

**CHENNAI – PONDICHERRY**

**Efficient Skew Handling for Outer Joins in a Cloud Computing Environment**

**Abstract:**

Outer joins are ubiquitous in many workloads and Big Data systems. The question of how to best execute outer joins in large parallel systems is particularly challenging, as real world datasets are characterized by data skew leading to performance issues. Although skew handling techniques have been extensively studied for inner joins, there is little published work solving the corresponding problem for parallel outer joins, especially in the extremely popular Cloud computing environment. Conventional approaches to the problem such as ones based on hash redistribution often lead to load balancing problems while duplication-based approaches incur significant overhead in terms of network communication. In this paper, we propose a new approach for efficient skew handling in outer joins over a Cloud computing environment. We present an efficient implementation of our approach over the Spark framework. We evaluate the performance of our approach on a 192-core system with large test datasets in excess of 100GB and with varying skew. Experimental results show that our approach is scalable and, at least of in cases of high skew, significantly faster than the state-of-the-art.

**Existing System:**

With data applications growing in scale, Cloud environments play a key role in application scale-out, exploiting parallelisation to speed up operation and extending the amount of memory available. In this light, efficient parallelisation of joins on shared-nothing systems is becoming increasingly desirable. Various distributed join algorithms have been studied however, there has been relatively little done on the topic of outer joins. In fact, outer joins are common in complex queries and widely used such as in data analytics applications. For example, in the Semantic Web domain, queries containing outer joins account for as much as 50% of the total number of queries, based on the analysis of DBPedia query logs. Moreover, in online e-commerce, customer ids are often left outer joined with a large transaction table for analyzing purchase patterns.

**Proposed System:**

We propose an efficient query-based approach, aiming at efficiently against data skew in massively parallel outer joins over shared-nothing systems. We implement our method over the Spark framework and conduct a performance evaluation ove r a data stored in HDFS [18] on an experimental configuration consisting of 16 nodes (192 cores) and datasets of up to 106GB with a range of values for skew. We summarize the contributions.

We present a new algorithm called query-based outer joins for directly and efficiently handling skew in parallel outer joins.

\_ We analyze the performance of two state-of-art techniques in currently DBMSs: (1) PRPD, for skew handling in inner joins; and DER, for optimizing inner join implementation of small-large table outer joins. We find that the composition of these methods (referred to as PRPD+DER) can potentially handle skew in large-large table outer joins. Our experimental results confirm this expectation. \_ We present the detailed implementation of our design over the Spark platform and our experimental evaluation shows that show that our algorithm outperforms PRPD+DER in the presence of high skew. Moreover, the results also demonstrate that our method is scalable, results in less network communication and presents good load balancing under varying skew.