TAREA 3: SISTEMAS DISTRIBUIDOS
Hadoop: Motor de búsqueda
Profesor: Nicolás Hidalgo
Ayudantes: Cristian Villavicencio, Joaquín Fernandez, Nicolás Núñez y Felipe Ulloa

LEA EL DOCUMENTO COMPLETO ANTES DE EMPEZAR A DESARROLLAR LA TAREA

Objetivo

El objetivo de este trabajo es introducir a los estudiantes a los *sistemas de procesamientos para grandes volúmenes de datos*. Para ello, los alumnos deberán trabajar con **Hadoop**, un proyecto de software de código abierto que se puede utilizar para procesar de forma eficaz conjuntos de datos de gran tamaño. Los alumnos deberán esta tecnología por medio de la generación de un código enfocado a map-reduce y reconocer las ventajas de su uso.

Conceptos previos

Hadoop nace con la necesidad de procesar grandes volúmenes de datos. En lugar de utilizar un equipo grande para procesar y almacenar los datos, Hadoop facilita la creación de clústeres de hardware de consumo para analizar conjuntos de datos masivos en paralelo. De la misma manera, **Hadoop** es confiable, tolerante a fallos, de bajo coste y escalable, siendo una solución completa en su área.

Si usted trabaja con Hadoop, hay ciertas cosas que debe conocer:

- **Índice Invertido**: Un índice invertido, también conocido como índice de búsqueda inversa, es una estructura de datos que se utiliza en los motores de búsqueda para facilitar la recuperación rápida de la información. La idea básica es que, en lugar de listar los datos y luego los documentos (o ubicaciones) donde se encuentran, se listan los documentos y se identifican los datos que contienen. Por lo tanto, si tienes una consulta de búsqueda, puedes identificar rápidamente los documentos que contienen esa consulta.
  Por ejemplo, en el caso de un motor de búsqueda de texto, un índice invertido almacenaría una lista de palabras y para cada palabra, una lista de los documentos en los que aparece. Entonces, si alguien busca la palabra "Hadoop", el motor de búsqueda consultaría el índice invertido y recuperaría rápidamente todos los documentos que contienen esa palabra.

- **HDFS**: Es el componente principal del ecosistema Hadoop. Esta pieza hace posible almacenar data sets masivos con tipos de datos estructurados, semi-estructurados y no estructurados como imágenes, vídeo, datos de sensores, etc. Está optimizado para almacenar grandes cantidades de datos y mantener varias copias para garantizar una alta disponibilidad y la tolerancia a fallos. Con todo esto, HDFS es una tecnología fundamental para Big Data, o dicho de otra forma, es el Big Data File System o almacenamiento Big Data por excelencia. La utilización del HDFS es muy similar al sistema de archivos existentes en unix:
  - **mkdir**: 'crea un directorio'
  - **ls**: 'lista los archivos de un directorio'
  - **cat**: 'lista el contenido de un archivo'
  - Puede encontrar más funcionalidades en el siguiente link: `https://aprenderbigdata.com/comandos-hdfs/`

- **Map Reduce**: Hadoop MapReduce es un paradigma de procesamiento de datos caracterizado por dividirse en dos fases o pasos diferenciados: Map y Reduce. Estos subprocesos asociados a la tarea se ejecutan de manera distribuida, en diferentes nodos de procesamiento o esclavos. Para controlar y gestionar su ejecución, existe un proceso Master o Job Tracker. También es el encargado de aceptar los nuevos trabajos enviados al sistema por los clientes. Para más información consulte: `https://aprenderbigdata.com/hadoop-mapreduce/`.

Problema

Después de una exhaustiva búsqueda, has logrado conseguir una posición como ingeniero en una prestigiosa compañía especializada en el análisis cinematográfico. La misión central de la empresa es ilustrar la evolución de los artistas a través de su trayectoria profesional. Para lograrlo, es esencial seleccionar un conjunto de películas que representen diversas fases de sus carreras.

La base de datos de IMDb, con su vasto catálogo de miles de películas y figuras del mundo del cine, presenta un desafío considerable. Buscar manualmente cada película en la que un individuo específico ha trabajado sería una tarea monumental, consumiendo gran cantidad de tiempo y recursos. Este desafío se multiplica al considerar la participación de múltiples actores, directores y guionistas.

Para abordar este desafío, propones la creación de un motor de búsqueda basado en la implementación de un índice invertido utilizando MapReduce en Hadoop. Esta técnica innovadora permite mapear nombres de actores, directores o guionistas a todas las películas en las que han trabajado. Así, al introducir el nombre de un individuo en la búsqueda, se puede obtener de manera rápida y eficiente una lista completa de todas las películas en las que ha participado, facilitando enormemente la tarea de análisis y selección de películas.

Qué hacer

- Descargar las bases de datos (en formato .tsv) de IMDb: `https://datasets.imdbws.com/`. La documentación: `https://developer.imdb.com/non-commercial-datasets/`
- Investigar sobre algún código de MapReduce que permita construir un índice invertido; la llave debe ser el *nombre del actor* y el valor los *ids* de las peliculas en los que ha participado. Es importante mencionar que este índice invertido puede ser construido a partir de los datos proporcionados por IMDb, en específico, el archivo *name.basics.tsv.gz* y *title.principals.tsv.gz*. El primero contiene información básica sobre las personas (incluyendo actores), y el segundo contiene información sobre los roles de cada persona en las películas, que incluye el identificador único de la persona y el identificador único de la película. Genere un diagrama que muestre el funcionamiento del código. En caso de que utilice alguna otra opción que sea factible, coméntelo. **(30 puntos)**
- Generar un trabajo utilizando Hadoop. Este trabajo de generará Key-values **(30 puntos)**:

Actor	IDs películas
Tom Hanks	”tt0075686”, ”tt0082766”, ”tt0110971”
Meryl Streep	”tt0075686”, ”tt0082766”, ”tt0110971”
Daniel Day-Lewis	”tt0469494”, ”tt0200720”, ”tt0167260”
...	....

  Explicación:

  - El actor *Tom Hanks* participa en las películas con *id*: ”tt0075686”, ”tt0082766”, ”tt0110971”.
  - El actor *Meryl Streep* participa en las películas con *id*: ”tt0075686”, ”tt0082766”, ”tt0110971”.
  - El actor *Daniel Day-Lewis* participa en las películas con *id*: ”tt0469494”, ”tt0200720”, ”tt0167260”.
  - ***Nota**: Los datos entregados en el ejemplo son inventados.*

- Guarde los valores anteriores en alguna base de datos (de cualquier tipo). Idealmente guarde también información sobre las películas. **(Sin puntaje)**
- Genere un buscador en el lenguaje que usted estime conveniente. Deberá utilizar como regla el buscar las coincidencias con el índice ingresado. Deberá poder seleccionar una película, mostrando todas las coincidencias existentes. **(20 puntos)**
- Genere un video de no más de 10 minutos mostrando el funcionamiento completo.**(10 puntos)**

Entregables

Al igual que las tareas pasadas, deberá entregar:

- Un breve informe mostrando los códigos y las justificaciones de los items correspondientes.
- Un video de no más de 10 minutos mostrando el trabajo realizado.
- Un repositorio con los códigos utilizados.

Observaciones

- El conjunto de datos de IMDB tiene un tamaño de 17 GB, por lo que si es necesario, siéntanse libres de reducirlo. Sin embargo, asegúrense de que el tamaño reducido siga siendo significativo, ya que si se reduce demasiado, puede resultar más difícil cumplir con los objetivos de la experiencia.

Aspectos formales de entrega

- Esta tarea es opcional. La entrega de esta tarea reemplazará la peor nota de las tareas.
- **Fecha de entrega:** 23 de Junio hasta las 23:59 hrs.
- **Número de integrantes:** Grupos de 2 personas las cuales deben estar claramente identificadas en la tarea.
- **Lenguaje de Programación:** para la implementación debe escoger entre los siguientes lenguajes: No existe limitación. Procure utilizar el que le brinde mayor confianza en cualquier caso.
- **Formato de entrega:** Repositorio público (Github o Gitlab), video de funcionamiento e informe en formato PDF.
- **Tecnologías complementarias:** En caso de usar tecnologías complementarias, añadir una descripción en el informe.
- Las copias de código serán penalizadas con nota mínima. Referente apropiadamente todo segmento de código que no sea de su autoría.
- No se debe implementar un front o interfaz, solo basta con implementar una API REST o una interfaz interactiva en la terminal.
- Consultas: **nicolas.nunez2@mail.udp.cl** o **nk#2258** y **joaquin.fernandez1@mail.udp.cl** o **Jøacø#9352**