International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August 2013
ISSN 2229-5518

Self Energy Optimizing Cluster Framework for Map-Join-Reduce Applications

Vinisha Sasidharan, P. Mohamed Shameem, Dr. R .S Shaji

Abstract— Cloud computing is emerging as a new paradigm of large-scale distributed computing. Cloud computing provides customers/users scalability, flexibility for their applications by providing collection of interconnected and virtualized servers. But as the requirement for cloud increases, so increase the consumption of huge amounts of electrical energy, contributing to high operational costs Green House Gas emission and carbon footprints to the environment. Therefore, we need a Green Cloud computing solutions that can minimize energy consumption and thereby operational cost. By Green Computing, it means reducing the environmental impact. This paper proposes an algorithm for achieving energy efficiency in cloud that runs Map-Join-Reduce applications for processing large amount of data.

Index Terms— Cloud Computing, Energy Efficiency, Green Computing, Map Reduce, Map-Join-Reduce, Virtualization, Scheduling.

1 Introduction

Cloud computing [1] is emerging as a new paradigm of large scale distributed computing. By using cloud, users don't need to reside on their workstations as it has moved data and processing away from desktop and PCs into large networked data centers. Most of the firms are now moving towards cloud computing for their business requirements. It is a framework for providing convenient, on-demand network access to a shared pool of computing resources. It provides users with scalability on demand and flexibility to meet their business changes based on a pay-per-use basis there by meeting the operational cost. Thus, cloud computing is a framework for providing a suitable, on-demand network access to a shared pool of computing resources (e.g. networks, servers, storage, applications, and services). Due to the exponential growth of cloud computing, it has been widely adopted by the industry and that results in a rapid expansion in data-centers. This expansion has caused the dramatic increase in energy use and environmental impact in terms of Green House Gas emission and carbon footprints. The link between energy consumption and carbon emission has given rise to an energy management issue which is to improve energy-efficiency in cloud computing to achieve Green computing [2].

Energy Efficiency is one of the critical issues in cloud computing [3]. To achieve energy efficiency in cloud environment, tasks running on nodes need to be scheduled efficiently. There are several resources need to be managed (CPU load, network bandwidth, memory, disk etc.). The increasing accumulation of greenhouse gases is changing the world's climate, creating serious problems such as droughts, floods and higher temperatures. In order to stop the accumulation of these gases in the atmosphere, it is necessary to stop the global growth of emissions, in which the generation of electricity plays a major role not only because of the carbon dioxide which results from the coal and oil used in this process, but also because it releases sulphurs and other pollutants into the atmosphere.

According to Amazon's [4] estimates, at its data centres (as illustrated in figure 1), expenses related to the cost and operation of the servers account for 53% of the total budget, while energy-related costs amount to 42% of the total, and include both direct power consumption (~19%) and the cooling infrastructure (23%) amortized over a 15-year period.

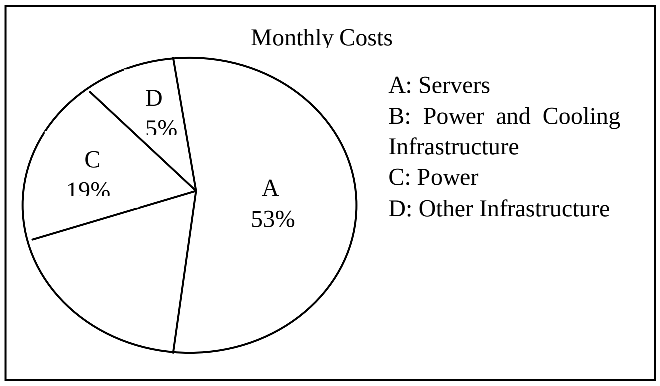

Fig 1: Energy distribution in the data centre.

Energy efficiency is increasingly important due to the increasing energy costs and the need to reduce greenhouse gas emissions and also to decrease the overall energy consumption, storage and communications.

Servers are unfriendly to environment [5] and IT industry contributes to 2% of worlds total CO2 emissions (eg: 2.8 tons from US power plants). A typical datacentre consumes as much as energy as 25000 households. Servers consume 0.5% of the world's total electricity usage. More than 15% [6] of the servers are running without being used actively. So we need energy efficient resource management methods that minimize energy consumption at the same time meet the job deadlines.

The energy consumption for computing could be divided according its use in two edges [7], the first regarding to the energy consumed by the clients conformed by PCs, peripherals and all types of mobile devices and the second refers to the energy consumed by servers, networks and cooling sys-

- *Vinisha Sasidharan is currently pursuing masters degree program in software engineering in Cochin University, Kerala, India, PH-9447797662. E-mail: vinisha.sasidharan@gmail.com*
- *P. Mohamed Shameem is working as an Associate Professor in TKM Institute of Technology, Kollam, India. Email: pms.tkmit@yahoo.in*
- *Dr. R.S Shaji is working as a Professor in Noorul Islam Centre for Higher Education, Nagarcoil, India. Email: shajiswaram@yahoo.com*

tems in data centers. Energy consumption in data center edge is increasing due to the increased number of users, storage and the need to maintain QoS.

Most datacenters today run large scale data intensive applications that store and process huge amount of data. Map Reduce [17] framework allows programmers without any prior experience with parallel and distributed systems to easily utilize the resources of a large distributed system and is hence adopted by a huge number of programmers and organizations worldwide. But the performance of Map Reduce is slower when it is adopted to perform complex data analysis tasks that require the joining of multiple data sets in order to compute certain aggregates. So a new algorithm, Map-Join-Reduce, a system that extends and improves Map Reduce framework to efficiently process complex data analysis tasks on large clusters is introduced.

This paper proposes an algorithm for achieving energy efficiency in cloud environment that runs Map-Join-Reduce applications. Algorithm configures the cluster based on the current utilization of cluster nodes and rebalances the cluster nodes. A Self Energy Optimizing technique is used in which, input data is split based on average throughput of each nodes.

The remainder of this paper is organized as follows. Section 2 describes the related works done in managing the energy in desktop and cloud environment. Section 3 describes proposed work for achieving energy efficiency in cloud. Section 4 describes experimental result. Section 5 describes conclusion.

2 RELATED WORKS

Different approaches for energy efficiency are

- Energy Efficient Hardware [4]
- Virtualization [8]
- Dynamic Voltage and Frequency Scaling [9]
- Energy-aware job Scheduling
- Request Batching [10]
- Multi-speed Disks [11]
- Server Consolidation [12]

Most local techniques try to reduce the energy consumption at a component level, like cpu, network, disk, storage, etc. by means such as reducing the clock frequency [9], voltage, disk speed or by saving on network interconnect components.

Energy savings for servers have received special attention at disk level. Several methods such as request batching [6], using both high performance and low performance disks, replacing high performance disk by several low performance disks, multi-speed disks [10] for servers are the various methods proposed for optimizing energy efficiency of disks. One method [13] used is applying a machine learning based technique to find the time at which a disk need to be spin down in order to the extend battery life of a mobile device. This means spinning down the disk when not in use. Unfortunately, network server disks have extremely short idle times and this method focus on energy reduction at single component (disk).

Dynamic Voltage Scaling (DVS) scheduling algorithms minimizes energy/ power consumption by controlling appropriate voltage levels. But the drawback is that it cannot be applied to real time applications.

Another method for reducing the energy used by the cpu is by using a new metric for cpu energy performance, millions-of-instructions-per joule (MIPJ) [14]. This is done by dynamic control of system clock by the operating system scheduler, but reducing the clock speed alone does not reduce the energy consumed by the cpu, since to do the same amount of work the system has to run longer.

Another technique is using virtualization. By using this method, physical servers can be shared. Most of the tasks do not use resources ideally. Only 5% of servers are needed by tasks. So rather than running tasks in several machines, these can be virtualized. But virtualized servers can lead to uncontrolled growth and more unused servers, a phenomenon called virtual server sprawl.

Rather than concentrating on a single system component for energy efficiency, cluster-level techniques were developed. The basic idea of a cluster-wide technique is to aggregate the system load and then determine the minimal set of servers which could handle the load. A number of tools and frameworks have been developed to process data at this scale and Map Reduce has emerged as the most prominent paradigm among them. It was first developed at Google to process web scale data on inexpensive commodity hardware. Methods for dynamically reconfiguring [15], [16] the cluster were developed. The algorithm dynamically adapts to the changes in the utilization levels of the machines in the cluster and reconfigures the nodes in the cluster in accordance with the workload experienced.

3. PROPOSED SYSTEM

This paper presents energy efficient algorithm for data placement and cluster reconfiguration, designed to reduce the energy consumption of datacenters running Map-Join-Reduce jobs. The proposed algorithm dynamically reconfigures the nodes in the cluster in accordance with the workload experienced.

3.1 Approach

In this paper, Master-Slave architecture is implemented that uses Map-Join-Reduce cluster framework for processing large data sets. Master node is Name nodes (Cloud Controller) and Slaves are Data nodes.

Each DataNode periodically sends a heartbeat message to the NameNode in order to inform the NameNode about its current state including the block report of the blocks it stores. The NameNode assumes DataNodes without recent heartbeat messages to be dead and stops forwarding any more requests to them. Cloud users submit their Map-Join-Reduce job and input file (that needs to be processed) to Cloud Controller.

A channel is introduced between Cloud Controller and Power Controller that scales the number of nodes in the cluster based on the current workload of cluster nodes. For eg, if cloud controller receives a request to find the sum of non-prime numbers in a file, it checks with power controller whether enough nodes available to handle this task. If no, power controller replies cloud controller with reconfiguration details.

3.2 Map-Join-Reduce

Map Reduce is a parallel programming model and an associated implementation introduced by Google. It is an attractive model for parallel data processing in high performance cluster computing environments. The scalability of Map Reduce is proven to be high, because a job in the Map Reduce model is partitioned into numerous small tasks running on multiple machines in a large-scale cluster. Map Reduce is a widely used method of parallel computation on massive data.

Map Reduce has two functions: Map and Reduce. Map takes an input pair and produces a set of intermediate key/value pairs. The Map Reduce library groups together all intermediate values associated with the same intermediate key and passes them to the Reduce function. The Reduce function accepts an intermediate key and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically just zero or one output value is produced per Reduce invocation.

But the performance of MapReduce is slower when it is adopted to perform complex data analysis tasks that require the joining of multiple data sets in order to compute certain aggregates. So in this paper, a new method called Map-Join-Reduce is used. In addition to map() and reduce() functions, a third join() function, i.e., joiner is added. To execute a Map-Join-Reduce job, the runtime system launches two kinds of processes: called MapTask, and ReduceTask. Mappers run inside the MapTask process while joiners and reducers are run inside the ReduceTask process.

In the first MapReduce job, the system splits the input data sets into key value pair and then invokes a set of MapTasks. Each MapTask executes a corresponding map function on each key value pair and writes intermediate key value pairs to local disk. When the MapTasks are completed, the system invokes a set of ReduceTasks. Each reduce task has joiner and reducer functions. Then, each ReduceTasks remotely reads (shuffles) partitions associated to it from all mappers. When a partition is successfully read, the ReduceTask checks whether the first joiner is ready to perform. A joiner is ready if and only if both its first and second input data sets are ready, either in memory or on local disk. When the joiner is ready, ReduceTask performs a join algorithm on its input data sets. The output of the final joiner is then fed to the reducer for partial aggregation.

The signatures of map, join, and reduce functions are as follows:

$\text{Map}_i(k1_i, v1_i) \rightarrow (k2_i, \text{list}(v2_i))$,

$\text{Join}_j((k2_{j-1}, \text{list}(v2_{j-1})), (k2_j, \text{list}(v2_j))) \rightarrow (k2_{j+1}, \text{list}(v2_{j+1}))$,

$\text{Reduce}(k2, \text{list}(v2)) \rightarrow \text{list}(v2)$

3.3 Proposed Algorithm

When Cloud Controller gets input file and Map-Join-Recue job from user, it splits input file based on average throughtput of cluster nodes which is backed up in databse. Input split is done based (1):

$$In*(Tav)node/Ttot \quad (1)$$

where In is the total input data,
(Tav)node is the average throughput of node,
Ttot is the total throughput.
(Tav)node = (Lavg)node/(tavg)node
Ttot = ∑(Tav)node(i)
(Lavg)node = Average Load of each node
(tavg)node = Average time taken by node to complete processing

Cloud Controller groups nodes as Mapper, Joiner and Reduce and assigns input sub-files and data job to Mapper along with task_id.

Upon receiving input key value pair, Mapper performs map function on input sub-files. Each node sends back job status to Cloud Controller. Cloud Controller checks whether all Mappers completed their map tasks by checking return status of mappers. Once a map task becomes completed, it sends back cloud controller the intermediate key-value pair.

Cloud Controller invokes Joiners for performing join function. Once join task is complete it invokes Reducers that perform reduce function on data and return result to Cloud Controller. Processed output can be obtained from Cloud Controller.

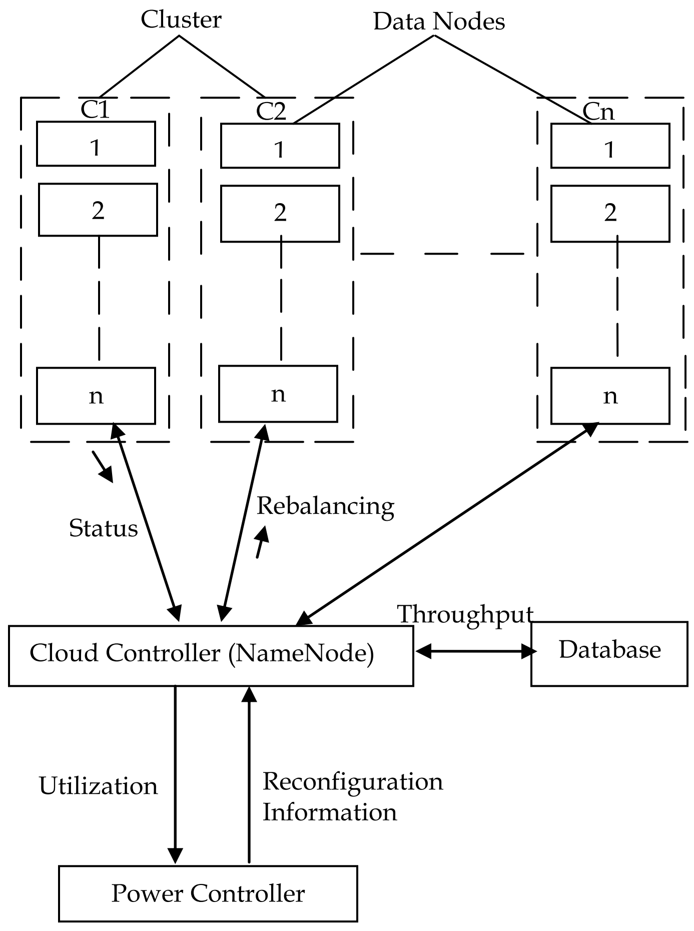

Fig 2: System Architecture

Every time when it receives return status from nodes, Cloud Controller checks whether node is underutilized or over utilized. It checks current utilization of each node with maximum and minimum threshold set by administrator. If nodes utilization is between maximum and minimum threshold, then nothing is done. Else cluster reconfiguration algorithm is user. If it is greater than maximum threshold, extra load needs to be moved and Cloud Controller checks whether other nodes in same cluster can handle extra task execution. If so, it selects best node from the same cluster and performs intra-cluster movement. Else it checks for nodes in other clusters and select best nodes from that cluster and performs inter-cluster movement. During intra-cluster movement, additional load is moved to other nodes in the same cluster. During inter-cluster additional load is moved to other nodes in different cluster.

Nodes in the same cluster are connected by a higher bandwidth link and hence Intra-Cluster transfers are cheaper as compared to Inter-Cluster transfers. Hence intra-cluster is preferred. If no such node is found, data is not transferred.

If node's utilization is less than minimum threshold, full load needs to be moved and the node is kept idle. Cloud Controller selects best node to move load using cluster reconfiguration algorithm and performs intra-cluster or inter-cluster movement.

Important challenge in a cloud environment is splitting load among nodes. Splitting is done based on the backed up data (average throughput value, average load/average time of each node). A node starts execution only after allocation of jobs to nodes.

4. EXPERIMENTAL RESULTS

The system that has been developed is intended to perform energy efficiency and reduce the time to process complex data tasks. Entire Cloud environment with Master Slave configuration using Map-Join-Reduce framework is implemented using Java (Net Beans 7.1).

In the first phase of the research work, simple Map Reduce Algorithm is used for processing input data. For improving the result and utilizing cloud more efficiently, new algorithm Map-Join-Reduce and self-energy optimizing technique is used. The performance comparison of two techniques reveals that Map-Join-Reduce with self-energy optimizing technique are more energy efficient.

Table below shows the performance of system towards processing different inputs. Here job is written to calculate sum of even numbers in input data file.

Different input data are given to the system. Performance graph is drawn based on the input given and time taken to complete processing. Figure 3 shows performance evaluation done using input data of length 4000 with various nodes. X-axis denotes number of nodes and Y axis denotes time taken (seconds) by the system to complete processing.

Performance graph shows that using Map-Join-Reduce Algorithm with self energy optimizing technique improves system performance by improving energy efficiency.

TABLE 1
TIME TAKEN BY SYSTEM FOR PROCESSING INPUT OF VARYING SIZE

		Time in seconds	
Input	No. of Nodes	Map Reduce	Map-Join-Reduce & Self Energy Optimizing
4000 numbers	5	6s	3s
	7	7s	3s
	9	8s	2s
	10	10s	2s
8000 numbers	5	10s	5s
	7	9s	4s
	9	8s	3s
	10	8s	3s

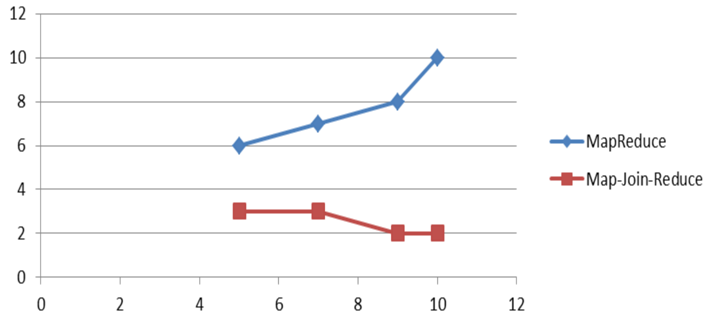

Fig 3: Performance graph for input data of length 4000

5. CONCLUSION

Most of the firms are moving to cloud environment now a days. Moving to cloud is clearly a better alternative as they can add resources based on the traffic according to a pay-per-use model. But for cloud computing to be efficient, the individual servers that make up the datacenter cloud will need to be used optimally. Even an idle server consumes about half its maximum power. With the emergence of cloud computing in the past few years, Map Reduce has seen tremendous growth especially for large-scale data intensive computing. The lack of a separate power controller in Map Reduce frameworks post an interesting area of research to work on. This paper addresses the issue of power conservation for clusters of nodes that run Map-Join-Reduce jobs. So an energy efficient cluster reconfiguration algorithm is proposed that dynamically reconfigures the cluster in accordance with the workload imposed on it. Self-Energy Optimizing technique is used which uses throughput history maintained in a database during workload distribution.

In future, applying machine learning approaches to study the resource usage patterns of jobs (CPU, memory) and model their power usage characteristics to efficiently schedule them on the cluster.

REFERENCES

[1] Alexa Huth and James Cebula, “The Basics of Cloud Computing”, US-CERT, 2011.

[2] Jayant Baliga, Robert W. A. Ayre, Kerry Hinton, and Rodney S. Tucker, Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport, IEEE 2010.

[3] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, ELSEVIER 2009.

[4] Andreas Berl, Erol Gelenbe, Marco di Girolamo, Giovanni Giuliani, Hermann de Meer, Minh Quan Dang and Kostas Pentikousis, Energy-Efficient Cloud Computing, Computer Journal 2010.

[5] Luiz André Barroso and Urs Hölzle, The Case for Energy Proportional Computing, IEEE 2007.

[6] Server Energy and Efficiency Report. Technical report, 1E, 2009.

[7] Ismael Solis Moreno, Jie Xu, Energy-Efficiency in Cloud Computing Environments: Towards Energy Savings without Performance Degradation.

[8] F. Tusa, M. Paone, M. Villari and A. Puliafito, CLEVER: A CLoud-Enabled Virtual EnviRonment IEEE 2010.

[9] G. von Laszewski, L. Wang, A. Younge, X. He, Power-aware scheduling of virtual machines in DVFS-enabled clusters, IEEE 2009.

[10] Enrique V. Carrera, Eduardo Pinheiro, and Ricardo Bianchini, Conserving disk energy in network servers, ACM.

[11] Sudhanva Gurumurthi, Anand Sivasubramaniam, Mahmut Kandemir, and Hubertus Franke, Drpm: Dynamic speed control for power management in server class disks, in: Computer Architecture, International Symposium 2003.

[12] Ching-Hsien Hsu, Shih-Chang Chen2, Chih-Chun Lee, Hsi-Ya Chang, Kuan-Chou Lai, Kuan-Ching Li and Chunming Rong, Energy-Aware Task Consolidation Technique for Cloud Computing, IEEE 2011.

[13] David P. Helmbold, Darrell D.E. Long, Tracey L. Sconyers and Bruce Sherrodm, Adaptive Spin Down For Mobile Computers, Journal on Mobile Network and Applications 2000.

[14] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya, Energy-Aware Resource Allocation Heuristics For Efficient Management Of Data Centers For Cloud Computing, ELSEVIER 2012.

[15] Eduardo Pinheiro, Ricardo Bianchini, Enrique V. Carrera, and Taliver Heath, Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems.

[16] Nitesh Maheshwari, Radheshyam Nanduri and Vasudeva Varma, Dynamic Energy Efficient Data Placement and Cluster Reconfiguration Algorithm for MapReduce Framework, ELSEVIER 2012

[17] Jeffrey Dean and Sanjay Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Magazine, Communications of the ACM, 2008.