Study of big data frame work with various issues and challenges involving in data transfer

Sasidhar K[1], Dr Rajesh Kulkarni[2]
[1]*Research Scholar, Career Point University*
[2]*Research Supervisor, Career Point University*

Abstract— **The Hadoop work scheduler can be arranged concerning the accompanying viewpoints: climate, need, asset mindfulness, (for example, CPU time, free opening, circle space, I/O use), time, and systems. The primary thought behind planning is to limit upward, assets, and consummation time, and to amplify throughput by designating position to the processor. Here, the order of schedulers is done in light of the planning techniques, time, and assets.**

Index Terms: **Big Data, Map Reduce, Hadoop.**

INTRODUCTION

Static planning techniques dispense a task to the processor before execution of a program starts. The handling asset and occupation execution time are perceived uniquely at the hour of accumulation. The fundamental motivation behind this sort of schedulers is to limit the general handling season at present running position. FIFO (First in First out), Delay, Capacity, LATE (Longest guess time to end), and Matchmaking booking techniques go under the classification of Static planning.

Dynamic planning methodologies allot a task to the processor at the hour of execution of the program. The scheduler has some information about the asset before execution, however the climate where the work will be executed is absolutely obscure, and the work will be executed during their life time. In a powerful climate, the choices are made and applied to the processor when the execution of the gig begins. Asset Aware and Deadline Constrain go under the classification of dynamic schedulers.

Fundamentally, this booking procedure depends on the asset prerequisite of the gig. Under this system, asset use (like I/O, memory usage, plate capacity, and CPU time), and occupation execution is moved along. Defer scheduler, Matchmaking scheduler, and Resource Aware schedulers are totally initiated on asset accessibility.

This booking system depends on schedule; here, work culmination relies upon the client cut off time. In this planning procedure, there is a period limit inside which the occupation should be finished. Two booking procedures, Deadline Constrain and Delay are utilized for time based work planning. FIFO booking strategy is the default strategy utilized in Hadoop. This approach gives more inclination to the positions coming in sooner than those approaching in later. At the point when new positions show up, the Job Tracker pulls the earliest work first from the line. Here, independent of the size of the gig or any sort of need, just the succeeding position is permitted into the line and the excess positions need to hold on until the main is executed. FIFO planning strategy is utilized when the request for execution of occupations has no significance.

Advantages

1. FIFO planning procedure is the least complex and generally effective among all the schedulers.
2. The positions are executed in similar request in which they are submitted.

Disadvantages

1. A significant downside of FIFO booking is that it isn't pre planned. Thusly, it isn't appropriate for intelligent positions.
2. Another downside is that a long-running interaction will defer for every one of the positions behind it.
3. FIFO Scheduler doesn't consider the equilibrium of asset designation between lengthy positions and short positions.
4. It lessens information area and starvation of occupations.

Map Reduce:

Map Reduce programming model, presented by Google, has now been the true handling motor for dissecting enormous scope datasets. These days, different executions of Map Reduce, including Hadoop, Spark, MR-MPI, and M3R definitely stand out enough to be noticed from both industry and the scholarly world. For instance, Hadoop, an open-source execution of Map Reduce, has been embraced by many driving IT organizations, including IBM, Intel, Amazon, Yahoo!, and so on, and applied by various associations to help various testing applications, such as large-scale graph processing, genomics computation, mining massive astrophysical datasets as well as facial similarity and recognition. According to the 2011 IDC report, the market of Map Reduce and its ecosystems will continue expanding and become a multi-billion business by 2016. Three superior characteristics of Map Reduce have greatly contributed to its success.

First of all, Map Reduce programming model intelligently exposes simple map and reduce interfaces to application developers, meanwhile hiding the complexities, such as fault-handling, intermediate data shuffling, etc. Therefore, it substantially relieves the development burden from the system designers who can then focus on business logic to satisfy the needs from customers. Secondly, Map Reduce aims to be fault-tolerant. Recognizing that failure is the norm, Map Reduce frameworks assume that the underlying systems are unreliable, thus has enabled fault handling throughout the entire system design. From a high-level perspective, contemporary Map Reduce systems generally rely on fine-grained job partitioning and task failure-over to resist common failure scenarios. Meanwhile, due to inherent task independence, Map Reduce can avoid expensive job abortions when failures occur. Thirdly, via minimizing the coupling among tasks, Map Reduce programming model exhibits superior scalability [2]. A recent report from Yahoo!, showing the deployment of YARN Map Reduce across 30,000 nodes is the best testaments of such characteristic. Under such unprecedented scale, Map Reduce applications are empowered to harness the computation capability of large clusters and address many challenges that will open up new horizons for making more discoveries. Following the far-reaching ubiquity of Map Reduce is the developing interest to improve its exhibition for different purposes. Specifically, speeding up the information development and improving the task are becoming basic since they on a very basic level which decides if the systems can accomplish the presentation objectives in various use cases. Redirecting from the underlying plan objective that Map Reduce is principally utilized for bunch occupations, contemporary Map Reduce systems are as a rule extensively utilized to process intuitive queries and streaming information on stages highlighting unmistakable execution qualities, including multi-inhabitant conditions and High-Performance Computing (HPC) groups.

In light of these difficulties, the focal point of this exposition is on examining the hindrances and chances to all the while upgrade Map Reduce structures from many sides, including accelerating single work execution, adjusting decency among simultaneous responsibilities. It aims to achieve this goal via shedding light on the inefficiencies and bottlenecks in Map Reduce frameworks, designing novel algorithms to exploit the performance from underlying infrastructures and balance the trade-off among multiple performance objectives, respectively. In particular, this work introduces several techniques to accelerate the intermediate data movement and enhance fairness within task management.

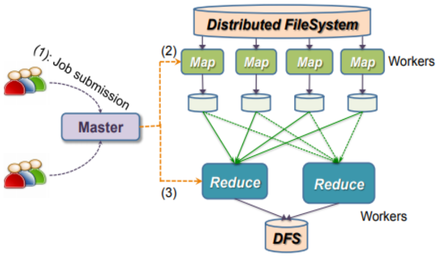


Figure : An Overview of Data Processing in Map Reduce framework

Productive information development is basic for Map Reduce systems. An overall Map Reduce structure comprises of two classifications to work with the information development: a Master, and numerous Workers, as displayed in the Figure. After clients submit occupations to the Master, the Master orders Workers to deal with information in equal through two fundamental capacities: map and diminish. Across this cycle, the Master is responsible for booking map errands and decrease undertakings from simultaneously running position to Workers by means of following specific planning strategies, like First-In-First-Out (FIFO) or Fair Sharing. In the interim, it likewise screens their advances, gathers run-time execution insights, and handles potential issues and mistakes through task re-execution. In the guide stage, the Master picks different Workers and schedules them to run the work. Every Worker dispatches a few guide assignments, one for each split of information that is recovered from the disseminated document framework, like Hadoop Distributed File System or Google File System. In each parted, client information is coordinated as many records of matches. The planning capacity in a guide task change over the first records into transitional outcomes, which are information records as matches.

In the mix stage, when some guide yields become accessible, the Master chooses one more arrangement of Workers to decrease errands. Each decrease task begins by bringing a parcel that is expected for it from a guide yield document (additionally called section). Regularly, there is one portion in each guide yield for each decrease task. In this way, a diminish task necessities to bring such fragments from all guide yield documents. Internationally, these activities lead to an all-to-all rearranging of information portions among all the guides and decrease errands. While the information portions are being rearranged, they are likewise converged inside each lessen task in light of the request for keys in the information records, as more remote sections are brought and blended locally, a diminish task needs to spill, i.e., store, a few fragments to nearby plates to ease memory pressure. Because of the simultaneous execution nature of mix and consolidation, this stage is additionally alluded as the mix/combine stage. In the decrease stage, each lessen task loads and cycles the blended sections utilizing the diminish work. The end-product is then put away back to the appropriated document framework.

Hadoop is an open-source execution of MapReduce, plan bears solid comparability to unique Map Reduce as depicted above with various names for significant parts: a Job Tracker, a.k.a. Ace and numerous Task Trackers, a.k.a. Laborers.

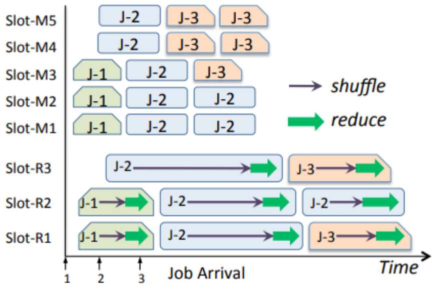


Figure : An Example of Managing Slots among Jobs

Memory-Resident Map Reduce

Flash is one more profoundly famous Map Reduce execution presented by UC Berkeley recently, it has likewise acquired expansive consideration from researchers at the administration processing offices as a promising answer for examine enormous re-enactment results. Like unique Map Reduce, it comprises of two classes of parts: a scheduler and numerous agents. The scheduler is responsible for booking assignments, checking their advancement, and shortcoming dealing through task re-execution. Such confinement of Data Nodes and Executors understands an information driven processing model to limit information development between calculation errands and the capacity framework. Contrasted with other Map Reduce executions, for example, Hadoop, Spark gives two key highlights. To begin with, Spark use the appropriated memory from all slave hubs to store middle of the road information during position execution and the last execution results at work fruition.

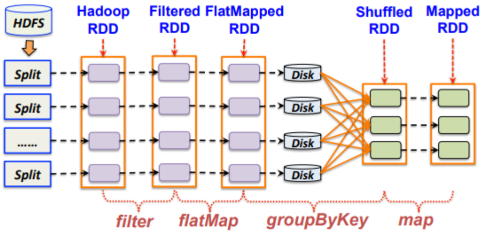


Figure : MapReduce processing pipeline via using RDDs.

At the point when Spark is sent on a bunch including register driven worldview, Hadoop RDD can be supplanted by framework subordinate RDD, like Lustre RDD, to recover input from HPC equal record framework. Flash activities incorporate decrease, count, gather, and so forth. An activity triggers Spark to build an execution plan addressed inside as a coordinated non-cyclic diagram (DAG) that comprises of numerous stages. To stay away from significant upward and give dependable work execution, Spark emerges parcels onto the neighbourhood record framework. Whenever a mix activity is experienced, Spark will attempt two stages for moving halfway information: putting away and rearranging. In the putting away stage, Spark plans a series of Shuffle Map Tasks to flush in-memory yield from the past stage to the record framework. Then in the rearranging stage, a Shuffled RDD is acquainted with move the middle of the road information across the organization.

Overview of Technical Challenges

This work aims to comprehensively optimize the design of Map Reduce frameworks from three aspects: (1) exploiting high-performance I/O layer for accelerating intermediate data movement, (2) enhance task management for provisioning quality-of-service to concurrent workloads in multi-tenant Map Reduce environments, (3) optimizing the adaptability of Map Reduce on HPC platforms. By addressing these three challenges, optimization can effectively enhance the performance from three dimensions, including job execution time, fairness, and cluster utilization.

Map Reduce frameworks have been exceptionally enhanced by many plans to lessen how much organization traffic while perusing input information for Map Tasks and composing yield from Reduce Tasks. For example, postpone booking further develops the information region and diminish information development in the organization. As indicated by their analysis report on many huge bunches, up to 98% Map Tasks can be sent off with inputs on neighbourhood plates. Furthermore, Reduce Tasks normally produce and store the last results to the circles neighbourhood in the appropriated record frameworks. Notwithstanding, moderate information rearranging causes a huge volume of organization traffic and stays as a basic bottleneck of Map Reduce frameworks. Each Reduce Task gets information portions from all guide yields, bringing about an organization traffic design from all Map Tasks to all Reduce Tasks, which fills in the request for O (N2), accepting that Map Tasks and Reduce Tasks are both a component of N all out undertakings. As announced by from Yahoo!, the middle information rearranging from 5% of huge positions can consume over 98% organization transfer speed in a creation group, and more awful yet, Map Reduce execution corrupts non-straight with the increment of halfway information sizes. Likewise, network transfer speed oversubscription can rapidly immerse the organization connections of those machines take part in the diminish stage, this halfway information rearranging basically turns into the predominant wellspring of organization traffic and execution bottleneck in Map Reduce. However, due to slow performance of disk devices, intermediate data merging is substantially detrimental to the performance of Reduce Tasks as pointed out by many studies, leading to severely degraded job performance. Slow merging process can significantly delay the progress of Reduce Task to enter into the reduce phase and also incur large amount of small random read during the computation stage.

Existing Map Reduce clusters are no longer solely used for single user single job environment. Instead, they are being shared among many users and running a mix of diverse types of concurrent workloads, including batch jobs and interactive queries in parallel. Such sharing is motivated by many desirable features, such as statistical multiplexing and data consolidation. To cope with such trend, many scheduling policies have been proposed to optimize multiple metrics, sometimes conflicting, simultaneously for the concurrent Map Reduce workloads so that they can deliver good quality-of-services. Improving the performance of a single job generally requires provisioning more resources to accelerate the processing of its tasks. While, on the contrary, maintaining the fairness deprives jobs of available resources, leading to degraded job execution times. Thus, it is challenging to balance between efficiency and fairness. Hadoop Fair Scheduler introduced by Facebook and Hadoop Capacity Schedulers initiated by Yahoo! are two notable efforts to improve both these metrics, however, studies show that they are still far away from delivering the optimal performance. Many issues are hindering Map Reduce schedulers from gaining both efficiency and fairness. Firstly, existing solutions assume tasks are short, thereby only assigning tasks when the resources are released by previous tasks. Secondly, current fair scheduling policies rigidly balance resources among jobs in a weighted fair sharing manner without considering the progress of each individual job and leveraging the lessons from previous scheduling research. However, when certain small jobs are very close to complete, prioritizing them can effectively improve the efficiency with negligible fairness violation. Therefore, current solutions lack the capability to enhance the efficiency. Thirdly, once tasks are assigned, they occupy the resource until completion or failure, and existing schedulers do not take account of the utilization of the taken resources, thus incurring resource underutilization when tasks are long running and exhibit intermittent I/O execution patterns.

Despite the fact that Map Reduce was at first presented for item groups, it has additionally acquired expansive consideration from researchers at the initiative registering offices as a promising answer for dissect huge recreation results. Subsequently, there is a developing interest for adjusting Map Reduce frameworks for High-Performance Computing (HPC) stages generally conveyed in the public labs throughout recent years. In total, there are two vital differentiations between Map Reduce structures and HPC frameworks. To begin with, there is a critical distinction on the organization of figure and capacity assets. Second, there is a significant contrast as far as the effect of undertaking planning and information territory. Given these differentiations, it is trying to adjust Map Reduce structures for HPC stages since just conveying Map Reduce on the frameworks can't completely take advantage of the exhibition benefits of HPC bunches. Accordingly, to improve the flexibility, portraying the exhibition of Map Reduce on HPC platforms is basic. Specifically, it is basic to measure the effect of HPC stockpiling frameworks on the info recovering, middle of the road information putting away and rearranging, as well as result sinking. Consequently, it is likewise vital to acquire understanding how this qualification can influence traditional Map Reduce task the board strategies. Besides, HPC frameworks are advancing to be appropriate for Map Reduce model. One model is that numerous HPC groups are embracing a progressive system of various sorts of capacity gadgets on conventional memory-just process hubs. Consequently, it is likewise of basic significance to concentrate on the presentation effect of such pattern on the Map Reduce systems.

Issues Preventing Efficient Data Movement

Current Hadoop design can uphold TCP/IP convention to move the middle information without the ability to use the superior exhibition network conventions usually utilized in High-Performance Computing people group. In this following, we talk about these three issues exhaustively.

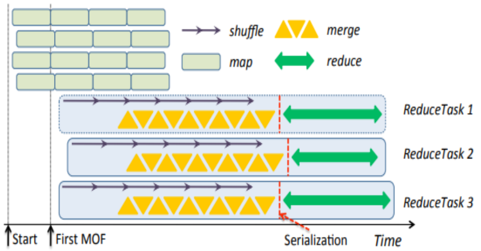


Figure : Serialization between Shuffle/Merge and Reduce Phases

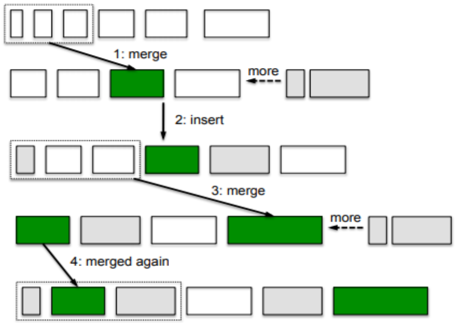


Figure : Repetitive Merging and Disk Access

Table : Profile of Excessive I/O During Merging

Total number of segments	480
Intermediate data per reduce task	5.69GB
Percentage of segments that are merged once	100%
Percentage of segments that are merged twice	98.1%

To represent the redundant, I/O brought about by the current consolidation calculation in Hadoop, we have directed a test running Tera Sort with 120GB information across 20 hubs. We count the quantity of allotments that are converged somewhere around once and measure the information size engaged with the consolidating system. Such exorbitant I/O exasperates the impedance among undertakings and defer the execution of whole Map Reduce programs.

Issues In Hadoop Task Management

To help numerous clients and occupations (enormous group occupations and little intelligent inquiries), Hadoop Map Reduce takes on a two-stage (map and diminish) plan to execute undertakings for information escalated applications. In any case, they don't turn out successfully for the two stages. Dissimilar to Map Tasks which are for the most part exceptionally brief and sent off each gathering in succession to deal with information parts, Reduce Tasks have an alternate execution design. As shown by the Facebook follow, the execution season of normal Reduce Task is longer than that of Map Tasks by one significant degree. Moreover, when a Reduce Task is sent off, it possesses the decrease space until consummation or disappointment.

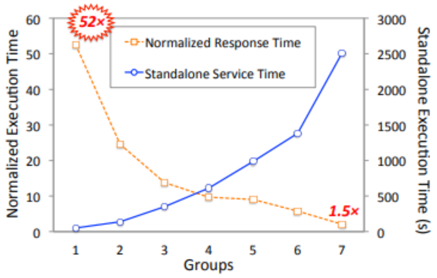


Figure : Unfair Execution among Different Size Jobs

Occupations in the modest gatherings have a lot of more awful standardized execution times, proposing that they should stand by extremely lengthy (however much 52× longer than the independent execution time). More terrible execution results have been seen with HCS, in this way overlook its presentation for compactness. Such booking conduct goes against clients' instinctive assumption that more modest positions ought to be finished quicker and pivoted all the more rapidly, showing serious shamefulness issue in existing Hadoop Map Reduce schedulers.

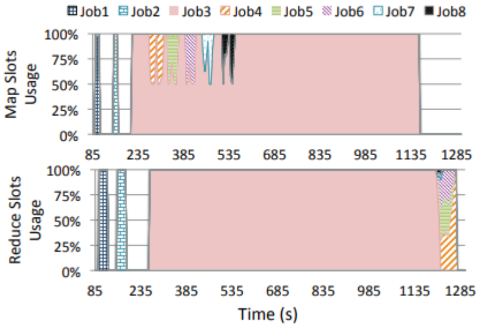


Figure : Run-Time Allocation Profile of Map and Reduce Slots.

To all the more intently analyse the issue between various positions, lead one more analysis on a group of 20 hubs. 40 guide spaces are made on 10 hubs, and 20 decrease openings on another 10 hubs. 8 positions are consecutively submitted into the group at regular intervals. Work 3 is a huge work that requires 20 Reduce Tasks. It shows the utilization of guide and lessen openings by 8 positions. Map openings are effectively divided between occupations over the long run as occupations show up and leave, however diminish spaces are completely involved by Job 3. Accordingly, Jobs 4-8 can't find an offer until Line of work 3 finishes, regardless of whether they have effectively completed all their Map Tasks. By and large, Jobs 4-8 are seriously dialled back by 1486%, contrasted with their independent execution times. This uncovers that Hadoop Fair Scheduler can't accomplish fair standardized execution times for all positions. A comparable way of behaving has additionally been accounted by an IBM study. Note that there exists a sensational fluctuation among the standardized execution for various positions in a similar pool and in various tests.

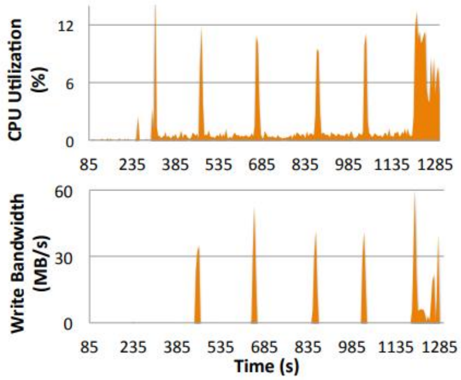
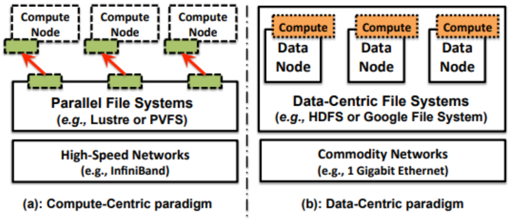


Figure : Inefficient Usage of Reduce Slot

Existing schedulers are likewise unaware of the asset underutilization issue brought about by lengthy running Reduce Tasks. Whenever the age pace of middle information is low, in any event, while long running Reduce Tasks are possessing the spaces, they don't proficiently use the assets, and Reduce Tasks occasionally go into the inactive state, causing serious asset underutilization. During the mix stage, Reduce Tasks possibly bring moderate information from far off Map Tasks when there are free guide yields. Whenever halfway information is inaccessible, a Reduce Task goes into the inactive status. Yet, it actually possesses the space, impeding different positions from obtaining the asset and debasing the general framework proficiency. Figure presents the normal CPU usage and plate compose transfer speed on the machines that host Job 3 in the past examination. We can see that, between 235th second and 1135th second (the execution time of Job 3's Reduce Tasks), CPUs and circles much of the time become inactive and are just intermittently initiated, despite the fact that another 5 positions are as yet holding up in the line. Likewise, additionally see that organization is exceptionally underutilized in this situation. By and large, during 87.6% of Job 3's Reduce Task execution time, CPUs and plates are inactive and hanging tight for the moderate information. This issue can be more exacerbated when a huge occupation is going after map openings with different positions. This opposition for map openings drags out the execution of the enormous work and slower its age pace of transitional information, consequently further postpones the arrival of diminish spaces, extending any remaining position's stand by time.

Issues In Adapting Map reduce For HPC Platforms

Numerous associations have been embracing Map Reduce and sending its various executions to address their issues of huge calculation and examination of colossal datasets, in this manner digging basic information for their business. In this advanced scramble for gold from information, various associations are confronting totally different contemplations with regards to a choice on their information investigation frameworks. With the commonness of cloud stages and business figuring administrations, numerous clients can pass on that choice to their framework suppliers. However, the framework suppliers truly need to shuffle between two decisions: would it be advisable for them to develop without any preparation devoted frameworks for information investigation, or would be advisable to advance their frameworks to fulfil the needs of information examination applications while proceeding to help existing applications and clients. Going with this predicament is that there is a secret change in perspective alongside the emanant centre around enormous information. For the initial not many years of PC history, figuring power has been a scant asset. The need to dissect enormous information has really driven the progress of PC frameworks into an information driven worldview for which the terrific goal is to achieve the quickest examination power as far as the quantity of bytes and records handled each second. In the accompanying, further portray the worldview qualifications between HPC frameworks and Map Reduce model.



(a): Compute-Centric paradigm

(b): Data-Centric paradigm

Figure : Data-centric and compute-centric paradigms.

Figure shows a correlation among register and information driven standards. There are two vital qualifications between these standards. To begin with, there is a vital distinction on the situation of figure and capacity assets. The regular register driven worldview has isolated process and capacity assets as a PC bunch and an equal record framework that are associated by means of fast organizations. Interestingly, the information driven world view gives co-found process and capacity assets on a similar hub. Second, there is a critical distinction as far as the effect of errand booking and information situation. In the figure driven world view, undertakings on process hubs are, as a general rule, similarly far off from the backend stockpiling framework. In the information driven world view, undertakings have solid partiality to the hubs containing their datasets. As a result of these qualifications, on figure driven worldview, applications dividing similar information frequently include tedious information development among the processing asset and the capacity backend. Interestingly, the information driven world view gives co-found figure and capacity assets on a similar hub to work with region situated task booking. These qualifications among register and information driven ideal models have critical execution suggestions to various sorts of utilization jobs. For framework suppliers who are anxious to help more Map Reduce-put together examination applications with respect to HPC stages, describing the exhibition of key engineering parts in these two distinct paradigms is basic.

Privacy Preservation on Big Data Using Map Reduce

Most recent Trends and Challenges in Anonymization Technology by Satoh &Takahashi et al (2014) "Offsetting utility with namelessness of information". It is feasible to handle individual information into an express that has essentially killed singularity remembered for information leaving just the vital least measure of data as indicated by the insightful reason. Measurable information is a genuine model. This is a predicament between the namelessness and utility of individual information. This can be accomplished by a technique called PK-Anonymity. This causes information unlimited as far as which they to have a place with through randomization, which is handling to change individual information probabilistically. In randomization, records are handled to be related to a likelihood of 1/k or less. This nature of Anonymization called PK-obscurity (probabilistic k-Anonymity). From that point onward, execute handling to assess the first condition of the information by utilizing an AI strategy called Bayesian deduction. Thusly, pragmatic unknown information for examination will be built and could say that this is pseudo-individual information in light of genuine individual information. PK-anonymization is viable for anonymization of large information while holding a comparable nature to k-anonymization. There is no such thing as flexible anonymization technique. It is dependent upon the situation as per the sorts, highlights and usage purposes.

CONCLUSION

This paper presents an overview of bigdata framework with the study of various parameters under bigdata. It gives map reduce technique, technical challenges with bigdata, issues with bigdata transferring, issues in Hadoop management and issues with map reduce techniques. It concludes with privacy preservation with the bigdata techniques.

REFERENCE

[1] S. Taneja, B. Suri, H. Narwal, A. Jain, A. Kathuria and S. Gupta, "A new approach for data classification using Fuzzy logic," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016, pp. 22-27, doi: 10.1109/CONFLUENCE.2016.7508041.

[2] J. Han, M. Kamber and J. Pei, Data mining concepts and techniques, Morgan Kaufmann Publishers, pp. 285-370, 2012, ISSN 1238-1489.

[3] P. Kromer, J. Platos, V. Snasel and A. Abraham, "Fuzzy classification by evolutionary algorithms", *Systems Man and Cybernetics (SMC) 2011 IEEE International Conference*, pp. 313-318, 2011, ISSN 1062-922X.

[4] P. Pendharkar, "Fuzzy classification using the data envelopment analysis", *Knowledge-Based Systems*, vol. 31, pp. 183-192, 2012, ISSN 0950-7051.

[5] I.H. Witten and E. Frank, Data mining: Practical machine learning tools and techniques- tutorial exercises for the weka explorer, Morgan Kaufmann Publishers, vol. 33, pp. 559-575, 2011, ISSN 0808-9035.

[6] Y. Gupta, A. Saini and A.K. Saxena, "A new fuzzy logic based ranking function for efficient information retrieval system", *Expert Systems with Applications*, vol. 42, no. 3, pp. 1223-1234, 2015, ISSN 0957-4174.

[7] L. Íñiguez, M. Galar and A. Fernández, "Improving Fuzzy Rule Based Classification Systems in Big Data via Support-based Filtering," 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1-8, doi: 10.1109/FUZZ-IEEE.2018.8491500.

[8] I. Witten, E. Frank, M. Hall and C. Pal, Data Mining Practical Machine learning tools and techniques., Elsevier, 2016.

[9] C. P. Chen and C.-Y. Zhang, "Data-intensive applications challenges techniques and technologies: A survey on big data", *Information Sciences*, vol. 275, pp. 314-347, 2014.

[10] A. Fernández, S. del Río, V. López, A. Bawakid, M. J. del Jesus, J. M. Benítez, et al., "Big data with cloud computing: an insight on the computing environment mapreduce and programming frameworks", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 5, pp. 380-409, 2014.

[11] X. Wu, X. Zhu, G. Q. Wu and W. Ding, "Data mining with big data", *IEEE Transactions on*

Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, Jan 2014.

[12] G. Michael, "A framework for considering comprehensibility in modeling", *Big Data*, vol. 4, no. 2, pp. 75-88, 2016.

[13] M. Gacto, R. Alcalá and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures", *Information Sciences*, vol. 181, no. 20, pp. 4340-4360, 2011.

[14] S. del Río, V. López, J. M. Benítez and F. Herrera, "A mapreduce approach to address big data classification problems based on the fusion of linguistic fuzzy rules", *International Journal of Computational Intelligence Systems*, vol. 8, no. 3, pp. 422-437, 2015.

[15] M. Elkano, M. Galar, J. Sanz and H. Bustince, "Chi-bd: A fuzzy rule-based classification system for big data classification problems", *Fuzzy Sets and Systems*, 2017.