Laboratorio 1 – Map Reduce y Hadoop

Miguel Alejandro González, José Ramón Alvarez, Fernando Arruza

Contexto de Presentación del documento

Universidad de los Andes, Bogotá, Colombia

{ma.gonzalez16, jr.alvarez, f.arruza[}@uniandes.edu.co](mailto:%7d@uniandes.edu.co)

Fecha de presentación: Septiembre 18 de 2017

Tabla de contenido

[1 Introducción 1](#_Toc493364200)

[2 Reto #1 2](#_Toc493364201)

[3 Reto #2 3](#_Toc493364202)

[4 Reto #3 5](#_Toc493364203)

[5 Reto #4 6](#_Toc493364204)

[6 Reto #5 (BONO) 6](#_Toc493364205)

[7 Conclusiones 6](#_Toc493364206)

# 

# Introducción

Este es un documento que describe las actividades realizadas y sus resultados obtenidos en la ejecución del Laboratorio 1 sobre Map Reduce y Hadoop, el cual persigue los siguientes objetivos:

* Desarrollar las habilidades básicas de uso de un *clúster* Hadoop.
* Desarrollar las habilidades básicas en procesamiento de información utilizando Map Reduce.
* Realizar una práctica de procesamiento básico paralelo sobre un dataset real.

El laboratorio incluye cuatro (4) retos en su alcance base y un (1) reto adicional reconocido como bono. Teniendo en cuenta esto, el documento se estructuró colocando una sección para cada reto que incluye el siguiente contenido:

* Alcance.
* Descripción de la solución.
* Código fuente.
* Resultado de las pruebas, que incluye dos (2) ejercicios; el primero utilizando como insumo un (1) *dataset* y el segundo utilizando como insumo cincuenta (50) *datasets*. El objetivo es comparar los resultados en cuanto a tiempo de procesamiento.

Para la implementación de los componentes de software se utilizó Java como lenguaje de programación tomando como base el proyecto de Maven que suministrado en la sesión presencial del laboratorio, el *dataset* fue suministrado por los profesores de la materia y contiene información de noticias de la agencia de prensa Reuters. Todas las pruebas fueron realizadas en el *clúster* Hadoop aprovisionado por la universidad como soporte para la materia.

# Reto #1

Alcance: Contar las noticas contenidas en el *dataset*.

Descripción de la solución: la revisión de los archivos de entrada permitió detectar que la estructura cumple con el formato XML, donde cada noticia está delimitada por un tag llamado “*REUTERS*”, por tanto se utilizó este criterio como expresión regular para identificar el inicio de cada noticia y así contarlas. El proceso inicia con la implementación de una función Map que actuará sobre cada línea de los archivos de texto procesados, identificando si es un encabezado de noticia y en caso positivo escribe una tupla en el contexto que tiene como llave la palabra “Noticias” y en el valor en número 1. En la revisión del código del laboratorio se identificó que la implementación de la función Reduce aplica para este reto ya que se persigue la misma meta, que consiste agrupar tuplas con la misma llave haciendo la sumatoria del valor.

Código fuente:

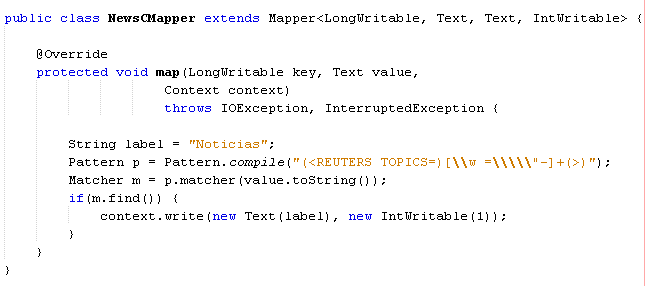


Figura 1. Implementación de Map del Reto #1.

Resultado de las pruebas:

Comando ejecutado:

*hadoop jar WordCountMR-0.0.1-SNAPSHOT.jar uniandes.reuters.job.NewsCounter /user/bigdata01/fernandofs/reuters21578.tar.gz /user/bigdata01/fernandofs/news\_counter*



Figura 2. Ejecución proceso del Reto #1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Prueba | Duración (segundos) | | | | | |
| **Ejec. 1** | **Ejec. 2** | **Ejec. 3** | **Ejec. 4** | **Ejec. 5** | **Promedio** |
| *Dataset* (1 lote) | 21.54 | 20.03 | 19.06 | 19.32 | 18.20 | 19.63 |
| *Dataset* (50 lotes) | 24.32 | 25.45 | 24.59 | 23.36 | 24.79 | 24.50 |

El tiempo promedio para las ejecuciones de un (1) lote es de 19.63 segundos detectando 21’597 noticias contenidas, llegando a procesar aproximadamente 704’000 líneas de información textual, a razón de 35’863 líneas por segundo, lo cual es un buen indicador considerando que el manejo de archivos es una operación costosa en tiempo.

El tiempo promedio para las ejecuciones de cincuenta (50) lotes es de 24.50 segundos detectando 1’078’950 noticias contenidas, llegando a procesar aproximadamente 35’200’000 líneas de información textual, a razón de 1’436’735 líneas por segundo.

Comparando los resultados de las dos (2) pruebas realizadas se puede concluir que la arquitectura tiene la capacidad de escalar ya que a pesar de que la cantidad de información a procesar se incrementó 50 veces la duración de la ejecución aumentó solo el **24.8 %**, en cambio el rendimiento aumento en el **4’006 %**.

# Reto #2

Alcance: Indique el título de cada noticia y cuántas palabras aparecen en el título de la misma, para los títulos que tiene en la primera línea más de 5 palabras.

Descripción de la solución: la revisión de los archivos de entrada permitió detectar que la estructura cumple con el formato XML, donde cada noticia está delimitada por un tag llamado “*REUTERS*” y contiene un tag llamado “*TITLE*”, por tanto se utilizó este criterio como expresión regular para identificar el título de cada noticia, contar sus palabras y en caso de que superaran el número de cinco (5) registrarla una tupla en el contexto que tuviera como llave el título de la noticia y como valor la cantidad de palabras del título. En la revisión del código del laboratorio se identificó que la implementación de la función Reduce aplica para este reto ya que se persigue la misma meta, que consiste agrupar tuplas con la misma llave haciendo la sumatoria del valor.

Código fuente:

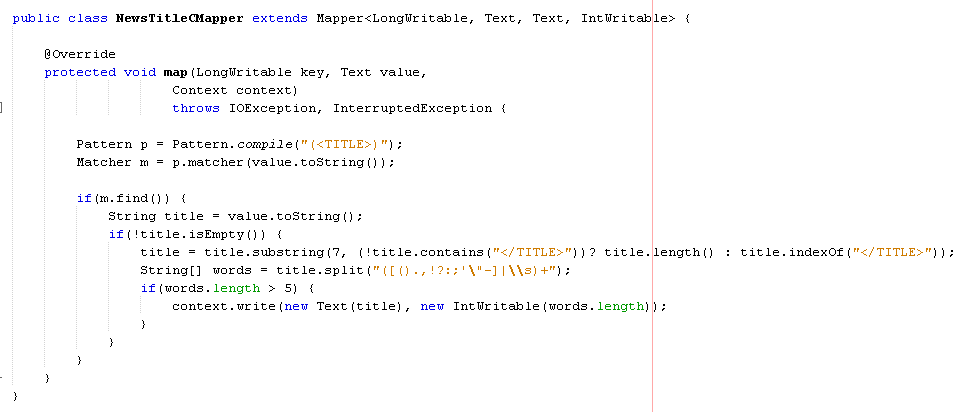


Figura 3. Implementación de Map del Reto #2.

Resultado de las pruebas:

Comando ejecutado:

*hadoop jar WordCountMR-0.0.1-SNAPSHOT.jar uniandes.reuters.job.*

*NewsTitleWCounter /user/bigdata01/fernandofs/reuters21578.tar.gz /user/bigdata01/fernandofs/news\_title\_wcounter*



Figura 4. Ejecución proceso del Reto #2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Prueba | Duración (segundos) | | | | | |
| **Ejec. 1** | **Ejec. 2** | **Ejec. 3** | **Ejec. 4** | **Ejec. 5** | **Promedio** |
| *Dataset* (1 lote) | 20.16 | 22.44 | 20.02 | 20.22 | 20.34 | 20.66 |
| *Dataset* (50 lotes) | 23.76 | 25.25 | 23.35 | 24.34 | 34.38 | 24.22 |

El tiempo promedio para las ejecuciones de un (1) lote es de 20.66 segundos, llegando a procesar aproximadamente 704’000 líneas de información textual, a razón de 34’076 líneas por segundo, lo cual es un buen indicador considerando que el manejo de archivos es una operación costosa en tiempo.

El tiempo promedio para las ejecuciones de cincuenta (50) lotes es de 24.22 segundos, llegando a procesar aproximadamente 35’200’000 líneas de información textual, a razón de 1’453’344 líneas por segundo.

Comparando los resultados de las dos (2) pruebas realizadas se puede concluir que la arquitectura tiene la capacidad de escalar ya que a pesar de que la cantidad de información a procesar se incrementó 50 veces la duración de la ejecución aumentó solo el **17.23 %**, en cambio el rendimiento aumento en el **4’265 %**.

# Reto #3

Alcance: Indique los lugares mencionados en las noticias y cuántas noticias hay para cada lugar.

Descripción de la solución: la revisión de los archivos de entrada permitió detectar que la estructura cumple con el formato XML, donde cada noticia está delimitada por un tag llamado “*REUTERS*” y contiene un tag llamado “*PLACES*”, por tanto se utilizó este criterio como expresión regular para identificar el lugar al que hace referencia cada noticia e incrementar el contador de noticias para dicho lugar. Durante la extracción de la información del lugar, se detectó que algunas noticias están asociadas a más de un lugar por tanto se debe afectar los contadores que sean necesarios. La tupla registrada en el contexto tiene como llave el nombre del lugar y como valor el número 1. En la revisión del código del laboratorio se identificó que la implementación de la función Reduce aplica para este reto ya que se persigue la misma meta, que consiste agrupar tuplas con la misma llave haciendo la sumatoria del valor.

Código fuente:



Figura 5. Implementación de Map del Reto #3.

Resultado de las pruebas:

Comando ejecutado:

*hadoop jar WordCountMR-0.0.1-SNAPSHOT.jar uniandes.reuters.job.*

*NewsPlaceCounter /user/bigdata01/fernandofs/reuters21578.tar.gz /user/bigdata01/fernandofs/news\_place\_counter*



Figura 6. Ejecución proceso del Reto #3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Prueba | Duración (segundos) | | | | | |
| **Ejec. 1** | **Ejec. 2** | **Ejec. 3** | **Ejec. 4** | **Ejec. 5** | **Promedio** |
| *Dataset* (1 lote) | 21.08 | 21.07 | 20.22 | 19.30 | 20.17 | 20.37 |
| *Dataset* (50 lotes) | 26.17 | 23.42 | 25.37 | 22.18 | 26.69 | 24.77 |

El tiempo promedio para las ejecuciones de un (1) lote es de 20.37 segundos, llegando a procesar aproximadamente 704’000 líneas de información textual, a razón de 34’561 líneas por segundo, lo cual es un buen indicador considerando que el manejo de archivos es una operación costosa en tiempo.

El tiempo promedio para las ejecuciones de cincuenta (50) lotes es de 24.77 segundos, llegando a procesar aproximadamente 35’200’000 líneas de información textual, a razón de 1’421’074 líneas por segundo.

Comparando los resultados de las dos (2) pruebas realizadas se puede concluir que la arquitectura tiene la capacidad de escalar ya que a pesar de que la cantidad de información a procesar se incrementó 50 veces la duración de la ejecución aumentó solo el **21.6 %**, en cambio el rendimiento aumento en el **4’112 %**.

# Reto #4

# Reto #5 (BONO)

# Conclusiones