

## **CERTIFICATE**

---

This is to certify that project entitled “**Improving Join Query Processing in MapReduce Environment**” has been completed by **Anwar Dilawar Shaikh, Roll No. 03/CTA/2K10** in the partial fulfillment of the requirement for the award of degree of **Master of Technology in Computer Technology & Application**.

This work is carried out by him under my supervision and support. This is a beneficial work in the field of large scale distributed computing.

**Dr. Rajni Jindal**

**Associate Professor & Project Guide**

(Dept. of Computer Engineering)

**Delhi Technological University**

Bawana Road, Delhi- 110042.

## ACKNOWLEDGEMENT

---

It is distinct pleasure to express my deep sense of gratitude and indebtedness to my learned supervisor **Dr. Rajni Jindal** (Associate Professor of Computer Engineering Dept., Delhi Technological University) for her precious guidance, encouragement and patient reviews. Her continuous inspiration only has made me complete this dissertation. All of them kept on boosting me time and again for putting an extra ounce of effort to realize this work.

I am also thankful to **Prof. (Dr.) Daya Gupta**, Head of the Department, Computer Engineering, Delhi Technological University, Delhi for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of the hour and provided with all the help and facilities, which I required for the completion of my thesis work.

Finally I am also thankful to my classmates and my family members for their unconditional support and motivation during this work.

**Anwar Dilawar Shaikh**

Master of Technology

(Computer Technology & Application)

Dept. of Computer Engineering

Delhi Technological University

Bawana Road, Delhi -110042

## Table of contents

Abstract	1
Organization of the Thesis	2
1. Introduction to MapReduce	3
1.1 General concepts	3
1.2 MapReduce	3
1.3 Hadoop	5
1.4 Hadoop Distributed File System	6
2. Join Query Processing Algorithms in MapReduce	8
2.1 Repartition Join	8
2.2 Broadcast Join	10
2.3 Optimized Broadcast Join	12
2.4 Trojan Join	14
2.5 Replicated Join	16
3. Comparison of Join Algorithms	18
4. Optimizations	20
4.1 Related Work	20
4.2 Proposed Optimization Techniques	20
4.2.1 Dynamic Hash Table Creation	20
4.2.2 Zip Broadcast Join	24
4.2.3 Hash Broadcast Join	26
5. Experiments	29
5.1 AWS Cloud	29
5.1.1 Elastic Compute Cloud	29
5.1.2 Simple Storage Service	30
5.1.3 Elastic MapReduce	30
5.2 Experimental Setup	30
5.3 Dataset	32
5.4 Results	33
5.4.1 Comparison of Default Join and Broadcast Join	33
5.4.2 Comparison of Broadcast Join and Semijoin	34
5.4.3 Comparison of Normal Broadcast Join & Dynamic hash table Broadcast	35
5.4.4 Comparison of Broadcast join and Zip Broadcast Join	37
5.4.5 Comparison of Default join and Hash Broadcast Join	38
6. Proposed Join Algorithm Selection Strategy	39
Conclusion	41
Future Work	42
References	43
Appendix A – Dataset Generation Script	44
Appendix B – Code	47
Appendix C – Steps to create MapReduce job	58

## Figure Index

Figure No.	Figure Caption	Page No.
1.	MapReduce framework	5
2.	Hadoop Distributed File System Architecture	7
3.	Repartition Join	8
4.	Broadcast Join	10
5.	Example of Semijoin	13
6.	Co-partitioning data of relations R1 and R2	14
7.	Trojan Join execution	15
8.	Replicated join	16
9.	Hash table creation in Normal Broadcast join	21
10.	Hash table creation based on the input size	21
11.	Zip Broadcast Join	24
12.	Hash Broadcast Join	28
13.	Experimental Setup	30
14.	Elastic MapReduce Job	31
15.	Comparison graph of Default Join and Broadcast Join processing time	33
16.	Comparison graph of Semi Join and Broadcast Join processing time	34
17.	Comparison graph of Broadcast Join & Dynamic Hash table processing time	35
18.	Comparison graph of Broadcast Join and Zip Broadcast join processing time	37
19.	Comparison graph of Broadcast Join and Hash Broadcast join processing time	38
20.	Join Selection Strategy	39

## Table Index

Table No.	Table Caption	Page No.
1.	Distribution of Tuples to Reduce processes	16
2.	Comparison of Join processing methods	18
3.	Result of Comparison of Default Join and Broadcast Join	33
4.	Result of Comparison of Broadcast Join and Semi Join	34
5.	Result of Comparison of Broadcast Join and Dynamic Hash table broadcast	35
6.	Result of Comparison of Broadcast Join and Zip Broadcast join	37
7.	Result of Comparison of Default join and Hash Broadcast join	38