

# A large data processing algorithm for energy efficiency in a heterogeneous cluster

*Lei Wang<sup>1,\*</sup>, Weichun Ge<sup>1</sup>, Zhao Li<sup>1</sup>, Zhenjiang Lei<sup>1</sup> and Shuo Chen<sup>1</sup>*

ICT Department, State Grid Liaoning Electric Power Co., Ltd, Shenyang, China  
wl@ln.sgcc.com.cn; gwc@ln.sgcc.com.cn; lz@ln.sgcc.com.cn; leizj@ln.sgcc.com.cn;  
cs@ln.sgcc.com.cn.

**Abstract.** It is reported<sup>1</sup> that the electricity cost to operate a cluster may well exceed its acquisition cost, and the processing of big data requires large scale cluster and long period. Therefore, energy efficient processing of big data is essential for the data owners and users. In this paper, we propose a novel algorithm MinBalance to processing I/O intensive big data tasks energy efficiently in heterogeneous cluster. In the former step, four greedy policies are used to select the proper nodes considering heterogeneity of the cluster. While in the latter step, the workloads of the selected nodes will be well balanced to avoid the energy wastes caused by waiting. MinBalance is a universal algorithm and cannot be affected by the data storage strategies. Experimental results indicate that MinBalance can achieve over 60% energy reduction for large sets over the traditional methods of powering down partial nodes.

## 1 Introduction

With the development and application of information technology, the data produced is presented. How to store, manage, and apply these data to become an explosive growth. A general concern of the business community and academia. You know, there is great value in big data, so research based on big data is also very much. Many scholars call it the fourth paradigm of scientific research [1]-[2]. Cloud computing as a kind of Emerging economies based on economies of scale have become big data the first platform for storage and processing. The open source cloud meter is the platform Hadoop, HBase, and HadoopDB have been widely studied and the application. More and more businesses are building their own big data points the platform deals with growing business data and even offers Various services based on big data [3]. A lot of hardware resources are required to handle big data. Include servers, PCS, and even mobile devices. The making of these devices takes a lot of energy, mainly electricity, to be used globally electricity is generated mainly by thermal power so the big data also has great challenges to energy and environment [4]. In 2005, show that a server is within the lifetime of its use the total amount of electricity consumed has exceeded the purchase cost. And research show that, in 2008, the world's 4400 servers consumed electricity 0.8% percent, if you go like that, at that rate By 2020, that proportion

---

\* Lei Wang: wl@ln.sgcc.com.cn

will be 3.2%. Epa (US Environmental Protection Agency) issued a report statement in 2006, the total electricity consumption of American IT agencies was 61 billion KWh, the electricity bill alone is \$4.5 billion [5]-[7]. So that's a concern Big data storage and processing performance must also be used for energy consumption Give enough attention [8].

This paper mainly discusses the large data processing tasks of I/O intensive. The computationally intensive tasks are affected by the real-time running state of the processor large, and different hardware and operating system provided processor control machines. There are differences in system, so this paper does not consider computationally intensive large Numbers According to the task. Because of the data-intensive task for the processor with a small dependency, for a server, the processing of each data block is reduced the time and power consumed can be regarded as basically the same. A cluster consisting of n heterogeneous nodes processes a Map-Reduce tasks, assume that the nodes involved in task processing are C, the total energy consumed during task processing is

$$Total\ cost = \max_{n_i \in C} \{T_i\} \sum_{n_i \in C} P_i \tag{1}$$

The  $T_i$  and  $p_i$  represent the processing time of the I node Power consumption. By type (1) the total energy is mainly affected by two factors: use the nodes that perform the tasks and the maximum processing time of the nodes. Type (1) The two kinds of high-efficiency data processing methods:

- 1) To select  
Some suitable nodes perform tasks to reduce total power consumption;
- 2) Equilibrium  
The load of a node reduces the maximum execution time.

According to the actual situation of the node, determine which tasks each node performs, namely Equalize the load of each node, reduce task execution time, and further Reduces total energy consumption of the system. The method has three distinct advantages:

- 1) Fully consider the heterogeneity of the nodes;
- 2) There is no copy storage strategy shut;
- 3) Comprehensively consider the total number of nodes and load balance the factor of consumption.

## 2 Problem description

I/O intensive large data processing tasks for heterogeneous clusters. Energy efficient processing problems can be formalized as follows: given a set, the group is composed of h isomeric nodes and  $N = \{n_1, n_2, n_3, \dots, n_h\}$ , of which A node  $n_i$  ( $1 \leq i \leq h$ ) takes the time and work required to process a block of data. The consumption is  $T_i$  and  $p_i$ . Given an I/ O intensive big data processing Task J, the data set that it needs to process contains m standard data blocks  $D = \{d_1, d_2, \dots, d_m\}$ , each block has r copies stored in h on a heterogeneous node. The actual storage location of each data block can be passed the metadata of the cluster is obtained, and the actual storage situation of the data set is  $L = \{(n_{1.1}, \dots, n_{1.r}), (n_{2.1}, \dots, n_{2.r}), \dots, (n_{m.1}, \dots, n_{m.r})\}$ ,  $n_{i.j} \in N (1 \leq i \leq m, 1 \leq j \leq r)$ , which represents the j of data block  $d_i$  a copy is stored on the node without loss of generality, assuming  $n_{i.1} < n_{i.2} < \dots < n_{i.r}$  ( $1 \leq i \leq m$ ), which can be known to be stored on node  $n_i$ . The set of data blocks can be represented as  $D_i = \{d_j \mid n_i \in (n_{j.1}, \dots, n_{j.r}), 1 \leq i \leq m\}$ . The solution of this problem is from node set N, select part of the nodes to handle task J and make these nodes work, the total energy consumed during processing is the least. Clearly, data set D a data block must have at least one copy stored on the selected node, the sample task can be successfully executed.

### 3 Processing algorithm

Through formula (1) and the analysis of the aforementioned issues shown the optimal target of high energy efficient data processing under the construction node is two:

- 1) Select the appropriate coverage node, i.e. the work in reduction (1) Part of the consumption  $\sum_{n_i \in NS} P_i$  ;
- 2) Reduce the actual processing time of critical nodes, reduction type (1) time  $\max_{n_i \in NS} \{VC(n_i)|t_i\}$

The analysis of the problem description section can find the exact solution to the problem is difficult, and it's hard to find the complexity in polynomial time solution.

In this paper, the approximate solution method is adopted to optimize the target decompose the problem into two relatively independent sub-problems: first find the appropriate coverage node and then allow the selected coverage node distribution, namely, load balancing each node to reduce critical nodes point processing time, finally achieve energy efficient target. The traditional way to achieve energy efficiency is by closing some nodes. The method, in fact, is the question of finding suitable coverage nodes after decomposition the topic. But the traditional approach does not take into account the heterogeneity of the nodes, therefore can be classified as hypergraph coverage problem or classic collection coverage problem to solve.

The problem with the node selection phase is that it is a given isomerism select the appropriate node in the node set as the overlay node performing I/O intensive large data tasks. In this question, by

the processing time of each node is determined by the number of new data blocks, the number of new data blocks is affected by the order of node selection the traditional WSCP solution does not apply to this problem.

In this section, we present a selection of selected cases based on the thinking of the greedy mind Greedy, as shown in algorithm 1.

#### 3.1 Algorithm

##### 3.1.1 Greedy algorithm

Input:  $N = \{n1, n2, \dots, nh\}$ ,  $D = \{d1, d2, \dots, dm\}$ ,  $\{t1, t2, \dots, th\}$ ,  $\{p1, p2, \dots, ph\}$ ,  $L = \{(n1.1, \dots, n1.r), (n2.1, \dots, n2.r), \dots, (nm.1, \dots, nm.r)\}$ ;

Output: overlay node set NS;

- (1) Initialization:  $NS = DS = \emptyset$ ;
- (2) For each node  $ni \notin NS$  do
- (3) The weight of weight ( $ni$ ) of each node is calculated;
- (4) The maximum value of the option value is the coverage node;
- (5) Update:  $DS = DS \cup NC(ni)$ ,  $NS = NS \cup \{ni\}$ ;
- (6) If the DS then go Step (2);
- (7) Else Return NS.

The selection of each node in algorithm 1 is based on a certain standard, at the current optimal node, the node may not be globally optimal, so, the solution is an approximate solution. The concrete choice idea is to calculate currently, the weight of each non-overridden node Weight (), the option value is the most. The large node ACTS as the new coverage node, and then updates the coverage node set. And its stored data set, continue the next calculation of weight and select nodes point, until the coverage node contains the entire dataset.

If the heterogeneity of the node is not considered at all, you only need to select the nodes that currently contain the new data blocks.

Overwrite the node so that the final selection can be approximated

The minimum number of cap nodes reduces energy consumption to a certain extent. But for heterogeneous nodes, the nodes that may be selected are inefficient and lead to eventual results, the energy expenditure is very high. The weight of each node in this scheme can be tabled shown for  $Weight(n_i) = \frac{|NC(n_i)|}{t_i \times P_i}$

### 3.2 Algorithm

#### 3.2.1 MinBalance algorithm

Specific load balancing methods: MinBalance as algorithm 2, The node selected by algorithm 1 is shown to load balancing to determine each what data is actually processed by the node to enable the task to process the energy consumed the least. The main idea of the MinBalance method is to find the current covering all the key nodes in the node, as described in algorithm 2, these key nodes are then loaded with other coverage nodes balance, then proceed to the next iteration, looking for key nodes again load balancing until all the key nodes are currently unavailable and other nodes are balanced by load, as shown in algorithm 2.

Input:  $N = \{n1, n2, \dots, nh\}$ ,  $D = \{d1, d2, \dots, dm\}$ ,  $\{t1, t2, \dots, th\}$ ,  $\{p1, p2, \dots, ph\}$ ,  $L = \{(n1.1, \dots, n1.r), (n2.1, \dots, n2.r), \dots, (nm.1, \dots, nm.r)\}$ ;

Output: total energy consumption TotalCost.

- (1) Find the overlay node set NS;
- (2) all key node KN in NS;
- (3) For each key node kni do {
- (4) For each cover node nj do {
- (5) Calculates kij;}
- (6) If  $\max \{kij\} > 0$  then Rebalance (kni, nj);
- (7) Remove kni form KN;
- (8) If  $KN = \emptyset$  then break;}

(9)  $TotalCost = \max_{n_i \in NS} \{ |VC(n_i)| t_i \} \sum_{n_j \in NS} P_j$  ;

- (10) Return TotalCost;
- (11) Else go Step (2).
- Rebalance(na,nb)
- (12) Calculates kab;
- (13) count=0;
- (14) For each data chunk  $d_i \in VC(na)$  do {
- (15) If  $d_i \in Db$  then {
- (16) Remove  $d_i$  from VC (na) and add  $d_i$  to VC (nb);
- (17) count+=1;}
- (18) If count=kab then Return.}

### 3.3 The energy consumption of MinBalance algorithm

This section focuses on the energy consumed by the MinBalance algorithm in handling big data tasks. In order to be easy to identify in the experimental results diagram, 4 kinds of

implementation MinBalance algorithm are used in BMNF, BHPF, BLPF, BLEF. Figure 1 shows the energy consumed by the algorithm in dealing with small tasks when the data is randomly distributed. It can be seen that in the same situation, MinBalance can reduce energy consumption by about 10% when  $|D| = 100$ , and  $|D| = 1000$ , MinBalance can reduce energy consumption by about 60%.

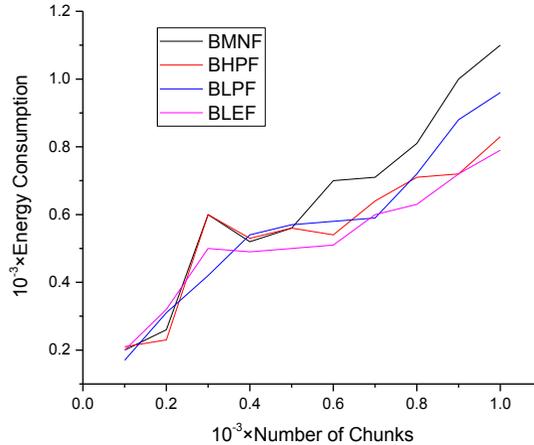


Fig. 1. MinBalance methods

## 4 Conclusion

This article addresses the I/O intensive large data points in heterogeneous clusters. In order to make a good calculation of the high efficiency MinBalance, this method is not subject to the shadow of the storage strategy. MinBalance decomposes complex problems into node selection and load. The equilibrium two steps are solved respectively, and the solution of the node selection is solved four different greedy strategies were adopted to fully consider the differences of the nodes Construct, select a small number of suitable nodes for task processing.

In the load balancing process focuses on the key section of the longest handling time point, as much as possible without increasing the maximum processing time. Its load is migrated to nodes with short processing time. The algorithm is in place when the data set is large, it is more than traditional closing node. The method reduces energy consumption by up to 60%. From the experimental results it can be seen that the MNF method with node-heterogeneity is the least energy efficient.

But the algorithm proposed in this paper is based on the idea of greed as an algorithm, it has not proved its exact approximate degree and above theoretically lower bound. The experiment in this paper is simulated, but the actual cluster is different time that nodes consume when processing big data is unpredictable and the property of the storage medium itself affects the same node. The power of data consumption of sample size may differ. So The next step will be to consider platforms such as real Hadoop conduct performance testing and analysis.

## References

- [1] Zhang L, Li K, Xu Y, et al. Maximizing reliability with energy conservation for parallel task scheduling in a heterogeneous cluster[J]. *Information Sciences An International Journal*, 2015, 319(C):113-131.
- [2] Pati S, Mehta M A. Job aware scheduling in Hadoop for heterogeneous cluster[C]// *Advance Computing Conference. IEEE*, 2015:778-783.
- [3] Botta A, Donato W D, Persico V. Integration of Cloud computing and Internet of Things [J]. *Future Generation Computer Systems*, 2016, 56(C):684-700.
- [4] Kumar S, Kumar S, Kumar S, et al. Security, Privacy Issues and challenges In Cloud Computing: A Survey[C]// *International Conference on Information and Communication Technology for Competitive Strategies. ACM*, 2016:47.
- [5] Zhang H, Li B, Jiang H, et al. A framework for truthful online auctions in cloud computing with heterogeneous user demands[C]// *INFOCOM, 2013 Proceedings IEEE. IEEE*, 2016:1510-1518.
- [6] Hu H, Wen Y, Chua T S, et al. Toward Scalable Systems for Big Data Analytics: A Technology Tutorial [J]. *Access IEEE*, 2017, 2(1):652-687.
- [7] Feller E, Ramakrishnan L, Morin C. Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study [J]. *Journal of Parallel & Distributed Computing*, 2015, s 79–80(C):80-89.
- [8] Majeed A, Shah M A. Energy efficiency in big data complex systems: a comprehensive survey of modern energy saving techniques [J]. *Complex Adaptive Systems Modeling*, 2015, 3(1):6.