Maestría en Computación: Tarea - MapReduce

Entregar el 28 de Septiembre 2020

tecDigital 12:pm

José Castro

Contents

Introducción

> *MapReduce* es un estilo de cómputo que se ha implementado en varios sistemas, incluyendo la implementación interna de Google (simplemente llamada Mapreduce) ...

Así empieza el apartado sobre MapReduce en el libro "Mining of Massive Data Sets". El MapReduce es un componente fundamental en la implementación de bases de datos NoSQL y cualquier aplicación que pretenda hacer minería/análisis de datos sobre archivos de volumen grande, que también tenga requerimientos fuertes sobre la escalabilidad del cómputo va a necesitar de utilizar alguna implementación de esta infraestructura.

Por otra parte las bases de datos distribuidas en compañías como Facebook, WhatsApp, Discord, y otras tienen un tamaño y volumen de transacciones que requieren de plataformas tolerantes a fallas y de alta disponibilidad que les permitan desarrollar tanto sus back ends de manera eficiente y confiable.

Todas estas empresas han optado en parte de su infraestructura el uso de la plataforma OTP (Open Telecom Platform), la cual implementa los principios vistos en el artículo "Why do Computers Stop and What Can Be Done About It", de separación de cómputo en procesos, uso de mensajes para implementar el modelo de Actores, falla rápida del sistema, y procesos y árboles de supervisión, como principios para generar el software escalable y tolerante a fallas.

A su vez, el *PageRank* es un algoritmo utilizado por todos los buscadores modernos para ranquear las páginas de internet, pero que también tiene mucha utilidad en análisis de grafos, redes sociales, interacciones monetarias, y todo lo que tenga que ver con minería de la estructura de un grafo.
Es el propósito de compuesto de esta tarea:

- Exponer al estudiante al Open Telecom Platform y sus principios de diseño para software tolerante a fallas.
- Desarrollar una aplicación en esta plataforma que sea fácilmente escalable a nodos distribuidos, con la posibilidad de hacer cómputo distribuido.
- Implementar el MapReduce dentro de esta plataforma.
- Utilizar la implementación de MapReduce para crear una versión respetable (escalable a muchos nodos, que pueda manejar grafos con millones de nodos) del PageRank.

La entrega de la tarea será por cada items, de forma secuencial, revisándose con citas de revisión con los miembros de cada grupo. Después de la primera entrega, las que siguen serán cada una, una semana después.

La primera entrega se fija para el 30 de Setiembre y corresponde al punto 1. La segunda son los puntos 2 y 3. La tercera entrega los puntos 4 y 5.

Problema 1

(15 pts)
Termine la implementación de MapReduce que se le da en erlang. Esta implementación la debe probar ejecutándola en varios nodos distribuidos (distributed erlang) tal que pueda configurar libremente la cantidad de tareas Map, tareas Reduce, tamaño de los chunks, y ubicación de cada una de las tareas en su red de nodos erlang.

La entrada de su mapReduce la debe hacer desde un archivo que contenga términos erlang de la forma `{llave, num1, num2}`, las tareas map deben sumar los números y dejar un solo registro de {llave, num}, las tareas reduce, deben sumar todos los números asociados a una misma llave.

Problema 2

(10 pts)
En este segundo ejercicio debe estructurar su código tal que lo que se comporte bajo el `behavior` de `gen_server`, esté comportamiento de OTP permite generar servidores robustos, con políticas de ejecución y cambio de código en caliente, si es necesario

Problema 3

(10 pts)
En este apartado de la tarea debe estructurar su código dentro de una aplicación (application) del OTP, las aplicaciones OTP, en el archivo de configuración de la aplicación puede definir si tiene una jerarquía de procesos de supervisión, los cuales le permiten definir políticas en caso de que alguno de los procesos se caiga.

Una vez hecho esto debe garantizar que su MapReduce funcione de manera distribuida, y que siga las políticas de restart o abort, definidas para los procesos map, reduce, y orquestador general que se esbozan en el capítulo 2 del libro MMDS.

Problema 4

(15 pts)
Implemente ahora la tarea de multiplicar una matriz por un vector, debe suponer que la matriz es rala y gigantesca, esto es, la matriz cumple con las propiedades de la matriz utilizada por el PageRank. Tanto el vector como la matriz de multiplicación las debe cargar de un archivo.

El formato de la matriz es el simple: cada linea es una página, que nada más contiene una lista pequeña de números de páginas a las cuales hace link. El vector es arbitrario.

Asuma que el tamaño del archivo tanto del vector como de la matriz puede que no le quepa en memoria, y que por lo tanto lo debe leer secuencialmente o en bloques.

Problema 5

15 pts
Ahora haga una versión iterativa del algoritmo de multiplicación tal que pueda implementar el PageRank completo. Agregue los parámetros de cantidad de iteraciones y β que tiene el algoritmo de PageRank.