IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

A Speculative Study on Hadoop Scheduling Algorithms

Vanika[1*], Aman Kumar Sharma[2]

[1] Department of Computer Science, Himachal Pradesh University, Shimla, India
[2]Department of Computer Science, Himachal Pradesh University, Shimla, India

[*]*Corresponding Author: vanika.vani92@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 11/Jun/2018, Published: 30/Jun/2018

***Abstract*—** Big Data is a term which mainly focuses on the use of techniques to capture, process, analyze and visualize large datasets in a reasonable time span. Different platforms, tools and software used for this purpose are known as "Big Data technologies". Hadoop is an open-source framework used to process large amount of data in an inexpensive and efficient way by using MapReduce which is used for processing and generating large data sets with a parallel, distributed algorithm on a cluster. Job scheduling is a key factor for achieving high performance in big data processing. The paper presents a comparative study of job scheduling algorithms in Hadoop environment. In addition, this paper describes the features, advantages and disadvantages of various Hadoop scheduling algorithms such as FIFO, Fair, Capacity, LATE, Energy-aware, Resource-aware, Matchmaking, Delay and Deadline Constraints.

***Keywords*—**Big Data, Hadoop, Mapreduce, Distributed Systems, Hadoop Scheduling Algorithms.

I. INTRODUCTION

Big Data

Big Data is defined as the voluminous amount of data that grow so fast that they are not manageable by traditional RDBMS (Relational Database Management System) or conventional statistical tools. Big Data is a term which mainly focuses on the use of techniques to capture, process, analyse and visualize large datasets in a reasonable time span. Big Data is when the size of the data itself becomes part of the problem [1]. This problem is now converted to an opportunity to use data stored in data warehouse to find some patterns in hidden data. Data collected by the real-time applications along with social media applications is in massive amount known as Big Data [2]. The new aspect of Big Data lies in the cost of storing and processing the large datasets. A Big Data analysis system must support input from multiple human experts and shared exploration of results [3]. The elements of Big data consists of 7 V's: value, volume, velocity, variety, veracity, visualization and variability.

Hadoop

Hadoop is one of the technologies used to process Big Data. It is an open-source platform that provides analytical technologies and computational power required to work with such large sets of data. Hadoop is an Apache Software Foundation project written in JAVA. The Hadoop core project provides the basic services for building a cloud computing environment with commodity hardware and the API for developing software that will run on that cloud [4]. The Apache Hadoop project was created by Doug Cutting and Mike Caferella in 2005. The name for the project came from the toy elephant of Cutting's young son [5]. Hadoop platform provides an improved programming model, which is used to create and run distributed systems quickly and efficiently. Hadoop distributes the data in advance. Data is replicated in a cluster of computers for reliability and availability. Processing occurs when the data is stored. A Hadoop cluster consists of single MasterNode and multiple SlaveNodes as shown in Fig. 1. The master node consists of a NameNode and a JobTracker. The slave node or worker node acts as both a DataNode and a TaskTracker. It performs the Heartbeat mechanism as each DataNode sends a "Heartbeat signal" to NameNode after every few minutes or default time set to make NameNode aware of the active/inactive status of DataNodes [6]. The processing in Hadoop is performed by MapReduce.

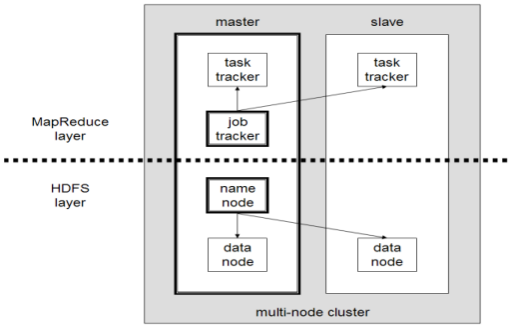

Fig. 1 Hadoop Cluster Architecture

MapReduce

MapReduce [7] is a programming model for processing large datasets. Hadoop MapReduce job consists of two user-defined functions: map and reduce [8]. The input of a Hadoop MapReduce job is given as a set of key-value pairs (k,v) and the map function is called for each of these pairs. The map function produces intermediate key-value pairs (k',v') shown in Fig. 2. Then, the Hadoop MapReduce framework groups these intermediate key-value pairs by intermediate key k' and calls the reduce function for each group. Then, the reduce function produces zero or more aggregated results. The Hadoop MapReduce uses a distributed file system to read and write its data. It uses Hadoop Distributed File System (HDFS), which is the open source counterpart of the Google File System [9].

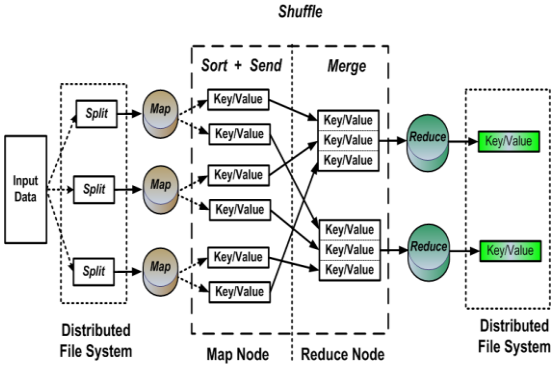

Fig. 2 Execution Overview of MapReduce

In this paper, Section I contains the introduction of big data, Hadoop and MapReduce, Section II contains the Hadoop scheduling algorithms, Section III contains comparison and analysis of different scheduling algorithms on the basis of parameters, Section IV concludes research work with future directions.

II. HADOOP SCHEDULING ALGORITHMS

Scheduling Algorithms

Scheduling is a management technique, which is used to govern the order of events. Scheduling can be influenced by priority, throughput and turnaround time requirements. The aim of scheduling of jobs [10] is to enable faster processing of jobs and to reduce the response time as much as possible by using better techniques for scheduling depending on the jobs, along with the best utilization of resources. The focus is on the scheduling of MapReduce jobs within Hadoop. The scheduler governs how the idle system accepts and executes jobs. In addition, it handles jobs which are submitted to the cluster, while a job execution is taking place. The goal of the scheduler is to optimize certain parameters. The different types of parameters related to Hadoop schedulers are:

1. Job Allocation
2. Fairness
3. Data Locality
4. Job Response Time
5. Resource Sharing
6. Environment
7. Workload
8. Resource Utilization
9. Mode
10. Performance
11. Execution
12. Priority in a job queue

Classification of Hadoop Scheduling Algorithms

The Hadoop job schedulers [11], [12] can be classified in terms of the following aspects: environment, priority, resource awareness (such as CPU time, free slot, disk space, I/O utilization), time and strategies. The main idea behind scheduling is to minimize overhead, resources and completion time and to maximize throughput by allocating jobs to the processor [13]. A number of Hadoop schedulers exist which are categorically segregated into various categories such as: Static, Dynamic, Resource based and Time based. Further these categories are subcategorised with number of algorithms as listed in Fig. 3.

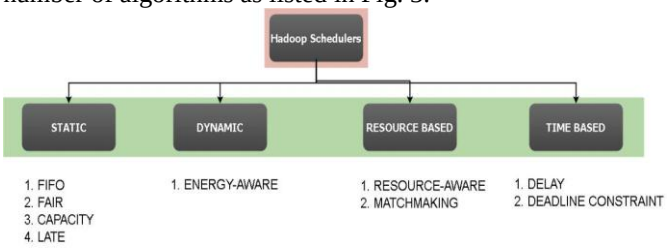

Fig. 3 Hadoop Schedulers

A. Static Scheduling - the allocation of jobs to processors is done before the program execution begins. The information regarding job execution time and processing of resources is known at compile time. The aim of static scheduling is to minimize the overall execution time of current programs. Different types of static scheduling algorithms are:

1. First In First Out (FIFO) Scheduler

FIFO is the default Hadoop scheduler. The job submitted first is given preference over jobs submitted later. Whenever a job arrives the JobTracker pulls the oldest job first from the job queue and processes it without considering the priority or size of the job. This scheduler is mostly used when the execution order of job is not important.

1.1 Advantages

1. FIFO scheduling technique is the simplest and most efficient among all the schedulers [14].
2. The jobs are executed in the same order in which they are submitted.
3. It is only suited for single type of job.

1.2 Disadvantages
1. Poor response time for short jobs in comparison to large jobs.
2. Reduces data locality and starvation of jobs.
3. Low performance when run multiple types of jobs.

2. Fair Scheduler

Fair scheduler is developed by Facebook. Fair scheduler groups jobs into named pools based on different attributes. It lets short jobs complete within a reasonable time while not starving long jobs [15]. The objective of Fair scheduling algorithm is to provide an equal distribution of resources among the users/jobs in the system [16].

2.1 Advantages
1. This scheduler makes a fair and dynamic resource reallocation.
2. It provides faster response to small jobs than large jobs.

2.2 Disadvantages
1. It ignores the node of the balance states and it will result in imbalance.
2. This scheduler does not consider the weight of each job, which leads to unbalanced performance in each pool/node.

3. Capacity Scheduler

Capacity scheduler is developed by Yahoo. It is developed for multiple organizations sharing a large cluster. In this scheduler, several queues are created instead of pools, each with defined map and reduce slots.

3.1 Advantages
1. It also supports the features of hierarchical queues, elasticity and operability.
2. The capacity scheduler has the potential to reuse unused jobs in the queue.

3.2 Disadvantages
1. Capacity scheduler is complex. There is difficulty in choosing proper queues.
2. With regard to pending jobs, it has some limitations in ensuring stability and fairness of the cluster from a queue and single user.

4. Longest Approximate Time to End (LATE) Scheduler

Speculative tasks are defined as those tasks that progress very slowly. This may happen due to load on the CPU, slow background process, contention for resources etc. LATE scheduler tries to locate a slow running task to start another equivalent task as a backup which is termed as speculative execution of task. The main objective of LATE scheduler [9] is to optimize the performance of jobs and to minimize job response time.

4.1 Advantages
1. Optimizes performance of jobs and minimizes job response time as much as possible.
2. This scheduler technique is highly robust in terms of heterogeneity [17].
3. It minimizes latency.

4.2 Disadvantages
1. LATE Scheduler does not ensure reliability.
2. This method does not break the synchronization phase between the map and reduce phases, but only takes action on appropriate slow tasks.

B. Dynamic Scheduling **-** allocation of jobs to the processors is done during execution time. A little basic knowledge is known about the resource needs of a job. It is also unknown in what type of environment the job will execute during its lifetime. The decision is made when a job begins its execution in the dynamic environment of the system.

1. Energy-Aware Scheduler

Energy-aware scheduler minimizes usage of energy during execution of MapReduce jobs. The map and reduce tasks while making scheduling decisions, the data centers can utilize their resources efficiently and reduce the energy consumption. It provides optimization of energy. Berkeley Energy Efficient MapReduce (BEEMR) represents a new design point that combines batching, zoning and data placement with new analysis-driven insights to create an efficient MapReduce system that saves energy while meeting these design requirements. The main focus is to develop a framework for systematically considering various MapReduce node power down strategies and their impact on the energy consumption and workload response time [18]. There are two techniques:

- The first technique is **Covering Subset (CS)**, which keeps only a small fraction of the nodes powered up during periods of low utilization.
- The second technique is **All-In Strategy (AIS)**, which uses all the nodes in the cluster to run a workload and then powers down the entire cluster.

1.1 Advantages
1. AIS strategy is suited only for production jobs.
2. BEEMR improves energy efficiency up to 40-50%.
3. BEEMR achieves minimal latency overhead for interactive jobs and some overhead for other job types.

1.2 Disadvantages
1. AIS is not well suited for time-sensitive interactive jobs.

2. The gap between ideal and BEEMR energy savings increases with cluster size which has high energy costs.

C. Resource Based Scheduling **–** the scheduling is on the basis of resource requirements of a job. This scheduling is for improving the resource utilization and job performance. The resources can be CPU time, disk storage, memory etc.

1. Resource-Aware Scheduler

Resource-aware scheduler focuses on resource utilization such as such as CPU utilization, I/O utilization, Disk utilization and Network Utilization when different kinds of workload run on the cluster. Resource-aware JobTracker scheduling mechanisms [19] are proposed, which make use of the resource metrics:

- **Dynamic Free Slot Advertisement -** In this, instead of having a fixed number of available computation slots configured on each TaskTracker node, the mechanism computes this number dynamically using the resource metrics obtained from each node.
- **Free Slot Priorities/Filtering -** In this, fixed maximum numbers of compute slots per node are retained. The ordering of free TaskTracker slots is done for advertising, according to the resource availability.

1.1 Advantages

1. Improves the performance of job management.
2. This scheduler also has better resource utilization in a cluster [20].

1.2 Disadvantages

1. This scheduler does not provide support for the pre-emption of reduce tasks.
2. Unable to confine the different resource utilization of each job in a multiuser environment.

2. Matchmaking Scheduler

The goal is to provide every slave node a fair chance to grab local tasks before any non-local tasks are assigned to them. Local task is defined as the task which can be executed on the node where its data is present. The matchmaking scheduler is related to the locality of the data [21], this scheduler works by finding matching between input data and the slave node which contain this data and then uses the locality marker as a guarantee that this node will take its local tasks. The data will be presented on that node before assignment of this task to other non-local nodes.

2.1 Advantages

1. Achieves highest data locality.
2. This scheduler also provides high cluster utilization.

2.2 Disadvantages

1. It works only in the homogeneous environment.
2. It has very low response time in case of finding matched data quickly.

D. Time Based Scheduling **-** in time constraint scheduling, the scheduling of jobs is done on the basis of a deadline, i.e. whether the job can complete its execution within the specified time or not. It includes delay scheduler and deadline constraint scheduler.

1. Delay Scheduler

Delay scheduling is handled by Facebook event logs saved in their Hadoop data warehouse. It uses the waiting approach for enhancing the locality. The goal is to wait for the right time if the data for the task is not present on the local node [24]. It tries to achieve fairness with locality. It also relaxes strict job order for task assignment. This scheduling [22] is introduced by applying changes to MapReduce with data locality to achieve better performance and lowest response time for the Map task.

1.1 Advantages

1. Improvement in performance of heterogeneous Hadoop system.
2. Throughput is increased.

1.2 Disadvantages

1. Delay scheduling technique is not effective when a majority of the tasks is much more than an average job [25].
2. There are limited slots per node.

2. Deadline Constraint Scheduler

Deadline constraint scheduler schedules jobs based on the deadline constraints mentioned by users [23]. This type of algorithms ensures that the jobs whose deadlines can be met are scheduled for execution. Dealing with deadline requirement, data processing is done by:

- **A job execution cost model -** it considers various parameters like map and reduce tasks runtimes, input data sizes, data distribution, etc.
- **A Constraint-Based Hadoop Scheduler** - it takes user deadlines as part of its input.

2.1 Advantages

1. Focuses on the optimization of Hadoop implementation [20].
2. Increases system utilization.

2.2 Disadvantages

1. There is a restriction that the nodes should be uniform in nature, which incurs cost.
2. There are some restrictions or issues of deadline, which are specified by the user for each job.

III. COMPARISON AND ANALYSIS

Comparison of scheduling algorithms is performed to analyse the working of different Hadoop schedulers. Task Scheduling is a factor that directly affects the overall performance of Hadoop platform and utilization of system resources. There are various algorithms designed to resolve this issue with different techniques and approaches. A number of them improves data locality and some provides synchronization processing. Also, numerous of them were designed to minimize the total completion time. While some other schedulers allocates capacity fairly among users and jobs. Also, some provide resource utilization like CPU utilization, I/O utilization, System utilization etc. Table-1 shows the comparison of different algorithms along with parameters.

Table-1 Comparison of Hadoop Schedulers

PARAMETERS/ ALGORITHMS	JOB RESPONSE TIME	DATA LOCALITY	JOB ALLOCATION	RESOURCE SHARING	ENVIRONMENT (HOMOGENEOUS/ HETEROGENOUS)	WORKLOAD	MODE (PREEMPTIVE/ NON-PREEMPTIVE)	PERFORMANCE	FAIRNESS	EXECUTION (SERIAL/ PARALLEL)	RESOURCE UTILIZATION	PRIORITY IN JOB QUEUE
FIFO	LOW	LOW	STATIC	NO	HOMOGENEOUS	YES	NON-PREEMPTIVE	HIGH (SMALL CLUSTER)	NO	SERIAL	LOW	NO
FAIR	HIGH	LOW	STATIC	YES	HOMOGENEOUS	YES	PREEMPTIVE	HIGH	YES	PARALLEL	HIGH	YES
CAPACITY	HIGH	LOW	STATIC	YES	HOMOGENEOUS	YES	NON-PREEMPTIVE	HIGH (LARGE CLUSTER)	YES	PARALLEL	HIGH	NO
LATE	HIGH	HIGH	STATIC	YES	BOTH	YES	PREEMPTIVE	LOW	YES	SERIAL	HIGH	YES
ENERGY-AWARE	HIGH	HIGH	DYNAMIC	YES	BOTH	YES	NON-PREEMPTIVE	HIGH	YES	PARALLEL	LOW	YES
RESOURCE-AWARE	HIGH	HIGH	RESOURCE BASED	YES	BOTH	YES	PREEMPTIVE	HIGH	YES	SERIAL	HIGH	YES
MATCHMAKING	LOW	HIGH	RESOURCE BASED	YES	HOMOGENEOUS	YES	NON-PREEMPTIVE	HIGH	YES	PARALLEL	HIGH	YES
DELAY	HIGH	HIGH	TIME BASED	NO	HOMOGENEOUS	YES	PREEMPTIVE	HIGH (SMALL CLUSTER)	YES	PARALLEL	HIGH	NO
DEADLINE CONSTRAINT	HIGH	LOW	TIME BASED	YES	BOTH	YES	NON-PREEMPTIVE	HIGH	YES	PARALLEL	HIGH	YES

The analysis of the above stated Hadoop scheduling algorithms shows that Deadline Constraint scheduler is the most optimal scheduling algorithm among all other scheduling algorithms. Deadline constraint algorithm has high resource utilization, executing jobs in parallel among all nodes in a non-pre-emptive scheduling mode. This results each task getting fair share of all the available resources in both homogeneous and heterogeneous cluster environment. Considering Matchmaking scheduler with low job response time contradicts the Deadline constraint scheduler with only drawback that it works for only homogeneous cluster environment.

IV. CONCLUSIONS AND FUTURE SCOPE

Big Data is a term used to depict the voluminous amount of data that grow large so fast that they are not manageable by traditional RDBMS (Relational Database Management System) or conventional statistical tools. Hadoop is an open-source java based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. The core of Hadoop consists of a storage part known as HDFS and a processing part known as MapReduce. The analysis of Hadoop scheduling algorithms shows that deadline constraint scheduler is the most optimal scheduling algorithm among all other scheduling algorithms.

It has high resource utilization, executing jobs in parallel among all nodes in a non-pre-emptive scheduling mode. This algorithm results each task getting fair share of all the available resources in both homogeneous and heterogeneous cluster environment. Considering Matchmaking scheduler with low job response time contradicts the Deadline constraint scheduler with only drawback that it works for only homogeneous cluster environment.

The future work may include Deadline Constraint and Matchmaking scheduling algorithms validation using tool and experiments may be constructed for the same for a particular environment.

REFERENCES

[1] O'Reilly Media, "*Big Data Now*", O'Reilly media, Inc., 2011.

[2] Rakesh. S. Srisath, Vaibhav A. Desale, Amol. D. Potgantwar, "*Big Data Analytical Architecture for Real-Time Applications*", IJSRNSC, Vol. 5, Issue 4, 2017.

[3] K. Parimala, G. Rajkumar, A. Ruba, S. Vijaylashmi, "*Challenges and Opportunities with Big Data*", International Journal of Scientific Research in Computer Science and Engineering, Vol. 5, Issue 5, pp. 16-20, 2017.

[4] Jason Venner, "*Pro hadoop*", Apress, ISBN 978-1-4302-1943-9, 2009.

[5] Jared Dean, "*Big data, data mining, and machine learning: value creation for business leaders and practitioners*", John Wiley & Sons, 2014.

[6] DT Editorial Services, "*Big Data*", Black Book, Dreamtech Press, ISBN 978-93-5119-931-1, 2016.

[7] Dean, Jeffrey, Sanjay Ghemawat, "*MapReduce: simplified data processing on large clusters*", Communications of the ACM, Vol. 51, Issue 1, pp. 107-113, 2008.
[8] Jens Dittrich, Jorge-Arnulfo and Quiane-Ruiz, "*Efficient big data processing in Hadoop MapReduce*", Proceedings of the VLDB Endowment, Vol. 5, No. 12, pp. 419-429, 2012.
[9] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung, "*The Google file system*", ACM SIGOPS operating systems review, Vol. 37, Issue 5, pp. 20-43, 2003.
[10] J.V. Gautam, H.B. Prajapati, V.K. Dabhi, S. Chaudhary, "*A survey on job scheduling algorithms in big data processing*", IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT'15), Coimbatore, pp. 1–11, 2015.
[11] Jiong Xie, FanJun Meng, HaiLong Wang, HongFang Pan, JinHong Ceng, Xiao Qin, "*Research on scheduling scheme for hadoop clusters*", Procedia Comput. Sci., Vol. 18, pp. 468-471, 2013.
[12] S. Suresh, N.P. Gopalan, "*An optimal task selection scheme for hadoop scheduling*", IERI Procedia, Vol. 10, pp. 70-75, 2014.
[13] Lisia S. Dias, Marianthi.G. Ierapetritou, "*Integration of scheduling and control under uncertainties: review and challenges*", Vol. 116, pp. 98-113, 2016.
[14] S. Divya, R. Kanya Rajesh, Rini Mary Nithila I, Vinothini M, "*Big data analysis and its scheduling policy - Hadoop*", IOSR J. Computer Engineering (IOSR-JCE), Vol. 17, Issue 1, pp. 36-40, 2015.
[15] B.P Andrews, A. Binu, "*Survey on Job Schedulers in Hadoop Cluster*", IOSR Journal of Computer Engineering, pp. 46-50, 2013.
[16] M. Pastorelli, A. Barbuzzi, D. Carra, M. Dell'Amico and P. Michiardi, "*Practical size based scheduling for MapReduce workloads*", IEEE International Conference on Big Data, pp. 51-59, 2013.
[17] J.C. Anjos, I. Carrera, W. Kolberg, A.L. Tibola, L.B. Arantes, C.R. Geyer, MRA, "*Scheduling and data placement on MapReduce for heterogeneous environments*", Future Generation Computer System, Vol. 42, pp. 22–35, 2015.
[18] Willis Lang, Jignesh M. Patel, "*Energy Management for MapReduce Clusters*", Department of Computer Sciences, University of Wisconsin Madison, USA, 2010.
[19] M. Yong, Shiwali Mohan, Nitin Garegrat, "*Towards a resource aware scheduler in hadoop",* ICWS, pp. 102-109, 2009.
[20] Dazhao Cheng, Jia Rao, Changjun Jiang, Xiaobo Zhou, "*Resource and deadline aware job scheduling in dynamic hadoop clusters*", IEEE 29th International Parallel and Distributed Processing Symposium, ISBN 978-1-4799-8649-1, 2015.
[21] C. He, Ying Lu, David Swanson, "*Matchmaking: A new MapReduce scheduling technique*", IEEE 3rd International Conference, Athens, Greece, 2011.
[22] Qiaomin Xie, Mayank Pundir, Yi Lu, Cristina L. Abad, Roy H. Campbell, Pandas, "*Robust locality-aware scheduling with stochastic delay optimality*", IEEE/ACM Trans. Netw., Vol. 25, Issue 2, pp. 662-675, 2016.
[23] K . Kc, K . Anyanwu, "*Scheduling hadoop jobs to meet deadlines*", IEEE 2nd Int. Conf. IEEE, Indianapolis, USA, pp. 388-392, 2010.
[24] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy Katz, Ion Stoica, "*Improving MapReduce performance in heterogeneous environments*", 8th USENIX conference on Operating systems design and implementation, pp. 29-42, 2008.
[25] Mohd Usama, Mengchen Liu, Min Chen, "*Job schedulers for Big data processing in Hadoop environment: testing real-life schedulers using benchmark programs*", Digital Communications and Networks, Vol. 3, Issue 4, pp. 260–273, 2017.

Authors Profile

Ms. Vanika pursed Diploma in Computer Science and Engineering from Himachal Pradesh Takiniki Shiksha Board of Dharamshala, India in 2011 and Bachelor of Technology in Computer Science and Engineering from Himachal Pradesh Technical University of Hamirpur, India in 2014. She is currently pursuing Master of Technology in Computer Science from Himachal Pradesh University of Shimla, India. Her main research work focuses on Big Data Analytics, Hadoop, MapReduce.

Dr Aman Kumar Sharma is a Doctorate in Computer Science from Himachal Pradesh University of Shimla, India with specialization in Software Engineering. His list of publications at international level has exceeded 100. He has represented India on the International Dais at United Arab Emirates, Dubai. He has delivered more than 80 lectures at national level. He has developed and implemented numerous software for H.P. University. He is currently working as Professor in Department of Computer Science Himachal Pradesh University of Shimla, India. His teaching experience accumulates to more than 20 years. He has authored few books on computer science both for UG and PG level.

