TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.3, March 2014, pp. 2088 ~ 2095
DOI: http://dx.doi.org/10.11591/telkomnika.v12i3.4282

Task View Reduction Algorithm Based on Rough Sets in Gloud Storage

Yiyi Xu*[1], Peihe Tang[2], Chen Yang[3]
[1,2,3]College of Computer Engineering, Guangxi University of Science and Technology, Liuzhou, 545006, China, Ph./Fax: 0772-2687372
[1]School of Information Engineering, Wuhan University of Technology, Hubei 430070, China
Corresponding author, e-mail: winxyy@qq.com[1], tangpeihe@163.com[2], chengyang@126.com[3]

Abstract

The Knowledge reduction is one of the important research issues in rough set theory, which applies knowledge reduction theory to reduction the massive task sets in Gloud storage. At first, an equivalence class evolved from subviews will be obtained after task update, Then, a parallel running strategy is designed for large-scale data, and calculate the optimal attributes based on the task set with minimal time overhead, to this end, delete redundant views according to the optimal attribute sets. Finally, the optimized task combination views are obtained. Simulation results shows it has better overall performance in time span, runtime, speed-up ratio and scalability when compared with the original algorithm that under same conditions, the actual examples used in analysis indicate the effectiveness of this method.

Keywords: *rough set, knowledge reduction, task view, MapReduce*

1. Introduction

Over the past decade, with the development and wide application of database technology, distributed network and distributed file systems, cloud storage, Gloud storage [1] and enterprise Gloud storage as an innovative storage mode, have become the academia and industry research hotspot. Gloud storage is designed to complement grid storage theory and cloud storage theory. It is based on virtualization technology, using the bounded internal network as the carrier to create a new grid-like structure of the loosely coupled architecture on or between the existing cheap device groups. Provide a more flexible , more economic and more effective storage platform to the enterprise.

Gloud storage basic storage services stressed basing on the existing physical device, converge the industry standard components (tight coupled disk, RAM and CPU), build a user-centric task scheduling view. The basic features are: First, massive task parallel; second, multiple task scheduler is deployed on the same physical host or multiple physical hosts, scheduler controls all the resources; third, the number of parallel task executed by the task scheduler is limited by the computing capacity (such as physical memory size) of the node; fourth, lack of a better collaboration strategy among each scheduler; fifth, the node may be limited to a single management domain, but the availability and computing capability the node showed is different and dynamic changing; sixth, task scheduling is based on the virtual machine or the shared middleware. Gloud storage with its numerous resource nodes, clear boundary, openness and dynamic nature, demands for different nodes are different, a reliable and efficient task scheduling technology has become one of the key technologies.

Therefore, in recent years some methods and tools for task combination services appeared, such as Aqualogic [2] and Damia [3], GPSO [4]. Such method is capable of handle the cross-domain task requests, and has a relatively good flexibility and ease of use. However, they are easy to fall into local optimal solution prematurely; especially there are challenges in the update and optimization of task view. The reason is that under the Gloud storage environment, parameters like the task arrival probability, distribution laws, the size of the storage space users' required, the credibility of resource nodes (reflecting the reliability and stability of the resource nodes), the demand for the reliability of users' data, the distance between resource nodes and application nodes, the urgency degree of users' expectation, the

Received August 31, 2013; Revised September 18, 2013; Accepted October 18, 2013

submit time of resources requesting task and the storage class of users' data (long-term archiving, temporary storage, frequently update) frequently changes when users' resources request task. In general, the low level tasks of enterprise storage grid resources are huge and dynamic changing. Its completion time will be growing exponentially with the growth of task scale. How to propose the corresponding update optimization scheme in the process of task service combination is very important.

In light of the above research background, we proposed a task combination view optimization method based on rough set reduction. With this method, view updates from bottom up when task parameters change, reduce the entire view update time and maintain data consistency. It retains and makes use of the time overhead and speed and other information, which is suitable for handle large-scale task under enterprise storage environment.

2. Problem Analysis

Current method of task combination view optimization still did not give a satisfying solution to the problem. Many proposed knowledge reduction algorithms, like attribute reduction algorithm based on information theory, or attribute reduction algorithm based on positive region, or attribute reduction algorithm based on the difference - similitude matrix all put how to improve knowledge reduction algorithm efficiency in the first place. But all these algorithms are based on the assumption that all data can be place into memory at one time, which obviously cannot handle the data generated during the massive task scheduling, and certainly cannot applied to distributed processing, parallel processing. In general, the reduction of the massive task view contains:

1. The abstract description and characteristics of scheduling task; reference the description mechanism of metadata from SNIA Cloud Data Management Interface standard to extend the demands of service quality, apply the formalized, standardized description, such as network bandwidth, response time, and context environment of the application to construct the description interface.
2. The association and evolution among each task at the same time point.
3. The task view generated at some point based on the macro characteristics and behavior characteristics of the task group.
4. Optimization of the task view based on rough set knowledge reduction algorithm.

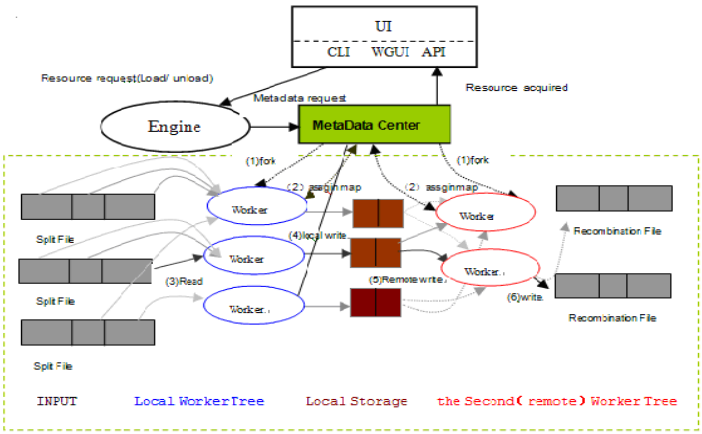

Figure 1. The Gloud Storage Resources Massive Task Scheduling Model

This paper only involves an in-depth study of the fourth content. Under the conditions of given task view construction demand and complex task request and update frequency, the actual task scheduling is divided into a number of parallel operation subviews on the base of MapReduce architecture. Designing the corresponding Map and Reduce functions (As shown in Figure 1), the Map function first takes in a set of input key/value pairs, then through some sort of calculation, generating intermediate key/value; Reduce function receives an intermediate key

and the corresponding set of value generating the final key/value through the merging process. Then run its own reduction algorithm in the distribution system to reduce each subview, thus obtain the program with least time overhead, highest actual running efficiency and minimum update cost. According to this program, an iterative method is used to determine the optimal candidate attribute, and then generate a set of equivalent optimal reduction to the task view at last. The algorithm is based on the simple theory of Rough set, combined with parallel program techniques in MapReduce, finally realized on the Hadoop open source platform. The experiment results show that this algorithm can effectively handle the task group of large-scale, hot inequality and under emergency.

3. Releated Theories

3.1. Knowledge Reduction Algorithm

The rough set theory is a mathematical tool which can quantitatively analyze the imprecise, inconsistent and incomplete information and knowledge. Its basic idea is to form concepts and rules through classification and summarization of the relational database, then realize knowledge discovery through equivalent classification. Its most significant feature is that it did not need to provide any prior information besides the required processing data. Therefore, the uncertainty description or processing of the problem is quite objective. Currently, research based on Rough set theory mainly focus on attribute reduction rule acquiring and algorithm research. Attribute reduction, as an NP-Hard problem, has become a hot topic for many scholars. Reduction theory based on rough set developed rapidly over the past several years, many new and effective methods have come forth. For example, for different information systems (coordinated and uncoordinated, complete and incomplete), Pawlak, Wong, Yao and Iwinski have proposed many methods by combining information theory, concept lattice and swarm intelligence algorithm technology, such as data analysis method, attribute reduction algorithm based on information entropy, dynamic reduction algorithm, incremental algorithm and identified matrix algorithm. They all achieved corresponding results [2-3], [5-7].

Below are some basic concepts of Rough set in this paper.

Definition 1:

Quintuple $S =< U,C,D,V,f >$ is a decision table, which $U = \{x_1, x_2,..., x_n\}$ represent the non-empty finite set of the objects, called the domain; subset C and D are called condition attribute set and decision attribute set; $C \cap D = \varnothing$, $V = \bigcup_{a=C \cup D} V_a$, V_a is the range of attribute a; $f : U \times (C \cup D) \to V$ is an information function, it specifies the attribute values of every object in U.

Definition 2:

To $\forall a \in C \cup D, x \in U$, $f(x, a) \in \forall a$; each attribute subset $A \subseteq C \cup D$ determines a binary i indistinguishable relation: $IND(A) = \{(x, y) \notin U \times U \mid \forall a \in A, f(x,a) = f(y,a)\}$. Relation $IND(A)$ constitute a division of U, denoted as $U / IND(A)$, abbreviated U / A.each of the element $[x]_A = \{y \mid \forall a \in A, f(x,a) = f(y,a)\}$ is called equivalent class.

Definition 3:

Assume U, V represent two domains. Elements u ∈ U and v ∈ V are compatible, denoted as $u \subset U$. Without loss of generality, it is assumed that for each u ∈ U, there will be a v ∈ V to ensure that they are associated, vice versa. Then the compatible relationship between U and V can be multi-value mapping, assign a value to each object, that is, to define it, i.e. $\forall(u) = \{v \in V \mid u \subset v\}$.

Definition 4:

Set the decision table information system $S =< U,C,D,V,f >$, for each subset $X \subseteq U$ and uncertain relation A. the lower approximation sets and upper approximation sets of X can be defined by the basic set of A respectively as follows:

Lower approximation sets: $\underline{A}(X) = \bigcup\{Y_i \in U \mid IND(A) : Y_i \subseteq X\}$

Upper approximation sets: $\overline{A}(X) = \bigcup\{Y_i \in U \mid IND(A) : Y_i \cap X = \varnothing\}$

Definition 5:

Assume C,D are attribute sets, no attribute of D can be omitted. If $D \subseteq C$, and $Ind(D) = Ind(C)$, then Q is a reduction of P , denoted as $Red(P)$. Furthermore, if $Core(C)$ is denoted as the attribute set that cannot be omitted, referred as the core of C .then all the reduction Red(C) just exactly equals the core of C,

That is $Core(C) = \cap Red(C)$. The formula not only reflects that the relation between nuclear and all the reduction are obtained by reduction, it also shows that core is the most important part of knowledge base, which cannot be deleted in the process of knowledge reduction.

Definition 6:

In decision table $S = < U,C,D,V,f >$, mark $U / C = \{[x_1^{'}]_c [x_2^{'}]_c ,...,[x_s^{'}]_c \}$.

$U^{'} = \{x_1^{'}, x_2^{'},..., x_s^{'}\}$ $U^{'}_{POS} = \{x_{i_1}^{'}, x_{i_2}^{'},..., x_{it}^{'}\}$, the object in $U^{'}_{POS}$ is compatible object, $U^{'}_{BND}$ equals $U^{'} - U^{'}_{POS}$, so $S^{'} = (U^{'} = U^{'}_{POS} \cup U^{'}_{BND}, C,D,V,f)$ is the simplified decision table.

Decision table can be divided into a consistent decision table and inconsistent decision table. When D is totally depend on ($C \Rightarrow D$), it is called consistent; when $C \Rightarrow kD(0 < k < 1)$, the decision table is inconsistent. Whether the decision table is reducible depending on whether it is a consistent decision table. This is because different reasons can cause the same results, but the same reason is not allowed to lead to different results.

3.2. Knowledge Reduction Algorithm Based on the Task Combination Views

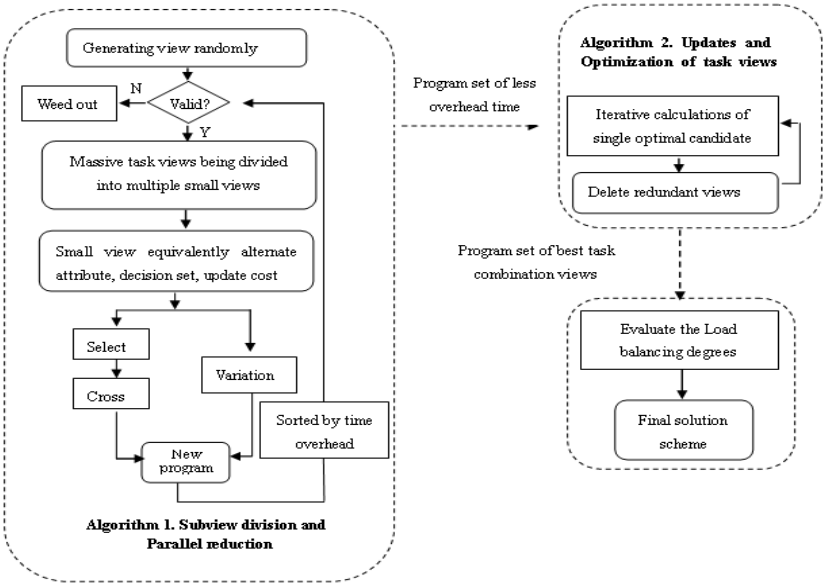

Figure 2. Specific Process of Knowledge Reduction Algorithm Based on Task Combination

Specific process of knowledge reduction algorithm based on the task combination under the Gloud storage environment, as shown in Figure 2. Below is a brief description of algorithm 1 and algorithm 2.

A. Algorithm 1: Subview division and parallel reduction algorithm

The traditional parallel reduction strategy assumes place all objects into the memory at one time. Yet this is not suitable for large-scale task view sets in Gloud storage system [8]. By using the MapReduce technology to handle massive amounts of data, we did not need to deal

with fault tolerance processing and data partitioning. We just need to divide the actual problem into a number of parallel sub-problems. Its main functions involve Map function and Reduce function. Map function mainly deals with the calculation of different sub-equivalence class, while reduce function mainly calculates the number of unrecognized objects in the same equivalence class [9].

First, assume there are k different decision attribute values in decision table S , the decision attribute value of compatible objects respectively mapping $1,2,...,k$. That of incompatible objects all mapping $k+1$. In this way, the entire decision table S can be seen as constituted by $k+1$ sub-decision table $D_1, D_2,...D_k, D_{k+1}$.

Each decision table contains objects of the same category; the number of the objects is $n^1, n^2,..., n^k$ respectively. Therefore, decision table S is "consistent" decision table.

Initialization: in the consistent decision table S, the recognizable objects in task combination views was generated by two objects with different decision attribute values and different condition attribute combination values. Assume $a \in C$, if the decision value of two objects is different, condition attribute a is also different, then a can identify these two objects, i.e. a recognizable object pair. In order to identify all the recognized objects in task scheduling views according to the above rules, for k different decision attribute values, mapped into $k+1$ sub decision table $T_1,T_2,...,T_k,T_{k+1}$.

Following process is the reduction of one of the component.

Step 1: Calculate the condition mutual information of condition attribute C_i and decision attributes D_i in the decision table T_i

Step 2: Calculate the relative core $C_0 = Core_{Di}(C_i)$ of C_i relative to D_i. Generally, $I(C_0, D_i) < I(C_i, D_i)$; sometimes $C_0 = \varnothing$,then $I(C_0, D_i) < I(C_i, D_i) = 0$;

Step 3: Order $I(B_i, D_i) = I(C_i, D_i)$ repeat in conditions attribute set $C_i - B_i$.

a) For each attribute $p \in C_i - B_i$, calculate the condition mutual information $I(p,D_i \mid B_i)$;

b) Choose the attribute that makes the condition mutual information $I(p,D_i \mid B_i)$ the biggest. Denoted as p (if the attribute are more than one, choose the one that has

The least combination with attribute B_i); and $B_i - B_i \cup \{P\}$.

c) If $I(B_i, D_i) = I(C_i, D_i)$,then terminate; otherwise turn to □;

Step 4: Finally B_i is a reduction of C_i relative to D_i

Following is an example of the knowledge reduction algorithm under the Gloud storage environment. Table 1 is a part of the typical decision table when the task view combines together, in which the condition attribute set $C=\{a_1,a_2,a_3,a_4,a_5,a_6,a_7\}$, a_1 indicates the order of requesting time, a_2 indicates the type of service, a_3 is the requirement of quality service, a_4 is the economic principle, a_5 is the size of the source file, a_6 is the length of task scheduling, a_7 is the security requirement.

Decision attribute set cc $D=\{d\}$ represents the preliminary results, domain $U = \{U_1,U_2,U_3,U_4,U_5,U_6,U_7\}$.

Table 1. Task view combines Decision table

task	a_1	a_2	a_3	a_4	a_5	$a_{6/ms}$	a_7
U_1	1	backup	3	maximize	high	4	Ordinary clients
U_2	2	Large Files division	2	maximize	low	2	VIP clients
U_3	3	backup	1	maximize	low	∞	Unexpected clients
U_4	4	download	1	null	medium	5	Trusted clients
U	5	Upload	3	null	high	600	Trusted clients
U_6	6	Search	3	null	Medium	1200	Unexpected clients
U_7	7	verify	2	null	High	100	New clients

Algorithm 1 is used in the attribute reduction of the object in Table 1 with the minimum time overhead. First calculate $I(C,D)$=1.761, then calculate the core C relative to D, $C_0 = \{a_1\}$ $B = \{a_1, a_2, a_3\}$ will be obtained through step 3 by algorithm 1. Next judge the conditions $I(B,D) = I(C,D)$. If the condition is true, then algorithm end; and output the reduction set $B = \{a_1, a_2, a_3, a_5\}$ which is a set C relative to a set D.

Analyze the relative reduction set B. Because $H(D|\{a_1, a_2, a_3\}) = H(D|\{a_1, a_2, a_3, a_5\})$, so the attribute a_5 is the redundant attribute of reduction B relative to decision attribute set D. Thus, the reduced decision table can have less condition attribute while with no loss of knowledge content.

B. Algorithm 2: optimization algorithm of the task combination views.

In order to optimize the task combination views, model it as a 0-1 programming problem. There are many ways to solve the problem quickly. Description of the 0-1 programming is as (1):

$$\min B_i \text{ s.t. } x^E \times ES' = I^S \quad (1)$$

B_i is the target function; x^E is the combination programs chosen from the equivalent subview, which is 1 when chosen, otherwise 0. Constraints are an original task can only choose one corresponding equivalent set in the task combination program. I^S is a column vector whose length is |S| and elements are all 1. Since the target function does not meet the principle of superposition, iterative calculation of the single-view optimal attributes is selected. Some of the specific process of the algorithm will be introduced in another paper.

4. Experimental results and analysis

The proposed algorithm in this paper was conducted on the open-source platform Hadoop 0.20.2 and Java 1.6.0_20 in the Gloud Environment built by the school distributed storage laboratory. We deployed a self-developed local Gloud storage system, which can support the users, including 105 client and 3 servers to do the task scheduling work, like upload, download and search. The screenshots of the system is shown in Figure 3, the parameters of task view equivalent class at some point is shown in Table 2:

Table 2. Task View Equivalence Class at Some Point View Combines Decision

Node ID	Number of tasks	Number of condition attributes	Set value of the number of decision attributes number
0	8249	10000	103
02	680	9	2
68	785	26	5
103	430	30	10
3	799	78	5
79	101	11	3

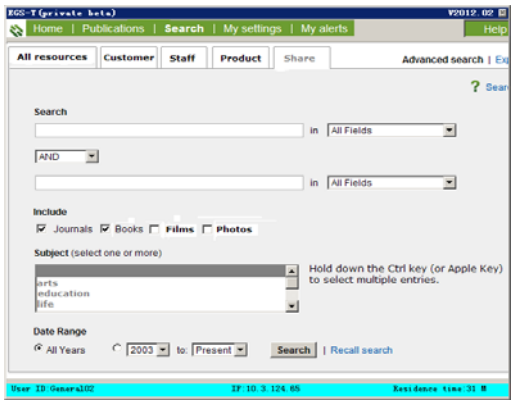

Figure 3. The Gloud Grid Test Platform

5. Experimental Results

We mainly measured the effects of reduced task combination views under the environment of Gloud computing from time span, runtime, speedup ratio and scalability. Figure 4 to 7 is the comparison before and after optimization. As we can see from the figure, runtime increased rapidly as the number of attributes goes up. When the scale of task set is fixed, this algorithm has better speedup ratio as the number of node increases. When the size of task set scale and number of node increase at the same time, the scalability of the algorithm is also very good. Therefore, the proposed task combination view reduction algorithm based on rough set knowledge is capable of applying in large-scale storage systems and has a better application prospects.

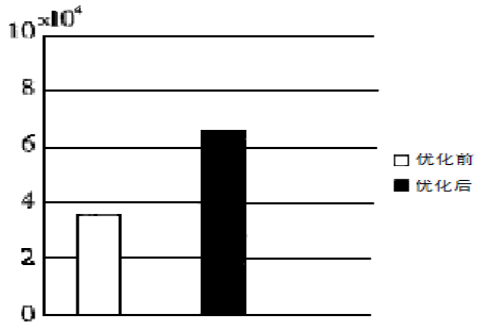

Figure 4. Comparison in Time Span

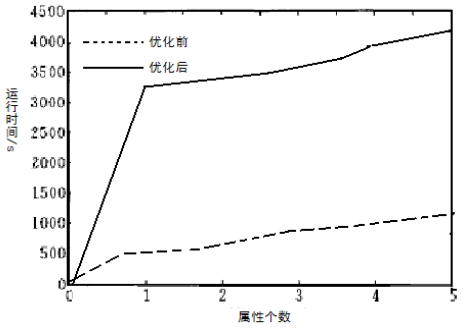

Figure 5. Comparison in Runtime

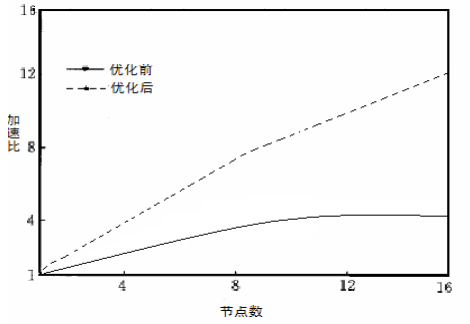

Figure 6. Comparison in Speedup Ratio

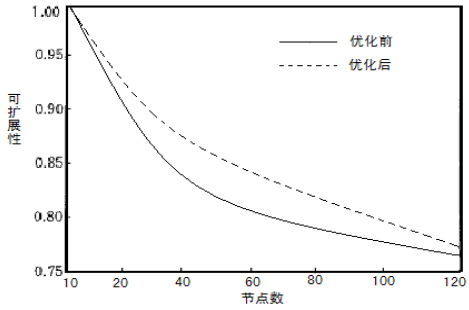

Figure 7. Comparison in Scalability

Acknowledgements

This work was supported by the Science Research Foundation of Guangxi University of Science and Technology (No.1261126), Guangxi Natural Science Foundation (No. 2013GXNSFBA019268, No.2011GXNSFA018162), Guangxi features professional and curriculum integration funded construction projects (No.GXTSZY217).

References

[1] Tang Peihe, Xu Yiyi. Resource Scheduling Strategy Based on Credibility in the Enterprise Gloud Storage. *Journal of Convergence Information Technology*. 2012; 7(16): 393-400.

[2] Mischa Schmidt, Jan Seedorf, Stefano Napolitano. Carey M. Experiences with large-scale operational trials of ALTO-enhanced P2P file sharing in an intra-ISP scenario. *Peer-to-Peer networking and Applications*. 2013; 6(2): 134-154.

[3] Destercke S. A k-nearest neighbour's method based on imprecise probabilities. *Soft Computing*. 2012; 16(5): 833-844.

[4] Gao Liqun, Li Ruoping, Zou spin. Global particle swarm optimization. *Northeastern University (Natural Science)*. 2011; 32(11): 1538-541.

[5] Xiao Fu, Jin Liu, Haopeng Wang, Bin Zhang, Rui GAO. Rough Sets and Neural Networks Based Aerial Images Segmentation Method. *Lecture Notes in Computer Science*. 2011; 76(66): 123-131.
[6] Qu Binbin, Lu Yansheng. Attribute reduction algorithm based on rough sets. *Hua zhong University of Science and Technology (Natural Science).* 2005; 33(8): 30–33.
[7] Qian Jin, Miao Dou Qian, Zhang Zehua. Knowledge Reduction Algorithms in Cloud Computing. *Chinese Journal of Computers*. 2011; 34(12): 2332-3333.
[8] Lin Zi-yu, Yang Dong-qing, Wang Teng-jiao, Song Guo Jie. Research on materialized view selection. *Chinese Journal of Software.* 2009; 20(2): 93–213.
[9] P Zhang, G Ang, Gi, C Liu. Optimization Update for Data Composition View Based on Data Service. *Chinese Journal of Software.* 2011; 34(12): 2344-2353.