2020 - 2021

ACTIVIDAD GUIADA 2

Máster Universitario en Big Data y Ciencia de Datos

03MBID – Procesamiento de datos masivos

Curso 2020-2021 – Ed. octubre

viu Universidad Internacional de Valencia

1 . Descripción general

Actividades Guiadas

DESCRIPCIÓN	
Introducción	Cuando se desarrollan programas Big Data es habitual hacer pruebas en local para ver si funciona como se espera. Sin embargo, el programa también debería funcionar en producción y ser escalable. Habitualmente la ejecución de los programas Big Data se delega en un framework que se encarga de asignar recursos en paralelo, re-ejecutar partes del programa si hay fallos de infraestructura, etc. Si nuestro programa no está diseñado correctamente según el modelo de procesamiento Big Data, va a fallar en alguna de las ejecuciones que haga el framework y va a ejecutarse correctamente en otras. Es importante conocer cómo diseñar correctamente los programas y ser capaces de detectar defectos de diseño en los programas desarrollados.
Objetivo	Conocer el modelo de procesamiento MapReduce, sus primitivas y cómo se ejecutan los programas Big Data. Ser capaz de entender defectos en los programas que estén mal diseñados y saber cómo se podrían corregir.
Trabajo previo	Lectura del material docente de la parte específica que se encuentra disponible desde el comienzo del curso en la carpeta: Recursos y materiales>1. Materiales docentes: Visualización de las videoconferencias sobre instalación del entorno y ecosistema Hadoop que se encontrarán disponibles en: Videoconferencias>grabaciones
Metodología	En las videoconferencias teóricas se expondrá al alumno conocimientos, material e indicaciones suficientes para que pueda elaborar una unidad didáctica basada en el aprendizaje y enseñanza por competencias en matemáticas e informática. En la videoconferencia de actividad guiada se establecerá las pautas concretas y la dinámica que el alumnado deberá seguir para realizar la actividad propuesta. Las actividades se centrarán en poner en práctica y asentar los conocimientos adquiridos en la videoconferencia teórica anterior.

Tarea para el e-portfolio	Dados dos programas Big Data que tienen defectos de diseño, se debe entender cuál es el defecto y hacer un informe que contenga: (1) Nombre y apellidos, (2) Tiempo empleado por el alumnado en entender cuál es el defecto, y (3) Descripción del defecto. IMPORTANTE: los defectos del programa son defectos del diseño. La sintaxis del programa es correcta, pero la funcionalidad del programa no se ha programado correctamente siguiendo el modelo de procesamiento Big Data. Por ello, puede que los programas los ejecutemos en nuestro ordenador y funcionen correctamente, pero al moverlos al cluster Big Data empiecen a fallar porque están ejecutando varias Mapper, Combiner, Reducer, algunas de ellas puede que se re-ejecuten, acaben antes, etc. Es decir, si ejecutamos dos veces en un cluster de producción el mismo programa con los mismos datos, podría una vez emitir la salida correcta y otra vez una incorrecta. Esto es porque el programa no se diseñó adecuadamente siguiendo el modelo de procesamiento Big Data. Un programa bien diseñado, debería ejecutarse correctamente independientemente de cómo el cluster Big Data decida ejecutarlo. A continuación, se describe cada programa y las preguntas que se tienen que responder: **Programa 1**: todos los archivos del programa se encuentran en la carpeta “ModaGastoPorPersona”. Conjunto de datos: cada fila representa el gasto que hizo una persona con la siguiente estructura: “persona gasto”. Por ejemplo “Alice 10” significa que Alice hizo una compra en la que gastó 10 euros. Descripción del programa: el programa tiene que calcular para cada persona, cual es el gasto más frecuente (la moda). Es decir, si se tiene como entrada “Alice 10”, “Alice 3” y “Alice 10”, el gasto más frecuente de Alice es 10 porque se repite 2 veces, mientras que el gasto 3 sólo se repite 1 vez. Código: el equipo de desarrollo ha creado los scripts mapperModaGastoPorPersona.py, combinerModaGastoPorPersona.py y reducerModaGastoPorPersona.py. Comando de ejecución: hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar -file ./mapperModaGastoPorPersona.py -mapper ./mapperModaGastoPorPersona.py -file ./combinerModaGastoPorPersona.py -combiner combinerModaGastoPorPersona.py -file ./reducerModaGastoPorPersona.py -reducer ./reducerModaGastoPorPersona.py -input casoDePrueba.txt -output ./misalida (notar que dependiendo de la versión de Hadoop, habría que cambiar el .jar y también las entradas y salidas)

Problema: el equipo de analistas ha observado que el programa no funciona correctamente. Según reportan, han ejecutado el programa con los mismos datos y unas veces proporciona la moda de forma adecuada, pero en otras ocasiones el programa emite un gasto que claramente no es la moda. De todos los datos que hay en producción, han reportado que el defecto se puede reproducir con sólo 102 datos que están disponibles en casoDePrueba.txt. La salida esperara es “Alice 10“ y “Bob 91”. Sin embargo, cuando el programa se ejecuta en producción hay ocasiones en las que emite correctamente esos valores, pero en otras ocasiones emite “Alice 16” y otras modas.

Se tiene que analizar el programa para entender el defecto y posteriormente realizar un informe llamado ModaGastoPorPersona.pdf que contenga lo siguiente:

1) Nombre y apellidos
2) Tiempo empleado en entender el defecto del programa
3) Descripción del defecto:
    a. Circunstancias bajo las que falla el programa: se tiene que indicar en qué ejecuciones podría fallar el programa.
    b. Motivos por los que falla el programa: se tiene que describir qué es lo que tiene erróneo el programa y que lo hace fallar.
    c. Directrices para corregir el defecto: se tiene que indicar a grandes rasgos lo que tendría que cambiar el equipo de desarrollo para eliminar el defecto del programa. No hace falta desarrollar el programa correcto, pero sí hay que indicar qué se tendría que cambiar.

Programa 2: Todos los archivos del programa se encuentran en la carpeta “PersonasQueCompranEnMuchasTiendas”.

Conjunto de datos: cada fila representa una compra que hizo una persona en una tienda. La fila tiene la siguiente estructura: “persona tienda”. Por ejemplo “Alice Nunc Corp.” significa que Alice compró en “Nunc Corp.”.

Descripción del programa: el programa tiene que obtener cuáles fueron las personas que compraron en 3 o más tiendas diferentes. Es decir, si se tiene como entrada “Alice Nunc Corp.”, “Alice Arcu Aliquam Company”, “Alice Pharetra Quisque Ac Company”, “Alice Nunc Corp.”, y “Bob Nunc Corp.”, el programa debería emitir Alice porque compró en 3 o más tiendas distintas. Concretamente, en el ejemplo anterior, Alice compró en 3 tiendas: en “Nunc Corp.” (dos compras), “Arcu Aliquam Company”, y “Pharetra Quisque Ac Company”. El programa no emite Bob porque sólo compró en una tienda.

Código: el equipo de desarrollo ha creado los scripts mapperPersonasQueCompranEnMuchasTiendas.py, combinerPersonasQueCompranEnMuchasTiendas.py y

reducerPersonasQueCompranEnMuchasTiendas.py.

Comando de ejecución: hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-2.4.0.jar -file ./mapperPersonasQueCompranEnMuchasTiendas.py -mapper ./mapperPersonasQueCompranEnMuchasTiendas.py -file ./combinerPersonasQueCompranEnMuchasTiendas.py -combiner combinerPersonasQueCompranEnMuchasTiendas.py -file ./reducerPersonasQueCompranEnMuchasTiendas.py -reducer ./reducerPersonasQueCompranEnMuchasTiendas.py -input casoDePrueba.txt -output ./misalida

(notar que dependiendo de la versión de Hadoop, habría que cambiar el .jar y también las entradas y salidas)

Problema: el equipo de analistas ha observado que el programa no funciona correctamente. Según reportan, han ejecutado el programa con los mismos datos y unas veces proporciona las personas que realmente compraron en 3 o más tiendas, pero en otras ocasiones el programa sólo emite alguna de esas personas. De todos los datos que hay en producción, han reportado que el defecto se puede reproducir con sólo 104 datos que están disponibles en casoDePrueba.txt. La salida esperara es Alice, Carol y Dave. Sin embargo, cuando el programa se ejecuta en producción hay ocasiones en las que emite correctamente a esas tres personas, pero en otras ocasiones sólo emite Alice y Dave.

Depuración: el equipo de pruebas ha utilizado una herramienta de localización y de reducción de datos para depurar el programa. Han obtenido lo siguiente:

- El defecto ocurre cuando se ejecutan >1 Combiners
- El defecto se manifiesta en la siguiente configuración con sólo 3 datos: ver imagen reduccion.jpg

Se tiene que analizar el programa para entender el defecto y posteriormente realizar un informe llamado PersonasQueCompranEnMuchasTiendas.pdf que contenga lo siguiente:

1) Nombre y apellidos
2) Tiempo empleado en entender el defecto del programa
3) Descripción del defecto:
    a. Circunstancias bajo las que falla el programa: se tiene que indicar en qué ejecuciones podría fallar el programa.
    b. Motivos por los que falla el programa: se tiene que describir qué es lo que tiene erróneo el programa y que lo hace fallar.
    c. Directrices para corregir el defecto: se tiene que indicar a grandes rasgos lo que tendría que cambiar el equipo de desarrollo para eliminar el defecto del programa. No hace falta desarrollar el programa correcto, pero sí hay que indicar qué se tendría que cambiar.

	4) ¿te fue útil la información de depuración (la imagen reduccion.jpg y que el defecto se encontraba en >1 Combiners) para entender el defecto? Sí/No e indicar el por qué.
Forma de entrega	La memoria de cada actividad se subirá al site de la asignatura en formato zip conteniendo dos documentos pdf (ningún otro formato será admitido). Cada documento pdf debe contener el informe realizado para cada uno de los programas. Las memorias se realizarán de manera individual y cada alumno es responsable de subir a la plataforma la actividad. En cualquier caso, las entregas se realizarán dentro de los plazos establecidos en el calendario de la asignatura en 1» o 2» Convocatoria. Las entregas sólo serán válidas si se realizan a través del site de la asignatura: • Actividades>Actividad 2: detectar defectos
Fecha de entrega	
1ª Convocatoria	**23/01/2022 hasta las 23:59**
2ª Convocatoria	**03/04/2022 hasta las 23:59**

RÚBRICA DE EVALUACIÓN DE LAS ACTIVIDADES GUIADAS (20%)			
	Circunstancias bajo las que falla el programa (15%)	**Motivos por los que falla el programa (50%)**	**Directrices para corregir el defecto (35%)**
Muy competente	Se indican todas las circunstancias por las que falla el programa de forma correcta	Se describen correctamente los motivos por los que falla el programe y de forma detallada	Se indican unas directrices que son adecuadas para que un desarrollador pueda arreglar el defecto sin realizar ninguna depuración adicional
Competente	Se indica alguna de las circunstancias correctamente, pero no se indican todas o hay alguna incorrecta	Se describen correctamente los motivos por los que falla el programa, pero se hace de forma vaga	Se indican correctamente las directrices y son adecuadas para el desarrollador pueda arreglar el defecto, pero aun así el desarrollador tendría que hacer un poco de depuración
Aceptable	Se indica sólo alguna de las circunstancias o se indica alguna de forma vaga o incorrecta	Se describen parte de los motivos por los que falla el programa	Se indican correctamente las directrices con las que se tiene que solucionar el problema, pero no son suficientes para que el desarrollador lo solucione
Aún no Competente	No se indican las circunstancias o se indican incorrectamente	No se describen los motivos por los que falla el programa.	No se indican las directrices o se indican incorrectamente