

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VI
Índice de Tablas	VII
Índice de Algoritmos	VIII
Resumen	IX
1. Introducción	10
1.1. Descripción de la Propuesta	11
1.1.1. Contexto del Proyecto	11
1.1.2. Definición del Problema	12
1.1.3. Trabajo Relacionado	12
1.1.4. Propuesta de Solución	13
1.2. Objetivos	13
1.3. Alcances	14
1.4. Metodología	14
1.5. Plan de trabajo	15
2. Marco Teórico	17
2.1. Grafos	18
2.1.1. Representación de Grafos	18
2.2. Administración de Grafos de Datos	20
2.2.1. Modelos de Base de Datos de Grafos	20
2.2.2. Procesamiento de Grafos	20
2.3. Sistemas de Procesamiento de Grafos	22
2.3.1. Hadoop	22

2.3.2.	YARN	22
2.3.3.	Stratosphere	22
2.3.4.	Apache Giraph	23
2.3.5.	GraphLab	24
2.3.6.	Neo4J	24
2.3.7.	Map Reduce vs Giraph	24
3.	Apache Giraph	26
3.1.	Orígenes	27
3.1.1.	Contribución de Facebook	27
3.2.	¿Porqué Giraph?	28
3.2.1.	Procesamiento de Algoritmos	29
3.2.2.	Vértices por Segundo (VPS)	30
3.2.3.	Aristas por Segundo (EPS)	31
3.3.	El Modelo de Programación de Giraph	32
3.3.1.	Complejidad de Computación Paralela y Distribuida	32
3.4.	Modelo de Datos de Giraph	33
3.4.1.	Modelo Tradicional vs Vertex Centric	35
4.	Diseño de Algoritmos en Apache Giraph	38
4.1.	Template de Estandarización de Algoritmos	39
4.2.	Ejemplos de aplicación del Template	39
4.2.1.	MaxValue	39
4.2.2.	TriangleClosing	40
4.2.3.	InOutDegree	42
4.3.	Algoritmos ya Diseñados	43
4.4.	Algoritmos a diseñar	43
4.4.1.	Kruskal	44
4.4.2.	Prim	45
4.4.3.	Borůvka modificado	46
5.	Implementación de Algoritmos en Apache Giraph	49
5.1.	Aspectos Generales	50
5.2.	Borůvka Modificado	51
5.2.1.	Clase Borůvka	53

5.2.2.	Clase BoruvkaTextWritable	56
5.2.3.	Clase BoruvkaNodeComputation	56
5.2.4.	Clase BoruvkaMasterCompute	56
5.2.5.	Enum BoruvkaMessages	57
5.2.6.	Enum TipoIteracion	57
5.2.7.	Clase BoruvkaVertexInputFormat	58
5.2.8.	Clase BoruvkaVertexOutputFormat	58
5.2.9.	Clase TextTextNullIntVertexReader	58
5.2.10.	Clase TextTextNullIntVertexWriter	58
6.	Pruebas	59
6.1.	Elementos a probar	60
6.2.	Metodología de pruebas	61
6.3.	Toma de Pruebas	61
6.3.1.	Entradas	61
6.3.2.	Configuración de ambiente	63
6.4.	Resultados	66
7.	Conclusión	69
7.1.	Conclusiones del Diseño de Algoritmos	70
7.1.1.	Dificultades del Modelo Centrado en Vértices	70
7.2.	Conclusiones de implementación de algoritmos	71
7.3.	Conclusiones del Algoritmo	72
7.4.	Trabajos futuros	75
	Glosario	76
	Bibliografía	78
	Anexos	
	chapterA: Implementación de MST81appendix.Alph1	
A.1.	Funciones implementadas	81
A.1.1.	Métodos relacionados con aristas	81

ÍNDICE DE FIGURAS

	página
1.1. Red Social con Grafo [11]	12
2.1. Representación de grafo no dirigido	18
2.2. Representación de grafo dirigido.	19
2.3. Modelo de Computación BSP	21
2.4. Arquitectura de Apache Giraph	23
3.1. Comparación de tiempos de ejecución de BFS [9]	29
3.2. Comparación de tiempos de ejecución de BFS [9].	30
3.3. Comparación de tiempos de ejecución de BFS [9]	31
3.4. Organización conceptual de Giraph	33
3.5. Ejecución maxValue en modelo Tradicional	35
3.6. Ejecución maxValue en Modelo Centrado en Vértices	36
3.7. Función compute para maxValue	37
4.1. Ejemplo de ejecución de Kruskal	44
5.1. Diagrama de Estado de Borůvka	51
5.2. Diagrama de Clases Borůvka	52
6.1. Grafo de prueba.	62
6.2. Grafo de prueba resuelto.	66
6.3. Tiempo de trabajo para distintos workers.	68
6.4. Tiempo de trabajo por cantidad de mappers.	68

ÍNDICE DE TABLAS

	página
1.1. Comparación entre Giraph y Hive [3]	13
1.2. Feriados dentro del período de trabajo	15

ÍNDICE DE ALGORITMOS

1.	AristaMenorPeso	81
2.	ComunicarNumerodeVotos	81
3.	NodosEnComponente	81
4.	NodoEnComponente	81
5.	ActualizarComponentes	82