IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 4, Ver. VII (Jul.-Aug. 2017), PP 06-11
www.iosrjournals.org

Dynamic Resource Allocation Scheme for Map Reduce Tool in Cloud Environment

Dr. M. Mohamed Surputheen[1], *M.Abdullah[2]
[1]*Associate Professor Department of Computer Science, Jamal Mohamed College,Trichy, Tamil Nadu, India*
[2]*Assistant Professor Department of Computer Science, Jamal Mohamed College,Trichy, Tamil Nadu, India*
*Corresponding Author: *M. Abdullah*

Abstract:*Most scientific data consists of analyzing huge amount of data collected from different resources. Hence, parallel algorithms and frameworks are the important which can process huge volumes of data and meeting the requirements of performance and scalability entailed in such scientific data analyses. In this paper, we proposed a concurrent VM reconfiguration mechanism for big data tool which is MapReduce on virtualized cloud environments. Our reconfiguration enhances the input data locality of a virtual MapReduce cluster. It adds cores to VMs to run local tasks temporarily by scheduling tasks based on data locality, and adjust the computational capability of the virtual nodes to contain the scheduled tasks unlike the traditional schemes which can leads to user-friendly configuration methods for cloud resources.*

Keywords:*Cloud Computing, Big Data, Map Reduce, Virtualization*.

Date of Submission: 14-08-2017 Date of acceptance: 26-08-2017

I. Introduction

Cloud computing comprises of resources and services produced through the Internet. The buzzword Cloud computing is considered as a new computing paradigm which can provide customized ,flexible , consistent, QoS guaranteed dynamic computing environments and its underlying capability to provide adjustable dynamic IT infrastructures and configurable software services. Grid computing has been replaced by cloud computing because of its system virtualization. System virtualization is the backbone of Cloud computing, has been liberalizing its services to distributed data-intensive platforms such as MapReduce and Hadoop. Cloud computing empowers consumers to access online resources using the internet, from anywhere at any time without considering the underlying hardware, technical management and maintenance problems of the original resources. Cloud services are obtained from data centres which are distributed throughout the world. Virtual resources are provided to clients through internet in cloud computing such as Gmail, provided by Google. The rapid growth in cloud computing has led to numerous advantages but at the same time it possess lack of security concerns which has been a major challenge. There is a growing set of large-scale scientific applications which are loosely-coupled in nature comprises many small jobs/tasks with much shorter durations and also encompass large volumes of data where these applications include those from data analytics, bioinformatics, data mining, astronomy, astrophysics, and MPI ensembles [1]. Cloud computing suffers from security issues like securing confidential data, cloud storage and examining the utilization of cloud by the cloud computing vendors and key questions arises like how safe is data in cloud ? .Besides, cloud computing has numerous advantages like scalable and dynamic. Cloud computing is independent computing and varies from utility computing and grid computing. Technologies involved in Cloud computing are still developing and evolving for example, Service Oriented Computing. Cloud computing environment is still lacking security features and large scale deployment and usage, which would rationalize the concept of Cloud computing and there exists no widely accepted definition. The cloud computing is divided into three forms [2].

A. Public

In a public cloud, the infrastructure and service are provided off-site over the internet. A public cloud accommodates storage services and application to the users over the internet.

B. Private

Private cloud is deployed for a single organization or by a third party externally. They consists high advanced security fault tolerant solutions. The private cloud is mainly employed for single organization. Comparing with other cloud models, private clouds provides power as a service within a virtualized environment employing the features of cloud computing resource. The private cloud can be used by a single organisation providing that organisation which guarantees privacy and control. From technical aspect, the underlying mechanisms are hard to define. The advantages of private clouds are more control, higher security and privacy, cost and efficiency.

C. Hybrid

It comprises both private and public cloud services with multiple providers. Cloud computing has been expanding its services to data-intensive computing on distributed platforms such as MapReduce [3], Dryad [4], and Hadoop [5]. In cloud-distributed platforms, virtualization takes place on the physical machines, and hence a virtual cluster is formed due to large collection of virtual machines. Data-intensive platform works on the virtual cluster unlike the traditional physical cluster. Such a virtual cluster provides adjustable environment, which can move up and down according to the changes in computation demands from different users. Cloud provider's combines virtual clusters from various users into a physical data centre, to increase the utilization of resources. In the virtual cluster, the need for computing resources for each node may change fluctuates, because of location of data and behaviour of task. However, a static cluster configuration is employed by current cloud services, manually adjusting the computing capability of each virtual machine deteriorate. The fixed homogeneous VM configuration may not react to varying demands in resources encountered in individual nodes. Hence we propose dynamic VM reconfiguration for mapreduce on cloud. It can adjust the computing capability of individual virtual machines to increase the consumption of resources and focuses on data-locality. Our proposed technique reacts and changes according to the computing capability of each node in order to improve locality-aware task scheduling. The proposed technique enhances computing resource of a VM, if the next task has its data on the VM. Simultaneously, an inactive virtual CPU is removed from another VM in the same virtual cluster, so that the size of the virtual cluster remains constant to provide a constant cost for the user. This dynamic reconfiguration of each VM improves the overall job throughput by improving data locality, while the total virtual CPUs in the cluster remain unchanged.

II. Related Work

Delay Scheduling presented locality-aware scheduling policies to improve data locality for Hadoop platforms [6]. A Resource Aware Scheduling Algorithm which enhances existing task scheduling algorithms, Min-min and Max-min, is described in [7]. In [8] an algorithm having good results on the compromise cost-execution time is presented. The tests showed that the cost decrease with over 15% while the execution time satisfies users' requirements or the execution time may be shorter with average 20% and the costs would remain almost the same. A heuristic genetic approach is described in [9] [10], with enhancement of execution time. Charm++ [11] is a machine independent parallel programming system, in which, load balancing can be performed in either a centralized hierarchical or distributed. Purlieus improved the locality of map and reduce tasks in MapReduce platforms on the cloud by locality-aware VM placement [12].

III. MapReduce

In this section, we provide a brief introduction of the MapReduce technique. Most scientific data consists of analysing huge amount of data collected from different resources. Hence, parallel algorithms and frameworks are the important which can process huge volumes of data and meeting the requirements of performance and scalability entailed in such scientific data analyses. One such popular and efficient framework is MapReduce which has acquired a lot of attention from the scientific community for its applicability in large parallel data analyses. Even though there are many assessments of the MapReduce tool employing huge amount of textual data collections, only a few evaluations for scientific data analyses have been made. The authors [13] explain the MapReduce programming model as: The calculation takes a group of input key/value pairs, and gives a group of output key/value pairs. The user of the MapReduce library states the computation as two functions: Map and Reduce. x Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library clusters intermediate values associated with the same intermediate key and passes them to the Reduce function. The Reduce function, also written by the user, 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.

Figure 1: MapReduce on virtualized environments

MapReduce with Hadoop implementation employs a distributed file system, known as Hadoop Distributed File System (HDFS) which stores data and also intermediate results. HDFS maps all the local disks to a single file system hierarchy enabling the data to be spread at all the computing nodes. Data locality possesses a serious challenge and affects the throughput of Hadoop job and they can be overcome by enhancing the locality by assigning tasks to the nodes with the corresponding data [14]. However, to improve locality, when a task is scheduled, the computing node with the corresponding data must have available computing slots to process the task. If a computing slot is not available, the task must be scheduled to a remote node, which must transfer necessary data from another node for the task. Whenever there is huge amount of data, tightly coupled parallel applications can encounters drawbacks. But MapReduce framework since its relaxed synchronization constraints does not impose much of an overhead for large data analysis tasks. And robustness and simplicity of the programming model can be regarded as major advantages. There are various schemes such as writing the data to files after a certain number of iterations or using redundant reduce tasks may remove this overhead and give a better performance for the applications. In order to react to varying demands on each VM in a virtual cluster, each VM may be dynamically reconfigured. Virtualization employs hot-plugging support such a dynamic reconfiguration of each virtual machine. As long as physical resources are available, each virtual machine is assigned with more virtual CPUs and memory while the virtual machine is running. However, the currently available cloud services, such a dynamic reconfiguration of VM is not available.

IV. VM Reconfiguration

In virtualization technology, "A virtual machine (VM) is an abstraction layer or the environment between hardware components and the end-user [15], virtualization is the key component which enables cloud computing. Virtualization is one of the technologies with the ability of providing abstraction which is an important feature of cloud computing, abstraction performs the actions and behaviour of the actual computer. Physical machine which consists of single software has multiple instances of virtual machines and it is known as Hypervisor. VMs can be transferred between physical hosts and it can be restarted by employing live-migration. This feature empowers an IP and it can be re-allocates the existing VM load across obtainable server resources during runtime. The contextualization procedure is not activated since the VM is not renewed during the migration. Therefore, any VM customization will stay unmoved. In order to configure virtual clusters, homogeneous static configuration of virtual machines is employed by existing or private or public clouds. A virtual cluster can be created by selecting a type of virtual machines along with a static number of virtual cores and a fixed memory size, and detects the number of virtual machines, by taking into account of the total cost of employs the virtual cluster. Different jobs may have various resource usages which lead to heterogeneous problem such as computation, storage and network load. There is always a limitation of resource size of a virtual cluster for each user and can be detected by varying resource demands for the VM. However, virtualization consists of major advantage to reconfigure virtual machines establishing a virtual cluster. Multiple virtual clusters share a set of physical machines, and within a physical system, multiple virtual machines for the same or different users co-exist.

V. Proposed Scheme

A static configuration consists of a major drawback where the resources needed for each virtual machine may change during the life time of the virtual cluster. A virtual machine may require more CPU resources while another virtual machine needs more network bandwidth and memory. Such a dynamic imbalance of resources in individual virtual machines, leads to the total inefficiency of cluster resources. The data locality problem happens in MapReduce platforms since input data are distributed in computing nodes. Inorder to enhance data locality, the master scheduler tries to allocate map task, which processes the data stored in the node with an existing slot. However, it is not always possible to find such a local task to the node. As reduce tasks commonly receive data equally from all nodes, the data locality is less critical for reduce tasks than for map tasks. In order to implement the proposed, cluster-level pricing options must be provided by the cloud providers for users so that the users can select VM configuration and also the number of VMs. Our proposed scheme consists of two important components, namely, Reconfiguration Controller (RC) and Resource Manager (RM). The Reconfiguration Controller component dispatches deallocation and allocation requests to the hypervisor based on the physical machine. The RC monitors resource and chooses virtual machine residing the cluster which demands resource reconfiguration, then it dispatches an allocation or de-allocation request to Resource Manager RM running in the physical system, where the VM to be reconfigured is running on. Upon obtaining requests from RC, RM re-assigns the demanded virtual CPUs to the VM. RM demands hypervisor of the physical machine, to hot-plug or to un-plug virtual CPUs for VMs running on the system. We also present the implementations of concurrent dynamic reconfiguration (DR), here we propose a concurrent dynamic reconfiguration implementation based on the Hadoop scheduler for fairness [14]. The concurrent DR assigns task to a virtual machine only when the physical system running the VM has an available core to allocate for the VM. Such synchronous reconfiguration requires free CPU resources in each system on the cloud. A task can be queued to a VM, even if the VM does not have an available virtual core immediately. According to the algorithm, whenever a VM consists of free slot, the scheduler selects a job for fairness, and tries to selects a local task for the VM. If no local task is available for the VM, it selects any task and finds the VM with the input data of the selected task. The scheduler in RC sends a CPU allocation request to the acquired virtual machine and core resource is transmitted from the source VM to the acquired VM.

A. Algorithm: Concurrent Dynamic Reconfiguration

Step 1: When a node x receives heartbeat:
Step 2: if x has vacant slot then,
Step 3: Arrange jobs in increasing order of the Number of executing tasks
Step 4: for i in jobs do
Step 5: if i has unlaunched task d with data on x
then launch d on x
    else if i has unlaunched task d
Step 6: then find node set e storing data of d
    to pick a node z from e
Step 7: Resource allocation request is sent
to RM
Step 8: Resource de-allocation request is
sent to RM
Step 9: Launch d on z
Step 10: end if
Step 11: end for
Step 12: end if

If the desired target does not contain any obtainable CPU resource, attaching a virtual core to the VM will decrease the CPU shares allocated virtual cores of the other VMs or same VM. Upon selection of VM, a CPU allocation request will be sent by RC to the target. The RM demands hypervisors to remove or add virtual cores. If no free core is available in the target system, the targeted task will be scheduled to the initial source VM.

VI. Experimental Methodology and Results

We evaluate our techniques on measuring the effect of data locality on MapReduce cloud service. A physical machine included in the 4-node private cluster has an AMD Phenom 6-core processor, 16GB memory. We allocate 4 virtual machines for each physical machine. Each virtual machine has 2 virtual cores, and 2GB memory. All of the physical machines are connected with a network switch, and we ran experiments on both 100Mbps and 1Gbps switches.For the test cases, we regarded tasks and resources with different demands, as support for different characteristics of workloads. We analyse the average execution time of tasks along a combination of scenarios using a certain configuration. The workloads employed in the cluster are described in the Table -1

Map	No Nodes	No Edges	CCR_m	CCR_s	Granularity	Type of workloads
1	616	18635	0.250	0.13	3.957	Invert
2	628	19491	0.245	0.10	4.000	Aggregation
3	681	22505	0.250	0.12	3.958	Join
4	686	22817	0.998	0.99	0.991	Inverted Index
5	688	22954	0.998	0.99	0.998	Aggregation
6	707	24118	1.007	1.00	0.988	Select
7	667	21637	3.654	6.60	0.029	Join
8	631	19539	3.640	6.61	0.031	Aggregation
9	627	19202	3.769	6.78	0.029	Select

Table -1 Workload Cluster

The workloads are from the Hive performance benchmark [16]. The workloads Select is I/O intensive workloads, while Join and aggregation are communication-intensive workloads whereas Inverted Index is regarded as highly communication-intensive workloads [17].The private cluster, has different map and reduce slot configurations considering their core resources. A VM in the cluster has 5 map and 3 reduce slots while a VM in the private cluster has 2 maps and 1 reduce slots.

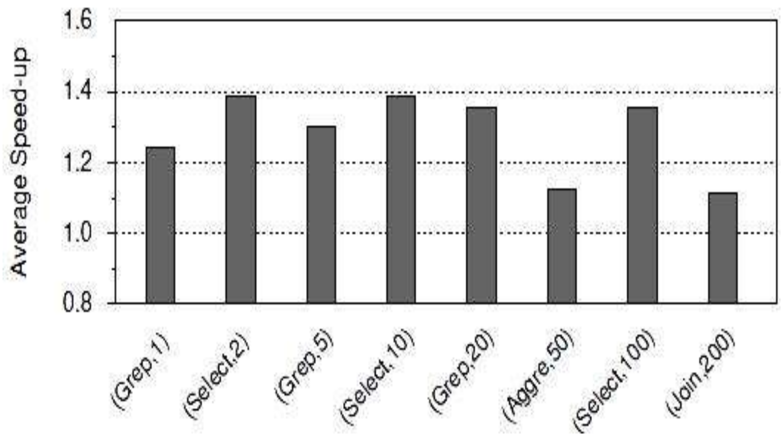

Figure -2: Rapid movement of resources using Dynamic Reconfiguration.

There is some variation of performance and from the figure - 2; pragmatic configuration cannot always impart supplementary additional cores right away, unlike the pseudo-ideal runs. The performance is increased with a significant improvement of 36% on average and performance improvement of 10% in the Hadoop.

VII. Conclusion and Future Work

System virtualization is the backbone of Cloud computing, has been liberalizing its services to distributed data-intensive platforms such as MapReduce and Hadoop. In this paper, we proposed a concurrent VM reconfiguration mechanism for big data tool which is mapreduce on virtualized cloud environments. Our reconfiguration enhances the input data locality of a virtual MapReduce cluster. It adds cores to VMs to run local tasks temporarily by scheduling tasks based on data locality, and adjust the computational capability of the virtual nodes to contain the scheduled tasks unlike the traditional schemes which can leads to user-friendly configuration methods for cloud resources. For future work we will consider, create a framework to build relationships between datasets.

References

[1] I. Raicu, I. Foster, Y. Zhao. "Many-Task Computing for Grids and Supercomputers", Invited Paper, IEEE MTAGS 2008.

[2] R.Udendhran, "New Framework to Detect and Prevent Denial of Service Attack in Cloud Computing Environment" Asian Journal of Computer Science And Information Technology 4 : 12 (2014) 87 - 91 ISSN 2249-5126.

[3] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. In Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pages 137–150, 2004.

[4] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: distributed data-parallel programs fromsequential building blocks. In Proceedings of the 2ndEuropean Conference on Coputer Systems (EuroSys), 2007.

[5] Apache Hadoop. http://hadoop.apache.org.

[6] M. Zaharia, D. Borthakur, J. Sen Sarma,K. Elmeleegy, S. Shenker, and I. Stoica. Delayscheduling: a simple technique for achieving localityand fairness in cluster scheduling. In Proceedings ofthe 5th European Conference on Computer systems (EuroSys), 2010.

[7] S. Parsa, R. Entezari-Maleki, RASA: A new task scheduling algorithm in grid environment, World Applied sciences journal.

[8] K. Liu, H. Jin, J. Chen, X. Liu, D. Yuan, Y. Yang, A Compromised-Time-Cost Scheduling Algorithm in SwinDeWC for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform, Int. J. High Perform. Computer Applications.

[9] K. Kaur, A. Chhabra, G. Singh, Heuristics based genetic algorithm for scheduling static tasks in homogeneousparallel system, Int. J. of Comp. Sci. and Sec.

[10] J. Kolodziej, F. Xhafa, Enhancing the genetic-based scheduling in computational grids by a structured hierarchical, P. Sempolinski, D. Thain, A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus, in: Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science, CLOUDCOM '10, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-4302-4, 417–426, doi: 10.1109/CloudCom.2010.42,

URLhttp://dx.doi.org/10.1109/CloudCom.2010.42, 2010.population, Future Generation Computer Systems 27 (8) (2011) 1035–1046, ISSN 0167-739X, doi:10.1016/j.future.2011.04.011, URL http://dx.doi.org/10.1016/j.future.2011.04.011.

[11] B. Palanisamy, A. Singh, L. Liu, and B. Jain. Purlieus: locality-aware resource allocation for MapReduce in a cloud. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2011.

[12] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," ACM Commun., vol. 51, Jan. 2008, pp. 107-113.

[13] M. Zaharia, D. Borthakur, J. Sen Sarma,K. Elmeleegy, S. Shenker, and I. Stoica. Delayscheduling: a simple technique for achieving locality and fairness in cluster scheduling. In Proceedings of the 5th European Conference on Computer systems (EuroSys), 2010.

[14] D. Kreuter, "Where server virtualization was born", Virtual Strategy Magazine, July 2004

[15] Hive Performance Benchmarks.

[16] https://issues.apache.org/jira/browse/HIVE-396.

[17] S. L. Faraz Ahmad and T. V. Mithuna Thottethodi MapReduce with communication overlap (marco). http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1412&context=ecetr, 2007.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Dr. M. Mohamed Surputheen. "Dynamic Resource Allocation Scheme for MapReduce Tool in Cloud Environment." IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 4, 2017, pp. 06–11.