

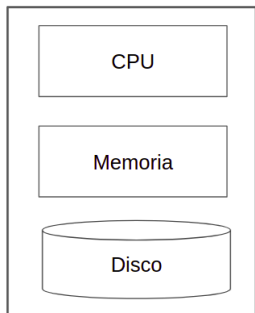
ANÁLISIS DE DATOS MASIVOS

SISTEMA DE ALMACENAMIENTO Y PROCESAMIENTO DISTRIBUIDO

Blanca Vázquez

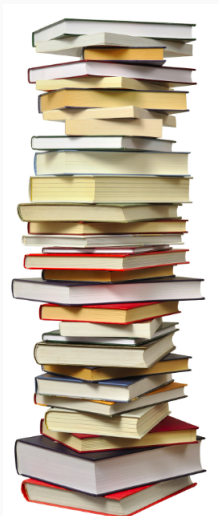
26 de agosto de 2024

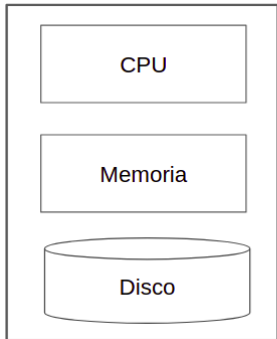
MODELO BÁSICO COMPUTACIONAL



- Este modelo también es conocido como:
 - Arquitectura de un solo nodo
- Ejemplos de uso: aprendizaje máquina, estadísticas

PROBLEMA





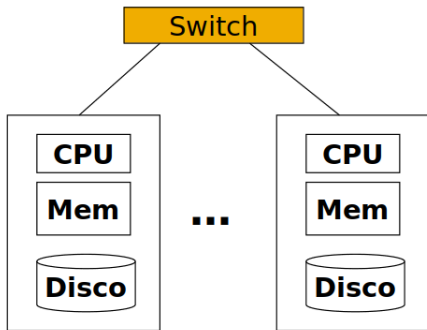
- Procesamiento de datos por lotes
- Algoritmos clásicos de minería de datos

- Total de páginas web = +20 billones
- Tamaño promedio de cada página web = 20 KB
- +20 billones * 20 KB = +400 TB
- Ancho de banda = 1 computadora lee 30-35 MB/sec desde disco
- Tiempo para leer = +11 millones de segundos = 4.6 meses!!

- Total de páginas web = +20 billones
- Tamaño promedio de cada página web = 20 KB
- +20 billones * 20 KB = +400 TB
- Ancho de banda = 1 computadora lee 30-35 MB/sec desde disco
- Número de discos = 10,000
- Tiempo para leer = 1,100 de segundos = 1 h aproximadamente!!

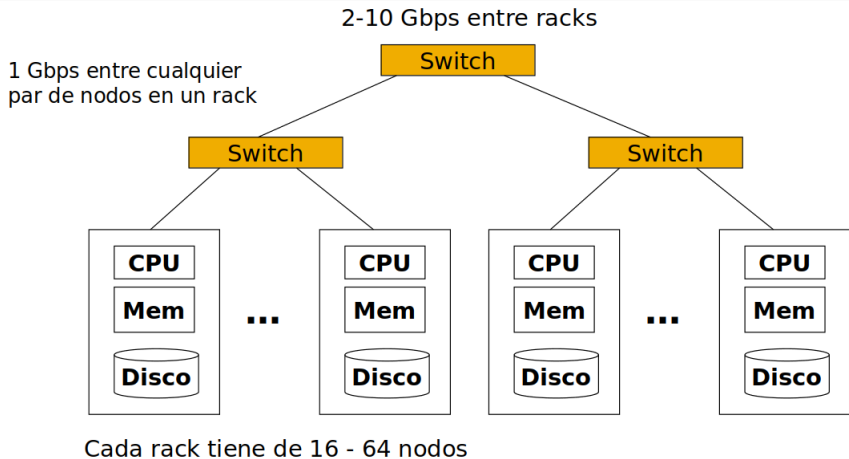
ARQUITECTURA DE CLÚSTER

1 Gbps entre cualquier
par de nodos en un rack



Cada rack tiene de 16 - 64 nodos

ARQUITECTURA DE CLÚSTER



- En el 2011, se estimó que google tenía un centro de datos con alrededor de 1 millón de nodos
<http://bit.ly/Shh0R0>

ARQUITECTURA DE 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

- 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?

- 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?
- ¿Cómo lidiar con las fallas de un nodo durante un cálculo de larga ejecución?

RETO 2: CUELLOS DE BOTELLA EN LA RED

- Ancho de banda de la red = 1 Gbps
- Supongamos que tenemos 10 TB de datos, ¿cuánto tiempo nos tomará moverlos?

RETO 3: LA PROGRAMACIÓN DISTRIBUIDA ES DIFÍCIL

- 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

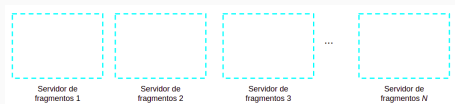
Solución: ¡¡MapReduce!!

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

- 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



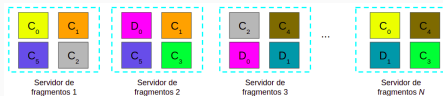
- Los datos se guardan en fragmentos (chunks) que se distribuyen entre los nodos
- Cada fragmento se replica en diferentes nodos
 - Se garantiza la persistencia y la disponibilidad



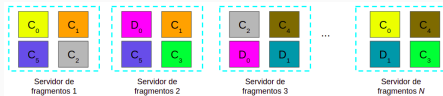
- Los datos se guardan en fragmentos (chunks) que se distribuyen entre los nodos
- Cada fragmento se replica en diferentes nodos
 - Se garantiza la persistencia y la disponibilidad



- Los 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 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
 - Un archivo se divide en fragmentos continuos (16-64 MB)
 - Cada fragmento es replicado (2 o 3 veces)
 - El sistema trata de mantener las réplicas en diferentes racks

- **Servidores de fragmentos**
 - Un archivo se divide en fragmentos continuos (16-64 MB)
 - Cada fragmento es replicado (2 o 3 veces)
 - El sistema trata de mantener las réplicas en diferentes racks
- **Nodo maestro**
 - Almacena metadatos
 - Este nodo maestro puede replicarse

- **Servidores de fragmentos**
 - Un archivo se divide en fragmentos continuos (16-64 MB)
 - Cada fragmento es replicado (2 o 3 veces)
 - El sistema trata de mantener las réplicas en diferentes racks
- **Nodo maestro**
 - Almacena metadatos
 - Este nodo maestro puede replicarse
- **Biblioteca cliente**
 - Habla con el maestro para encontrar los servidores de fragmentos
 - Conecta directamente hacia los servidores de fragmentos para acceder a los datos

¿POR QUÉ ES IMPORTANTE UN HDFS?

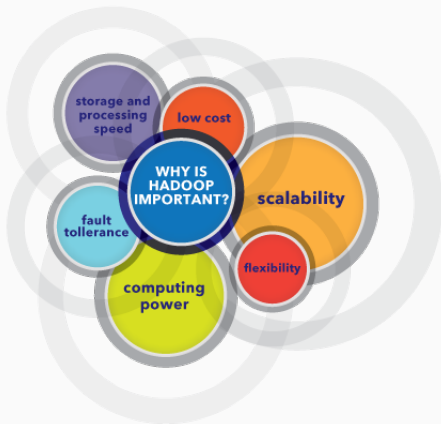


Imagen tomada de https://www.sas.com/es_pe/insights/big-data/hadoop.html

COMPONENTES DEL ECOSISTEMA DE HADOOP

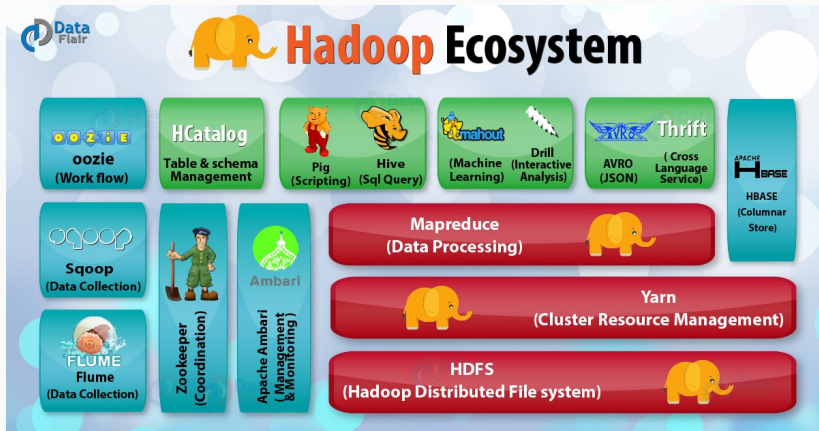


Imagen tomada de <https://bit.ly/2JgZwjx>

APACHE HIVE

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

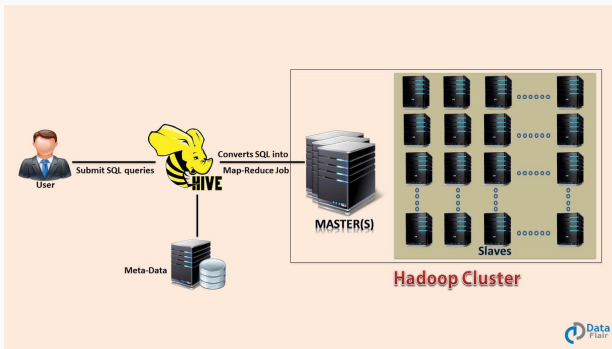


Imagen tomada de <https://bit.ly/2JgZwjx>