

**CHENNAI – PONDICHERRY**

**IMPROVING THE EFFICIENCY OF MAPREDUCE SCHEDULING**

**ALGORITHM IN HADOOP**

**Abstract:**

In a distributed computing environment, to support the processing of large data sets a free Java-based programming framework Hadoop plays a vital role. In Hadoop, MapReduce technique is used for processing and generating large datasets is used with a parallel distributed algorithm on a cluster. The benefit of using MapReduce is to automatically handle failures and hides the complexity of fault tolerance from the user. The Scheduling algorithm of FIFO(FIRST IN FIRST OUT) is used in Hadoop as default in which the jobs are executed in the order of their arrival. This method suits well for homogeneous cloud and results in poor performance on the heterogeneous cloud. Later the LATE (Longest Approximate Time to End) algorithm has been developed which reduces the FIFO's response time by a factor of 2. It gives better performance in heterogeneous environments. The three principles of LATE algorithms are i) prioritizing tasks to speculate ii) selecting fast nodes to run on iii) capping speculative tasks to prevent thrashing. It takes action on appropriate slow tasks and it could not compute the remaining time for tasks correctly and can't find the real slow tasks. Finally, an SAMR (Self-Adaptive MapReduce) scheduling algorithm is being introduced which can find the slow tasks dynamically by using the historical information recorded on each node to tune parameters. SAMR reduces the execution time by 25% when compared to FIFO and 14% when compared to LATE.