

UNIDAD 2: MODELO DE MAPEO Y REDUCCIÓN

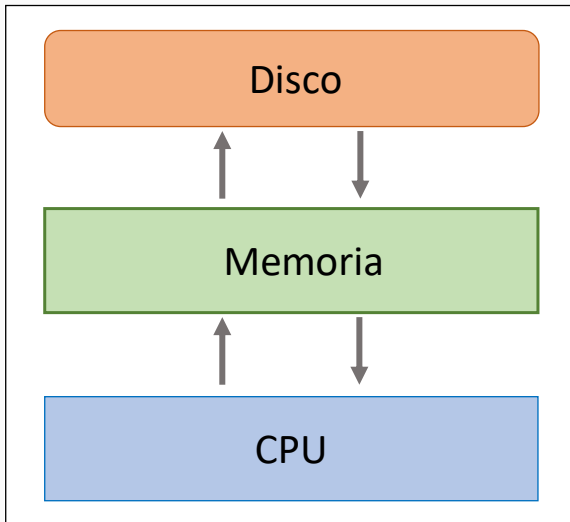
SISTEMA DE ARCHIVOS DISTRIBUIDO

Gibran Fuentes-Pineda

Diapositivas basadas en las de la M. en C. Blanca Vázquez

Febrero 2021

CÓMPUTO CON UN SOLO PROCESADOR



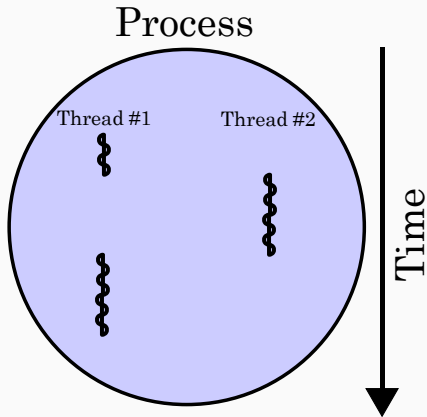


Figura del usuario Cburnett de Wikipedia, tomada de Wikipedia Commons. CC BY-SA 3.0.

COMPUTACIÓN PARALELA

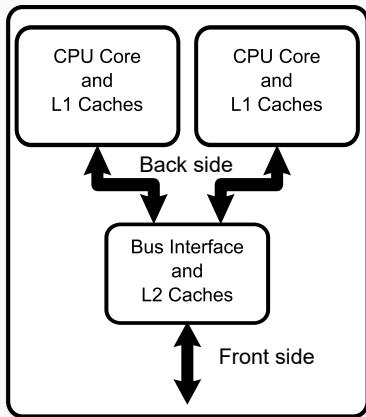


Figura del usuario Dennis Schmitz de Wikipedia, tomada de Wikipedia Commons.

SUPERCOMPUTADORA

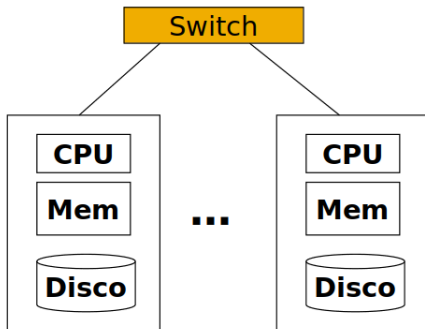
- Computadora especializada con múltiples procesadores
 - Costosas
 - Difíciles de mantener



Figura de Argonne National Laboratory, tomada de Wikipedia Commons. CC BY-SA 2.0.

CÓMPUTO EN CLÚSTER

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 CLÚSTER

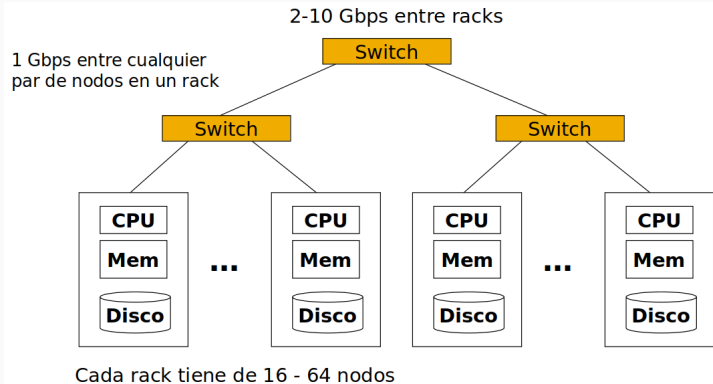


Imagen tomada de J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, <http://www.mmids.org>

CÓMPUTO EN CLÚSTER



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

1. Fallo en los nodos
2. Cuellos de botella en la red
3. La programación distribuida es difícil

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

RETO 1: FALLO EN LOS NODOS

- Contexto
 - Un nodo puede estar activo hasta por 3 años (1,000 días)
 - 1000 nodos en un clúster >> 1 fallo por día
 - 1M de nodos en un clúster >> 1000 fallas por día
- ¿Cómo almacenar los datos persistentemente y mantenerlos disponibles si los nodos fallan?

RETO 1: FALLO EN LOS NODOS

- Contexto: en clústeres grandes suelen ocurrir fallos en nodos y en la red
 - Un nodo puede estar activo hasta por 3 años (1,000 días)
 - 1000 nodos en un clúster >> 1 fallo por día
 - 1M de nodos en un clúster >> 1000 fallas por día
- Objetivos
 - Almacenar los datos persistentemente y mantenerlos disponibles aún si los nodos fallan
 - Lidar con fallas de nodos en cómputo de larga duración

RETO 2: 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 1Gbps, tomaría aproximadamente 1 día transferir 10TB de datos de un nodo a otro
- Objetivo
 - Minimizar transferencia de datos a través de la red

RETO 3: LA PROGRAMACIÓN DISTRIBUIDA ES DIFÍCIL

- Contexto: programación distribuida requiere considerar sincronización, carga de trabajo, comunicación, etc.
- Objetivo
 - Es necesario un modelo que oculte la complejidad posible de la programación distribuida

1. Fallo en los nodos
2. Cuellos de botella en la red
3. La programación distribuida es difícil
4. Solución: ¡**MapReduce**!

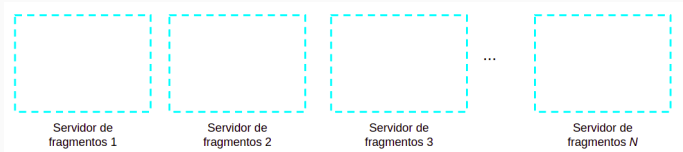
MapReduce es un sistema de programación que permite atender los tres retos del cómputo en clúster.

1. Almacenamiento redundante en múltiples nodos para garantizar persistencia y disponibilidad
2. Minimiza los problemas de cuello de botella
3. Proporciona un modelo simple de programación, ocultando las cuestiones complejas inherentes

- Sistema de archivo distribuido
 - Proporciona un nombre de archivo global, persistencia y disponibilidad.
 - Ejemplos: Google GFS, Hadoop HDFS
- Patrones de uso típico
 - Archivos grandes (100s de GB o TB)
 - Los datos raramente son actualizados en su lugar

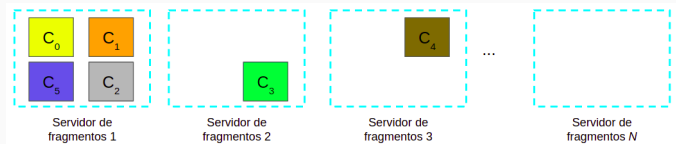
SISTEMA DE ARCHIVOS DISTRIBUIDO HDFS

- 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



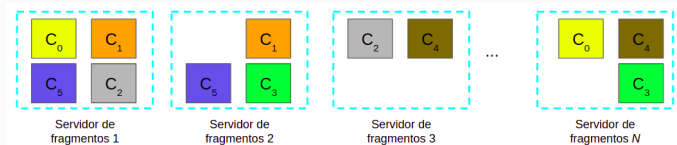
SISTEMA DE ARCHIVOS DISTRIBUIDO HDFS

- 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



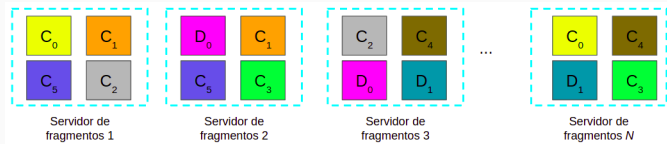
SISTEMA DE ARCHIVOS DISTRIBUIDO HDFS

- Los datos se guardan en fragmentos (*chunks*) que se distribuyen entre los nodos
- Cada fragmento se replica en diferentes nodos
 - Se garantiza la persistencia y la disponibilidad



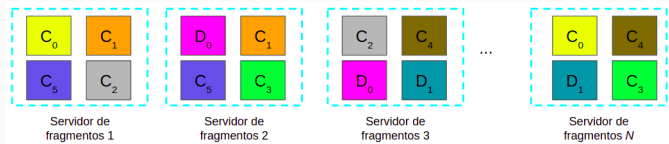
SISTEMA DE ARCHIVOS DISTRIBUIDO HDFS

- Los datos se guardan en fragmentos (*chunks*) que se distribuyen entre los nodos
- Cada fragmento se replica en diferentes nodos
 - Se garantiza la persistencia y la disponibilidad



SISTEMA DE ARCHIVOS DISTRIBUIDO HDFS

- Los datos se guardan en fragmentos (*chunks*) que se distribuyen entre los nodos
- Cada fragmento se replica en diferentes nodos
 - Se garantiza la persistencia y la disponibilidad



- 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*

- Servidores de fragmentos
 - Archivos se dividen en fragmentos continuos (16-64 MB)
 - Cada fragmento se replica (2 o 3 veces)
 - Sistema trata de mantener réplicas en diferentes *racks*
- Nodo maestro
 - Almacena metadatos
 - También se replica

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

VENTAJAS DE HADOOP

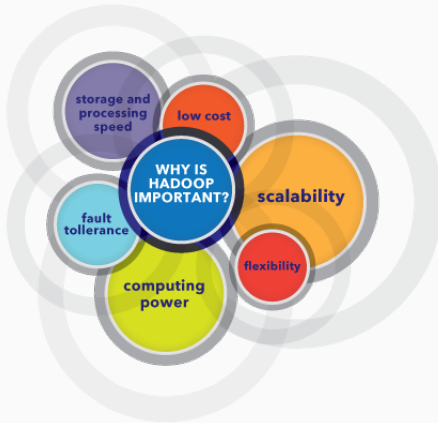


Imagen tomada de https://www.sas.com/es_pe/insights/big-data/hadoop.html

COMPONENTES DEL ECOSISTEMA DE HADOOP

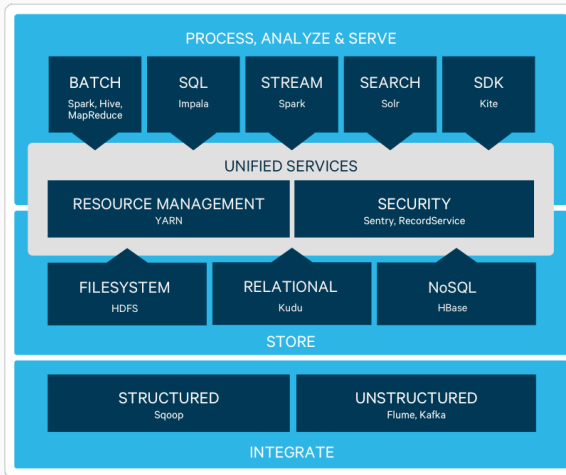


Imagen tomada de <https://www.cloudera.com/products/open-source/apache-hadoop.html>

APACHE HIVE

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

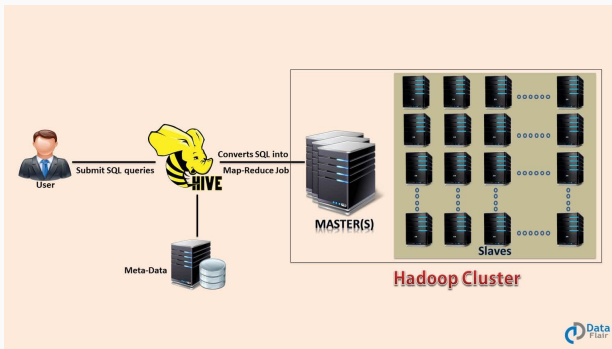


Imagen tomada de <https://bit.ly/2JgZwjx>