Acta Technica 62 (2017), No. 7A, 1191–1202

Data envelopment analysis method based on manquist index model and its application

Yanrui Liu[1], Di Yao[1], Jinpei Li[1]

Abstract. Existing cloud storage task scheduling algorithm based on particle swarm cannot perceive QoS preference of user, which means that the same fitness function is adopted for user tasks with different QoS requirements, so user satisfaction rate is low. This paper incorporates multi-QoS factor into assessment by redefining fitness function of particle swarm optimization and adjusts QoS weight corresponded by task of different type to make it adapt to different QoS requirements of different tasks by integrating analytic hierarchy process. Experiment shows that the algorithm can adapt to QoS requirement changes among tasks better, and although some execution time is sacrificed through weight adjustment, user satisfaction rate increases obviously.

Key words. Cloud storage, Task scheduling, Particle swarm optimization, Analytic hierarchy process.

1. Introduction

In recent years, with rapid increase of information of various types and development of Internet technology, more and more services are provided to user in the form of the cloud. The user can access to various services provided by cloud service providers through the network when he has a thin client (the most common one is browser), from common online document editing and video audio to virtual host and scientific calculation etc. As a kind of special cloud system, cloud storage system provides storage service, which means that local data is stored at online storage space provided by storage service provider through the network. The user needing storage service does not need to establish his own data center again, and what he shall do is to apply for storage service to service provider to access to haul storage, and on the basis of storage demand convenience, repeated construction of storage platform is avoided, and expensive software and hardware cost and maintenance

[1]Antai College of Economics & Management, Shanghai Jiaotong University, Shanghai 200030

charge are saved[1]. Task scheduling algorithm decides efficiency of the entire cloud system largely, being one of research hotspots of the field. Aimed at the problem, this paper improves existing PSO scheduling algorithm through analytic hierarchy process. Fitness function of PSO algorithm is redefined firstly and multi-QoS factor is incorporated into fitness calculation after classification. Different weighted values are endowed to these factors to reflect preference of user to different QoS. Weight setting is determined based on analytic hierarchy process and for weight after conformance inspection, logical error in other subjective setting method is avoided. Experiment shows that QoS requirement changes among tasks can be accommodated better when different QoS weight is adopted for task of different type, and although some execution time is sacrificed, user satisfaction rate increases obviously.

2. PSO task scheduling supporting multi-QoS factor

2.1. Introduction to task scheduling based on PSO

In particle swarm optimization (PSO), particle searches in solution space, and particle position represents a solution, and it represents a task scheduling strategy in cloud system scheduling. Each particle updates its own position according to local experience and global experience to search for globally optimal solution. Standard PSO initializes an initial position randomly, and searches for optimal solution under the guidance of fitness function.

Task scheduling researched in this paper is "batch scheduling", which means that within a unit time set, n new tasks will be generated and wait for scheduling, instead of the phenomenon that a new task is generated and is scheduled immediately. V= [v1, v2,. . . .,vn] represents task scheduling vector, i.e. a scheduling scheme. For cloud storage system, v$i$represents that data of the ith task is provided by resource node represented by vi value and length of the vector is the total quantity of task needing to be scheduled within unit time.

For PSO algorithm, fitness value of PSO corresponded by the V shall be calculated after a group of V is generated, and following content shall be inspected: 1) whether the value researches to threshold value; 2) whether iterations reach to set value, and in the case of reaching of any one, stopping shall prevail, and V corresponded by the minimum fitness value shall be output, which is the solution generated finally. In cloud storage system, invalid solution shall be restricted. Refer to our work in the past for details, which is described in detail in literature [7]. But fitness function of existing algorithm is the same for all tasks, which is unreasonable obviously for task with different QoS requirements.

2.2. Definition of PSO fitness function constrained by multi-QoS factor

Specific computing method of fitness is defined differently in different application fields. To adapt to multi-QoS preference in cloud storage, PSO fitness function is redefined as follows:

$$f_{ij} = w_t\delta_t + w_c\delta_c + w_q\delta_q + w_s\delta_s\,. \tag{1}$$

Where $w_t + w_c + w_q + w_s = 1$.

Where t represents time, c represents cost, q represents quality, and s represents security. Factors in cloud system can be classified into the four types basically. Normalization processing of factor in the classification can eliminate dimension effect. W represents importance of various factors, and different QoS preference can be met by adjusting weight w of each type, which means that fitness function value is affected by different weight distribution. V corresponded by the minimum fitness function value shall be output similarly as scheduling scheme.

Present problem is division of task type and determination of specific w of each type.

3. User preference perception through analytic hierarchy process

3.1. Division of user task type

Existing cloud storage system task is classified as follows on the basis of difference of QoS requirements after analysis:

(1) Low **cost** requirement: quality and time are considered less, and cost shall be the minimum, and a great deal of ordinary task belong to the type. For example, when ordinary free network disk user downloads document slowly or wrongly without serious problems, he shall wait patiently or download again, which shall be marked as type C1.

(2) Task with strict **quality** requirement: cost can be relatively high and time can be relatively more as long as data is complete and correct. The typical example is transmission task of important but not urgent document, such as backup data, which shall be marked as type C2.

(3) Strict **time** requirement: the representative of the task is video where frame can be omitted in the middle part, but live transmission speed requirement is strict, which shall be marked as type C3.

(4) Task with strict **quality** and **time** requirement: cost shall not be considered, such as transmission of important and urgent document, which shall be marked as type C4.

(5) Task with strict **quality** and **security** requirement: the task is task with confidentiality requirement mostly, and quality guarantee will be required simultaneously necessarily, and it is few for individual confidentiality without quality requirement, and cost and time are not stressed, which shall be marked as type C5.

(6) Strict **quality** and **security** and urgency requirement: this kind of task corresponds to transmission of important and urgent document with confidentiality requirement, and generally speaking, cost will not be considered, which shall be marked as type C6.

3.2. Introduction to steps of weight setting through analytic hierarchy process

Analytic hierarchy process (AHP)[11] is to arrange complex evaluation object as an ordered hierarchical structure entirety, and to make two-two comparison and judgment among each evaluation item, and calculate relative importance factor of each evaluation item, i.e. weight. Because this paper uses analytic hierarchy process to weight determination instead of solution selection, construction of objective hierarchy can be omitted and it can be utilized directly to judge matrix calculation index weight. Implementation step for weight determination is introduced briefly as follows:

Step I, establish judgment matrix A, and make two-two comparison to evaluation index and its initial weight will form judgment matrix A, and element xij in the ith line and the jth column in judgment matrix A represents scale factor gained after comparison of xi index xj and. Scale means quantitative concept to each evaluation index importance level difference by evaluator. Common Saaty[11] level-9 scale is adopted in this paper, which is as shown in Table 1.

Table 1. Saaty level-9 scale method and its meaning

Scale aij	Definition (comparative factor i and j)
1	Factor i and j is equally important.
3	Factor i is more important than j slightly.
5	Factor i is more important than j in relatively strong degree.
7	Factor i is more important than j strongly.
9	Factor i is more important than j absolutely.
2, 4, 6, 8	The median of above 2 adjacent judgment.
Reciprocal of 1 $\sim$ 9	Judgment value gained is **aji=1/aij and aii=1** when factor j is compared with factor i.

Step II, calculate the maximum characteristic value and its corresponding characteristic vector of each judgment matrix compared in pairs, and make conformance inspection through conformance index, random conformance index and conformance rate. In the case of conformance of inspection, normalization characteristic vector $\{w_1, w_2, ..., w_n\}$ and $\sum\limits_{i=1}^{n} w = 1$ of the maximum characteristic root λ of matrix are weight vectors that represent each factor weight and that are required by us. In the case of failure, paired comparison matrix shall be reconstructed. Conformance is of great importance, and its meaning is that if factor a is more important than factor b, factor b is more important than factor c, then factor a shall be more important than factor c. If judgment matrix violates the conformance, then the judgment violates common sense, which will cause decision fault. Mathematical definition is that judgment matrix has transmissibility, which means that equality aij*ajk=aik is met, and when above equality is applicable to all elements, then the judgment matrix will be called as conformance matrix. In reality, nonconformance of certain degree is allowed.

Conformance inspection method is:

Conformance index is defined as: $CI = \frac{\lambda - n}{n-1}$

When CI=0, complete conformance will be gained; when CI is close to 0, satisfying conformance will be gained; the larger the CI is, the more serious the nonconformance will be.

Conformance rate is defined as: $CR = \frac{CI}{RI}$, and generally speaking, when conformance rate $CR = \frac{CI}{RI} < 0.1$, it is considered that nonconformance degree is within permissible range and satisfying conformance will be gained, so conformance inspection is conforming. Its normalization characteristic vector can be used as weight vector, and otherwise paired comparison matrix A shall be reconstructed, and aij shall be adjusted. Valuing of RI is as shown in Table 2.

Table 2. Random conformance index RI

n	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.58	0.90	1.12	1.24	1.31	1.41	1.45	1.49	1.5

3.3. Calculation of weight vector of task of each type

Four types of factors shall be arranged in the order of cost, time, quality and security to constitute two-two comparison judgment matrix. Finally, normalization characteristic vector of matrix passing conformance inspection is weight vector and value at its corresponding position is weight of corresponding factor.

Judgment matrix of C1-C6 is shown as follows:

$$c1 = \begin{bmatrix} 1 & 7 & 7 & 7 \\ 0.143 & 1 & 1 & 1 \\ 0.143 & 1 & 1 & 1 \\ 0.143 & 1 & 1 & 1 \end{bmatrix}.$$

Weight vector W1=[0.7 0.1 0.1 0.1]

$$c2 = \begin{bmatrix} 1 & 1 & 0.143 & 3 \\ 1 & 1 & 0.111 & 3 \\ 7 & 9 & 1 & 7 \\ 0.333 & 0.333 & 0.143 & 1 \end{bmatrix}.$$

Weight vector W2=[0.1202 0.1148 0.7088 0.0561]

$$c3 = \begin{bmatrix} 1 & 0.143 & 1 & 3 \\ 7 & 1 & 9 & 9 \\ 1 & 0.111 & 1 & 1 \\ 0.333 & 0.111 & 1 & 1 \end{bmatrix}.$$

Weight vector W3=[0.1248 0.7216 0.0869 0.0667]

$$c4 = \begin{bmatrix} 1 & 0.143 & 0.143 & 1 \\ 7 & 1 & 1 & 7 \\ 7 & 1 & 1 & 7 \\ 10.143 & 0.143 & 1 & \end{bmatrix}.$$

Weight vector W4=[0.0625 0.4375 0.4375 0.0625]

$$c5 = \begin{bmatrix} 1 & 0.333 & 0.111 & 0.111 \\ 3 & 1 & 0.1110.111 & \\ 9 & 9 & 1 & 1 \\ 9 & 9 & 1 & 1 \end{bmatrix}.$$

Weight vector W5=[0.0386 0.0682 0.4466 0.4466]

$$c6 = \begin{bmatrix} 1 & 0.111 & 0.111 & 0.111 \\ 9 & 1 & 1 & 1 \\ 9 & 1 & 1 & 1 \\ 9 & 1 & 1 & 1 \end{bmatrix}.$$

Weight vector W6=[0.0357 0.3214 0.3214 0.3214]

All above matrixes pass conformance inspection and specific adjustment process will not be described in detail because of restriction of the length. Weights of each type and conformance inspection value are as shown in Table 3.

Table 3. Conformance inspection value of judgment matrix of each type

	The maximum characteristic value	CI	CR
c1	4.0007	2.50E-04	2.81E-04
c2	4.1994	0.0665	0.0747
c3	4.1263	0.0421	0.0473
c4	4.001	3.33E-04	3.74E-04
c5	4.1533	0.0511	0.0511
c6	3.9992	-2.50E-04	-2.81E-04

3.4. PSO task scheduling integrating analytic hierarchy process

Traditional scheduling based on PSO algorithm is modified as shown in Fig.1, briefly called as AHP-PSO scheduling algorithm. Total flow is the same with that of traditional PSO algorithm, but weight in fitness function changes with change of task classification, which means that weight is adjusted in real time. To save adjustment time, in task transmission, we will input task on the basis of little classification after classification of task according to type for scheduling.

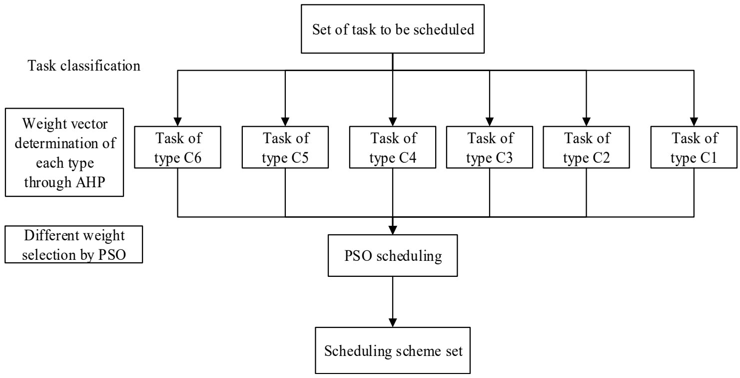

Fig. 1. Schematic diagram for algorithm flow

4. Experiment and analysis

4.1. Experimental environment

Experiment is implemented on the basis of analog system developed by Matlab. System is as shown in Fig.2, which is divided into 6 parts, i.e. "task generator", "node generator", "failure and disturbance generator", "updater for QoS value between node", "scheduler" and "evaluator".

"Task generator" generates task vector and QoS factor including task size and requirement. "Node generator" generates relationship between network node and node, i.e. network topology and attribute between nodes. "Failure and disturbance generator" simulates failure and disturbance that may appear in network, and these failures shall be abstracted as attribute between nodes directly, which means that aimed at attribute between nodes, numerous random numbers of Poisson distribution (Matlab internally installed) shall be set at the part, while "updater for QoS value between node" updates QoS value of node according to attribute value of node at previous time and random failure and disturbance generated here. "Scheduler" assigns node to task according to scheduling algorithm result, and updates node value through updater for scheduling again.

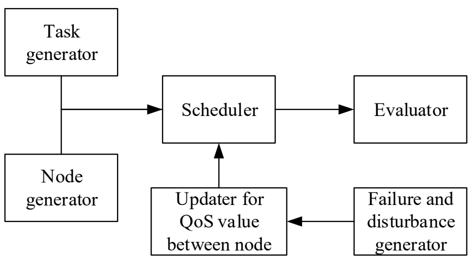

Fig. 2. Matlab simulation experiment system structure

Finally, "evaluator" judges whether user QoS requirement is met according to task vector requirement and scheduling distribution node attribute. Measurement index: 1) user satisfaction is equal to number of task with requirement met/total of task scheduled; 2) makespan is equal to total time for task completion.

4.2. Result and analysis

This paper mainly researches effect of AHP-PSO scheduling that integrates AHP and that adjusts weight within fitness function on the basis of task of different type in aspect of user satisfaction rate, algorithm execution time and total execution time of task set when compared with PSO scheduling with fixed weight value for the same task set. To avoid effect of some random factors, we repeat each experiment for three times. Number of task to be scheduled is defaulted as 100 and the maximum iterations are set as 100.

User satisfaction rate mostly concerned shall be researched firstly:

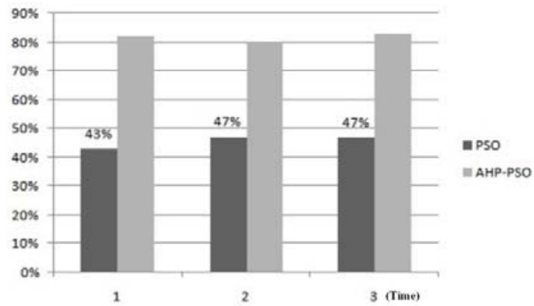

Fig. 3. Comparison of user satisfaction rate (Unit: Percentage)

It can be seen that user satisfaction rate increases obviously after AHP adjustment weight is integrated, increasing from original 40% to 80% roughly, which shows that in PSO scheduling, fitness function reflects QoS preference of task of different type effectively with changes of task of different type.

Execution time comparison of algorithm is as shown in Fig.4, and the time means the time from the moment when algorithm starts receiving to input task vector to be scheduled to the moment when an acceptable scheduling sequence is finally output.

Seen from Fig.4, compared with pure PSO, AHP-PSO needs more time, which is mainly the consumption by AHP-PSO algorithm because of different weight for different task in implementation. But time increases not in great degree, and with enlargement of task scale, the effect from weight adjustment will be less and less, and most time consumption is from PSO calculation.

We estimate weight adjustment consumption through difference of execution time of 2 algorithms, because difference of AHP-PSO algorithm and PSO scheduling lies in here. Then weight adjustment effect shall be measured through "time difference/PSO time value". Comparison is as shown in Table 4, and with increase of task scale index, the effect from weight adjustment will be less and less. Cloud system is aimed at a great deal of task scheduling generally, and therefore, effect of weight adjustment on time consumption of the entire scheduling process is not great.

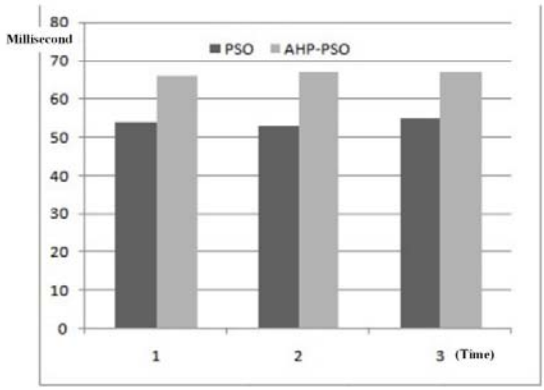

Fig. 4. Execution time comparison of algorithm (Unit: Millisecond)

Table 4. Relationship between time consumption proportion of adjustment weight and task scale

Task scale	10	100	1000
PSO	4	56	645
AHP-PSO	6	67	689
Proportion	50.00%	19.64%	6.82%

Finally, compare total execution time of task in scheduling strategy calculated by 2 algorithms, and as shown in Fig.5, total execution time of AHP-PSO scheduling is longer than that of traditional PSO algorithm generally, but the extent is not great, which is because objective of AHP-PSO algorithm for weight adjustment is to choose strategy with the minimum total execution time under the premise that user QoS is met while original PSO is at the cost of the minimum total execution time directly. Cloud system is a kind of service in essence, and in our opinion, if user satisfaction cannot be realized, no matter how minimum total execution time is, it will be meaningless. Seen from Fig.5, total execution time of AHP-PSO scheduling is 10% longer than that of traditional PSO maximally, and we think that it is worthy to increase user satisfaction rate with relatively little execution time.

5. Conclusion

Aimed at the low user satisfaction rate problem caused by the fact that existing PSO algorithm ignores different requirements of user task to QoS when applied to cloud storage, a kind of PSO scheduling algorithm adapting to different user task QoS requirements is proposed. Firstly, redefine fitness function of PSO algorithm and integrate multi-QoS factor support. Secondly, endow different weight values to these factors to reflect preference of user to different QoS. Determination of weight shall be based on analytic hierarchy process (AHP), and weight after conformance inspection can avoid logical error in other subjective setting method effectively. Experiment shows that QoS requirement changes among tasks can be accommodated better when

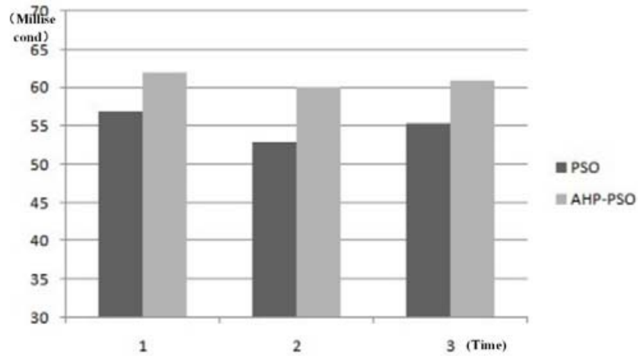

Fig. 5. Total execution time comparison of task (Unit: Millisecond)

different QoS weight is adopted for task of different type, and although some total execution time of task is sacrificed, user satisfaction rate increases obviously. This paper helps user to set weight through AHP, and seen from the experiment, it has better user satisfaction rate than that of traditional algorithm without adjustment weight. But whether AHP is the optimal weight determination method shall be researched further. We will compare different weight setting methods and their advantages and disadvantages to provide better guidance for weight setting.

References

[1] K. Matsumoto, T. Hayase, N. Ikeda: *Data Mining of Generalized Association Rules Using a Method of Partial-Match Retrieval*[J]. Lecture Notes in Computer Science *1721* (1999), 160–171.

[2] R. A. Alghamdi, M. Taileb, M. Ameen: *A new multimodal fusion method based on association rules mining for image retrieval*[C]// Melecon 2014 - 2014, IEEE Mediterranean Electrotechnical Conference. IEEE (2014), 493–499.

[3] A. Kouomou-Choupo, L. Berti-Equille, A. Morin: *Multimedia indexing and retrieval with features association rules mining*[C]// IEEE International Conference on Multimedia and Expo. IEEE Xplore *2* (2004), 1299–1302.

[4] D. W. Cheung, K. Hu, S. Xia: *An Adaptive Algorithm for Mining Association Rules on Shared-Memory Parallel Machines*[J]. Distributed and Parallel Databases *9* (2001), No. 2, 99–132.

[5] M. X.. Ribeiro, P. H. Bugatti, C. Traina, Jr: *Supporting content-based image retrieval and computer-aided diagnosis systems with association rule-based techniques* [J]. Data & Knowledge Engineering *68* (2009), No. 12, 1370–1382.

[6] C. K. Chiou, J. C. R. Tseng: *An Incremental Mining Algorithm for Association Rules Based on Minimal Perfect Hashing and Pruning*[M]// Web Technologies and Applications. Springer Berlin Heidelberg (2012).

[7] X. Liu, C. Hu, Z. Liu: *An Efficient Metadata Information Retrieval Model Based on Association Rules and Mutual Information*[J]. International Journal of Advancements in Computing Technology *4* (2012), No. 18, 40–48.

[8] S. Anekritmongkol, M. L. K. Kasamsan: *Boolean Algebra and Compression Technique for Association Rule Mining*[C]// Advanced Data Mining and Applications -, International Conference, Adma 2010, Chongqing, China , Proceedings. DBLP (Nov 2010), 150–157.

[9] J. Liu, S. Liu: *Semantic retrieval for remote sensing images using association rules mining*[J]. IEEE International Geoscience & Remote Sensing Symposium, IGARSS (2015), 509–512.

[10] J. Liu, S. Liu: *Semantic retrieval for remote sensing images using association rules mining*[C]// Geoscience and Remote Sensing Symposium. IEEE (2015), 509–512.

[11] R. Bueno, M. X. Ribeiro, A. J. M. Traina, et al.: *Improving medical image retrieval through multi-descriptor similarity functions and association rules*[C]// IEEE, International Symposium on Computer-Based Medical Systems. IEEE Computer Society (2010), 309–314.

[12] D. W. Cheung, Y. Xiao: *Effect of Data Distribution in Parallel Mining of Associations*[J]. Data Mining and Knowledge Discovery *3* (1999), No. 3, 291–314.

[13] R. He, N. Xiong, T. Kim, et al.: *Mining Cross-Modal Association Rules for Web Image Retrieval*[C]// International Symposium on Computer Science and ITS Applications. IEEE Computer Society (2008), 393–396.

[14] S. T. Klein: *Processing queries with metrical constraints in XML-based IR systems*[J]. Journal of the Association for Information Science and Technology *59* (2008), No. 1, 86–97.

Received May 7, 2017