

Enhanced Dynamic Data Placement and Virtual Machine Creation for MapReduce

X.Z. Lu, K.K. Phang

Abstract—In this paper, we proposed a novel mechanism, namely Enhanced Dynamic Data Placement (EDDP). There are two components in EDDP: data partitioning and virtual machines (VMs) optimization. The first component is adapted from [Lee et al, 2014] whereby data placement and their size at the computing nodes must be proportional with their computation capability. In the second component, the configurations of the virtual machines created to handle the incoming jobs are optimized based on benchmarking. Experimental results show that EDDP managed to shorten job completion time.

Keywords—Big data, cloud computing, data placement, virtual machine.

I. INTRODUCTION

Hadoop is an implantation framework supply for parallel computing organized by multiple clusters. The most design for this framework are Hadoop distributed file system(HDFS) and MapReduce.

HDFS is a leading framework provides storage for massive data[7]. It cut a file into many unified unit of blocks and stored on this distributed system. On each node of this HDFS have it's own computation power, by this reason running files on HDFS must be processing faster than running files on a single computing node.

MapReduce is processing large-scale data via the distributed, parallel programming approach [2, 3]. However, the map and reduce processes are not optimized for heterogeneous environment [4]. Various approaches have been proposed to improve MapReduce performance in heterogeneous environment [1, 4, 5, 6].

[1] proposes a data placement algorithm, namely Dynamic Data Placement (DDP), to resolve the unbalanced node workload problem in heterogeneous environment. DDP assigned portion of a job to a node based on the computing capacity of the node.

Table 1 below illustrates the concept. As an example, for a block size of 64MB, if the total job size is 7 blocks. Job is distributed as follows:

TABLE 1 JOBS ASSIGNMENT OF DDP

	Slave 1	Slave 2	Slave 3
Speed	4X	2X	1X
Blocks data assigned	4 Blocks	2 Blocks	1 Blocks

II. OBSERVATIONS AND PROBLEM STATEMENT

A. Observation : Relationship between RAM-size allocated to VM and computational speed

Figure 1 depicts the outcome of the execution time of the WordCount program versus the RAM size allocated to a VM. The CPU of A and B are respectively 1024 and 3392 in Megahertz. It can be seen that the computation speed improves (i.e. lower) as the RAM size increases. However, when the RAM size goes above the region around 1024MB threshold, the performance almost stall and no significant improvement in the performance as the RAM size increases.

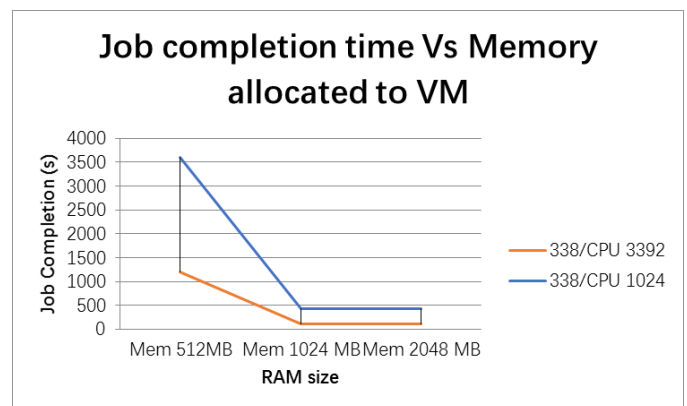


Fig. 1 Execution time versus RAM size (Job size 338 MB)

B. Observation 2: Increasing the number of VM assigned to a job will improve system performance.

A file partitioned on 7 node cluster must processing faster be processing on a cluster with 4 nodes, if we presuming every computing power are the same. Figure 1 provide us a clue that until the resource allocation reaches a threshold value no matter how many resources assigned to computing node will not changing their computing power.

We trying to test and count out the thresholds of every node's in the cluster and regarding this threshold value to rebuild a new node, this new node must the minimal resource consumption but

X.Z. Lu, China, Email: Luxinzhu198198@hotmail.com
 K.K.Phang, Malaysia, Email: phang@fsktm.um.edu.my

the computing speed will not reduce. Finally , the new cluster must have more nodes than the old one but each of them the computing speed must not be reduced.

C. Problem Statement

TABLE II: NODE CAPACITY

Each node specification.

Node	CPU	Memory	Disk
Node A (Master)	2	2 GB	50 GB
Node B (Slave)	4	4 GB	50 GB
Node C (Slave)	2	2 GB	50 GB
Node D (Slave)	1	1 GB	50 GB

For DDP, even though, experiment results show that DDP has improved, on average Hadoop WordCount performance by about 20%. DDP does not fully utilize the computing resources in each node. In the above scenario, only 3 VMs are used in parallel to complete a job. In fact, more VMs could be created to handle the job given.

III. PROPOSED ENHANCED DYNAMIC DATA PLACEMENT (EDDP) ALGORITHM

A. Motivation of EDDP

DDP is able improve system performance by shortening the job completion time. DDP allocates bigger job portion to VM which is hosted in more powerful host with higher computation capacity. However the VMs created are not optimized in terms of RAM size. These VMs utilizes the entire RAM available within the host.

For example, node B and C are more powerful nodes (in terms of CPU and RAM). These nodes could support 4 and 2 machines (e.g. with RAM size of 1 GByte) respectively. Due to system heterogeneity, 3 different types of VM are created as depicted in table 3. Note that EDDP has 7 VMs instead of the 3 VMs for DDP.

TABLE III: HETEROGENEOUS NODES WITH CAPACITY RATIO (A, B, C)

	Node A	Node B	Node C
DDP	1 VM	1 VM	1 VM
EDDP	1 VM of type A	4 VM of type B	2 VM of type C
Capacity ratio	A	b	c

In EDDP, given a job and a set of physical hosts, multiple VMs will be created within a host to handle the job assigned. The capacity ratio of each type of VMs is obtained from historical run time the job.

Algorithm EDDP:

If historical table exists

ComputingCapacityRatio of each type of VMs← obtain from historical record

for each DataNode in the cluster do

NodeCapacity according to VM type← obtain from ComputingCapacityRatio;

$$BlockNumber = TotalBlockNumber * \left[\frac{NodeCapacity}{\sum \text{each node capacity}} \right]$$

Allocate BlockNumber data blocks to the DataNode;

Else // no historical record

ComputingCapacityRatio of each type of VMs← set 1 for each node;

Add JobTypewith ComputingCapacityRatio to RatioTable; foreach DataNode in the cluster do

NodeCapacity=1;

$$BlockNumber = TotalBlockNumber * \left[\frac{NodeCapacity}{\sum \text{each node capacity}} \right]$$

IV. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

TABLE 4 EXPERIMENTAL SETUP

Machine	Specifications	
HP Compaq Elite 8300 SFF	4 CPUs x Processor Type: Intel Core i5-3570	16258.02 MB memory
HP Compaq Elite 8300 CMT	4 CPUs x Processor Type: Intel Core i7-3770	20354.00 MB memory
HP EliteDesk 800 G1 TWR	4 CPUs x Processor Type: Intel Core i7-4770	8103.47 MB memory
Hadoop version	Hadoop-0.2 (stable version)	
Virtual Machine Management	VMware vSphere 5.5	

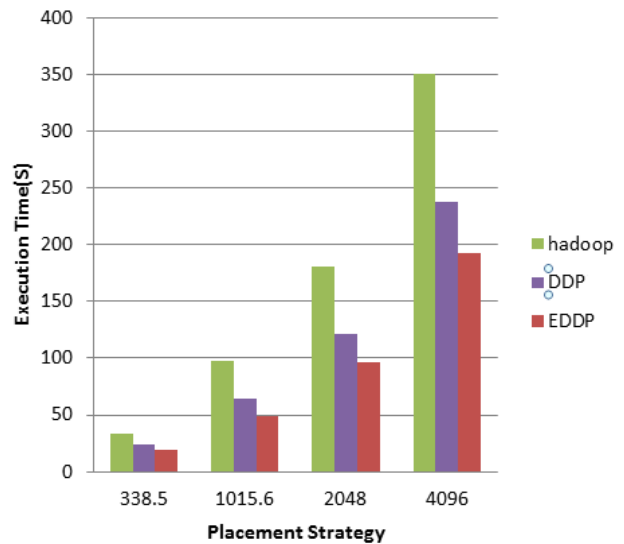


Fig. 2 Performance

In this experimental setup, the proposed EDDP prototype is implemented in Hadoop. In figure 2, it can be seen that EDDP out performed Hadoop and DDP by an average of about 20% and 8% respectively. This is because, in the DDP scenario, only 3 computing nodes are created; whereas in EDDP 7 nodes are created and are assigned to handle the job. In EDDP, it is observed that on one hand, more VMs within a host will decrease the overall system performance as each VM requires computing resources to run. However, on the other hand, more

VMs within a host implies more computational node and power. The overall effect is a shorter job completion time.

V.CONCLUSION

In this paper we proposed a dynamic data placement algorithm for Hadoop in heterogeneous environment. The proposed EDDP algorithm allows the creation of more computing nodes to enhance Hadoop performance. Using the WordCount applications, EDDP is able to improve the job completion time by about 8% compared to DDP. The limitation of current research is that it only involved the WordCount application, other type of applications will be included in the future work.

REFERENCES

- [1] Lee, C.-W., et al., A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments. *Big Data Research*, 2014. 1: p. 14-22. <https://doi.org/10.1016/j.bdr.2014.07.002>
- [2] Welcome to Apache™ Hadoop®! 2015; Available from: <http://hadoop.apache.org/>.
- [3] MapReduce Tutorial. 2015; Available from: http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.
- [4] Sujitha, S. and S. Jaganathan. Aggrandizing Hadoop in terms of node Heterogeneity & Data Locality. in *Smart Structures and Systems (ICSSS)*, 2013 IEEE International Conference on. 2013.
- [5] Improving MapReduce performance through data placement in heterogeneous Hadoop clusters. in *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, 2010 IEEE International Symposium on. 2010.
- [6] Quan, C., et al. SAMR: A Self-adaptive MapReduce Scheduling Algorithm in Heterogeneous Environment. in *Computer and Information Technology (CIT)*, 2010 IEEE 10th International Conference on. 2010.
- [7] HDFS Users Guide 2016; Available from: <http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html>