International Conference on Computer Applications 2016 [ICCA 2016]

ISBN	978-81-929866-5-4	**VOL**	05
Website	icca.co.in	**eMail**	icca@asdf.res.in
Received	14 – March– 2016	**Accepted**	02 - April – 2016
Article ID	ICCA005	**eAID**	ICCA.2016.005

Data Dimensional Reduction by Order Prediction in Heterogeneous Environment

P Suganya[1], Thirupurasundari D R[2]

[1,2]Department of Computer Science and Engineering, Meenakshi College of Engineering, Tamil Nadu, Chennai.

Abstract – *Equalizing the amount of processing time for each reducer instead of equalizing the amount of data each process in heterogeneous environment. A lightweight strategy to address the data skew problem among the reductions of MapReduce applications. MapReduce has been widely used in various applications, including web indexing, log analysis, data mining, scientific simulations and machine translations. The data skew refers to the imbalance in the amount of data assigned to each task.Using an innovative sampling method which can achieve a highly accurate approximation to the distribution of the intermediate data by sampling only a small fraction during the map processing and to reduce the data in reducer side. Prioritizing the sampling tasks for partitioning decision and splitting of large keys is supported when application semantics permit.Thus providing a reduced data of total ordered output as a result by range partitioner. In the proposed system, the data reduction is by predicting the reduction orders in parallel data processing using feature and instance selection. The accuracy of the data scale and data skew is effectively improved by CHI-ICF data reduction technique. In the existing system normal data distribution is calculated instead here still efficient distribution of data using the feature selection by χ 2 statistics (CHI) and instance selection by Iterative case filter (ICF) is processed.*

The decision tree classifier is used to classify the data stream to produce an appropriate reduced data set.

Keywords – *MapReduce, data skew, sampling, partitioning, CHI-ICF, data reduction.*

Nomenclature

CHI	χ 2 Chi-square statistic
ICF	Iterative Case Filter
LIBRA	Lightweight Implementation of Balanced Range Assignment
IS	Instance Selection
FS	Feature Selection

I. INTRODUCTION

Parallel programming is developed as a means of improving performance and efficiency. In a parallel program, the processing is broken up into parts, each of which can be executed concurrently. The parallel programs are faster, they can also be used to solve problems on large datasets using non-local resources.

MapReduce is inspired by these parallel programming concepts. It was developed by Google as a mechanism for processing large amounts of raw data, for example, crawled documents or web request logs. This data is so large, it must be distributed across

Cite this article as: P Suganya, Thirupurasundari D R. "Data Dimensional Reduction by Order Prediction in Heterogeneous Environment". *International Conference on Computer Applications 2016*: 22-28. Print.

thousands of machines in order to be processed in a reasonable time. This distribution implies parallel computing since the same computations are performed on each CPU, but with a different dataset. MapReduce is an abstraction that allows Google engineers to perform simple computations while hiding the details of parallelization, data distribution, load balancing and fault tolerance.

MapReduce has proven itself to be an effective tool to process large datasets. If one task takes significantly longer to finish than others are called straggler, it can delay the progress of the entire job due to various reasons, among which data skew is an important one. The data skew refers to the imbalance in the amount of data assigned to each task. To mitigate the data skew problem by Lightweight Implementation of Balanced Range Assignment for reduce-side applications in MapReduce.

The algorithm adjust its work load allocation in heterogeneous platform to deliver improved performance. In the proposed system, data reduction is carried out on the basis of the most efficient feature selection of data distribution by χ 2 statistics(CHI) and instance selection by Iterative case filter(ICF) instead of Zipf's distribution .The decision tree classifier is used to classify the data stream to enhance the parallelism and performance of data processing.

II. Related Work

Qi Chen, Jinyu Yao, and Zhen xiao, [1] proposed a new sampling method for general user-defined MapReduce programs. The method has a high degree of parallelism and little overhead, which can achieve a much better approximation to the distribution of the intermediate data by zipfian distribution. It uses an innovative approach to balance the load among the reduce tasks in heterogeneous environment. The system can adjust its work load allocation and delivers improved performance even in the absence of data skew. Speculative execution is not effective for straggler, as it reduces the job execution time. A common measurement for data skew is the coefficient of variation is calculated.

Y.kwon, M.Balazinska, B.Howe and J.Rolia, [2] the focus of this paper was on UDOs in the form of MapReduce applications. In particular, skew is a significant challenge in many applications executed on this platform. When skew arises, some partitions of an operation take longer to process their input data than others, slowing down the entire computation. Load imbalance can occur either during the map or reduce phases. Such an imbalanced situation referred as map-skew and reduce-skew respectively. Skew can lead to longer job execution times and signicantly lower cluster throughput. The two very common types of skew: (1) skew caused by an uneven distribution of input data to operator partitions and (2) skew caused by some portions of the input data taking longer to process than others. The skewTune's approach to planning mitigators is presented to find a contagious order-preserving assignment of intervals. This performs scan in parallel than local scan to effectively keep the overheads low with large datasets to be repartitioned.

G.Benjamin, A.Nikolaus, R.Angelika, and K.Alfons, [3] addressed the problem of efficiently processing MapReduce jobs with complex reducer tasks over skewed data. It defined a new cost model that takes into account non-linear reducer-tasks and provided an algorithm to estimate the cost in a distributed environment. The two load balancing approaches proposed are fine partitioning and dynamic fragmentation, that are based on cost model and can deal with both skewed data and complex reduce tasks allowing improved load balancing. Varying execution times result in low resource utilisation and high overall execution time since the next MapReduce cycle can only start after all reducers are done. The empirical calculations are used to evaluate the solution on both synthetic data and real data on scientific applications.

Y.kwon, M.Balazinska, B.Howe and J.Rolia, [4] presented SkewReduce, a new API for feature-extracting scientific applications to generate parallel processing plans that leverages user-supplied cost functions. This method employs a static optimizer which reduces the impact of computational skew inherently. A high fidelity cost function benefits SkewReduce's optimization and offers a more general skew-resistant solution to improve application (scientific domains like Astronomy Simulation, Flow Cytometry and image processing) run times.

S.Ibrahim, J.Hai, L.Lu, W.Song, H.Bingsheng, and Q.Li, [5] investigated the problem of Partitoning Skew in MapReduce system which degrades the performance of huge data transfer during the shuffle phase particularly in reduce phase. Based on this a novel algorithm was developed named LEEN for Locality/ fairness-aware key partitioning in MapReduce. It improved the data locality by guaranteeing near optimal balanced reducer's input execution efficiency and fairness distribution of intermediate data during and after the shuffle phase which reduced the network congestion and achieves acceptable data distribution fairness. LEEN achieved both fair data distribution and performance under moderate and large key's frequency variations by the asynchronous map and reduce scheme among different data nodes.

Jeffrey Dean and Sanjay Ghemawat described about MapReduce is a programming model and an associ-ated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

III. Existing Approach

In this paper the system which implements the LIBRA approach [1] to solve data skew for general applications like MapReduce framework. The design goals of LIBRA includes

Cite this article as: P Suganya, Thirupurasundari D R. "Data Dimensional Reduction by Order Prediction in Heterogeneous Environment". *International Conference on Computer Applications 2016*: 22-28. Print.

- Transparency - Data skew mitigation should be transparent to the users who do not need to know any sampling and partitioner details.
- Parallelism - It should preserve the parallelism of the original MapReduce framework as much as possible. This precludes any pre-run sampling of the input data and overlaps the map and the reduce stages as much as possible.
- Accuracy - Its sampling method should be able to derive a reasonably accurate estimate of the input data distribution by sampling only a small fraction of the data.
- Total order - It should support total order of the output data. This saves applications which require such ordering an extra round of sorting at the end.
- Large cluster splitting - When application semantics permit, it should be able to split data associated with a single large cluster to multiple reducers while preserving the consistency of the output.
- Heterogeneity consideration - When the performance of the worker nodes is heterogeneous, it should be able to adjust the data partition accordingly so that all reducers finish around the same time.
- Performance improvement - Overall, it results in significant improvement in application level performance such as the job execution time.

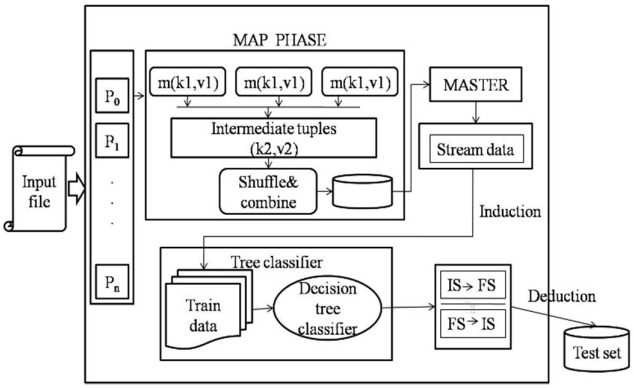

Fig.1 System Architecture

The architecture of the system is shown in Fig.1. Data skew mitigation of Straggler with LIBRA in a MapReduce system, 1) The input file is divided into multiple parts and assigned to a group of map tasks for parallel processing. 2) Each map task transforms its input (K1, V1) tuples into intermediate (K2, V2) tuples according to some user defined map and combine functions, and outputs them to the local disk. 3) Each reduce task copies its input pieces from all map tasks, sorts them into a single stream by a multiway merge, and generates the final (K3, V3) results according to some user defined reduce function.

As a result, the number of clusters is equal to the number of distinct keys in the input data. Each reduce task copies its partition (containing multiple clusters) from every map task and processes it locally. This require a total order of the output data.

The SkewTune technique tackles the data skew problem from a different angle. It does not aim to partition the intermediate data evenly at the beginning. Instead, it adjusts the data partition dynamically after detecting a straggler task, repartitioning the unprocessed data of the task and assigning them to new tasks in other nodes which reconstructed the output by concatenating the results from those tasks according to the input order. SkewTune and LIBRA are complementary to each other. When load change dynamically or when reduce failure occurs, it is better to mitigate skew lazily using SkewTune. On the other hand, when the load is relatively stable, LIBRA is better to balance the copy and the sort phases in reduce tasks and its large cluster split optimization improves the performance further when application semantics permit. The effective $\chi 2$ statistic distribution (CHI) and Iterative Case Filter (ICF) algorithm is proposed to improve the data reduction which is based on feature and instance selection.

Feature selection is a pre-processing technique for selecting a reduced set of features for large-scale data sets according to their feature values and a given number of words with large values are selected as representative features. An Instance selection provides a reduced data set by removing non-representative instances like noisy and redundant instances.

The tree classifier technique is employed to predict the reduction orders among parallel data processing by deducting from the stream of data to mitigate the data skew. Thus the performance of data reduction and scalability of the system is enhanced.

IV. Random Sample Allocation

The Data Skew mitigation process of the system consists of following steps as shown in the below fig.2. we provide an effective cluster split strategy which allows large clusters to be split into multiple reduce tasks when appropriate. We modify the partition decision to

Cite this article as: P Suganya, Thirupurasundari D R. "Data Dimensional Reduction by Order Prediction in Heterogeneous Environment". *International Conference on Computer Applications 2016*: 22-28. Print.

include both the partition keys and the partition percentage. For example, a partition decision record (k,p) means that one of the partition point is p percent of key k. For map tasks issued before the partition decision is made, we can easily find these percentage partition points from the total-ordered intermediate outputs by adding some fields to the sparse index record.

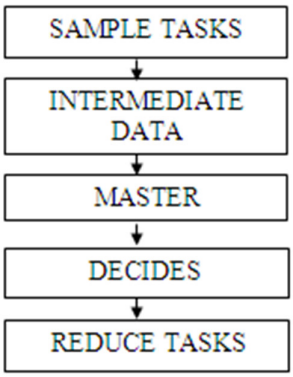

Fig.2 Data Skew Mitigation Process

The new fields we add are the current record number in the key cluster Kbi and the total record count of Kbi. By calculating the ratio of current record number in Kbi to the total count of Kbi, we can get the key and the percentage in its cluster for the start record in each index block. In this way, we can quickly locate the index blocks which contain the partition points. For map tasks issued after the partition decision is made, we calculate a random secondary key in the range [0, 100] for each record and compare (key, secondary key) to partition decision records to decide which partition it belongs to (the order within the key may not be the same as input order). Using this cluster split strategy, the solution of the example shown can be optimized with 60 percent of the large key A to reducer1 and the rest keys to reducer2. By doing so, the data skew is mitigated.

A. Chunk Index for Partitioning

After the master notifies the worker nodes of the partition decision ready event, the worker nodes take responsibility for partitioning the intermediate data previously generated by the sampling tasks and already launched normal map tasks accordingly. This in general involves reading all the records from the intermediate output, finding the position of each partition key, and generating a small partition list which records the start and the end positions of each partition. When a reducer is launched later, the worker nodes can use the partition lists to help the reducer to locate and copy the data associated with its allocated key range from the map outputs quickly. The challenge here is how to find the partition breakpoints in a large amount of intermediate data. Since the intermediate data can be too large to fit into the memory, a brute force method using linear or binary search can be very time consuming.

When application semantics permit, cluster splitting provides substantially more flexibility in mitigating the data.

A small percentage of the map tasks are selected as the sample tasks. They are issued to the system when it has free slots. Sample tasks collects statistics on the intermediate data distribution during map process and transmit a digest of that information to the master after completion of all processes. The master collects all the sample information to derive an estimate of the data distribution, makes the partition decision and notifies to the reduce tasks.

V. Data Reduction by Load Balancing

The sampling and partitioning algorithm in existing system is to balance the load across reduce tasks. The algorithm consists of three steps as given in the below diagram fig.3

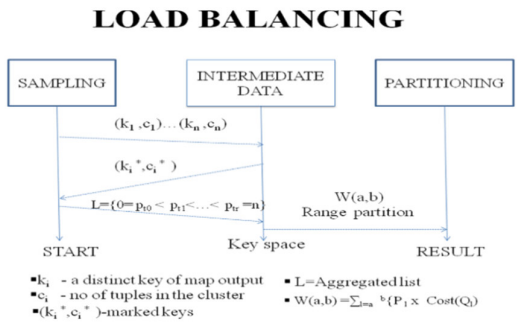

Fig.3 Load balancing sequences

Cite this article as: P Suganya, Thirupurasundari D R. "Data Dimensional Reduction by Order Prediction in Heterogeneous Environment". *International Conference on Computer Applications 2016*: 22-28. Print.

B. LIBRA-Balanced Range Partitioning Algorithm

The approach of integrating sampling into a small percentage of the map tasks are prioritized for the execution of those tasks. An effective cluster split strategy which allows large clusters to be split into multiple reduce tasks.

The optimized cluster split strategy with 60 percent of the large key to reducer (P1) and rest keys to the reducer(P2). By doing so, the data skew is mitigated. Here, the translation of English dictionary to other languages like tamil, hindi and french is performed with the MapReduce concept.

- Sampling Strategy To Control Data Skew

A specific map task of English words 'j' is chosen for sampling its normal execution with a lightweight sampling procedure (ki j ,ci j) where ci j is the frequency(no of records of ki j). A sample set SSample contains the following two parts:

- Slargest: p tuples with the largest ci j.
- Snormal: q tuples randomly selected according to uniform distribution

The master generates the total number of records S Sample = S largest U S normal.

The ratio of p/q is positively related to the degree of the data skew when it is heavier, the larger ratio should be chosen for good approximation.

By following a Zipf distributions with varying σ parameters from σ = 0.2 to 1.4 to control data skew. The COV is calculated for the English words with the frequency of words value.

For the optimal data reduction average coefficient variation (avg COV) of all σ for each p/q ratio as follows:

$$\text{Coefficient of variation} = \frac{stddev \quad \overline{(x)}}{mean \quad \overline{(x)}}$$

$$\text{Avg COVp/q} = \frac{\sum_{aEs} COV\sigma^{p/q}}{|S|}$$

- Estimating the Intermediate Data Distribution

Assuming two sample map tasks ,let m1 & m2 with their sample sets

m1={ (A,10),(B,5),(C,3),(D,2),(E,2)}

m2={(A,20),(B,3),(D,1),(F,1),(H,1)}

by summing up the frequencies of the same key, the merged sample set

S sample ={ (A,30),(B,8),(C,3),(D,3),(E,2),(F,1),(H,1)}.

There are m1 =50 keys &1000 records and m2= 60 keys &1500 records.

Therefore, Aggregated total records TR=2500 is used to estimate overlap degree of sample

$$\text{Degree} = \frac{2*\left(|L|+|S_{Sample}|+L-p\right)}{|L|+|S_{Sample}|-2p}$$

Where,

Cite this article as: P Suganya, Thirupurasundari D R. "Data Dimensional Reduction by Order Prediction in Heterogeneous Environment". *International Conference on Computer Applications 2016*: 22-28. Print.

L=Aggregated list , p=large tuple value.

The performance meteric for the existing system is the moving average of the process bandwidth and approximate the intermediate data.

- Range partition

Range partition : 0= pt0 < pt1<…< ptr =n of data distribution to get approximate data reduction and skew mitigation solution by generating a list of partition lists and it is minimized. The list of English words is partitioned logically for 26 alphabets and special characters.

$$\text{Minimize} = \max i=1,2,\ldots r \left\{ \frac{\sum_{j=p_{ti-1}+1}^{p_{ti}} \{P_j \; XCost \quad (Q_j)\}}{e_i} \right\}$$

C. Accuracy of Sampling Method

The sampling method achieves a good approximation to the data distribution, which follows the Zipf distribution. The TopCluster sampling method and random sampler is used to split the data values.

D. Degrees of the Data Skew

The coefficient of variation changes when the skew increases. The performance of the system with and without cuslter split strategy across reducers is produced.

E. Sort, Grep, Inverted Index and Join

- Sort. We use the sort benchmark in our main workload because it is widely used and represents many kinds of data-intensive jobs. We generate 10 GB synthetic datasets following Zipf distributions with varying s parameters to control the degree of the skew. We choose Zipf distribution workload because it is very common in the data coming from the real world, e.g., the word occurrences in natural language, city sizes, many features of the Internet, the sizes of craters on the moon.
- Grep. Grep is a popular application for large scale data processing. It searches some regular expressions through input text files and outputs the lines which contain the matched expressions. We modify the grep benchmark so that it outputs the matched lines in a descending order based on how frequently the searched expression occurs. The dataset we used is the full English words.
- Inverted Index. Inverted indices are widely used in search area. We implement a job that builds an inverted index from given documents and generates a compressed bit vector posting list for each word. We use the Potter word stemming algorithm and a stopword list to pre-process the text during the map phase, and then use the RADIX partitioner to map alphabet to reduce tasks in order to produce a lexicographically ordered result. The dataset we used is also the full English Dictionary.
- Join. Join is one of the most common applications that experience the data skew problem. We implement a simple broadcast join job in which partitions a large table in the map phase, while a small table is directly read in the reduce phase to generate a hash table for speeding up join operation. When the small table is too large to fit into the memory, we use a buffer to keep only a part of the small table in memory and use the cache replacement strategy to update the buffer. We use synthetic datasets which follow Zipf distribution to generate the large tables, while use data-sets which follow either the uniform distribution or the Zipf distribution to generate the small tables.

F. Heterogeneous Environment

To show the system can fit well with the variable environments, we set up a heterogeneous test environment by running a set of CPU and I/O intensive processes (e.g., heavy scientific computation and dd process which creates large files in a loop to write random data) to generate background load on two of the servers. We use sort bench-mark with (s ¼ 0:2). We intentionally choose a small s value so that all methods can partition the intermediate data quite evenly. This allows us to focus on the impact of environment heterogeneity.

Cite this article as: P Suganya, Thirupurasundari D R. "Data Dimensional Reduction by Order Prediction in Heterogeneous Environment". *International Conference on Computer Applications 2016*: 22-28. Print.

LIBRA: with or without considering environment heterogeneity, and with or without cluster split enabled. Thus the light weight data skew mitigation is reduced by the above explained algorithm and data reduction is performed.

VI. Feature and Instance Selection

In the proposed system, the data size reduction is based on efficient feature selection of data distribution by χ 2 statistics(CHI) and instance selection by Iterative case filter(ICF) instead of Zipf's distribution .The decision tree classifier is used to classify the data stream to enhance the parallelism and performance of data processing. By predicting the reduction orders based on CHI and ICF for better performance and accurate result. The pseudo code for feature selection will be,

Select features(D,c,k)
V ← Extract vocabulary (D)
for each t V
do A(t,c) ← computer feature utility(D,t,c)
append(L, {A(t,c),t}
return features with largest values(L,k)

VII. Performance Analysis -Libra vs CHI - ICF

The analysis of the system is based on their performance, accuracy and scalability is obtained. The comparison of both existing and future system will be evaluated and thus the later system is the best approach to data reduction is proven. At various platform the system performance will be processed by applying the enhanced feature and instance selection technique.

VIII. Conclusion and Future Work

In this paper, the proposed data dimensional reduction by predicting the order using selection technique witnesses the explosive growth of data processing to routinely generate hundreds of tera-bytes of logs and operation records by MapReduce. Data skew mitigation has improved the MapReduce Performance by the implementation of LIBRA which supported a large cluster split and its adjustment for heterogeneous environments. In the future work, the MapReduce performance has been enhanced by chi-square χ 2 statistics (CHI) of feature selection and Iterative case filter (ICF) of instance selection instead of Zipf's distribution.The decision tree classifier is used to classify the data according to its stream to improve the parallelism and accuracy. The exact word in the file is retrieved by feature to instance selection.

References

1. Qi Chen, Jinyu Yao, and Zhen xiao, "LIBRA:Lightweight data skew mitigation in MapReduce", IEEE Transactions on Parallel and Distributed Systems,VOL. 26,NO.9,September 2015.
2. Y.kwon,M.Balazinska,B.Howe and J.Rolia,"Skewtune:Mitigating skew in map reduce applications,"in proc.ACM SIGMOD Int.Conf.Manage.Data,2012,pp.25-36.
3. G.Benjamin,A.Nikolaus,R.Angelika, and K.Alfons,"Handling data skew in mapreduce,"in Proc.Int.Conf.Cloud Comput.Serv Sci.,2011,pp.574-583.
4. Y.kwon,M.Balazinska,B.Howe and J.Rolia,"Skew-resistant parallel processing of feature extracting scientific user-defined functions,"in Proc.ACM Symp.Cloud comput.Technol.sci.,2010,pp.75-86.
5. S.Ibrahim,J.Hai,L.Lu,W.Song,H.Bingsheng,andQ.Li,"Leen:Locality/fairness-aware key partitioning for mapreduce in the cloud,"in Proc.IEEE Int. Conf .cloudcomput .Technol .Sci .,2010,pp.17-24.
6. J. Dean and S. Ghemawat, "Mapreduce: Simplified data process-ing on large clusters," Commun. ACM, vol. 51, pp. 107–113, Jan. 2008.
7. https://dzone.com/articles/how-hadoop-mapreduce-works
8. Data Resource: http://www.tamildict.com, http://www.wordreference.com

Cite this article as: P Suganya, Thirupurasundari D R. "Data Dimensional Reduction by Order Prediction in Heterogeneous Environment". *International Conference on Computer Applications 2016*: 22-28. Print.