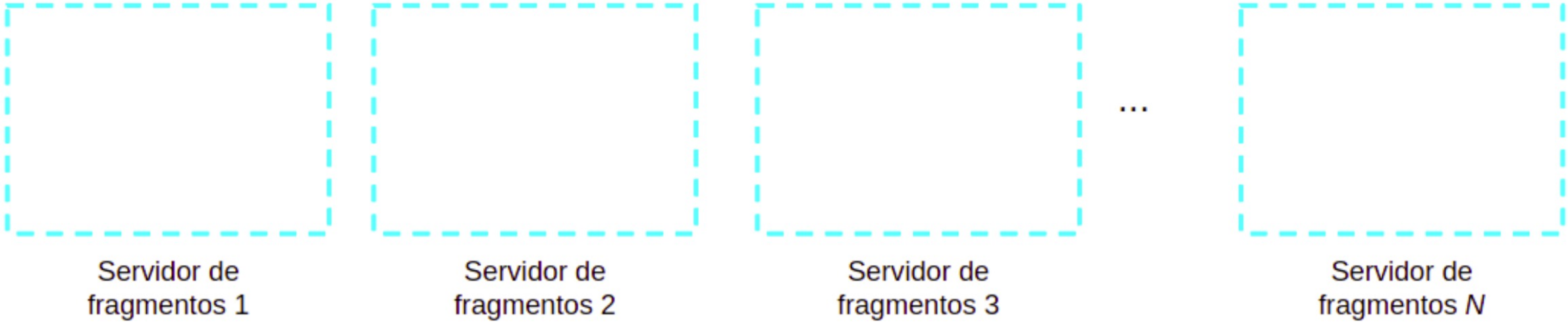
Patrones de uso típico.

- Archivos grandes (*Cientos* de *GB* o *TB*).
- Los datos raramente son actualizados en su lugar.

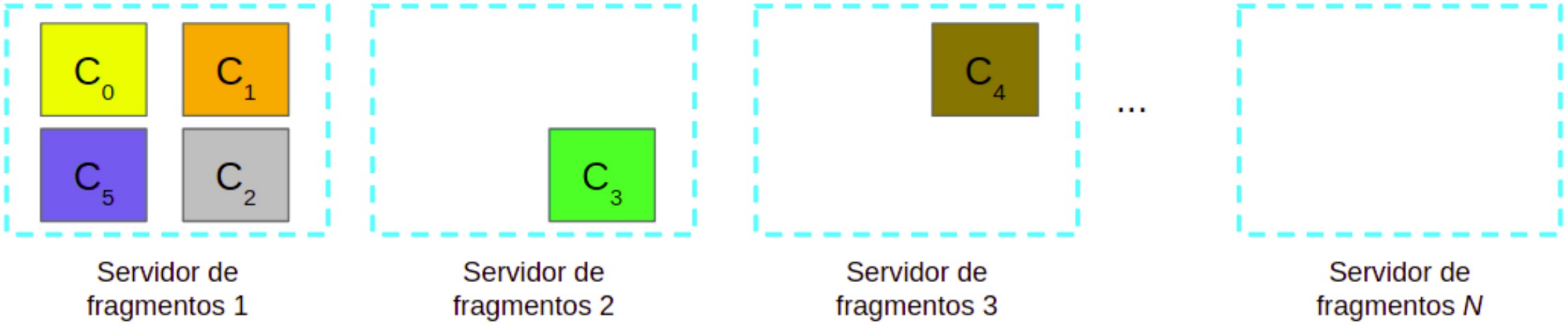
Sistema de Archivos Distribuidos – HDFS

- Los datos se almacenan en fragmentos o chunks que se distribuyen entre los nodos.
- Cada fragmento se replica en diferentes nodos.
- Se garantiza la persistencia y la disponibilidad.

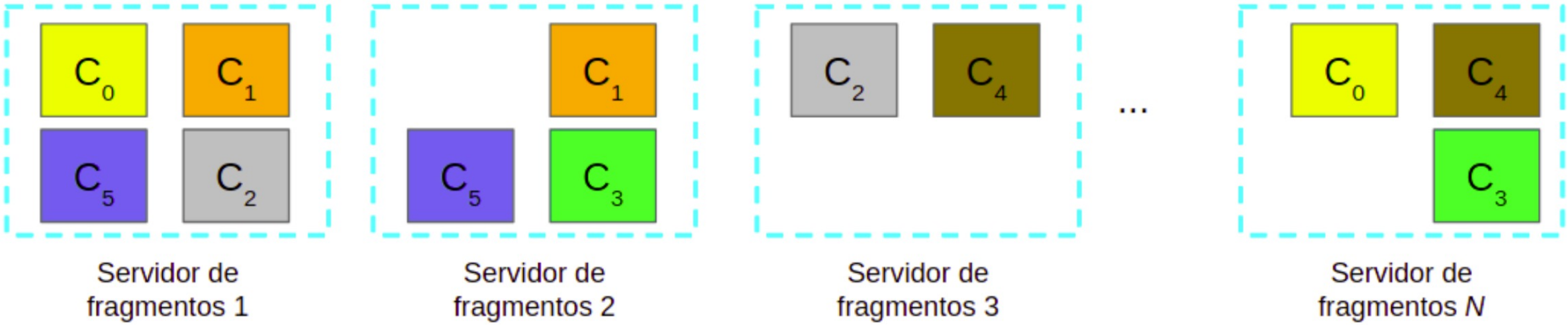
Sistema de Archivos Distribuidos – HDFS



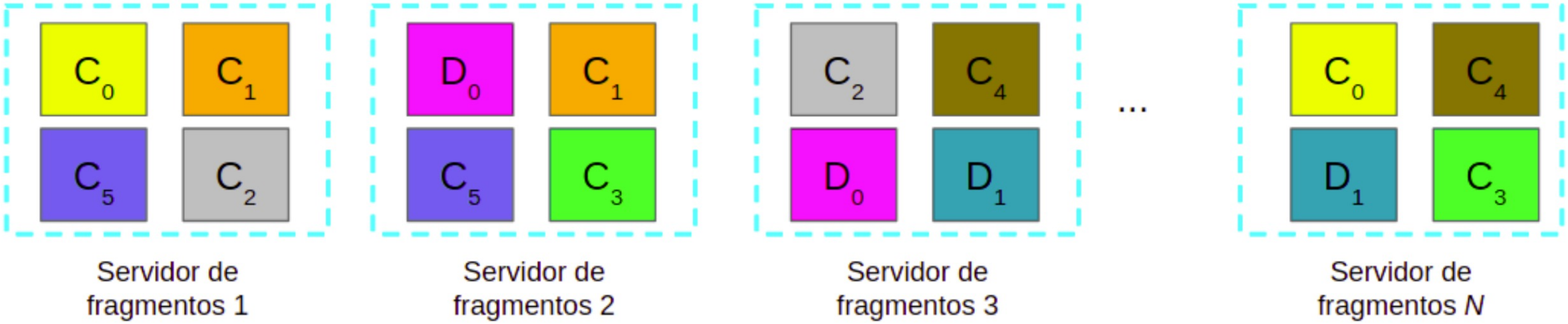
Sistema de Archivos Distribuidos – HDFS



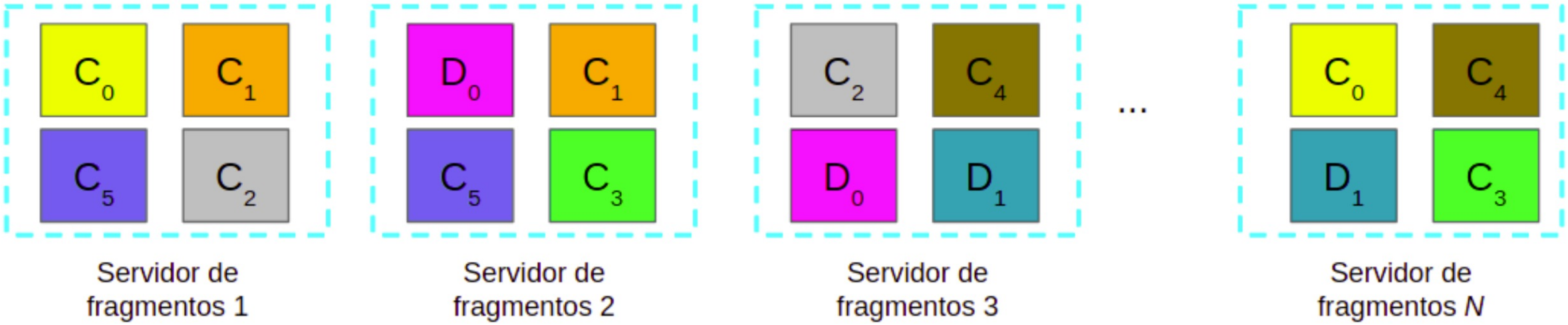
Sistema de Archivos Distribuidos – HDFS



Sistema de Archivos Distribuidos – HDFS



Sistema de Archivos Distribuidos – HDFS

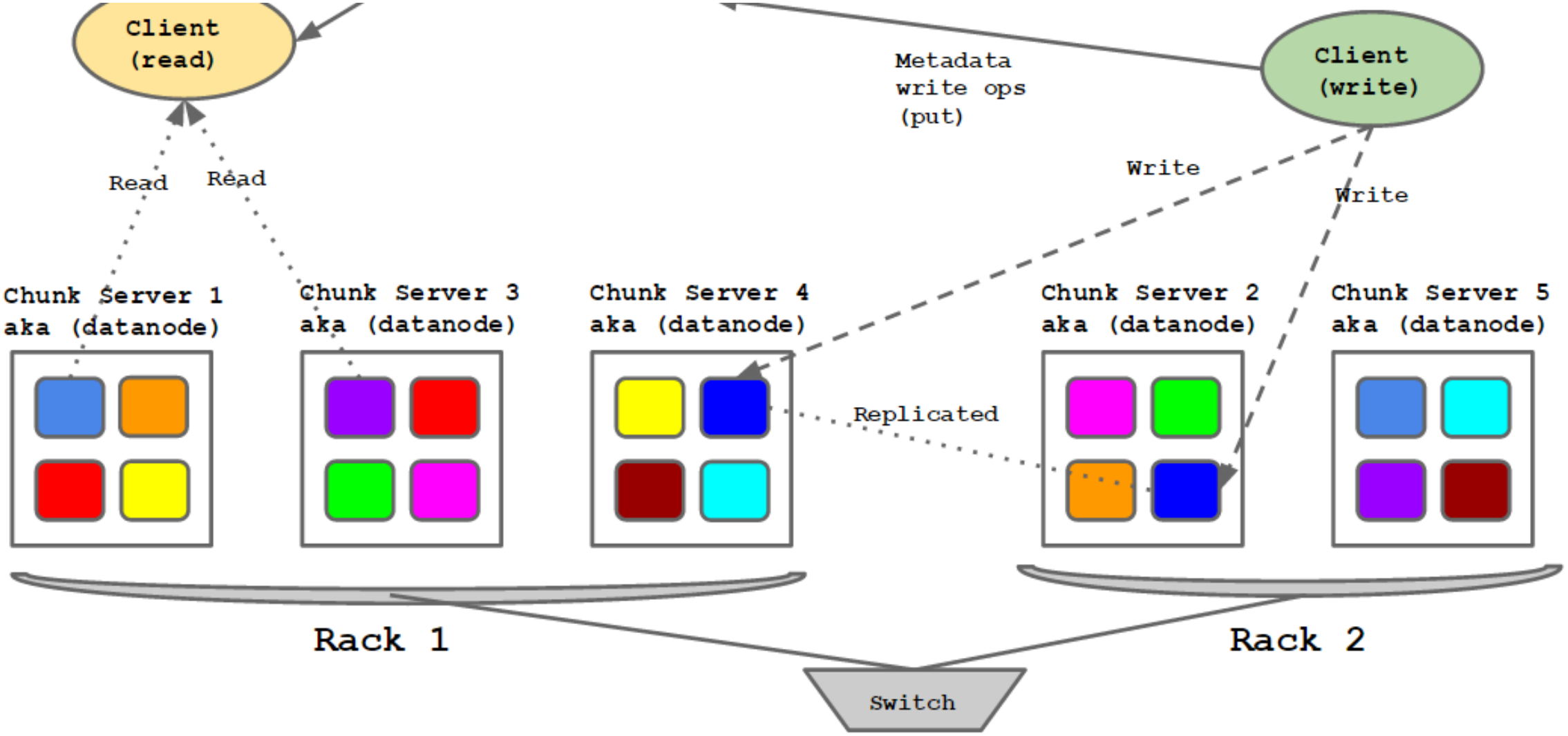


Los servidores de fragmentos actúan como servidores de cómputo.

Servidores de fragmentos.

- Archivos se dividen en fragmentos contínuos (16 - 64 *MB*).
- Cada fragmento se replica (2 o 3 veces).
- Sistema trata de mantener réplicas en diferentes racks.

Sistema de Archivos Distribuidos – HDFS



Sistema de Archivos Distribuidos – HDFS

Nodo maestro.

- Almacena metadatos.
- También se replica.

Biblioteca cliente.

- Se comunica con nodo maestro para encontrar los servidores de fragmentos.
- Conecta directamente hacia los servidores de fragmentos para acceder a los datos.

Hadoop: Ventajas

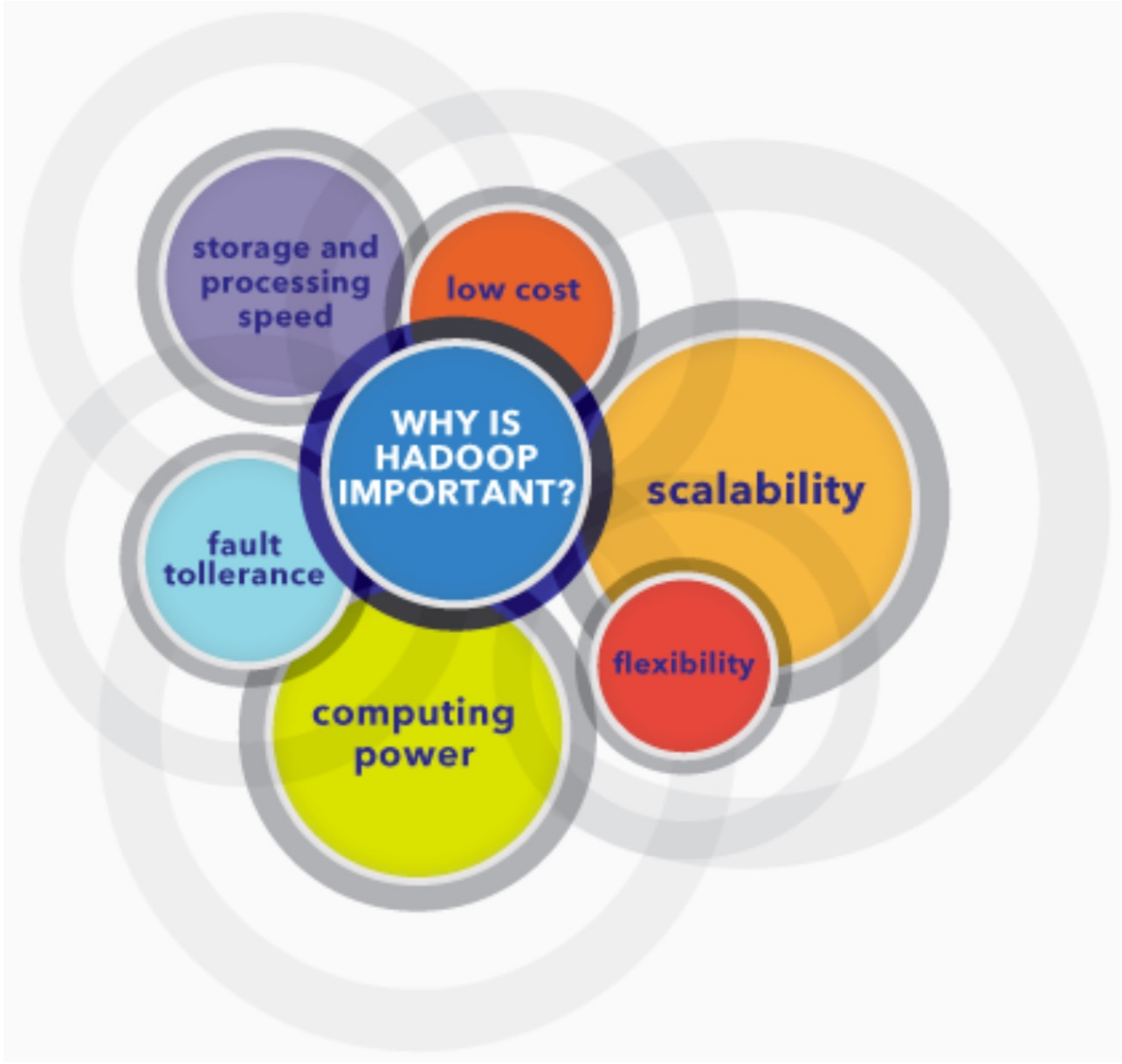


Imagen tomada de https://www.sas.com/es_pe/insights/big-data/hadoop.html

Hadoop: Componentes del Ecosistema

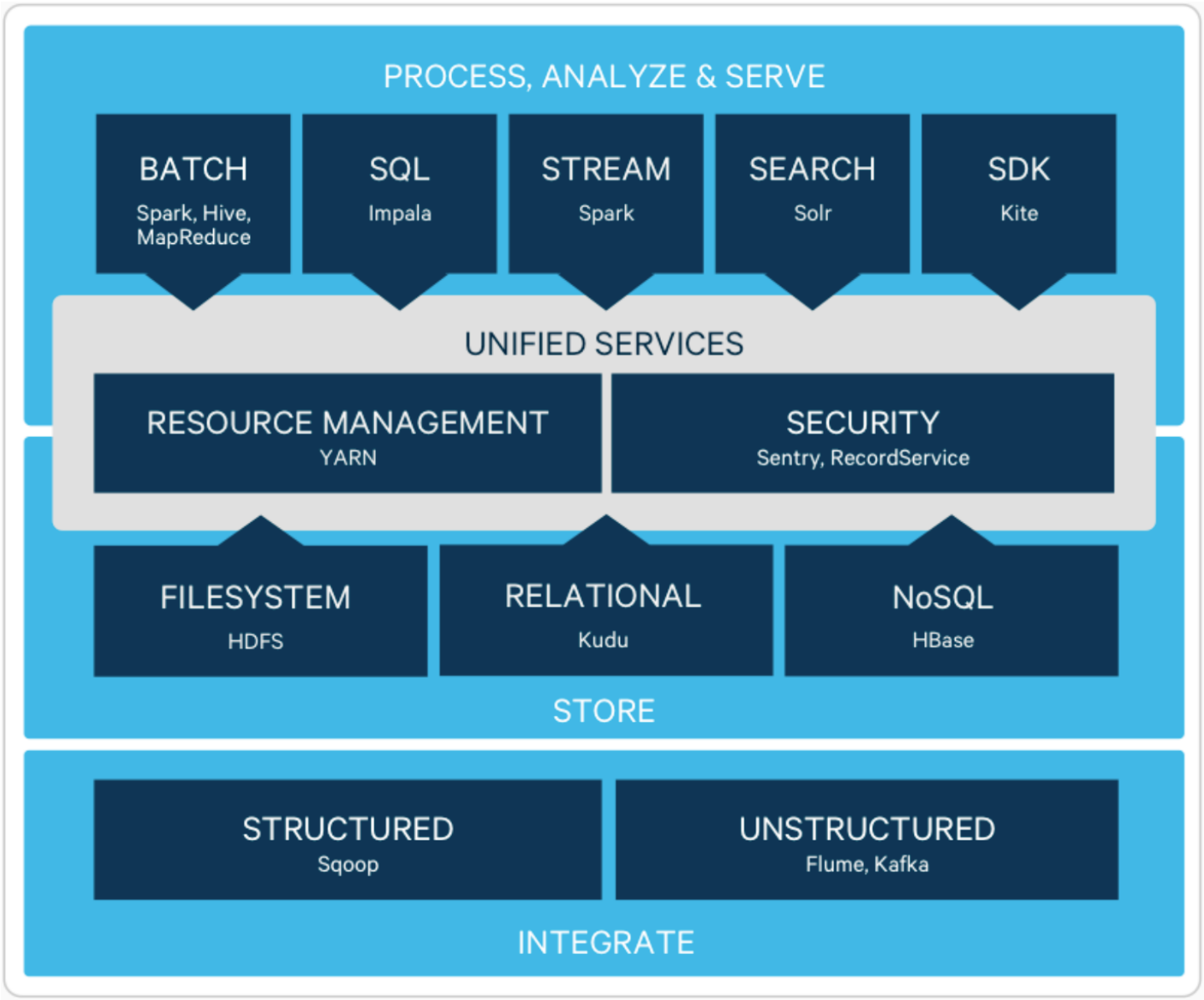


Imagen tomada de https://www.cloudera.com/products/open-source/apache-hadoop.html

Apache HIVE

Es un sistema de almacenamiento de datos de código abierto para consultar y analizar grandes conjuntos de datos almacenados en archivos Hadoop.

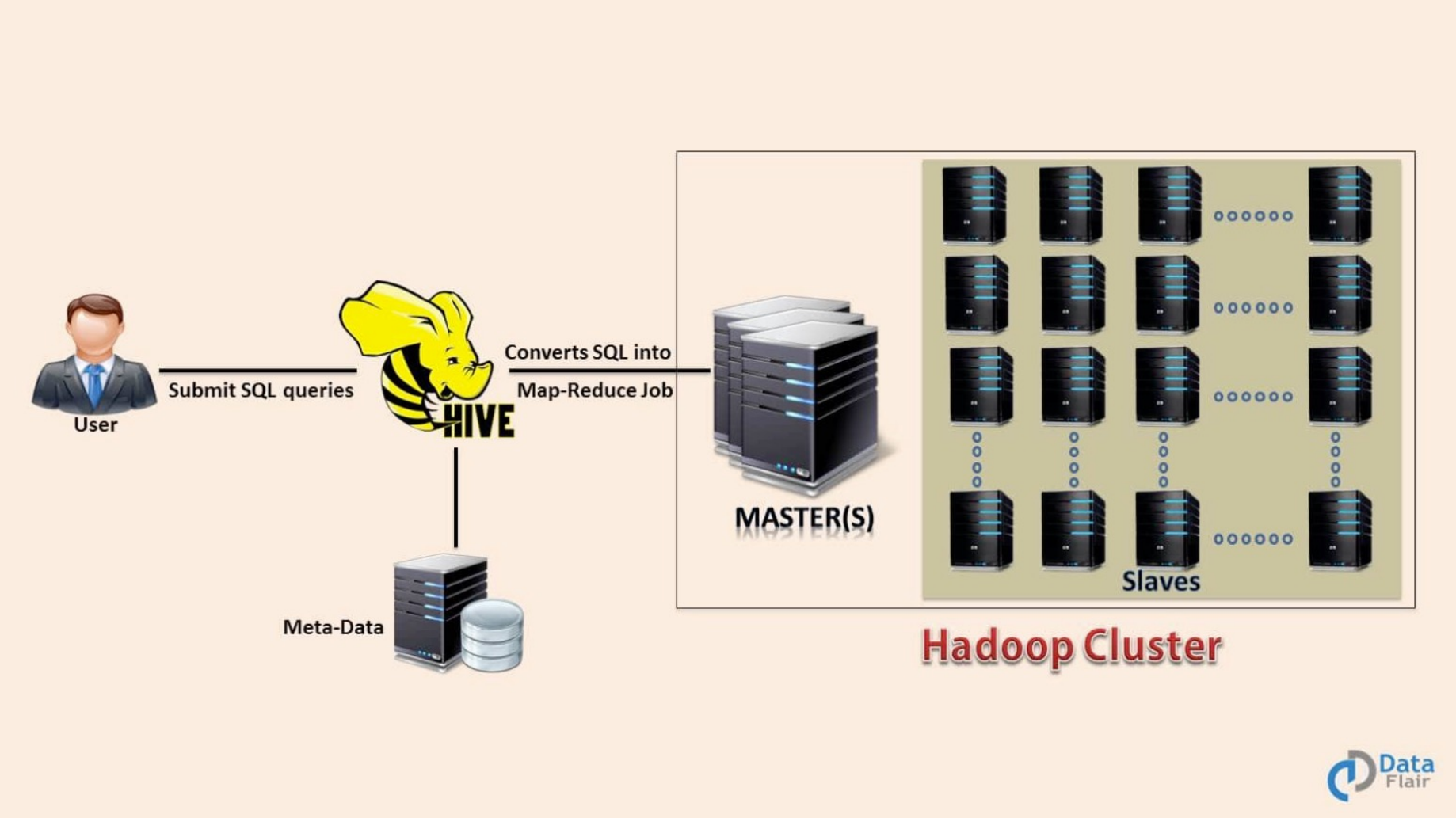


Imagen tomada de https://bit.ly/2JgZwjx