ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN FIEC

Evaluación, análisis y comparación del rendimiento de programas de procesamiento masivo implementados usando lenguajes de programación Java, Python y C++ sobre la plataforma Hadoop para clústeres de varios tamaños

Presentado por:

- Mayra Alejandra Mendoza Saltos
- Betsy Tatiana Trujillo Miranda

Agenda

- Introducción
- Objetivos del Proyecto
- Hadoop: Plataforma de Procesamiento Masivo de Datos
  - Hadoop
  - Streaming
  - Pipes
- Ejecución de un trabajo MapReduce en Hadoop
- Diseño e Implementación
  - WordCounter
  - Bigramas
  - Escala de Gris
  - Hit Log FIEC
- Resultados y Análisis
- Conclusiones y Recomendaciones

Introducción

- Selección de un lenguaje de programación para desarrollo de una aplicación
  - Considerar: facilidad de uso, simplicidad, rendimiento y portabilidad
- Hadoop → herramienta popular para procesamiento masivo de datos
  - Varios APIs: Java nativo, Pipes, Streaming
    - Permiten trabajar con diferentes lenguajes
    - No existe evaluación que compare su rendimiento

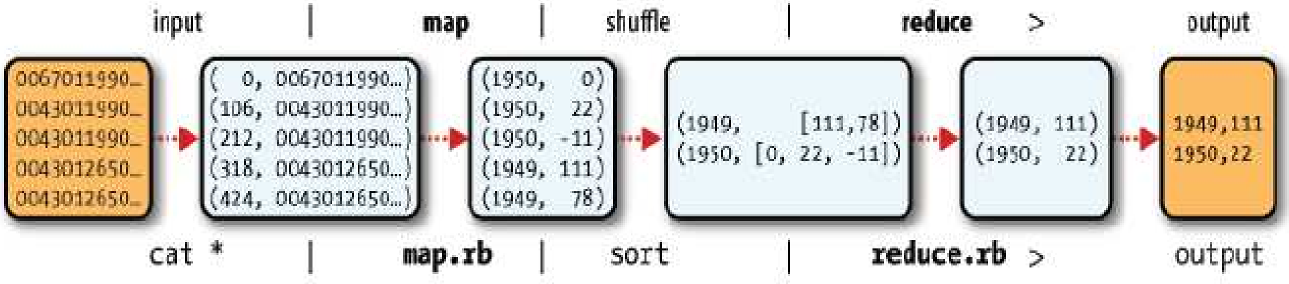
Objetivos

- **Objetivo General**
  - El objetivo general del presente trabajo es realizar una comparativa del rendimiento de programas desarrollados usando lenguajes de programación Java, C++ y Python sobre la plataforma Hadoop.

- **Objetivos Específicos**
  - Implementar los programas WordCounter, Bi-Gramas, Escalado de Grises de Imágenes y Hit Log FIEC-ESPOL en los lenguajes de programación Java, Python y C++.
  - Obtener tiempos de respuestas para cada aplicación ejecutada en 2, 4, 6, 10, 15 y 20 nodos.
  - Realizar evaluación y comparación de los resultados obtenidos.
  - Elaborar gráficas comparativas del tiempo de respuesta de cada aplicación.

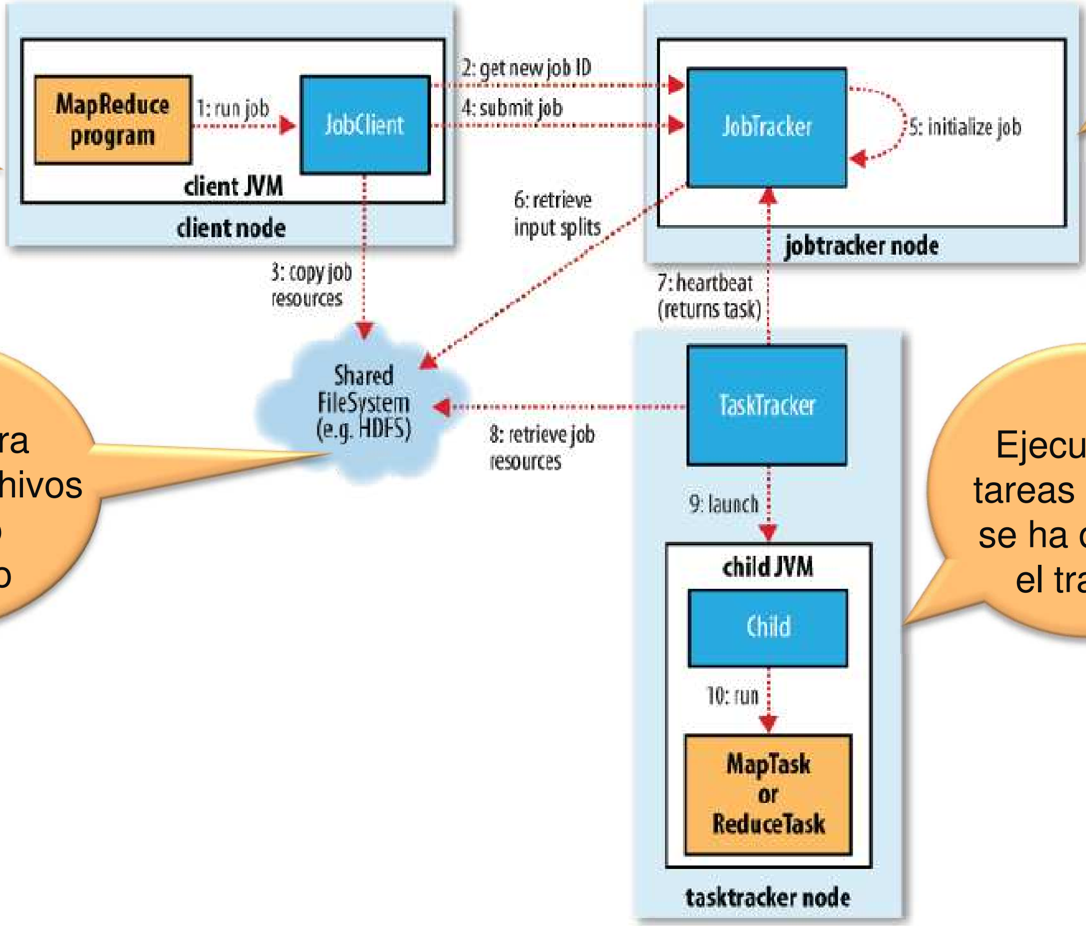
HADOOP: Plataforma de Procesamiento masivo de datos

- Procesamiento distribuido y masivo de datos
- Gran escalabilidad
- Modelo de programación Map/Reduce



- Código abierto
- Reduce complejidad en desarrollo de aplicaciones distribuidas

Ejecución de un trabajo Map/Reduce en Hadoop



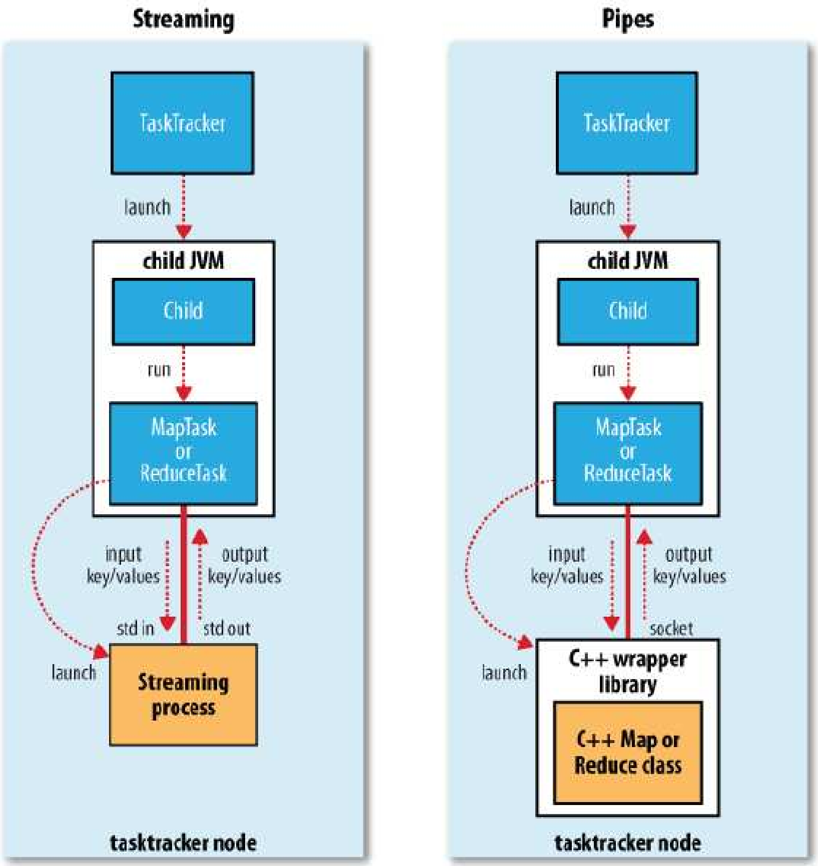
Cliente que envía el trabajo MapReduce.

Aplicación java que coordina la ejecución del trabajo

HDFS se utiliza para Almacenar archivos de trabajo compartido

Ejecuta las tareas en que se ha dividido el trabajo

Streaming y Pipes



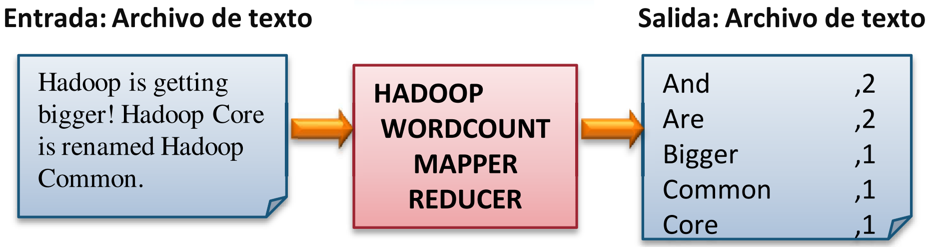
Streaming: Permite que programas mapper/reducer escritos en cualquier lenguaje (ejecutables y scripts) puedan ser ejecutados sobre Hadoop.

Pipes: Interfaz de C++ para Hadoop MapReduce.

PROBLEMAS A RESOLVER

RESULTADOS Y ANÁLISIS

WordCounter



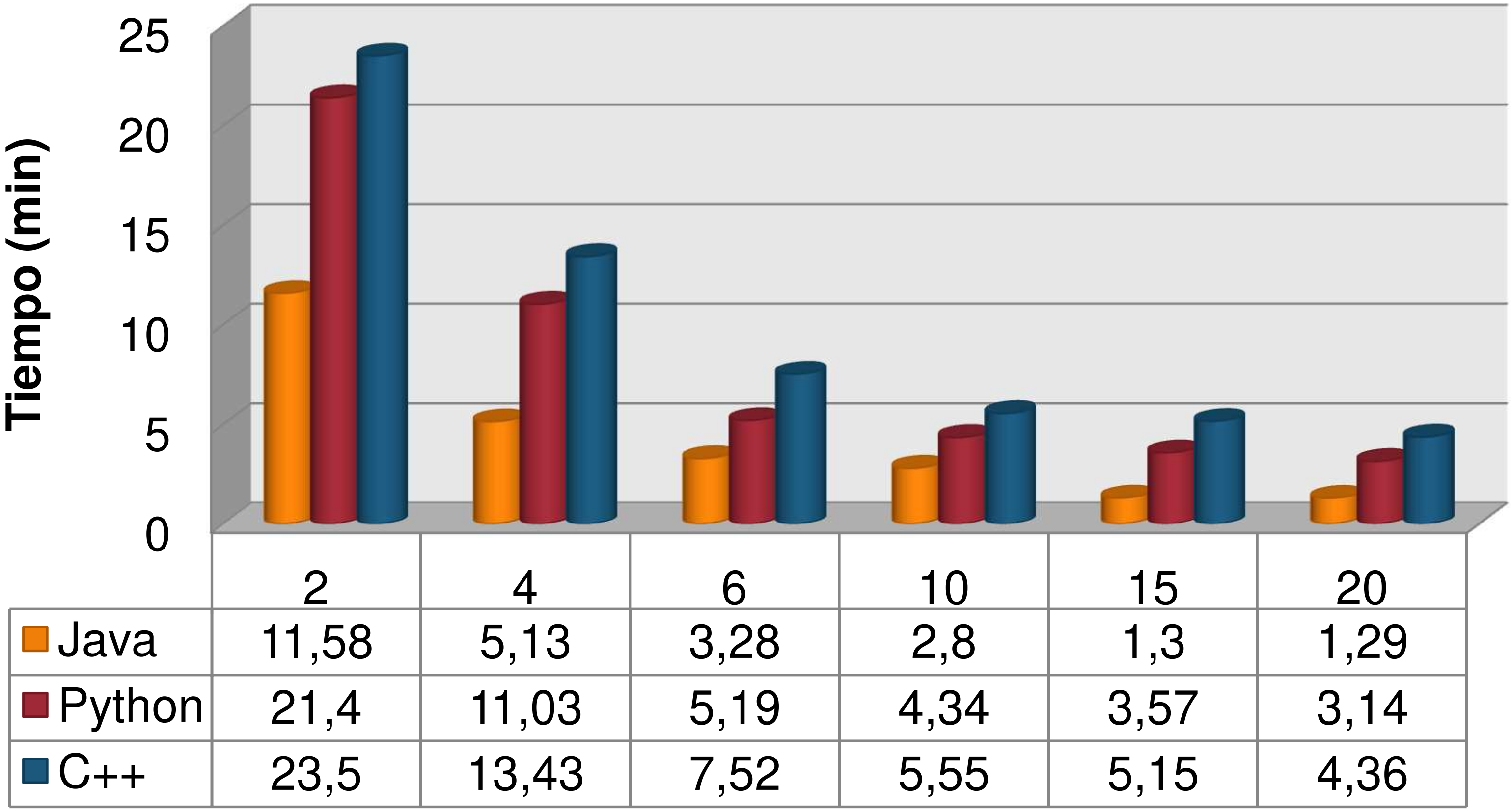
Entrada: Archivo de texto

Hadoop is getting bigger! Hadoop Core is renamed Hadoop Common.

HADOOP WORDCOUNT MAPPER REDUCER

Salida: Archivo de texto

And	,2
Are	,2
Bigger	,1
Common	,1
Core	,1



	2	4	6	10	15	20
🟧 Java	11,58	5,13	3,28	2,8	1,3	1,29
🟥 Python	21,4	11,03	5,19	4,34	3,57	3,14
🟦 C++	23,5	13,43	7,52	5,55	5,15	4,36

Bigramas

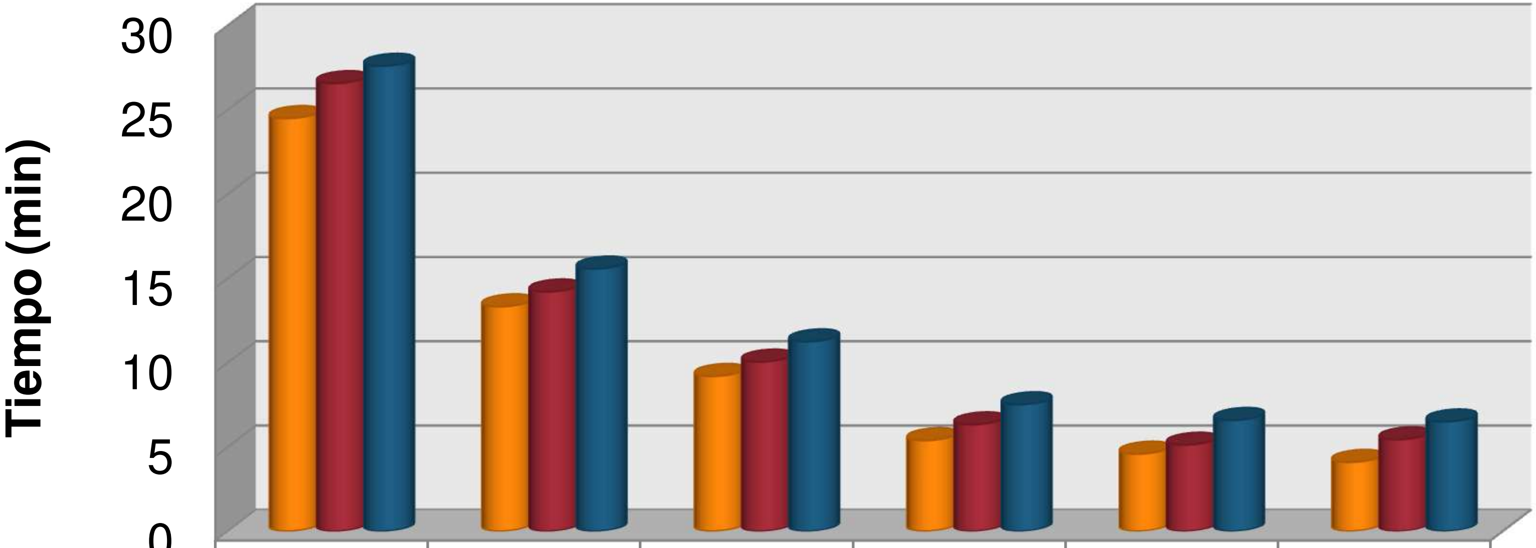
Entrada: Archivo de texto

Hadoop is getting bigger! Hadoop Core is renamed Hadoop Common.

HADOOP BIGRAMA MAPPER REDUCER

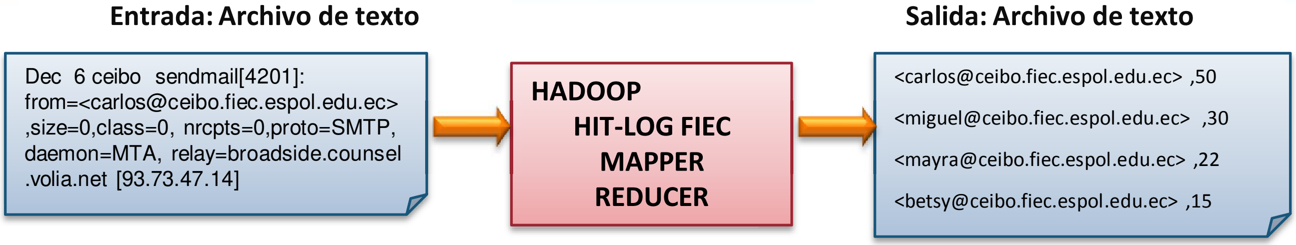
Salida: Archivo de texto

```
Hadoop::is        ,1
is::getting       ,1
getting::bigger   ,1
bigger::Hadoop ,1
Hadoop::Core    ,1
```



	2	4	6	10	15	20
🟧 Java	24,50	13,33	9,20	5,39	4,59	4,10
🟥 Python	26,57	14,21	10,05	6,33	5,12	5,44
🟦 C++	27,60	15,57	11,25	7,52	6,59	6,50

Hit-log FIEC



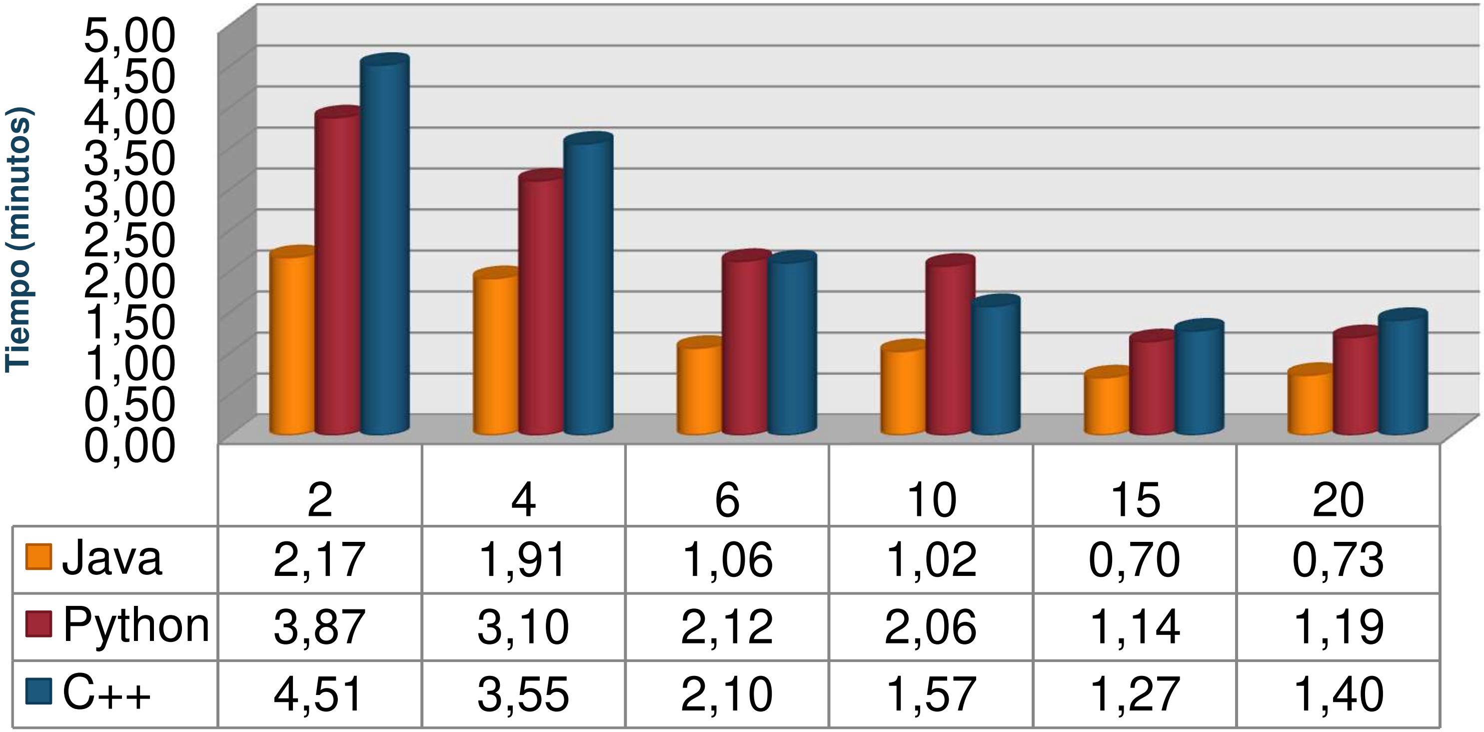
Entrada: Archivo de texto

Dec 6 ceibo sendmail[4201]:
from=<carlos@ceibo.fiec.espol.edu.ec>
,size=0,class=0, nrcpts=0,proto=SMTP,
daemon=MTA, relay=broadside.counsel
.volia.net [93.73.47.14]

HADOOP
HIT-LOG FIEC
MAPPER
REDUCER

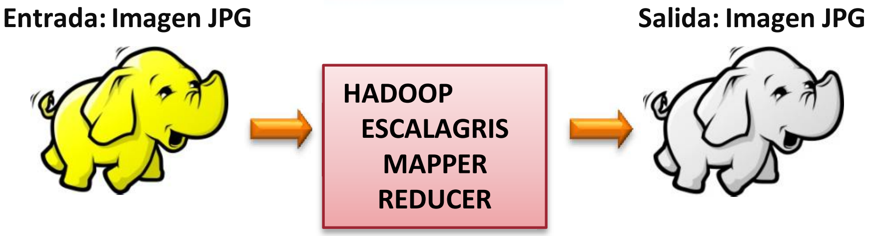
Salida: Archivo de texto

<carlos@ceibo.fiec.espol.edu.ec> ,50
<miguel@ceibo.fiec.espol.edu.ec> ,30
<mayra@ceibo.fiec.espol.edu.ec> ,22
<betsy@ceibo.fiec.espol.edu.ec> ,15



	2	4	6	10	15	20
🟧 Java	2,17	1,91	1,06	1,02	0,70	0,73
🟥 Python	3,87	3,10	2,12	2,06	1,14	1,19
🟦 C++	4,51	3,55	2,10	1,57	1,27	1,40

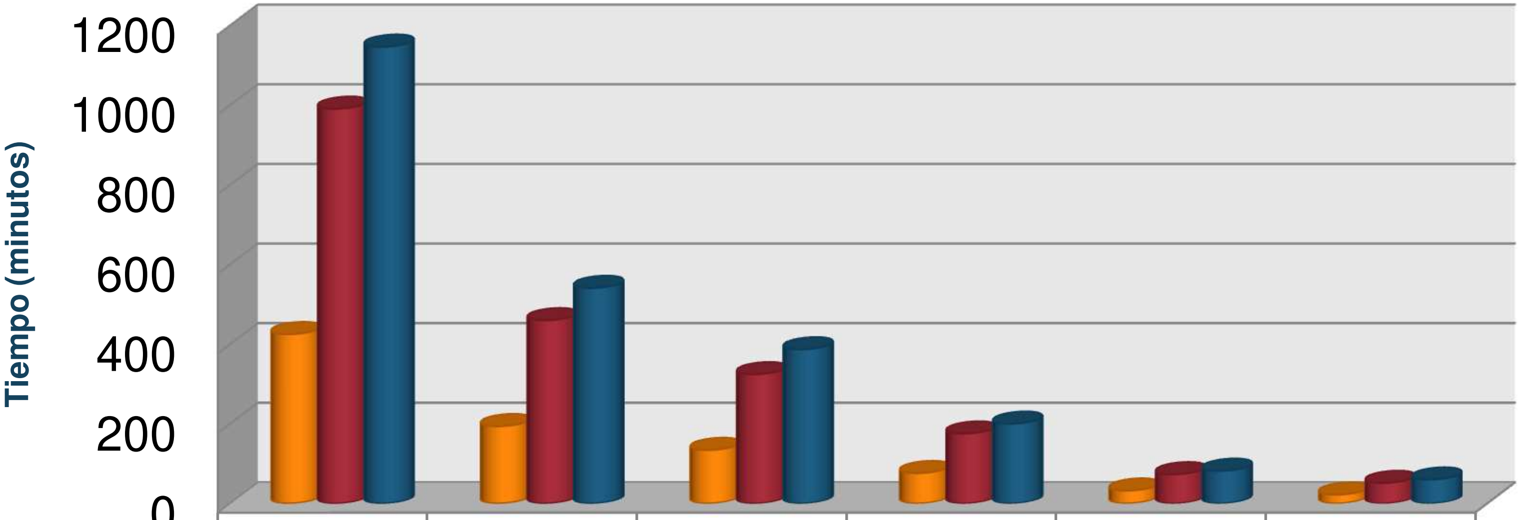
Escala de grises en imágenes



Entrada: Imagen JPG

HADOOP
ESCALAGRIS
MAPPER
REDUCER

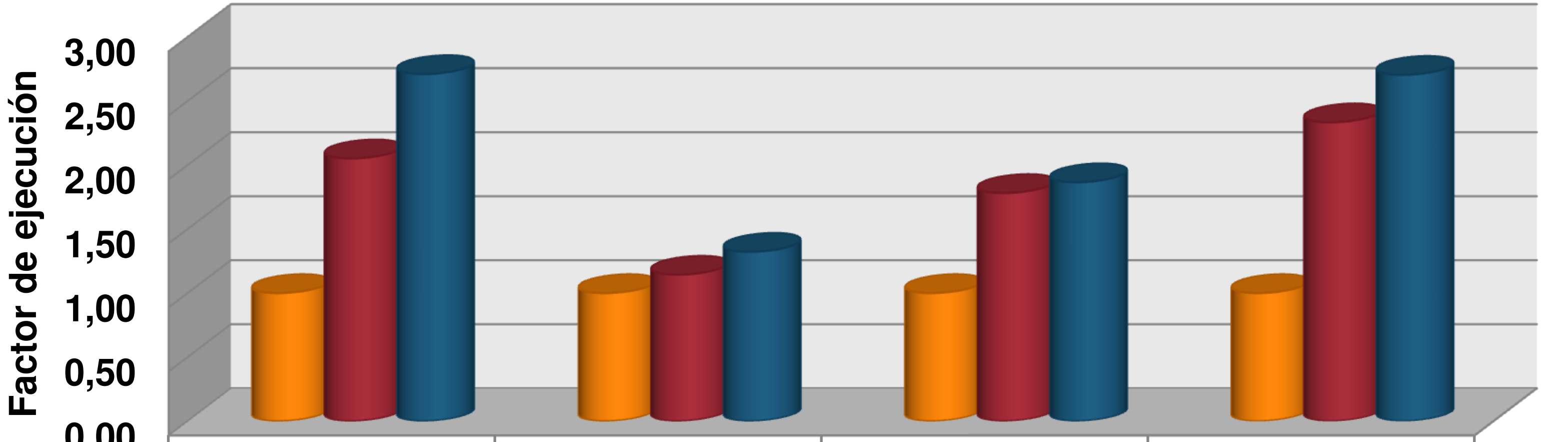
Salida: Imagen JPG



	2	4	6	10	15	20
🟧 Java	425,20	194,10	134,30	76,20	32,40	22,20
🟥 Python	991,20	460,08	324,42	175,59	73,29	51,48
🟦 C++	1147,00	541,20	387,00	200,20	82,00	60,00

Factor de ejecución Java vs Python vs C++

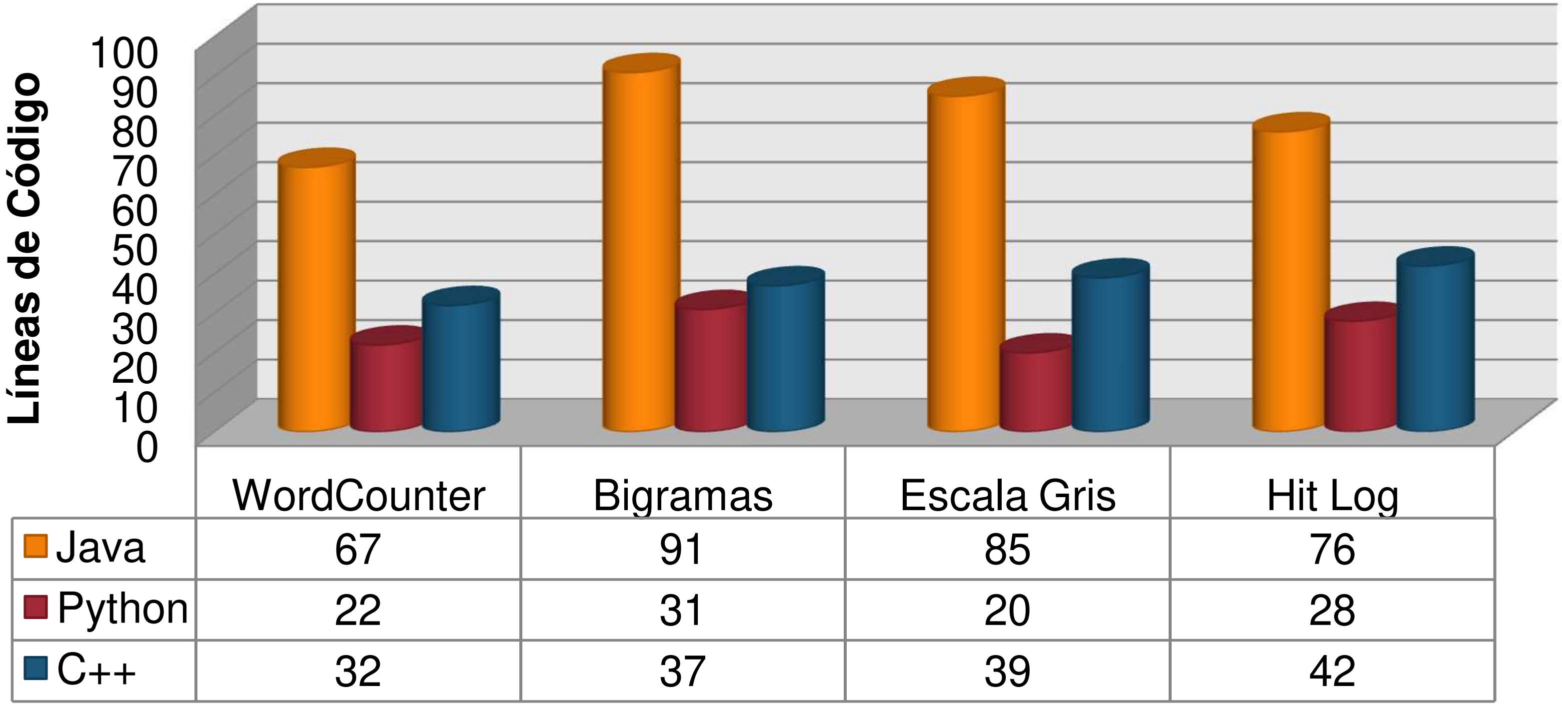
Streaming (Python)	**1,83 tiempo Java**
Pipes (c++)	2,15 tiempo Java



	WordCounter	Bigramas	Escala Gris	Hit Log
🟧 Java	1	1	1	1
🟥 Python	2,05	1,14	1,78	2,33
🟦 C++	2,71	1,32	1,87	2,70

Líneas de código vs Lenguaje de programación

Lenguaje	% Código
Python	18%
C++	26%
Java	56%



	WordCounter	Bigramas	Escala Gris	Hit Log
🟧 Java	67	91	85	76
🟥 Python	22	31	20	28
🟦 C++	32	37	39	42

Conclusiones

1. El API de programación de Hadoop de mejor rendimiento en tiempo de respuesta fue Java nativo seguido por Streaming (evaluado con Python) y finalmente Pipes (C++).
2. Debemos también considerar también las preferencias/conocimientos del desarrollador y el soporte disponible a fin de equilibrar rendimiento vs facilidad de desarrollo.
3. El procesamiento de texto fue realizado con el uso de cadenas, que básicamente consistió en separar el texto y analizarlo para cada lenguaje existen diferentes formas de hacerlo.
4. El procesamiento de archivos pequeños demora la tarea de ejecución debido a que Hadoop está desarrollado para ser más eficiente manipulando archivos de gran tamaño.

Recomendaciones

1. Hadoop es mucho más eficiente trabajando con archivos de gran tamaño. El SequenceFile permite unir muchos archivos pequeños en un solo archivo grande. Es recomendable hacer uso de este InputFormat cuando sea factible.
2. Si se trabajan con InputFormat propio, es recomendable empaquetarlos junto a los demás InputFormat provistos por Hadoop.
3. Al momento de realizar concatenación de cadenas de palabras es mejor utilizar los métodos más eficientes que brinde cada lenguaje para hacerlo (ej.: usar StringBuffer vs el operador “+” en Java).

Referencias

- Andrew Pavlo, Erick Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, Michael Stonebraker, “A Comparison of Approaches to Large-Scale Data Analysis”, Brown University, University of Wisconsin.
- Xiaoyang Yu, “Estimating Language Models Using Hadoop and Hbase”, University of Edinburgh, 2008.
- Apache Lucene, http://lucene.apache.org/, último acceso 17-Feb-2010.
- Dean, J. y Ghemawat, S. “MapReduce: Simplified Data Processing on Large Clusters”. En memorias del Sixth Symposium on Operating System Design and Implementation (OSDI 2004), San Francisco, CA-EE.UU. Diciembre, 2004.
- HDFS Architecture, http://hadoop.apache.org/common/docs/current/hdfs_design.html, último acceso 17-Feb-2010.
- Ghemawat, S., Gobioff, H., y Leung, S. “The Google File System”. En Memorias del 19th ACM Symposium on Operating Systems Principles. Lake George, NY-EE.UU., Octubre, 2003.
- Welcome to Apache Hadoop, http://hadoop.apache.org/core/ , último acceso 17-Feb-2010.
- Anatomy of a MapReduce job run with Hadoop, http://answers.oreilly.com/topic/459-anatomy-of-a-mapreduce-job-run-with-hadoop/, último acceso 20-Abr-2010.
- Hadoop The definiitive Guide, Tom White, publicado por O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Pag. 192.
- Java Advance Imaging (JAI) API, http://java.sun.com/javase/technologies/desktop/media/jai/ , último acceso 17-Feb-2010.
- Python Imaging Library (PIL), http://www.pythonware.com/products/pil/, último acceso 17-Feb-2010.
- Magick++ -- C++ API for ImageMagick, http://www.imagemagick.org/Magick++/, último acceso 17-Feb-2010.
- Sequencefile , http://hadoop.apache.org/common/docs/current/api/org/apache/hadoop/io/SequenceFile.html, último acceso 5-Mar-2010.
- Hadoop The definiitive Guide, Tom White, publicado por O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Pag. 184.
- Hadoop The definiitive Guide, Tom White, publicado por O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Pag. 20.
- http://IP de maquina virtual:50030 ← para monitorear el status de trabajos.
- http://IP de maquina virtual:50070 ← para ver el contenido del HDFS.
- Dumbo, http://www.audioscrobbler.net/development/dumbo/
- Python API to HDFS: libpyhdfs, http://code.google.com/p/libpyhdfs/
- C API to HDFS:libhdfs, http://hadoop.apache.org/common/docs/r0.20.1/libhdfs.html

¿Preguntas?

¡Gracias por su atención!