

**CHENNAI – PONDICHERRY**

**Data Transfer Scheduling for Maximizing Throughput of Big-Data Computing in Cloud Systems**

**Abstract:**

Many big-data computing applications have been deployed in cloud platforms. These applications normally demand concurrent data transfers among computing nodes for parallel processing. It is important to find the best transfer scheduling leading to the least data retrieval time – the maximum throughput in other words. However, the existing methods cannot achieve this, because they ignore link bandwidths and the diversity of data replicas and paths. In this paper, we aim to develop a max-throughput data transfer scheduling to minimize the data retrieval time of applications. Specifically, the problem is formulated into mixed integer programming, and an approximation algorithm is proposed, with its approximation ratio analyzed. The extensive simulations demonstrate that our algorithm can obtain near optimal solutions.

**Existing System:**

The existing method to retrieve data, which is used in current HDFS systems and DCN, cannot obtain the least data retrieval time. In the existing method, when a non-local data is required, a request is sent to any one of the closest replicas. Then, the data is transferred from the selected node through any one of the shortest paths, determined by routing protocols like Equal-Cost Multipath Routing (ECMP) or per-flow Valiant Load Balancing (VLB). It is noted that many tasks are retrieving data concurrently. This method may result in heavy congestions on some links, leading to long data retrieval time, because it ignores link bandwidths and the overlaps of selected nodes and paths. Some researchers proposed flow scheduling systems to avoid path collisions. However, they only exploited path diversity but not data replica diversity.

**Proposed System:**

The data retrieval time (i.e., to maximize the throughput) of an application consisting of concurrent tasks, we propose a max-throughput data transfer scheduling, utilizing both replica and path diversities. In our method, the problem is formulated into mixed integer programming, and an approximation algorithm is proposed, with its approximation ratio analyzed. We also solve the data retrieval problem for the case of multiple applications. Our simulations demonstrate that the approximation results are almost as good as the optimal solutions. We also show that the availability of a small number of additional data replicas can be greatly beneficial in many cases, regardless of path diversity.