Chapter 7
Conclusion & Future Work

This research work aims to optimize Big Data processing time on a heterogeneous distributed environment. Hadoop MapReduce framework is the most popular framework for Big Data processing. To improve Hadoop performance, HDFS is an imperative component. HDFS not only allows to store the big data but also provides handy support to distribute and process the huge amount of data. Default HDFS block placement policy requires a major improvement in terms of placing of data replicas and processing. This work contributes to the block rearrangement algorithm along with the proposed "Saksham" model for performance optimization of Big Data processing.

We studied default block placement policy and found a bottleneck in handling the heterogeneous environment and managing load balancing issue. Default HDFS block placement policy does not consider the cluster heterogeneity and processing capability of nodes while placing data blocks. MapReduce job tries to place the process, where data blocks are stored but it might be possible that the few nodes which are having data blocks may not have processing capability or may be slow in nature. These straggler nodes affect the overall performance of Hadoop. Therefore, it is required to shift the process, where processing capability is available and in that case, it may be required to move data blocks too where the process is put for execution. This would affect the overall performance in Hadoop, which is a matter of concern, particularly while processing big data.

The proposed approach, Saksham: block rearrangement policy leverages the processing capacity of CPU, during block placement. This approach helps MapReduce to minimize the internode and inter-rack transfer. The research work demonstrates that data blocks of a specific file can be placed on the specified nodes. This approach will not affect the overall load balancing of a cluster as other remaining files will not be affected.

Considering all cluster nodes' processing capability along with node labelling and scheduling is a key feature in "Saksham" policy. It is proved that based on nodes processing capability, replica rearrangement on desired cluster nodes achieves better data locality and turns into better job execution time.

We conducted two case studies to evaluate the performance of proposed "Saksham" model. We implemented all four scheduling policy in "Saksham" model and compared the results with two default approaches used primarily for Hadoop. Our results show that with the use of the proposed scheme and Fair-Fair scheduler achieves better data locality, equal load balancing, and low job completion time.

Thus, "Saksham" block rearrangement policy successfully achieves better load balancing compared to default policy and "Saksham" policy combined with node labelling and scheduling yields a greater performance over Hadoop default performance. We achieved more than 90% data locality for all test cases and also total time is reduced to near about 30% which is tremendous.

Furthermore, as a future enhancement, it is proposed to use "Saksham" policy for direct process scheduling, without using Hadoop Scheduler for actually scheduling of the job. However, that would require a lot of modification in the existing API. It is also possible to think of implementing assigning priorities to jobs dynamically, based on processing capability or hardware specifications.