INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION & MANAGEMENT

A Monthly Double-Blind Peer Reviewed (Refereed/Juried) Open Access International e-Journal - Included in the International Serial Directories

Indexed & Listed at:

Ulrich's Periodicals Directory ©, ProQuest, U.S.A., EBSCO Publishing, U.S.A., Cabell's Directories of Publishing Opportunities, U.S.A.,

Open J-Gage, India [link of the same is duly available at Inflibnet of University Grants Commission (U.G.C.)],

Index Copernicus Publishers Panel, Poland with IC Value of 5.09 & number of libraries all around the world.

Circulated all over the world & Google has verified that scholars of more than **4700 Cities** in **180 countries/territories** are visiting our journal on regular basis.

Ground Floor, Building No. 1041-C-1, Devi Bhawan Bazar, JAGADHRI – 135 003, Yamunanagar, Haryana, INDIA

http://ijrcm.org.in/

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the areas of Computer Science & Applications; Commerce; Business; Finance; Marketing; Human Resource Management; General Management; Banking; Economics; Tourism Administration & Management; Education; Law; Library & Information Science; Defence & Strategic Studies; Electronic Science; Corporate Governance; Industrial Relations; and emerging paradigms in allied subjects like Accounting; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Rural Economics; Co-operation; Demography: Development Planning; Development Studies; Applied Economics; Development Economics; Business Economics; Monetary Policy; Public Policy Economics; Real Estate; Regional Economics; Political Science; Continuing Education; Labour Welfare; Philosophy; Psychology; Sociology; Tax Accounting; Advertising & Promotion Management; Management Information Systems (MIS); Business Law; Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labour Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; International Relations; Human Rights & Duties; Public Administration; Population Studies; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism & Hospitality; Transportation Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic; Web Design and emerging paradigms in allied subjects.

Anybody can submit the **soft copy** of unpublished novel; original; empirical and high quality **research work/manuscript** ***anytime*** in ***M.S. Word format*** after preparing the same as per our **GUIDELINES FOR SUBMISSION**; at our email address i.e. infoijrcm@gmail.com or online by clicking the link **online submission** as given on our website (***FOR ONLINE SUBMISSION, CLICK HERE***).

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____________

THE EDITOR

IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF _______________________.**

(e.g. Finance/Mkt./HRM/General Mgt./Engineering/Economics/Computer/IT/ Education/Psychology/Law/Math/other, please specify)

DEAR SIR/MADAM

Please find my submission of manuscript entitled '____' for possible publication in one of your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication elsewhere.

I affirm that all the co-authors of this manuscript have seen the submitted version of the manuscript and have agreed to their inclusion of names as co-authors.

Also, if my/our manuscript is accepted, I agree to comply with the formalities as given on the website of the journal. The Journal has discretion to publish our contribution in any of its journals.

NAME OF CORRESPONDING AUTHOR	:
Designation	:
Institution/College/University with full address & Pin Code	:
Residential address with Pin Code	:
Mobile Number (s) with country ISD code	:
Is WhatsApp or Viber active on your above noted Mobile Number (Yes/No)	:
Landline Number (s) with country ISD code	:
E-mail Address	:
Alternate E-mail Address	:
Nationality	:

NOTES:

a) The whole manuscript has to be in ***ONE MS WORD FILE*** only, which will start from the covering letter, inside the manuscript. ***pdf. version* is liable to be rejected without any consideration**.

b) The sender is required to mention the following in the **SUBJECT COLUMN of the mail**:

New Manuscript for Review in the area of (e.g. Finance/Marketing/HRM/General Mgt./Engineering/Economics/Computer/IT/ Education/Psychology/Law/Math/other, please specify)

c) There is no need to give any text in the body of mail, except the cases where the author wishes to give any **specific message** w.r.t. to the manuscript.

d) The total size of the file containing the manuscript is expected to be below **1000 KB**.

e) **Abstract alone will not be considered for review** and the author is required to submit the **complete manuscript** in the first instance.

f) ***The journal gives acknowledgement w.r.t. the receipt of every email within twenty four hours*** and in case of non-receipt of acknowledgment from the journal, w.r.t. the submission of manuscript, within two days of submission, the corresponding author is required to demand for the same by sending a separate mail to the journal.

g) The author (s) name or details should not appear anywhere on the body of the manuscript, except the covering letter and the cover page of the manuscript, in the manner as mentioned in the guidelines.

2. **MANUSCRIPT TITLE:** The title of the paper should be **bold typed**, **centered** and **fully capitalised**.

3. **AUTHOR NAME (S) & AFFILIATIONS:** Author (s) **name**, **designation**, **affiliation** (s), **address**, **mobile/landline number** (s), and **email/alternate email address** should be given underneath the title.

4. **ACKNOWLEDGMENTS:** Acknowledgements can be given to reviewers, guides, funding institutions, etc., if any.

5. **ABSTRACT:** Abstract should be in **fully italicized text**, ranging between **150** to **300 words**. The abstract must be informative and explain the background, aims, methods, results & conclusion in a **SINGLE PARA**. ***Abbreviations must be mentioned in full***.

6. **KEYWORDS:** Abstract must be followed by a list of keywords, subject to the maximum of **five**. These should be arranged in alphabetic order separated by commas and full stop at the end. All words of the keywords, including the first one should be in small letters, except special words e.g. name of the Countries, abbreviations.

7. **JEL CODE:** Provide the appropriate Journal of Economic Literature Classification System code (s). JEL codes are available at www.aeaweb.org/econlit/jelCodes.php, however, mentioning JEL Code is not mandatory.

8. **MANUSCRIPT:** Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. *It should be free from any errors i.e.* ***grammatical, spelling*** *or* ***punctuation.*** *It must be thoroughly edited at your end*.

9. **HEADINGS:** All the headings must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

10. **SUB-HEADINGS:** All the sub-headings must be bold-faced, aligned left and fully capitalised.

11. **MAIN TEXT:**

THE MAIN TEXT SHOULD FOLLOW THE FOLLOWING SEQUENCE:

INTRODUCTION
REVIEW OF LITERATURE
NEED/IMPORTANCE OF THE STUDY
STATEMENT OF THE PROBLEM
OBJECTIVES
HYPOTHESIS (ES)
RESEARCH METHODOLOGY
RESULTS & DISCUSSION
FINDINGS
RECOMMENDATIONS/SUGGESTIONS
CONCLUSIONS
LIMITATIONS
SCOPE FOR FURTHER RESEARCH
REFERENCES
APPENDIX/ANNEXURE

The manuscript should preferably range from ***2000 to 5000 WORDS***.

12. **FIGURES & TABLES:** These should be simple, crystal **CLEAR**, **centered**, **separately numbered** & self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. *It should be ensured that the tables/figures are referred to from the main text*.

13. **EQUATIONS/FORMULAE:** These should be consecutively numbered in parenthesis, horizontally centered with equation/formulae number placed at the right. The equation editor provided with standard versions of Microsoft Word should be utilised. If any other equation editor is utilised, author must confirm that these equations may be viewed and edited in versions of Microsoft Office that does not have the editor.

14. **ACRONYMS:** These should not be used in the abstract. The use of acronyms is elsewhere is acceptable. Acronyms should be defined on its first use in each section: Reserve Bank of India (RBI). Acronyms should be redefined on first use in subsequent sections.

15. **REFERENCES:** The list of all references should be alphabetically arranged. ***The author (s) should mention only the actually utilised references in the preparation of manuscript*** and they are supposed to follow Harvard Style of Referencing. **Also check to make sure that everything that you are including in the reference section is duly cited in the paper**. The author (s) are supposed to follow the references as per the following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parenthesis.
- ***Headers**, **footers**, **endnotes** and **footnotes** should **not be used** in the document.* However, **you can mention short notes to elucidate some specific point**, which may be placed in number orders after the references.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio" Ohio State University, Nigeria.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg, Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–23

UNPUBLISHED DISSERTATIONS

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITES

- Garg, Bhavet (2011): Towards a New Gas Policy, Political Weekly, Viewed on January 01, 2012 http://epw.in/user/viewabstract.jsp

HADOOP MapReduce

MANISHA AGARWAL
SR. ASST. PROFESSOR
AIM & ACT DEPARTMENT
BANASTHALI VIDYAPITH
TONK

SOURABH MUKHERJEE
ASSOCIATE PROFESSOR
AIM & ACT DEPARTMENT
BANASTHALI VIDYAPITH
TONK

SUSHMA GREWAL
STUDENT [M. Tech. (IT)]
BANASTHALI VIDYAPITH
TONK

ABSTRACT

MapReduce is a program foundation which helps the builders in writing instructions which processes large quantity of disordered information parallelly over a scattered group of processors and separate or unconnected systems [1]. In 2004, google develops the concept of MapReduce. MapReduce software is comprised of a Map process whose work is to conduct refinement and resolving and a Reduce process whose work is to perform an review of working performance. A MapReduce foundation coordinates the actions by assembling the scattered servers, executing different jobs parallelly, handles each and every communiqué along with transfer of information among the disparate sections of the network, and provides tautology with defect forbearance and scalability also. GridGain and Hadoop are two open source java implementations of MapReduce.

KEYWORDS

MapReduce, hadoop, gridgain.

INTRODUCTION

MapReduce is a computer instruction writing prototype that was developed through the agency of Google [1]. MapReduce was formulated so that corresponding information handling on huge groups of systems can be simplified.

This planning and scheduling prototype is stirred through Map and Reduce basics that can be seen in Lisp or many other functional languages. Google had done a large number of various implementations for processing and computing large datasets then introduces the MapReduce framework. Mostly the given information found to be extremely huge but the calculations were comparatively simpler. Therefore the computations were needed to be scattered all over hundreds of systems so that the calculations can be finished within a smaller duration.

MapReduce can be used for processing large datasets because of its efficiency and scalability. When the concept of MapReduce came into exixtence, Google rewrite its entire web searching index system for using this modish planning and scheduling prototype. The categorization structure generates the information configuration for the use through Google network forage. The input data here exceeds 20 terabytes. In starting, the categorization structure move like a cycle of eight MapReduce functions, however various novel stages were included later. Altogether, the mid point of thousands of MapReduce processes are running everyday on Google cluster, processes more over and above 20 Petabytes of information everyday [2]. The concept of MapReduce is used for hiding the problem in data correspondance, defect forebearnce, data supply and weight balancing. In addition with calculational problems, the software enigneer hardly requires to explain framework to control the information categorization and correspondance. Google implemented for designing the huge clusters of interconnected machines. For example, Hadoop is one of the open-source implementations of MapReduce, written in Java programming language. Hadoop utilizes various system in groups for distributing information management same as google MapReduce.

The parallelization does not need to be performed across various interconnected machines. There are different applications of MapReduce for parallel computing in various surroundings. Phoenix is another implementation of MapReduce that is intended to distributed cache, multiple hub and parallel processing structure. Mars is also a MapReduce implementation for graphic processing machine (GPUs). GPUs are massive coresspondent actionist having very high computational capacity and cache baud rate as compared to CPUs, but it is persistant to plan because of its construction and interfaces which are formulated especially for graphic implementations. MapReduce foundation can hide this difficulty, so that programmers can smoothly bind together the computational capacity of the GPU for information management task.

MapReduce

MapReduce is a planning and scheduling prototype originated by Google. MapReduce foundation comprises of customer donated Map and Reduce function and an implementation of MapReduce repository, which automatically manages data distribution, parallelization, load balancing, fault tolerance, etc.

With this, a user has to write few patterns, such as name of the input and output files, or a few alternative elective regulating parameters. The patterns additionally explains the breaking of given information to key/value pairs. In MapReduce prototype, customers describes their computations in the form of two tasks, one is Map and the other is Reduce. A key or value set is given to the Map task and it produces the output as a group of intermediary key/value sets.

Reduce accepts a key in the form of input and a inventory of values allotted to it. Input values for Reduce are spontaneously clustered from intermediary produce from the MapReduce repository. When essential Map jobs are finished, the repository acquires an intermediary key and batched it jointly with every other values related with it. The Reduce task acquires an intermediary key and the value repository allotted to it in the form of an input. It amalgamates the values in a manner that the customer has described in the execution of the Reduce task, and generates small groups of values. Commonly zero or one output or response is generated at every Reduce job. This prototype has some limits, because it only supplies map and reduce tasks to be executed by customer. Due to these drawbacks, MapReduce offers a simpler connection for customers for parallel and distributive computations. A programming model with limitations is good because due to that developer can concentrates on formulation of the real problem with only two simpler tasks.

Nevertheless, drawbacks reach it hard to utter some issues with this prototype. Yet many of information management jobs can be executed using MapReduce with efficiency. It is not difficult to include modish MapReduce phase to surviving MapReduce operation. By adding MapReduce sections, extra difficult issues can be uttered with this prototype.

The very common illustration of this prototype is a problem of calculating a quantity of recognizable words in a huge collection of documentation. The illustration is from [2].

```
/* key: page name
* value: page contents*/
map(key, value){
for each word w in value:
emit Intermediate(w, "1");}
/* key: a word
* values: list of counts for the word*/
reduce (key, values){
int result = 0;
for each v in values:
result += ParseInt(v);
emit(result);}
```

APPLICATIONS

After the introduction of MapReduce framework, research areas with various types of difficulty domains has increased using a MapReduce. Various calculations can be finished only by utilizing the

MapReduce prototype, but there are a few problems which cannot be uttered by using the Map and Reduce tasks.

Take an instance, the recurrences method of Genetic Algorithms cannot instantly be uttered by using Map and Reduce tasks. Genetic Algorithms are a type of development algorithms that are used in streams of chemistry and biology. Parallel Genetic Algorithms are adopted in order to achieve a better effectiveness, because in processing Genetic Algorithms for huge problem domains require a very long duration. MapReduce requires to be considerable to assist these type of algorithms, that can be acquired by including a second reduce section after the rounds, and a client for cooperating

the implementation of rounds. MapReduce is utilized in SMS message fedge. SMS messages are mainly utilized for a uncomplex communication among users. In any country a large number of messages are sent in every month. MapReduce is utilised as a foundation in message mining [3]. The messages are processed in three steps. Firstly, the pre-processing of original dataset is done and assembled by sender mobile numbers. It is finished by initiating MapReduce method. Second MapReduce method does a change for the regrouping of the datasets by small contented keys, and lastly the third MapReduce method is required for the mining of the famous messages.

Error-correcting ciphers are also helpful in several ways. If information files are needed to be stored on a defective channel, then the file is ciphered with an error-correcting cipher. If the file gets taint when storing, then it is restored when decoding the error correcting cipher. Encoding a huge number of files or huge files is the main issue which needs attention. For standard encoding and decoding algorithms, it is difficult to manage huge block size and it does not permit careless access of the data. Encoding should not be done on small pieces of a file error correction can be achieved with a improving working on huge documents.

Google's computing infrastructure is utilized by Feldman and also Google's MapReduce execution for encoding and decoding a huge twister cipher. Twister cipher are the cipher utilised for correcting errors. These ciphers contain some linear time algorithm to decode and encode. Mapreduce offers a concept named parallelization by the help of which twister codes can be used for large files.

Particle Swarm Optimization algorithm can be naturally utilised with MapReduce. To enhance tasks which have to check huge data, Particle Swarm Optimization algorithm is used on parallelization.

Generalized Stochastic Petri nets are a popular graphical modeling formalism which is used in the operational scrutiny of system and communiqué systems. The reply times can be calculated in parallel with the help of MapReduce.

There is an instance in astronomy related to this, in which the Large Synoptic Survey Telescope generates nearly twenty Terabytes data every night. Spatial queries are decayed and processed with the help of MapReduce for optimizing the performance, like reply time. Spatial queries includes spatial selection query, spatial join query, nearest neighbor query, etc. Parallelization is a better solution for these kind of problems.

IMPLEMENTATIONS

- **Google's MapReduce**

The target of google's MapReduce is huge groups of interconnected machines. The MapReduce library manages parallelization and data distribution in a mechanical way. The developers can concentrate on the main issue that is the presentation of computing problem using map and reduce functions because they have no need to be anxious about parallel and network programming. Local disk of interconnected machines are used for the distribution and saving of data.

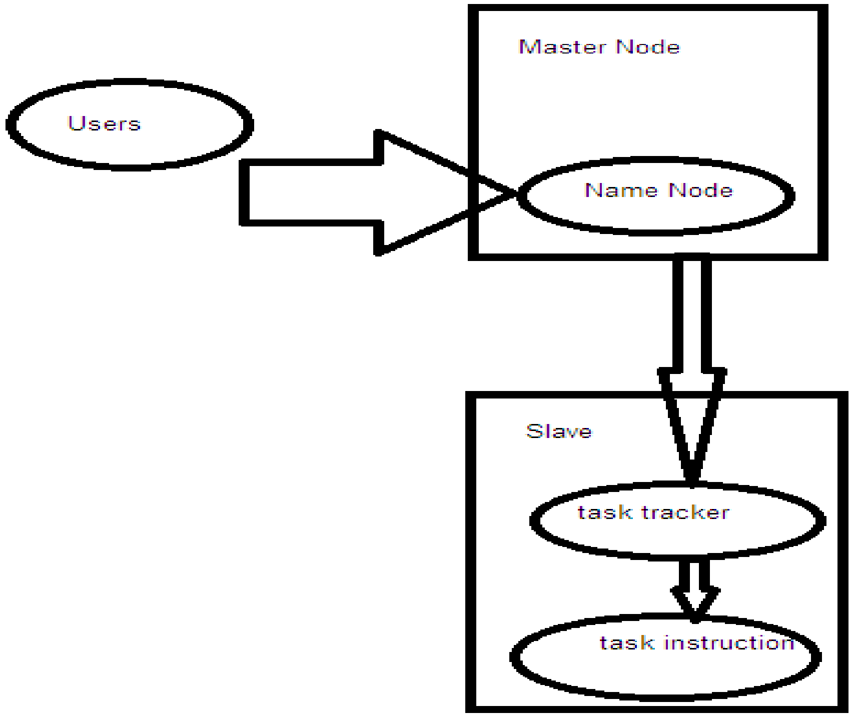
Google File System (GFS) [1] is a scattered file arrangement which is utilised for supervising saved information over the groups. For making the node more reliable and fault tolerant, GFS generates replicas of data blocks on various nodes.

MapReduce has high scalability; hence it can run on groups consisting of hundreds of cheaper cost machines.

- **Hadoop**

Hadoop is a MapReduce application whose developer is Apache [4]. The construction of Hadoop is necessarily the same as similar Google's application but the major dissimilarity is that Hadoop is an open-source application. Hadoop Distributed File System (HDFS) [4] is used of distributing the data across the interconnected machines. And it can generate a number of duplicate copies of data blocks to make it more reliable. Data storage is done on local disks on interconnected machines that make the data to access from other machines in the network.

FIGURE 1: HADOOP MapReduce [2]



It comprises of two main methods, one is the Name node and another is Data nodes.The Name node executes on a isolated leading system. It contains all information regarding every system in the network, and also the information of the data blocks that are saved on the machines in the group. Data node process executes on all the remaining machines in the group, and they communicates to the Name node for knowing that when data is to be obtained on their local hard disk.

Hadoop comprises of one Job Tracker and a number of Task Tracker methods or processes. The Job Tracker generally executes on the equivalent leading system as the Name node. Users drive their MapReduce functions to the Job- Tracker, that's job is to divide the exertion among the systems in the group. Every other system in group executes a Task Tracker method. The

TaskTracker interacts with the JobTracker that will allot it a Map or Reduce task whenever possible.

- **GridGain**

The GridGain has a similarity with hadoop that it is also an open source map reduce implementation [5]. From technical aspect, the major distinction is in the allotment operation of map function to various junctions. In the MapReduce algorithm the resposibility is divided into subtasks and operator pulls the divided fragments as soon as they have leisure processing time. In GridGain the subtasks are propelled towards the junctions. It enhances the load balancing property. In this, the developers have to preplan so that no operator will remain idle without any cause. It is unpopular as compared to Hadoop, but it has improved documentation and is good for novice users.

- **Phoenix**

Phoenix is a MapReduce application designed for distributed-cache machines [7]. It comprises of MapReduce programming prototype and related runtime repository which manages resources, fault tolerance and other related issues in a mechanical way. Threads are used by phoenix for generating parallel Map and Reduce functions. It can be utilized for parallelizing information concentrated computing on multiple hub and multiple processors machines. Overheads that are due to job reproduction and information interactions can be reduced while functiong in a distributed cached surroundings.

SCHEDULING

Here the main question arises that how the map and reduce tasks of thousands of jobs, that are present on a hadoop cluster for their processing, are scheduled. For this purpose

Hadoop uses a FIFO scheduler out of the box. Afterwards, two more schedulers have been developed. First one is the Fair Scheduler developed by facebook for fast response time for smaller jobs and a moderate finish time for the production jobs. Second one is Capacity Scheduler that was developed by Yahoo and this scheduler contains named queues in which jobs are submitted. A fraction of the total computing resource is assigned to the queues and jobs are weighted. There is no single scheduling algorithm that is suitable for different types of jobs or tasks. So we can conclude that a mixture of scheduling algorithms can be a better option depending upon various workload features.

ISSUES OF SCHEDULING IN MAP REDUCE

The following are the various issues in map reduce scheduling [6] :

- **Locality**

Out of the major problems in MapReduce scheduling, one is locality. The space between the input data junction and task -assigned node is termed as locality. Data transfer cost is minimum if the input information junction is near to the computational junction. Locality adversely affects the performance in a shared clustered or grouped environment, because of limited bisection bandwidth of network. The throughput of tasks is directly proportional to the locality. High the locality more will be the throughput of the tasks. The processing of a task or job on a junction having the information, called junction locality is the effectual case of locality. If node locality is not possible to attain, then running a task or job on the identical frame, known as rack locality, is used. If the requirements of locality are not complete, the transfer of data input output costs can affects the production because of the cached baud rate of network.

- **Synchronization**

It is the method of transfer of the intermediary output of the map process to the reduce process in the form of an input, is consider as a element too that adversely affects the performance. Mapper has to wait unless every map tasks are completed for initiating sending intermediary output. Because of the dependent nature of map and reduce sections of processing on each other, a isolated junction can deaccelerates the whole operation, which causes the other junctions to wait for its completion. There are many factors that lead to fall in the performance in the synchronization step, some of these are heterogeneity of the groups, junction abortion miss-configuration, and serious raised burden of the input output quotation.

- **Fairness**

Many MapReduce tasks are executed in distributed information repository of organization such as facebook, Amazon, Google and Yahoo. A map-reduce task with a hefty assignment may influence the use of the distributed groups, so a few small computing tasks may not have the required response time. When every map-reduce task has nearly an equivalent percentage of the junctions and the input files are scattered in distributed file system, a few map process has to stack information from the network. Due to this, there is a large fall in the throughput and response time. Synchronization raised burden could affect the fairness. Let us consider an instance, reduce processes have to wait unless the map functionsare completed or that results in idle nodes and starvation of remaining tasks. Because of this, there is no proper utilization.

CONCLUSION

Hadoop is in a great demand in the market nowadays. As there is a large amount of data is lying in the industry but there is no proper tool that can manage it and hadoop can be implemented on lower cost hardware and can be used by a huge number of users. Map reduce is the most important component in hadoop. In this paper we have studied various methods for making the scheduler more efficient for the map reduce so that we can increase systems processing power, computational power, speed and performance.

REFERENCES

1. The survey on MapReduce, V.Vijayalakshmi, Ass. Professor, department of computer science and engineering, A.Akila, m.tech final year, department of computer science, S.Nagadivya, m.tech final year, department of computer science and engineering, Christ college of engineering and technology, Puducherry.
2. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. ACM, 51(1):107–113, 2008.
3. T. Xia. Large-scale sms messages mining based on map-reduce. Computational Intelligence and Design, 2008. ISCID '08. International Symposium on, 1:7–12, Oct. 2008.
4. Apache. Hadoop documentation, http://hadoop.apache.org/core. 2008.
5. Gridgain: http://www.gridgain.com/
6. Job Scheduling for Multi-User MapReduce Clusters, Matei Zaharia ,hruba Borthakur,Joydeep Sen Sarma,Khaled Elmeleegy,Scott Shenker,Ion Stoica ; Electrical Engineering and Computer Sciences, University of California at Berkeley.
7. C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis. Evaluatingmapreduce formulti-core and multiprocessor systems. High Performance Computer Architecture, 2007. HPCA 2007. IEEE 13th International Symposium on, pages 13–24, Feb.2007.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application & Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you to supply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mail **infoijrcm@gmail.com** for further improvements in the interest of research.

If you have any queries, please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-
Co-ordinator

DISCLAIMER

ABOUT THE JOURNAL

In this age of Commerce, Economics, Computer, I.T. & Management and cut throat competition, a group of intellectuals felt the need to have some platform, where young and budding managers and academicians could express their views and discuss the problems among their peers. This journal was conceived with this noble intention in view. This journal has been introduced to give an opportunity for expressing refined and innovative ideas in this field. It is our humble endeavour to provide a springboard to the upcoming specialists and give a chance to know about the latest in the sphere of research and knowledge. We have taken a small step and we hope that with the active co-operation of like-minded scholars, we shall be able to serve the society with our humble efforts.

Our Other Journals