Fundamentos de los RDDS

BigData - UBO2022

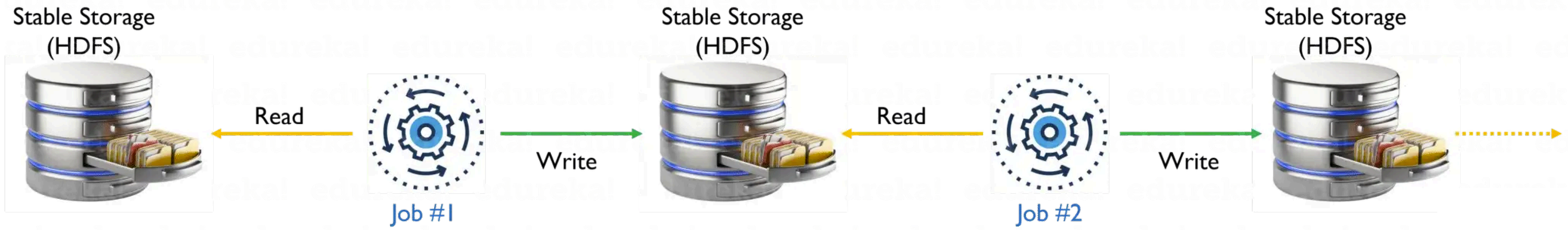
Sebastian Ulloa Quezada

La computación distribuida iterativa, es decir, el procesamiento de datos en múltiples trabajos requiere la reutilización y el intercambio de datos entre ellos.

Antes de que los RDD entraran en escena, los marcos como Hadoop se enfrentaban a dificultades para procesar múltiples operaciones/trabajos.

Además, se necesitaba un almacén de datos intermedio estable y distribuido, como HDFS o Amazon S3.

Estos medios para compartir datos ayudaron a realizar varios cálculos como la regresión logística, la agrupación de K-means, los algoritmos de clasificación de páginas, las consultas ad hoc, etc. Pero nada es gratis, el intercambio de datos conduce a un procesamiento lento de datos debido a múltiples operaciones de E/S como la replicación y la serialización.



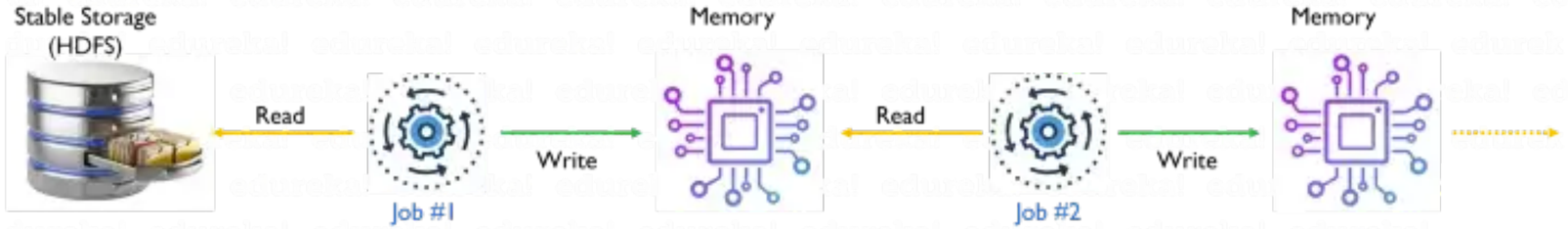
Por lo tanto, había una necesidad de algo que pudiera superar el problema de las múltiples operaciones de E/S a través del intercambio de datos y reducir su número…

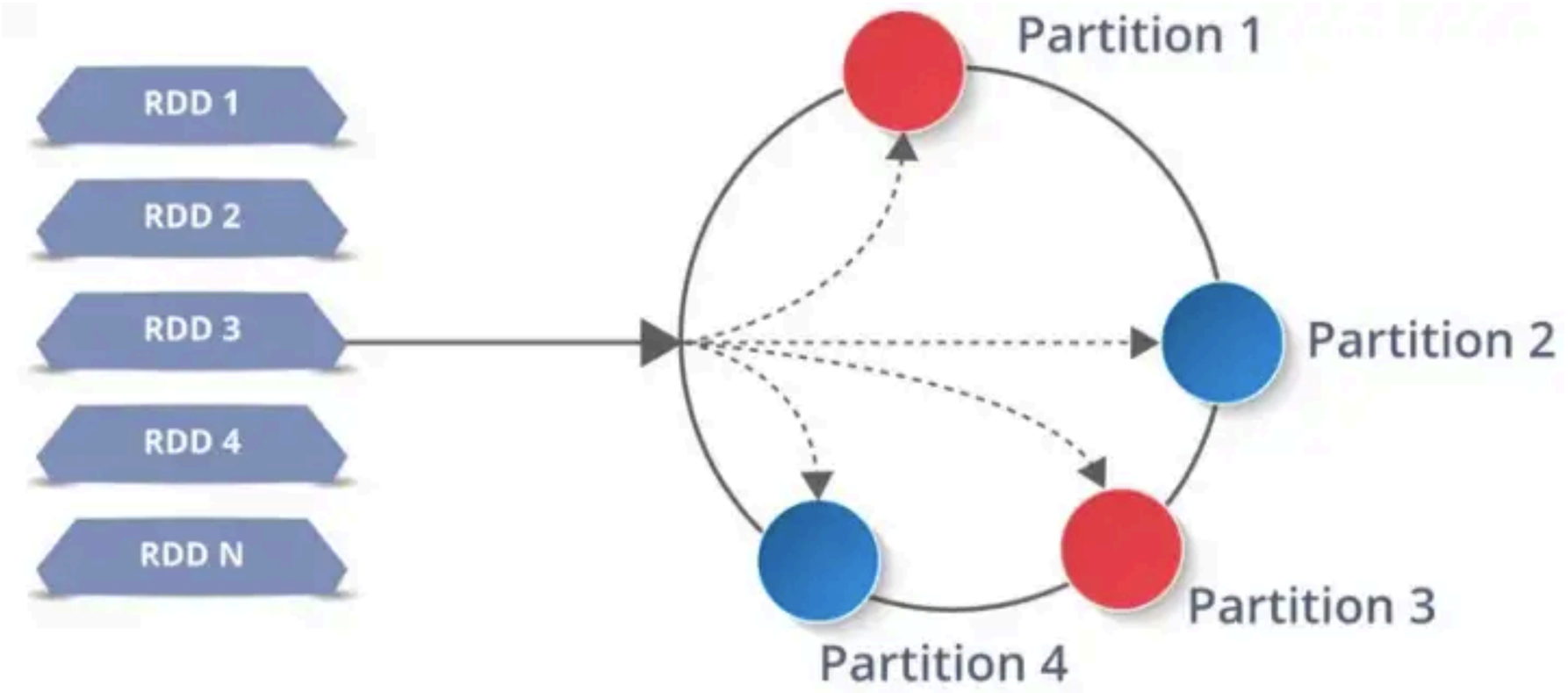
Que es RDDS?

RDDS - Resilient Distributed Dataset

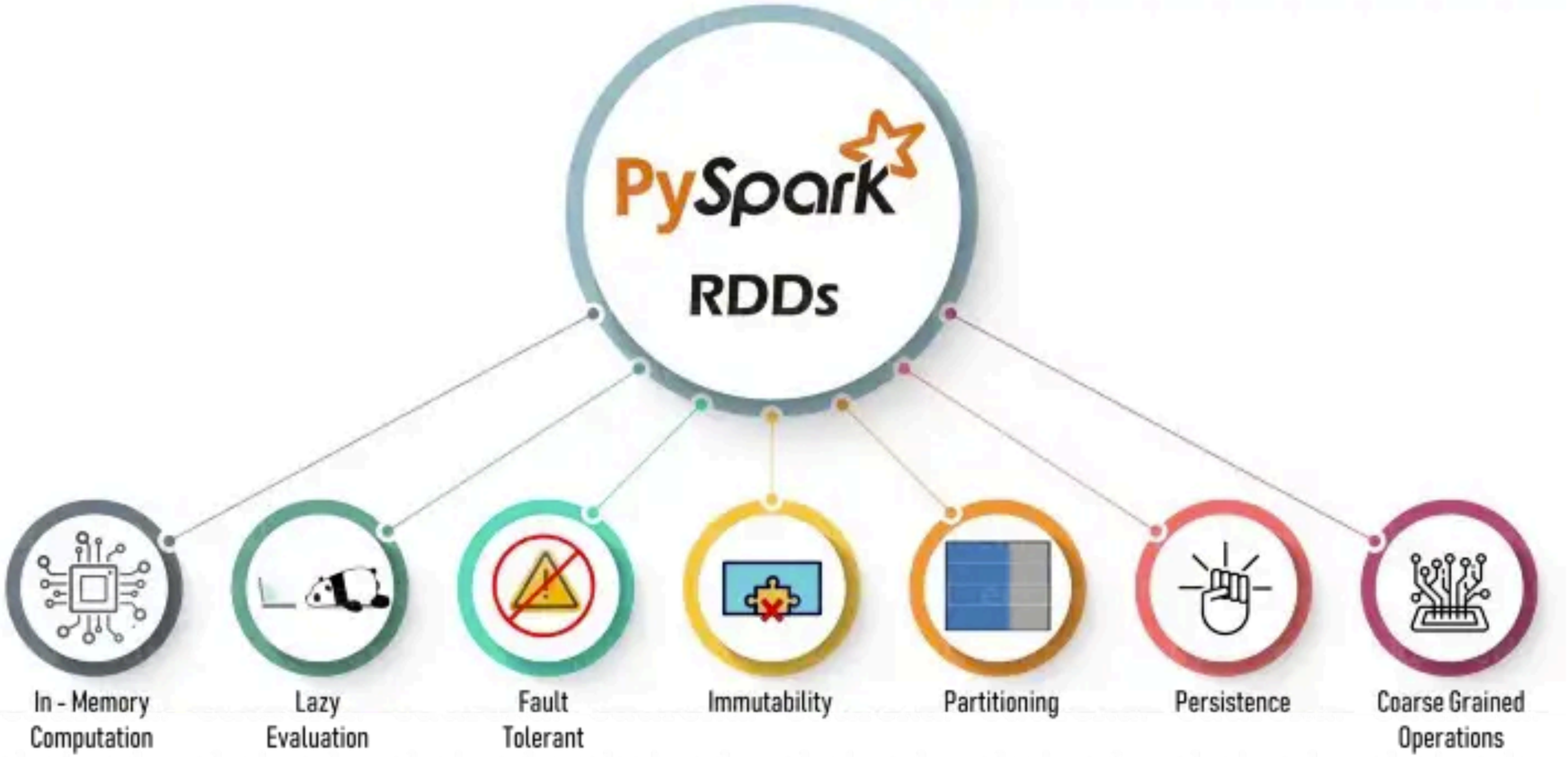
- **Resilient Distributed Datasets (RDDs) Son una abstracción de memoria distribuida que ayuda a un programador a realizar cálculos en memoria en grandes clústeres que también de una manera tolerante a fallos.**

Los RDD se consideran la columna vertebral de PySpark. Es uno de los pioneros en la estructura de datos fundamental sin esquemas, que puede manejar datos estructurados y no estructurados. El intercambio de datos en memoria hace que los RDD sean 10-100 veces más rápidos que el uso compartido de redes y discos.





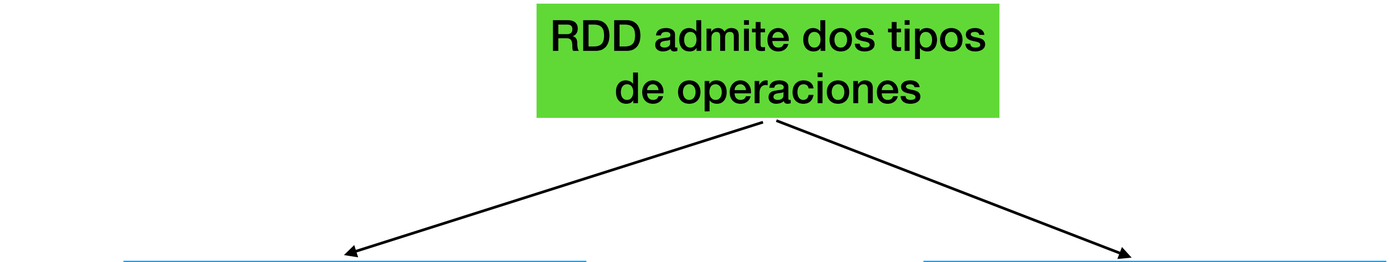
los datos de un RDD se dividen en trozos basados en una clave. Los RDD son altamente resistentes, es decir, pueden recuperarse rápidamente de cualquier problema, ya que los mismos trozos de datos se replican en múltiples nodos ejecutores. Por lo tanto, incluso si un nodo ejecutor falla, otro seguirá procesando los datos. Esto le permite realizar sus cálculos funcionales contra su conjunto de datos muy rápidamente aprovechando el poder de múltiples nodos.


PySpark
RDDs
In - Memory Computation
Lazy Evaluation
Fault Tolerant
Immutability
Partitioning
Persistence
Coarse Grained Operations

Características de los RDDS

- **Cálculos en memoria:** mejora el rendimiento en un orden de magnitudes.
- **Evaluación perezosa:** Todas las transformaciones en los RDD son perezosas, es decir, no calculan sus resultados de inmediato.
- **Tolerante de fallos:** los RDD rastrean la información de linaje de datos para reconstruir los datos perdidos automáticamente.
- **Inmutabilidad:** los datos se pueden crear o recuperar en cualquier momento y, una vez definidos, su valor no se puede cambiar.
- **Partición:** es la unidad fundamental del paralelismo en PySpark RDD.
- **Persistencia:** los usuarios pueden reutilizar los RDD de PySpark y elegir una estrategia de almacenamiento para ellos.
- **Operaciones grano grueso :** estas operaciones se aplican a todos los elementos de los conjuntos de datos a través de mapas o filtros o grupos por operación.

Operaciones de los RDD en PySpark



Estas son las operaciones que se aplican a un RDD para crear un nuevo RDD. Las transformaciones siguen el principio de las evaluaciones perezosas (lo que significa que la ejecución no comenzará hasta que se active una acción). Esto le permite ejecutar las operaciones en cualquier momento simplemente llamando a una acción sobre los datos.

Las acciones son las operaciones que se aplican en un RDD para indicar a Apache Spark que aplique el cálculo y devuelva el resultado al controlador.

Transformaciones

Algunas transformaciones de los RDDS

- map
- flatMap
- filter
- distinct
- reduceByKey
- mapPartitions
- sortBy

Acciones

Algunas Acciones de los RDDS

- collect
- collectAsMap
- reduce
- countByKey/
  countByValue
- take
- first

Ventajas de los RDDS

- **Data resilience**. El mecanismo de autorrecuperación garantiza que los datos nunca se pierdan, independientemente de si una máquina falla.
- **Data consistency**. Dado que los RDD no cambian con el tiempo y solo están disponibles para su lectura, la coherencia de los datos se mantiene a lo largo de las diversas operaciones.
- **Performance speeds**. Almacenar datos en RAM siempre que sea posible en lugar de en el disco. Sin embargo, los RDD mantienen la posibilidad de almacenamiento en disco para proporcionar un aumento masivo de rendimiento y flexibilidad.

Desventajas de los RDDS

- **No schematic view of data**. Los RDD tienen dificultades para lidiar con los datos estructurados. Una mejor opción para manejar datos estructurados es a través de las API de DataFrames y Datasets, que se integran completamente con los RDD en Spark.

- **Garbage collection**. Dado que los RDD son objetos en memoria, dependen en gran medida de la gestión y serialización de la memoria de Java. Esto causa limitaciones de rendimiento a medida que los datos crecen.

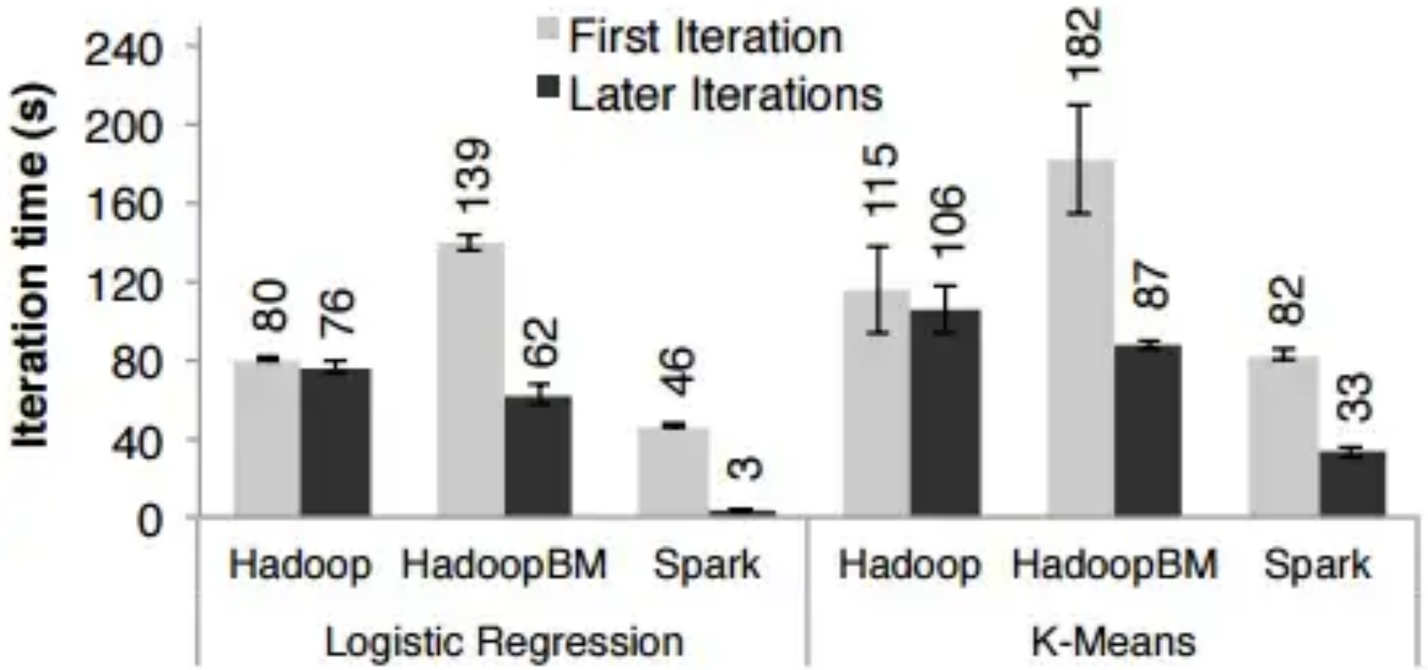
- **Overflow issues**. Cuando los RDD se agotan la RAM, la información reside en un disco, lo que requiere RAM y espacio en disco adicionales para superar los problemas de desbordamiento.

- **No automated optimization**. Un RDD no tiene funciones para la optimización automática de la entrada. Mientras que otros objetos de Spark, como DataFrames y Datasets, utilizan el optimizador Catalyst, para los RDD, la optimización se realiza manualmente.

-

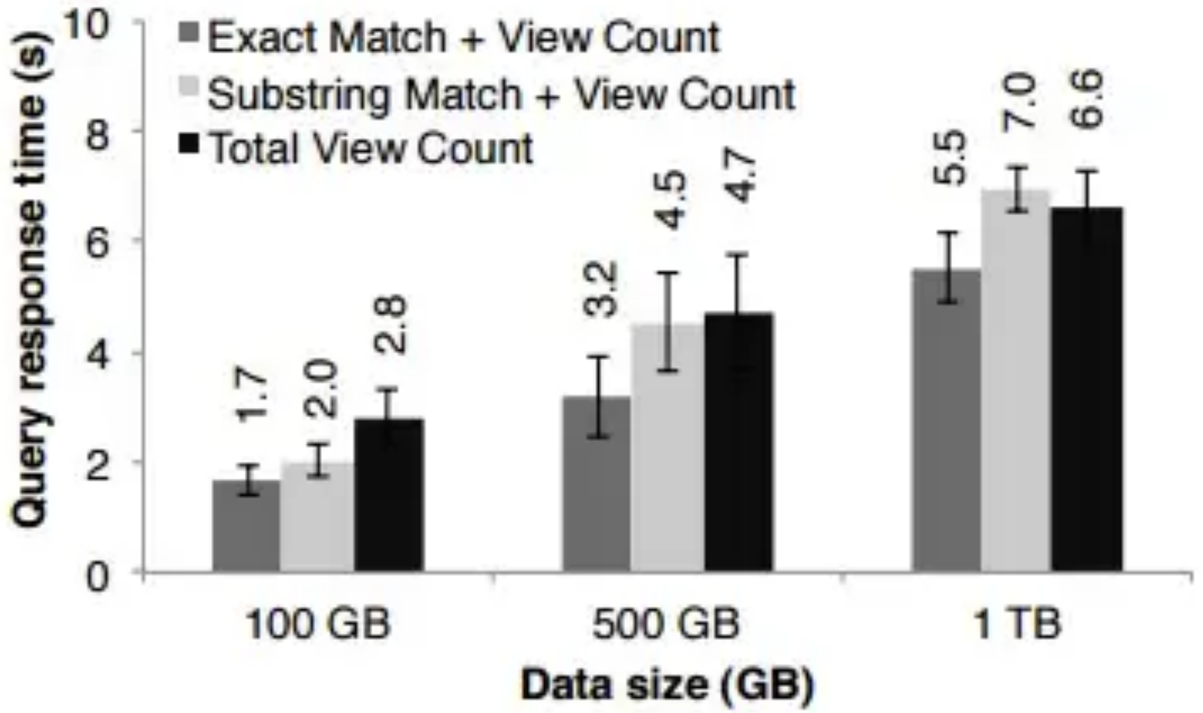
Evaluación RDDS

Aplicaciones Machine learning



Evaluación RDDS

Queries Tiempo de respuesta



PySpark RDD

Caso de Uso

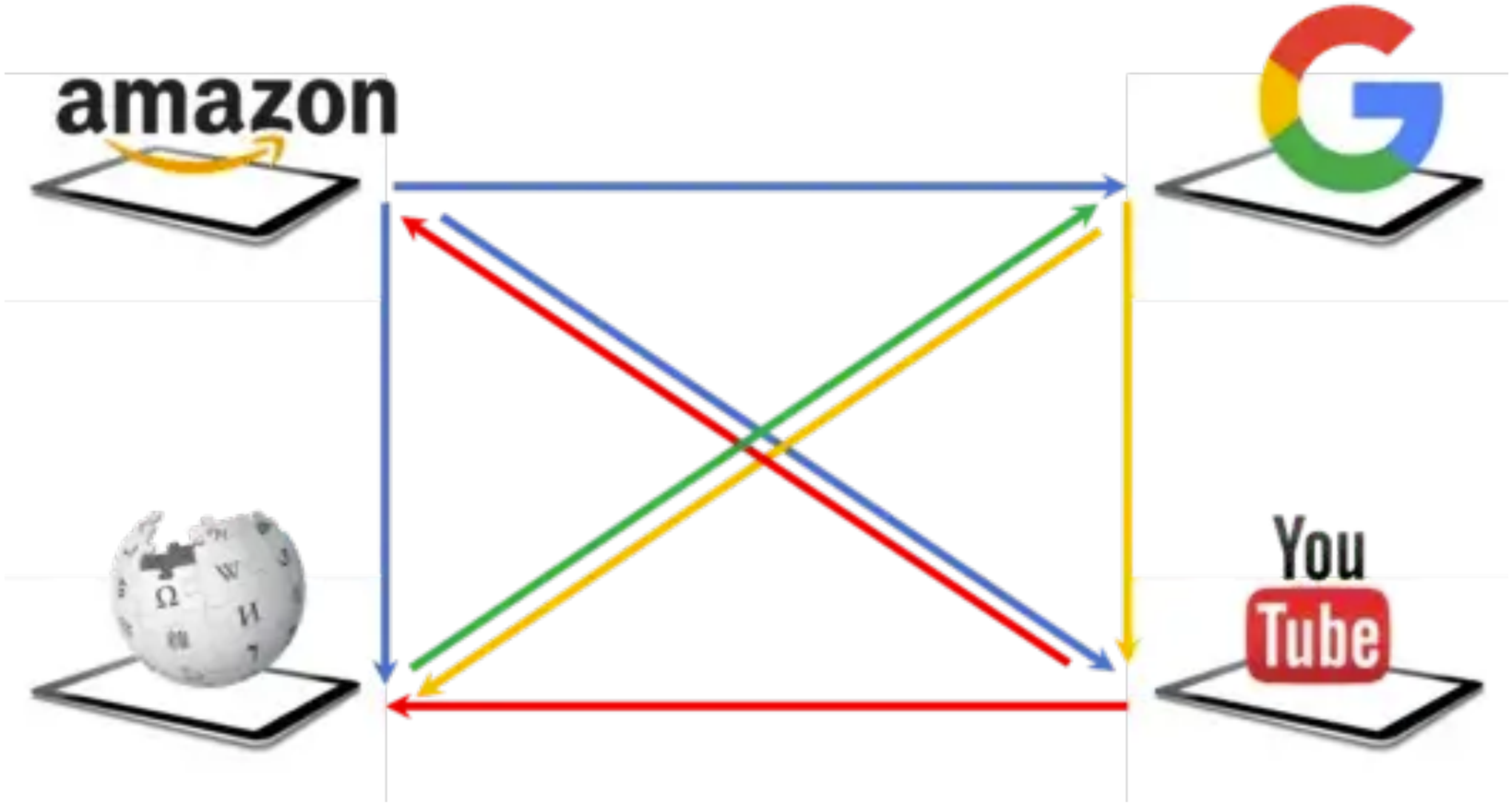
- Problema

Tienes que calcular el rango de página de un conjunto de páginas web basadas en el sistema de páginas web ilustradas. A continuación se muestra un diagrama que representa cuatro páginas web, Amazon, Google, Wikipedia y YouTube, en nuestro sistema. Para facilitar el acceso, nombremoslos a, b, c y d, respectivamente.

Aquí, la página web "a" tiene enlaces salientes a las páginas b, c y d. Del mismo modo, la página "b" tiene un enlace saliente a las páginas d y c. La página web "c" tiene un enlace de salida a la página b, y la página "d" tiene un enlace de salida a las páginas a y c.

PySpark RDD

Caso de Uso



PySpark RDD

Caso de Uso - Solución

Para resolver esto, implementaremos el algoritmo de rango de página desarrollado por Sergey Brin y Larry Page.

Este algoritmo ayuda a determinar el rango de una página web en particular dentro de un grupo de páginas web.

Cuanto mayor sea el rango de la página, más alto aparecerá en una lista de resultados de búsqueda. Por lo tanto, tendrá más relevancia.

$$PR_{t+1}(P_i) = \sum_{P_j} \frac{PR_t(P_j)}{C(P_j)}$$

PRt+1(Pi) = Page rank of a site
PRt(Pj) = Page rank of an inbound link
C(Pj) = Number of links on that page

Que es un page rank?

El PageRank es un algoritmo de Google lanzado en el año 1999 por los fundadores de la compañía, Larry Page y Sergey Brin. Su función era la de medir la importancia y la calidad de una página web en un rango de 0 a 10, siguiendo varios criterios

La idea del PageRank está poderosamente inspirada en el *Science Citation Index* (SCI), el índice de citación más conocido en todo el mundo, desarrollado por el químico Eugene Garfield alrededor de la década de los 60.

Que es un page rank?

PageRank funciona de manera muy similar, pero aplicado al mundo de la Web.

El valor de una página web –llamémosle X– es determinado a través de los enlaces que le son otorgados por un sitio Web Y, teniendo en cuenta, además, otros factores como la calidad y la importancia del sitio web que otorga el enlace.

Por consiguiente, tener muchos enlaces no quería decir tener un PageRank de 10, sino que su calidad también era un factor más que importante en el cálculo de cuan influyente era un portal o sus diferentes páginas.

En nuestra declaración de problema, se muestra que la página web "a" tiene tres enlaces salientes. Por lo tanto, de acuerdo con el algoritmo, la contribución al rango de página de página d por página a es PR(a) / 3.

Ahora tenemos que calcular la contribución de la página b a la página d. La página b tiene dos enlaces salientes: el primero a la página c y el segundo a la página d. Por lo tanto, la contribución de la página b es PR(b) / 2.

Por lo tanto, el rango de página de la página d se actualizará de la siguiente manera, donde s se conoce como el factor de amortiguación:

$$PR(d) = 1 - s + s \times (PR(a)/3 + PR(b)/2)$$