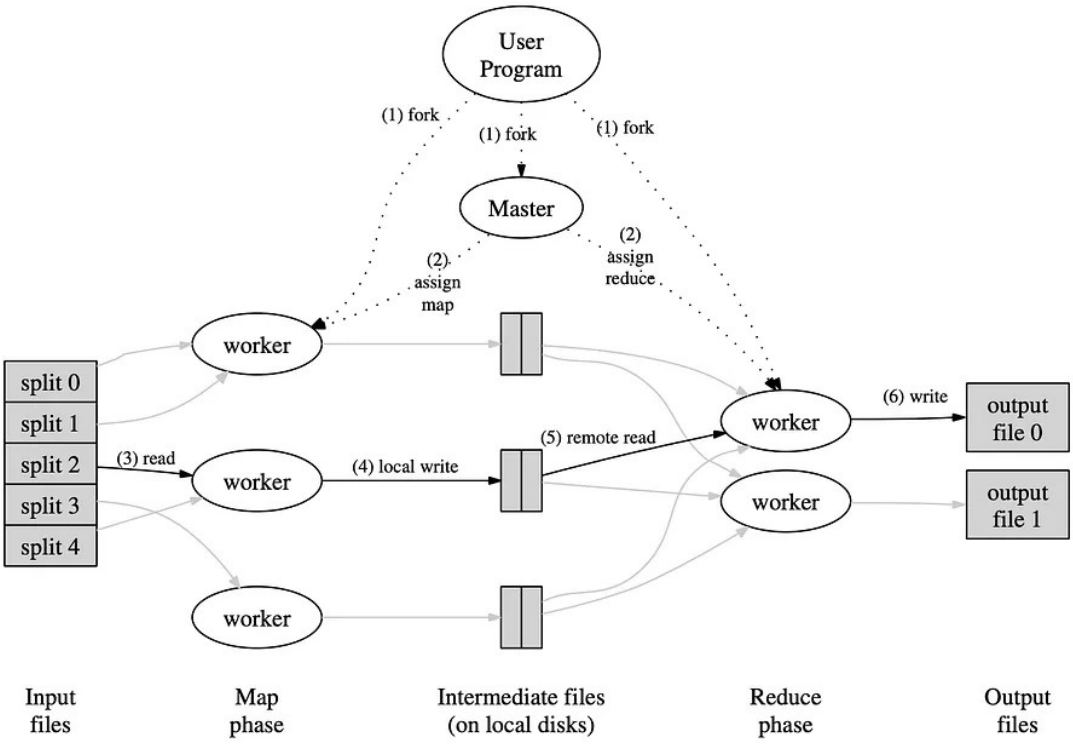
Tarea 1 MapReduce

Revisar en este enlace: Introduccion a MapReduce

La tarea consiste en el uso del algoritmo MapReduce para construir, dado un conjunto de paginas web, un sistema de conteo de palabras.

En MapReduce los archivos de entrada están divididos. El nodo maestro crea varios nodos trabajadores para asignar estos archivos. Las palabras se colocan en un archivo intermedio. Después de la fase de mapeo, el nodo maestro crea múltiples trabajadores reductores. El resultado final es el recuento de cada palabra que se encuentra dentro de los archivos.

En la siguiente figura se muestra la arquitectura de MapReduce:



El algoritmo esta basado en dos funciones:

Un Maestro: que maneja todas las E/S hacia y desde el archivo. Crea clases de mapeador y combinador para el uso de MapReduce

Los datos de partida serían un conjunto de paginas web con el formato “(*documento, lista de palabras contenidas en el documento*)”

- Se define una función ***map*** que transforma esos pares en otros diferentes donde la clave es una palabra concreta y el valor siempre es 1, es decir, genera un elemento “(*palabra, 1*)” por cada palabra que ve en los documentos que procesa.

- Se define una función ***reduce*** que se encarga de procesar todos los elementos "(*palabra, 1*)" de la misma palabra y calcular la suma final para cada palabra.

La lógica que funciona por debajo del modelo es la que va asignando a las diferentes máquinas la tarea de hacer ***map*** o hacer ***reduce*** (y los datos con que hacerlas) y de ir integrando todos los resultados.
****

Los siguientes 6 pasos resumen cómo funciona el algoritmo:

1. Verifique los archivos de entrada y divídalos en varias partes.

2. Iniciar copias del programa MapReduce en un grupo de máquinas. Una de las copias será la maestra y el resto de trabajadores. El maestro asigna tareas de Map y Reduce a los trabajadores.

3. El trabajador asignado a una tarea de Mapa lee el contenido de su parte de la entrada. Analiza pares clave/valor de la entrada y los pasa a la función Map proporcionada por el usuario. La salida de la función Map son pares clave-valor intermedios.

4. Los pares clave-valor se particionan y su ubicación de almacenamiento se proporciona al maestro. Las ubicaciones de las particiones de pares intermedios son las entradas para los trabajadores de Reduce.

5. Un trabajador de Reduce lee los pares y agrupa todas las apariciones de la misma clave. Luego, pasa la clave y el conjunto de valores correspondiente a la función Reduce proporcionada por el usuario. La salida de la función Reduce se adjunta a un archivo de salida.

6. Una vez que se completan todas las tareas de Map y Reducce, el maestro devuelve el control al usuario. Luego, el usuario puede acceder al resultado de la ejecución de MapReduce a través de los archivos de salida. Hay disponible un archivo de salida para cada uno de los trabajadores de Reduce.

****

Su tarea es:

1. Implemente este algoritmo usando un solo hilo / proceso. Contabilice el numero de palabras y el rendimiento del algoritmo secuencial.
2. Implemente la version distribuida de este algoritmo. Utilice RPC para comunicar al archivo Maestro con sus trabajadores.

   Debe construir un sistema de archivos distribuidos: pueden usar el protocolo NFS para el manejo de los archivos distribuidos.

   - NFS (sistemas de archivos distribuidos) NFS es un protocolo cliente-servidor para compartir archivos distribuidos que se utiliza comúnmente para sistemas de almacenamiento conectados a la red.

- Tambien podria usar Apache Hadoop para crear el sistema de archivos distribuidos

Como lenguaje de implementacion: le sugiero Java o C++ o GO.

Que debe entregar?

1. Documento en pdf con:

- diseno del algoritmo en la version secuencial.
- la version distribuida del algoritmo.
- Para ambos casos debe detallar la solucion: la arquitectura, protocolo (su implementacion y uso) .

2. Ejecucion de las soluciones y el rendimiento de cada algoritmo

3. El documento de entrada: puede usar un grupo de paginas de wikipedia o publicaciones academicas de Redalyc

4. Video con la presentacion de su trabajo. Maximo 10 minutos.

5. Codigo en Github o Gitlab