Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc445388848)

[1.1 Autores 2](#_Toc445388849)

[1.2 Planificación 2](#_Toc445388850)

[1.3 Entrega 2](#_Toc445388851)

[2. Descripción de las tecnologías 3](#_Toc445388852)

[2.1 Descripción de la tecnología 1 3](#_Toc445388853)

[2.2 Descripción de la tecnología 2 3](#_Toc445388854)

[3. Criterios de comparación 4](#_Toc445388855)

[3.1 Categoría A: Nombre 4](#_Toc445388856)

[3.1.1 Criterio A.1: Nombre 4](#_Toc445388857)

[3.1.2 Criterio A.2: Nombre 5](#_Toc445388858)

[3.1.n Criterio A.n: Nombre 5](#_Toc445388859)

[3.2 Categoría B: Nombre 5](#_Toc445388860)

[3.2.1 Criterio B.1: Nombre 5](#_Toc445388861)

[3.2.2 Criterio B.2: Nombre 5](#_Toc445388862)

[3.2.n Criterio B.n: Nombre 5](#_Toc445388863)

[3.3 Categoría Z: Nombre 5](#_Toc445388864)

[3.3.1 Criterio Z.1: Nombre 5](#_Toc445388865)

[3.3.2 Criterio Z.2: Nombre 5](#_Toc445388866)

[3.3.n Criterio Z.n: Nombre 5](#_Toc445388867)

[4. Evaluación de los criterios por tecnología 6](#_Toc445388868)

[4.1 Evaluación de los criterios para la tecnología 1 6](#_Toc445388869)

[4.2 Evaluación de los criterios para la tecnología 2 6](#_Toc445388870)

[5. Comparación de las tecnologías 7](#_Toc445388871)

[6. Recomendaciones 9](#_Toc445388872)

[6.1 Situación 1 9](#_Toc445388873)

[6.1.1 Descripción de la situación 9](#_Toc445388874)

[6.1.2 Recomendación de tecnología a utilizar 9](#_Toc445388875)

[6.2 Situación 2 9](#_Toc445388876)

[6.2.1 Descripción de la situación 9](#_Toc445388877)

[6.2.2 Recomendación de tecnología a utilizar 9](#_Toc445388878)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

En este apartado se debe indicar el número de grupo y los nombres de los autores, poniendo en primer lugar al coordinador del grupo.

## 1.2 Planificación

En este apartado se debe incluir un enlace (URL) compartido a la planificación del trabajo utilizando una herramienta online de diagramación Gantt (por ejemplo, GanttPro, versión gratuita).

Hay que tener en cuenta que cada participante del grupo debe tener asignadas tareas que sumen al menos 15 horas. El peso de este trabajo en la calificación total de la asignatura es de un 10%, por tanto requiere de una dedicación de 15 horas del total de 150 horas de la asignatura.

## 1.3 Entrega

En este apartado debe incluirse un enlace (URL) a un repositorio en GitHub o en BitBucket creado para el trabajo.

En dicho repositorio debe encontrarse, al menos los siguientes archivos en la rama máster:

* Trabajo terminado: del trabajo terminado con el nombre TG2\_final.docx
* Presentación del trabajo: TG2\_final.pptx

Dichos archivos serán los que se tendrán en cuenta para la calificación del trabajo.

# 2. Descripción de las tecnologías

En los siguientes apartados se debe describir brevemente cada tecnología a comparar.

Se pueden incluir imágenes copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada una se indique la fuente (al menos la URL).

## 2.1 Descripción de Hadoop

La tecnología Apache Hadoop es una de las soluciones pioneras en el ámbito de Big Data, nacida en el seno de las principales empresas tecnológicas de Internet (inspirada en Google MR y Google FS, extendida por Yahoo!). Es ya una solución madura y robusta, y es utilizada por organizaciones como eBay, LinkedIn, Facebook o Twitter.

Apache Hadoop es un framework que permite el tratamiento distribuido de grandes cantidades de datos y trabajar con miles de máquinas de forma distribuida. Es decir,

Las principales características de Hadoop son:

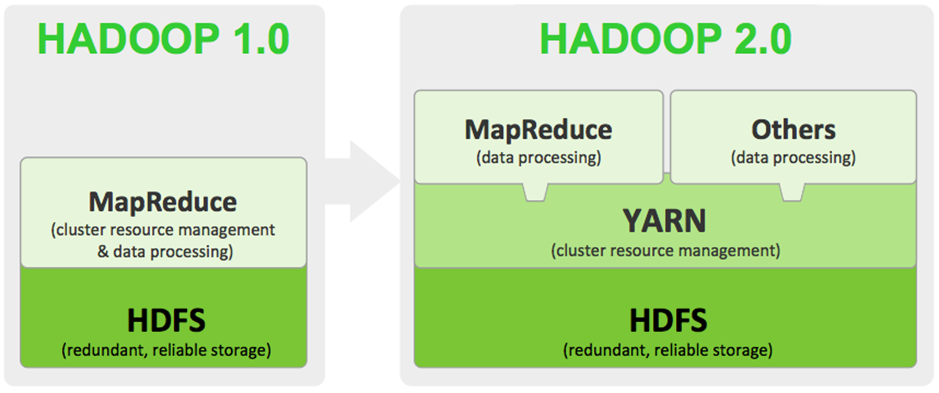
* Económico: Está diseñado para ejecutarse en equipos de bajo coste formando clústeres. Estos clústeres pueden llevarnos a pensar en miles de nodos de procesamiento disponibles para el procesado de información.
* Escalable: Si se necesita más poder de procesamiento o capacidad de almacenamiento solo hay que añadir más nodos al clúster de forma sencilla.
* Eficiente: Hadoop distribuye los datos y los procesa en paralelo en los nodos donde los datos se encuentran localizados.
* Confiable: Es capaz de mantener múltiples copias de los datos y automáticamente hacer un redespliegue de las tareas. El aspecto clave de Hadoop es que en lugar de mover los datos hacia donde se hace el procesamiento, Hadoop mueve el procesamiento a donde están los datos.
* Consta de 2 Componentes centrales: HDFS (almacenamiento de archivos) y MapReduce (infraestructura de programación)

## Apache Hadoop busca la consecución de los siguientes objetivos:

* Almacenamiento: Mediante un repositorio de almacenamiento independiente y autónomo.
* Performance: Aprovechando las capacidades de multiprocesamiento en paralelo.
* Robustez: Mediante mecanismos de fallo parcial y recuperación.
* Actualización: Utilizando las capacidades de migración de datos en tiempo real.
* Productividad: Implementando los procesos en lenguajes ampliamente extendidos
* Funcionalidad: Proporcionando algoritmos y técnicas de inferencia de conocimiento.

Tiene muchas herramientas que permiten interactuar con él:

* HDFS: Es el sistema de archivos como tal en el cual se almacenan los datos, está compuesto por el nodo maestro, que se encarga de almacenar los metadatos, y por una serie de nodos esclavos que se encargan de almacenar los datos.
* MapReduce: Es la herramienta propia de Hadoop para procesar los datos que están almacenados en HDFS, en este trabajo es bastante importante ya que Hadoop no va a competir con Spark en todos sus aspectos, sino principalmente en este (cuando expliquemos Spark veremos a que nos referimos).
* Flume: Sirve para mover grandes cantidades de logs a un respositorio central
* Hive: Sirve para organizar los datos de una manera que sean más accesibles para su procesamiento.
* Sqoop: Sirve para transferir datos desde bases de datos tradicionales (con datos estructurados) a HDFS, y viceversa.
* Y un amplio etcétera de herramientas que sirven para interactuar con HDFS.



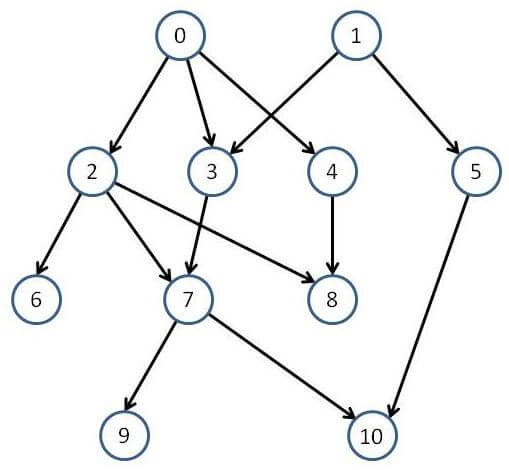
1http://www.dineshonjava.com

## 2.2 Descripción Spark

Spark es un framework orientado al procesamiento de grandes cantidades de datos. Es un proyecto liderado por Databricks e ideado para reducir los tiempos de ejecución de MapReduce mediante la carga de datos en memoria.

Spark mantiene la escalabilidad lineal y la tolerancia a fallos de MapReduce, pero amplía sus bondades gracias a varias funcionalidades: DAG y RDD.

DAG (Directed Acyclic Graph)



2 geekytheory.com

DAG (Grafo Acíclico Dirigido) es un grafo dirigido que no tiene ciclos, es decir, para cada nodo del grafo no hay un camino directo que comience y finalice en dicho nodo. Un vértice se conecta a otro, pero nunca a sí mismo**.**

En comparación con MapReduce, el cual crea un DAG con dos estados predefinidos (Map y Reduce), los grafos DAG creados por Spark pueden tener cualquier número de etapas. Spark con DAG es más rápido que MapReduce por el hecho de que no tiene que escribir en disco los resultados obtenidos en las etapas intermedias del grafo. MapReduce, sin embargo, debe escribir en disco los resultados entre las etapas Map y Reduce.

RDD (Resilient Distributed Dataset)

Apache Spark mejora con respecto a los demás sistemas en cuanto a la computación en memoria. Surge debido a que las herramientas existentes tienen problemas que hacen que se manejen los datos ineficientemente a la hora de ejecutar algoritmos iterativos y procesos de minería de datos. En ambos casos, mantener los datos en memoria puede mejorar el rendimiento considerablemente.

El almacenamiento de los datos en memoria caché hace que los algoritmos de machine learning ejecutados que realizan varias iteraciones sobre el conjunto de datos de entrenamiento sea más eficiente. Además, se pueden almacenar versiones transformadas de dichos datos.

Las características principales de Spark son:

* Open Source
* Integrado con YARN (Hadoop 2.0)
* Evita accesos a disco y permite compartir datos entre procesos. Mejores tiempos de respuesta en procesos batch.
* Posibilidad de trabajar en Streaming.
* Acceso de datos ‘online’.
* Ideal para tratamiento de grafos y minería de datos.

Para concluir y que las cosas queden claras vamos a hacer un pequeño resumen:

¿QUÉ ES SPARK?

* No es una versión modificada de Hadoop
* Sistema de almacenamiento In-Memory para consultas iterativas muy rápidas.
* De 10 a 40 veces más rápido que Hadoop.
* Sistema escalable y con tolerancia a los fallos.
* Integrado en procesos Batch e Interactivos.
* Compatible con la API del sistema de almacenamiento de Hadoop

# 3. Criterios de comparación

En los sub-apartados de este apartado se deben indicar cada uno de los criterios (también llamados factores, propiedades, características, indicadores, etc.), que se usarán en la comparación.

Los criterios deben organizarse en categorías (al menos 3 categorías). El número de criterios totales no puede ser inferior a 20.

Las categorías dependerán del tipo de tecnología, pueden ser categorías como “General”, “Utilidades”, “Rendimiento”, etc.

Los criterios a definir en cada categoría también dependerán del tipo de tecnologías a comparar. En el siguiente apartado hay algunos ejemplos.

## 3.1 Categoría A: Nombre

### 3.1.1 Criterio A.1: Nombre

Por cada criterio hay que indicar el nombre, una breve descripción, y el tipo de valor a asignar al criterio. El número de criterios no puede ser inferior a 20.

Por ejemplo, si se comparan dos herramientas CASE, un criterio podría ser el autor, entonces en este apartado se podría incluir este texto:

*3.1.1 Criterio A.1: Autor de la herramienta*

*Nombre del criterio: Autor.*

*Descripción: Nombre de la persona, institución o empresa que ha creado la herramienta.*

*Tipo de valor: Texto libre.*

Otro ejemplo de criterio con valor Si/No:

*3.1.1 Criterio A.1: Generación de código Java*

*Nombre del criterio: Generación código Java.*

*Descripción: Indica si la herramienta CASE incluye funcionalidad para generar archivos fuente .java a partir de diagramas de clase.*

*Tipo de valor: Booleano (Si/No).*

Otro ejemplo de criterio con valor numérico:

*3.1.1 Criterio A.1: Número de proyectos simultáneos*

*Nombre del criterio: Proyectos simultáneos.*

*Descripción: Indica el número máximo de proyectos de modelado que deja tener abiertos la herramienta de forma simultánea.*

*Tipo de valor: Numérico.*

### 3.1.2 Criterio A.2: Nombre

### 3.1.n Criterio A.n: Nombre

## 3.2 Categoría B: Nombre

### 3.2.1 Criterio B.1: Nombre

### 3.2.2 Criterio B.2: Nombre

### 3.2.n Criterio B.n: Nombre

## 3.3 Categoría Z: Nombre

### 3.3.1 Criterio Z.1: Nombre

### 3.3.2 Criterio Z.2: Nombre

### 3.3.n Criterio Z.n: Nombre

# 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para la tecnología 1

Debe incluir al menos una tabla con la siguiente estructura.

|  |  |
| --- | --- |
| CRITERIOS | EVALUACIÓN |
| Criterio A.1: Nombre |  |
| Criterio A.2: Nombre |  |
| Criterio A.n: Nombre |  |
| … |  |
| Criterio B.1: Nombre |  |
| Criterio B.2: Nombre |  |
| … |  |

Y algunos comentarios aclaratorios sobre aquellos criterios cuyo valor indicado en la tabla no sea suficiente para entenderlo.

La tabla anterior es obligatoria y deben completarla los autores del trabajo, aunque se pueden incluir otros gráficos o tablas complementarias copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada uno se indique la fuente (al menos la URL).

## 4.2 Evaluación de los criterios para la tecnología 2

# 5. Comparación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

Esta tabla anterior es obligatoria y deben completarla los autores del trabajo, aunque se pueden incluir otros gráficos o tablas complementarias copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada uno se indique la fuente (al menos la URL).

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERIOS | TECNOLOGÍA A | TECNOLOGÍA B | COMENTARIOS |
| A.1 |  |  |  |
| A.2 |  |  |  |
| … |  |  |  |
| B.1 |  |  |  |
| B.2 |  |  |  |
| … |  |  |  |

# 6. Recomendaciones

Deben platearse posibles situaciones de uso, y recomendar justificadamente una u otra tecnología en función de la situación. Al menos 2 situaciones diferentes.

## 6.1 Situación 1

### 6.1.1 Descripción de la situación

*Una posible situación en el caso de comparar dos herramientas CASE, podría ser el caso de una empresa de desarrollo muy interesada en tecnologías open source, que programa sólo en Java, con equipos de desarrollo pequeños, que utiliza UML como notación, etc, etc*…

Nos encontramos ante el caso de que el mayor banco de inversión chino, el ZHUZHU, decide hacer uso del Big Data para posicionarse por delante se sus rivales más directos.

Han decidido que cogiendo los valores de la bolsa (cada minuto) a través del Machine Learning serán capaces de predecir con mucha exactitud el siguiente valor de bolsa diferente. Es decir predecir el futuro.

### 6.1.2 Recomendación de tecnología a utilizar

Debe indicarse la tecnología propuesta para esa situación.

Debe incluirse una tabla como la siguiente, mostrando las ventajas, respecto a los criterios, que ofrece cada tecnología en esa situación concreta.

Incluir sólo los criterios sobre los que se aprecien ventajas de una de las tecnologías frente a otra. No incluir criterios que no sean relevantes para la decisión (por ejemplo, el criterio “autor” seguramente no será relevante).

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Hadoop | Spark |
| Madurez | Hadoop es un sistema mucho mas maduro | Aunque en la situación actual es un sistema que ha madurado mucho, no llega alos niveles de Hadoop |
| Online | No | Acceso de datos en streaming |
| Adecuado para machine learning | Los algoritmos de Machine Learning proporcionan capacidades que no son fáciles de explotar en Hadoop Map Reduce | Tiene una biblioteca (MLIB) de Machine Learning |
| Sistema de almacenamiento | Hadoop posee el sistema de almacenamiento distribuido HDFS | No |
| Rapidez | MAPREDUCE tiene que mover datos dentro/ fuera de disco | Sistema de almacenamiento In-Memory para consultas iterativas muy rápidas.  De 10 a 40 veces más rápido que Hadoop. |

Aquí ante la posibilidad de una gran inversión por parte del banco para la implementación y poder contar con expertos en la tecnología Spark y Hadoop, la solución propuesta es ejecutar la herramienta Spark en la parte “superior” de Hadoop. No obstante, cabe recordar que Spark no tiene un sistema de almacenamiento independiente y Spark debe cargar los datos desde algún lugar para procesarlos. Por lo tanto al integrarlo con Hadoop, el banco Zhuzhu puede aprovechar las ventajas de ambas tecnologías.

## 6.2 Situación 2

### 6.2.1 Descripción de la situación

Una empresa de paquetería decide realizar un estudio de todas las direcciones de envío donde van repartiendo para realizar una optimización de los recorridos que realizan los transportistas. Con ello reducirán el tiempo de entrega, algo vital en esta industria.

### 6.2.2 Recomendación de tecnología a utilizar

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Hadoop | Spark |
| Madurez | Hadoop es un sistema mucho mas maduro | Aunque en la situación actual es un sistema que ha madurado mucho, no llega alos niveles de Hadoop |
| Online | No | Acceso de datos en streaming |
| Sistema de almacenamiento | Hadoop posee el sistema de almacenamiento distribuido HDFS | No |
| Rapidez | MAPREDUCE tiene que mover datos dentro/ fuera de disco | Sistema de almacenamiento In-Memory para consultas iterativas muy rápidas.  De 10 a 40 veces más rápido que Hadoop. |

Como en este caso no depende de tanta velocidad de procesamiento, los accesos a los datos no tienen por qué ser en streaming, el procesamiento no debe ser en tiempo real.

La solución adoptada es Spark.

---------------------------

(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)