Workflow Scheduling in Mapreduce Environment by Local and Global Metaheaurstic

Ambika Rani[1], Mr.Karambir[2]

[1]*Assistant Professor, Department of CSE, UIET, KUK, India*
[2]*Student of Department of CSE, UIET, KUK, India*

---***---

Abstract - *In this period of data explosion, parallel handling is essential to processing a vast size of data in a well-timed method. Mapreduce is a accessible and fault-tolerant data handling tool that allows to process a huge size of data in similar through several low-end computing nodes, which has been interpreted by Google. In big data main problem scheduling the jobs and utilize the resources in efficient manner, which reduce the cost and time of task computation or processing. In this paper use amount of task i form of work flow and optimize scheduling by metaheuristics like Ant colony optimization and FCFS algorithm.*

Key Words: Mapreduce, Ant Colony Optimization, FCFS, Metaheaurstic Technique, Job Scheduling, F_ACO, Workflow.

1. INTRODUCTION

Hadoop is meant to expeditiously method massive size of data by linking several trade goods computers along to figure in corresponding. The hypothetical 1000-CPU device delineated previous would price a really great total of cash, much further than one, 250 quad-core machines. Hadoop can attach these minor and a lot of affordable machines along into one cost-efficient reckon cluster.

Activity working out on massive data of knowledge has been finished before, sometimes in an exceedingly circulated setting. A easy programming model make Hadoop different that permits the user to rapidly write and check circulated systems, and its economical, automatic sharing of knowledge and work obliquely machines and successively utilizing the essential correspondence of the central processor cores.

In an exceedingly hadoop bunch, knowledge is circulated to all or any the slot of the cluster because it is being encumbered in. The Hadoop circulated classification system can tear massive knowledge records into chunk that area unit managed by totally different slots within the bunch. Additionally to the current every chunk is simulated across much machinery, in order that one system breakdown doesn't lead to any knowledge being untouchable. A full of life observance structure then re-replicates the information in reply to system failure which may lead to fractional storage. Even if the folder chunks area unit simulated and circulated across much machinery, they type one name space, therefore their filling area unit universally reachable .So their fillings are universally reachable.

1.1 MAPREDUCE

In this era of information explosion, parallel handling is crucial to process an enormous volume of information during a timely methodology. Mapreduce could be a accessible and fault-tolerant data processing implement that lets to process a huge volume of information in parallel with several low-end computing nodes, that has been promoted by Google.

With the goodness of its simplicity, quantify ability and fault-tolerance, Map reduce is turning into universal attaining necessary momentum from each trade and world on the opposite hand Mapreduce has essential limitations on its performance and productivity. Thus, some studies have tried to beat the boundaries of the Mapreduce framework. [1]

HADOOP MAPREDUCE

Hadoop Mapreduce is also [2] a code package framework for merely writing applications. A Map-Reduce job generally separations the input data-set into temporary chunks that component of measurement managed by the map jobs in associate totally similar manner. The framework kinds the outputs of the maps, that unit of measurement at that time input to the reduce tasks. The framework takes caution of programming tasks, observance them and executes the unsuccessful tasks.

A Mapreduce work is also a component of labor that the patron wishes to be performed: it consists of the pc file, the Mapreduce program, and configuration information. Hadoop goes the work by separating it into tasks, of that there are units of two categories: map tasks and reduce tasks. There is couple of nodes that management the effort execution process: a job tracker and style of task trackers. The job tracker synchronizes everything the roles run on the structure by programming tasks to run on task trackers.

Advantages: first off, straightforward and simple to use:-

The Mapreduce model is communicative and easy. A coder describes his job with Map and scale back functions, while not specify physical sharing of his job through nodes. Secondly, flexible:- Mapreduce doesn't obsessed with information model and schema. A coder will simply traumatize irregular or unstructured information. Mapreduce is largely freelance from underlying storage layers. Moreover, Fault tolerance:-Mapreduce is actual fault-tolerant. Besides, Mapreduce cannot be used if the computation obsessed with antecedently computed values.

Mapreduce isn't sensible fit ad-hoc looking or full texts categorization as a result of scale back doesn't manufacture great deal of information. Map and scale back task run severally and in isolation expressed in Mapreduce. In Mapreduce text categorization don't want international synchronization

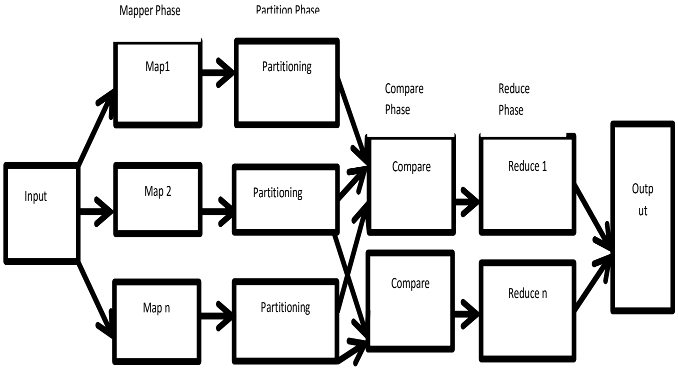

Figure 1:Mapreduce Framework

Sample paragraph, The entire document should be in cambria font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes. The entire document should be in cambria font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

2. RELATED WORK

Shekhar Gupta [3] planned Hadoop was that the de-facto general place for big data analytics applications. They offered schedulers for hadoop clusters appointed tasks to nodes whereas not connection the attainable of the nodes. It condensed the final job action time on a cluster of various nodes by actively programing tasks on nodes supported optimally matching job requirements to node capabilities. The following model was to optimize allocation of tasks to servers. Weikuan Yu [4] represented Hadoop-A, associate acceleration framework that optimizes hadoop with plug-in elements for quick information movement. A unique network-levitated merge formula was familiarized to merge information while not repetition and memory access. Furthermore, a complete pipeline was considered to overlay the shuffle, merge, and scale back phases. Experimental results show that Hadoop-A considerably races information movement in mapreduce and doubles the output of Hadoop. C.P.Saranya [5] analyzed the performance pattern time interval factors in Hadoop distributed arrangement. Supported the dataset size and vary of nodes that were created in Hadoop cluster, the performance of individual jobs unit of measurement legendary. By pattern Johnson's rule, the optimum declare individual jobs for varied disks unit of measurement been calculated. Nidhi Tiwari [6] examined that Mapreduce has turn into ubiquitous for method huge data size jobs. This presented a survey of variety of the Map reduce programing algorithms planned for such advanced eventualities. Taxonomy was provided for Map-reduce algorithms supported their runtime nature. Energy efficiency was typically achieved at the worth performance and availability. Data distribution scheme was worn out each of the key factors for rising the Hadoop Mapreduce Energy efficiency. Chanwit Kaewkasi [7] presented Hadoop cluster for method Brobdingnagian data designed a primary twenty 2 ARM boards. These examined Brobdingnagian process with Hadoop had been rising simply, every on the computing cloud and enterprise preparation. A cluster for big data was completely different from associate MPI-based cluster in terms of the planet of applications and additionally the code package stack. Associate MPI-based cluster targeted on CPU-bound procedure tasks, however a large data cluster performed process that was I/O-bound. Dan Wang [8] planned a framework given Map reduce is also a fairly code package framework for merely writing applications that methodology Brobdingnagian amounts of information on large clusters of artifact hardware and urge higher allocation of tasks and loading deed, the map reduce work mode and task programming rule of Hadoop platform was analyzed. It indicates that it had been effective in making task allocation and achieving wise balance once it's applied into the Hadoop platform that used exclusively Job skilled worker programming. Supriya Pati [9] planned a unique job aware programming formula. Programming formula was necessary for best utilization of cluster resources. Mapreduce word count benchmark was wont to analyze the performance of programming formula. This formula regular job supported 3 criteria i.e. job execution time, earliest point in time 1st and employment of the work. Programming formula was wont to increase resource utilization and reduced the common waiting time by seventy nine in best case and twenty third in average case.

3. PROPOSED WORK

Ant Colony optimization (ACO) is also a fairly Meta heuristics method that's replicated from the social behavior of ants. It may be associatehonest completely different approach to resolve NP exhausting combinatorial optimization problems like 0-1 bag draw back and thus the voice draw back (TSP). ACO can get resolution that is quite nearer to the simplest solution; however premature input bogs the system down. Parallelization is economical due to solve large-scale hymenopterous insect colony optimization problems, since higher resolution wants larger style of ants and iterations that consume longer. The matter is resolved by Map reduce based ACO approach. ACO is native optimization. The capacitated arc routing draw back (CARP) is representative for style of wise applications like gas pipeline bobbing up with and vehicle routing jointly this approach is used in routing of packets throughout a wireless network. CARP's scope is continuous by furthermore as total service time and affixed speculated costs. The planned hybrid technique is used to resolve instances of associate extended CARP and TSP (Travelling Salesmen Problem). This approach is given by the accommodative parameters, native optimization techniques and exploitation of heuristic information. a pair of kinds of heuristic information like arc cluster information and arc priority information obtained infinitely from the solution that's utilized to guide succeeding optimization methodology. The accommodative parameter reduces the problem of choosing initial values and facilitates improvement in hardiness of the results. Finally, the planned technique exploitation the mixture of two heuristic approaches is employed to spice up the final optimization performance of combinatorial problems. Mapreduce is also a programming model associated a related application for method and generating big data sets. Users can require a map perform that processes a key/value try and get a lot of transitional key/value sets, and it is a reduce perform that mixes all intermediate values regarding the same intermediate key. Many globe tasks is expressed with Mapreduce framework, Programs written in Mapreduce technique area unit automatically parallelized associated dead on an outsized cluster of computing machine. The run-time analysis of the strategy handles the tiny print of partitioning the computer file, programming execution of the program across a bunch of machines, handling hardware failures at intervals the term of machine failure, and it jointly controls the inter-machine Communication.

Pseudo code for ACO Metaheaurstic is:

```
Step 1: Set parameters, initialize pheromone trails.
Step 2: SCHEDULE_ACTIVITIES
Step 3: Construct Ant Solutions
Step 4: Daemon Actions    {optional}
Step 6: Update Pheromones
Step 7: END_SCHEDULE_ACTIVITIES
```

The SCHEDULE_ACTIVITIES construct does not specify still the three components area unit regular and synchronized. In most uses of ACO to NP-hard problems however, the three components of algorithm bear a loop that consists in (i) The development of solutions by all ants, (ii) The (optional) improvement of these answer via the utilization of a part search rule, and (iii) The update of the pheromones. These three components unit of measurement presently explained in extra details. [10]

Advantages of the Ant Colony Optimization [11]

ACT provides regeneration accounts for quick discovery of fantastic solutions. Economical for representative draw back and similar problems. It may be utilized in dynamic applications (adapts to changes like new distances etc.)

Disadvantages of the Ant Colony Optimization:-

In this Theoretical analysis is hard and a sequence of random selections (not independent) ACT is likelihood distribution variations by reiteration and during this analysis is experimental rather than theoretical. Time to convergence unsure (but convergence is guaranteed!)

Workflow Ensembles

Scientific workflows, usually represented as Directed Acyclic Graphs (DAGs), are an imperative class of utilizations that lead to testing issues in asset administration on framework and utility processing frameworks. Work processes for vast computational issues are regularly made out of a few interrelated work processes grouped into ensembles. There are several applications that need scientific workflow ensembles.

Randomized groups utilizing work processes accessible from the work process generator gallery [12]. The gallery contains engineered work processes demonstrated utilizing structures and parameters that were taken from genuine applications. Ensembles were made utilizing engineered work processes from two genuine applications: GENOME, LIGO.

Using this gathering of workflows, there are five different ensemble types: steady, uniform sorted, uniform unsorted, Pareto sorted and Pareto unsorted. In the unsorted ensemble, work processes of many sizes are joined and the priorities are allocated arbitrarily. For several applications, huge workflows are more important to users than small workflows because they represent more significant computations. Constant ensembles have workflows that all must the similar number of tasks. The number of tasks is selected randomly from the usual of likely workflow sizes. When the size is gritty, at that time N workflows of that size are certain arbitrarily for the ensemble from the set of synthetic workflows. Uniform ensembles hold workflows through sizes that are uniformly distributed among the set of likely sizes. Each workflow is selected through first arbitrarily choosing the size of the workflow. Pareto ensembles have a lesser number of large workflows and a large number of small workflows. Their sizes are selected according to a Pareto distribution. The distribution was adjusted so that the number of large workflows (of size≥ 900) is improved through a small amount to produce a “heavy tail”. This origins Pareto ensembles to have a slightly larger number of large workflows, which imitates performance detected in many computational workloads.

Time to complete has been later as per Pareto Distribution. With a specific end goal to watch the interesting qualities of the proposed calculations, for every group, we chose ranges for deadline and budget that cover an expansive parameter space: from tight imperatives, where just a little number of work processes can be finished, to more liberal imperatives where all of the work processes can be finished. We figured imperative extents in view of the qualities of every group. The budget imperatives are ascertained by recognizing the littlest spending plan vital to implement one of the work processes in the gathering (MinBudget), and the smallest budget plan required executing all work processes in the group. (MaxBudget):

$$\text{MinBudget} = \min_{w \in e} \square\ Cost(w) \qquad \text{Eq.1}$$

$$\text{MaxBudget} = \sum_{w \in e} Cost(w) \qquad \text{Eq. 2}$$

This range—[MinBudget; MaxBudget]—is then divided into equal intervals to determine the budgets to use in each experiment. In the similar way, the deadline restraints are calculated by finding the smallest volume of time essential to achieve a single workflow in the ensemble (MinDeadline), which is the space of the grave path for the workflow through the shortest critical path, and by finding the minimum amount of time essential to execute totally workflows (MaxDeadline), which is the calculation of the critical paths of all the workflows:

$$\text{MinDeadline} = \min_{w \in e} \square\ Critical\ Path(w) \qquad \text{Eq. 3}$$

$$\text{MaxDeadline} = \sum_{w \in e} Critical\ Path(w) \qquad \text{Eq. 4}$$

This range—[MinDeadline; MaxDeadline]—is then divided into equal intervals. By computing the budget and deadline constraints in this way we ensure that the experiments for each ensemble cover the most interesting area of the parameter space for the ensemble.

F_ACO ALGORITHM:-

Step 1: Initialize the workflow (0……n) by 0

Step 2: After that workflow input.

Step 3: Parse the workflow that changes the workflow into task

Step 4: Use Pareto Distribution which distribute the task into normal form.

Step 5: Apply FCFS

Step 6: After that Map Mapreduce that reduces the map which task provided by FCFS

Step 7: Mapreduce that divided the task into n Data node.

Step 8: Call F_ACO.

Step 9: After completing the F_ACO we apply the fitness function.

Step 10: In the fitness function we check it is optimized or not. If optimized then find the cost and time otherwise go to step 6.

Step 11: End

Pseudo Code:-

```
Set algorithm parameter ()
Begin
Set FCFS (Workflow)
Begin
Task ← i
Scheduled (Task)
Queue (Scheduled Task)
End
i, j ← 0
For j = 1 to colonies do
Ant $S_0$ ← Create Sub-Colony and release agent
While no-termination conditions
On sub colony do
unitcost =amount to execute the task in one second.
unitcost=10
Cost=Runtime*unitcost
i=i+1
Manage_antsactivity ()
Manage_pheromone ()
Manage_DemonAction ()
Selection Procedure ()
Compute Solution Quality ()
End While
J = j+1
$S_{best}$ ← Candidate to be optimal Solution
Update Pheromone on arc ()
End
```

Table1:-Simulation Environment

Parameter	2
Virtual Machine	10

Table 2: Input Table

Job id	File Size	Runtime
1	308078825	41.19
2	512375947	56.69
3	397681391	36.03

4	355294122	81.23
5	397328995	50.14
6	473376657	48.54
7	179677370	24.71
8	316900579	39.38
9	252169932	36.34
10	283032733	30.68
11	340650120	38.08
12	414832252	37.16
13	494817476	68.95
14	306885075	43.25
15	226157082	26.04
16	384438616	55.55
17	480232270	53.59
18	396111458	41.05
19	234447149	28.64
20	439558498	74.13

4. RESULT AND DISCUSSION

Programs written in Mapreduce technique are mechanically parallelized and dead on an oversized cluster of digital computer. The proposed algorithm F_ACO is tested on a Workflow. Cost of F_ACO was compared with ACO Cost in order to prove the effectiveness of F_ACO algorithm. Figure 3 describes the cost respect to LIGO and GENOME. Initially the cost is low. LIGO provide better result a compare to GENOME.

Application	F_ACO	ACO	FCFS
LIGO	00.03	14.05	00.63
GENOME	30.27	28.68	36.05

Table 3: Cost between LIGO and GENOME

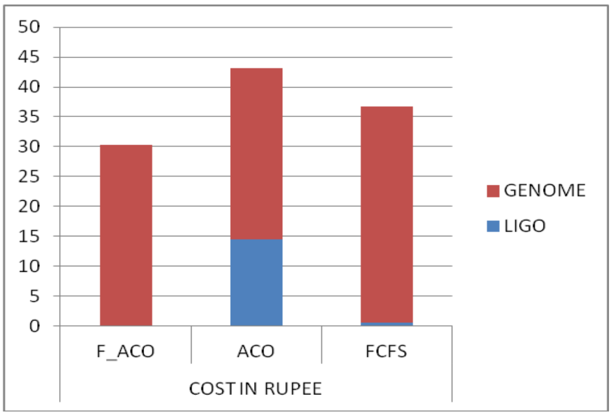

Fig 2: Cost between LIGO and GENOME

We analyze cost and time on different workflows on mapreduce environment and compare with each other. F_ACO method gives better results to other.

Application	F_ACO	ACO	FCFS
LIGO	67.96907	1395.094	199.7565
GENOME	11184.8936	9911.427	13749.38

Table 4: Time between LIGO and GENOME

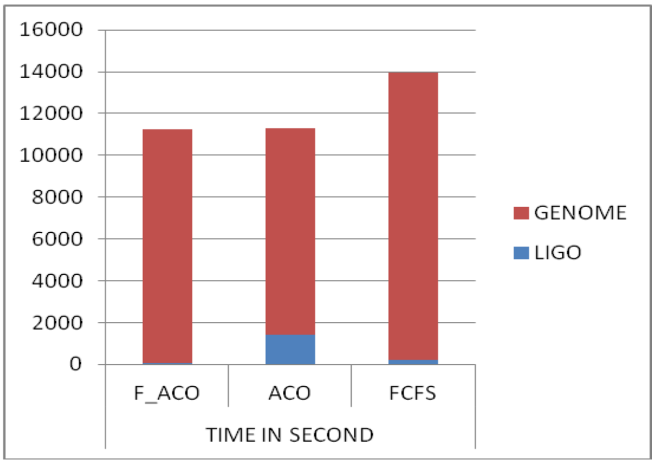

Fig 3: Time between LIGO and GENOME

5. CONCLUION

Programs written in Mapreduce technique are mechanically parallelized and dead on an oversized cluster of digital computer. The run-time analysis of the strategy handles the small print of partitioning the input file, programming execution of the program across a group of machines, handling hardware failures within the term of machine failure, and it conjointly controls the inter-machine Communication. In this paper we analyze cost and time on different workflows on mapreduce environment and compare with each other. Ant colony method gives better results to other.

REFERENCES

[1].Kyong-Ha Lee , Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, Bongki Moon, “Parallel Data Processing with Mapreduce: A Survey”, SIGMOD Record, Vol. 40,Issue(4),pp:11-20,2011

[2].Duan, Ailing and Hai Fang Si., "Research and Practice of Distributed Parallel Search Algorithm On Hadoop_MapReduce", IEEE International Conference on Control Engineering and Communication Technology (ICCECT), 2012

[3]. Gupta Shekhar, Christian Fritz, Bob Price, Roger Hoover, and Johan declarer, “Throughput Scheduler: Learning to Schedule on Heterogeneous Hadoop Clusters”, 10th International Conference on Autonomic Computing ICAC, 2013

[4]Weikuan Yu, Yandong Wang and XinyuQue, “Design and evaluation of network-levitated merge for Hadoop Acceleration”, IEEE Transactions on Parallel and Distributed Systems, Vol5, Issue (3), pp: 602-611, 2014

[5]R. Manopriya and C.P.Saranya, “Optimal Resource Allocation and Job Scheduling to Minimize the Computation Time under Hadoop Environment”, International journal of latest trends in engineering and technology (IJLTET),Vol 3,issue (3),pp:75-82, 2014

[6] Nidhi Tiwari, Santonu Sarkar, Umesh Bellur, and Maria Indrawn ,"An empirical study of hadoop's energy efficiency on a HPC cluster",14th International Conference on Computational Science,pp:62-72, 2014

[7]Kaewkasi Chanwit, and Wichai Srisuruk, "A study of big data processing constraints on a low-power Hadoop cluster", IEEE International Conference on Computer ,Science and Engineering (ICSEC), 2014

[8]Maria Dania and Naomi Osato, "A Comparative Study of Scalable Implementations of the Alternating Least Squares Algorithm for Collaborative Filtering Recommendation", 2015

[9]Pati Supriya and Mayuri A Mehta, “Job Aware Scheduling in Hadoop For Heterogeneous Cluster”, IEEE International Advance Computing Conference (IACC), pp: 778-783, 2015

[10]Gajanan Babura Aochar and Prof.Omprakash Tambourine, “An Approach to Combinatorial Problems by Mapreduce Based Ant Colony Optimization”, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Vol. 4,Issue(1),pp:1009-1014,2014

[11]V.Selvi, Dr.R.Umarani, “Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques”, International Journal of Computer Applications Vol. 5, Issue(4),pp.: 0975 – 8887,August 2010

[12]“WorkflowGenerator.”[Online].Available: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator