http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (21): 6273-6276, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.6273.6276

Task Scheduling Strategies of Etl for Banking Off-site Audits

Wang Shan and Li Yue-Ping
Computer Engineering College, Shenzhen Polytechnic College, Shenzhen, 518055, China

Abstract: Facing massive data processing of off-site auditing systems in commercial bank, an optimal task scheduling method of ETL based on greedy algorithm is proposed. The shortest total task processing time can be achieved by this task scheduling method. Through the actual test, the high efficiency and stability of our task scheduling method based on greedy algorithm is proven.

Key words: Off-site audits for banks, ETL (extraction, transformation and loading), task scheduling, greedy algorithm, efficiency

INTRODUCTION

ETL (Extraction, Transformation and Loading) technology is not only the basis for building a data warehouse technology but also the basis for bulk data exchange technology (El Akkaoui *et al*., 2013). ETL is the process for data from the source repository into target format, including data extraction, transformation, integration, cleaning and loading. For a given application scenario, this process may contain several smaller task units and these units are divided into several task groups executing concurrently (Song *et al*., 2010).

A task unit may include a number of actions in ETL, such as: A Table of data from the source extraction, transport, transformation, cleaned and loaded into the target of the whole process, can be considered a unit of work; A certain data's integration process can be considered is a unit of work. In this article, a unit of work is considered to be a task. A unit of work is a relatively complete whole, contains a number of a time sequence of actions; other hand, there is no mandatory chronological order between different tasks, no matter which task is executed first, will not affect the final result. Shown in Fig. 1, we propose and define the three tasks.

As off-site audit system for banks are faced with massive data processing, the processing speed should be considered an important issue in the system development and how to solve task grouping strategy in order to get the optimal ETL workflow is an important measure to raise the running speed of the system.

In order to improve the efficiency of ETL, you need to optimize all aspects of ETL procedures and processes (Karagiannis *et al*., 2013). In this study, we analyze the execution time of each task to a reasonable allocation of tasks grouped so that all tasks will have a shortest total execution time.

Following, through the development research project of an off-site audit of a commercial bank ETL proposes a task scheduling design method based on greedy algorithm and tests various scheduling methods in the implementation of the project carried out, the results are analyzed and empirical.

TASK SCHEDULING ALGORITHM DESIGN

Here through a practical application of an off-site project audit system of a commercial bank, to study the greedy algorithm for task scheduling in order to achieve optimization of the ETL (Luo *et al*., 2012).

Basic principles of the greedy algorithm: Greedy algorithm is an approximation solution to the optimization problem. Each optimization problem contains a set of constraints and an optimization function. The problem-solving program meeting the constraints is called as feasible solution. The feasible solution which let the optimization function to obtain the optimum value is called as optimal solution.

In the greedy algorithm used gradually construct optimal approach. At each stage, we have made a seemingly optimal decision (under some certain criteria). Once you make a decision, it can't be changed any more. The guidelines that the greedy decisions based on is called greedy criterion, also known as the greed factor. The key lies in the setting of greedy criterion (Zhang, 2003) of a greedy algorithm.

Task scheduling algorithm based on greedy algorithm
Task scheduling problem description: An ETL include a number of tasks and know the time required for each task. In an ETL process, each task must and can only be performed once. Because the system

Corresponding Author: Wang Shan, Computer Engineering College, Shenzhen Polytechnic College, Shenzhen, 518055, China

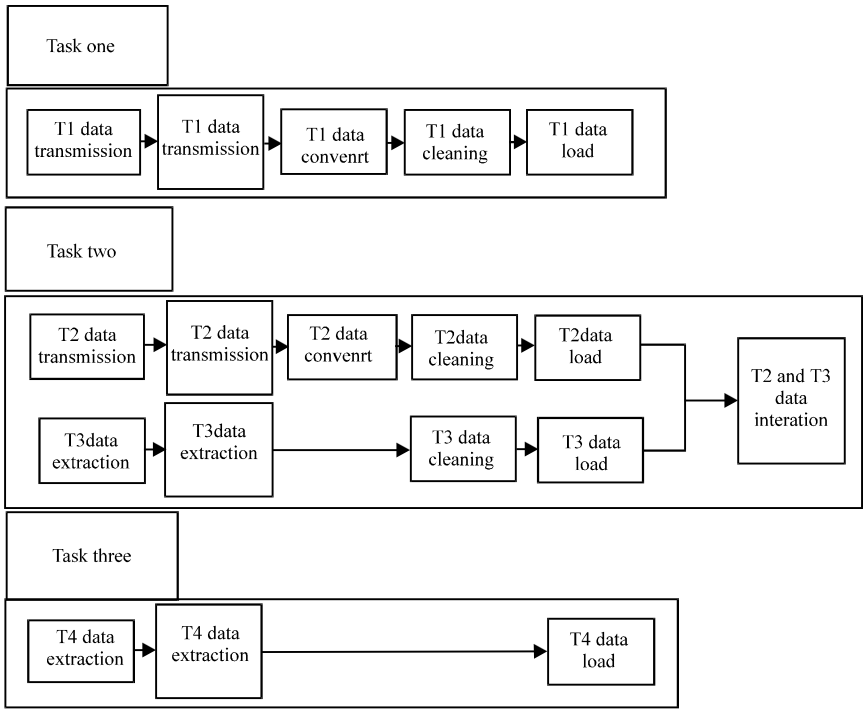

Fig. 1: Tasks of ETL

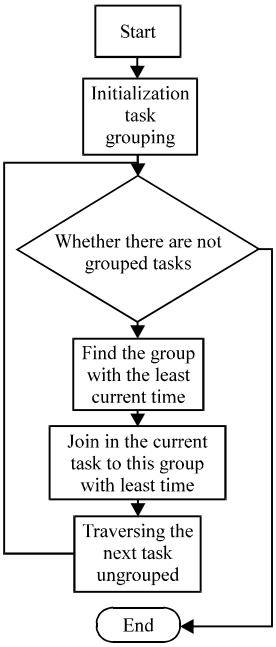

Fig. 2: Task scheduling flowchart designed basing on the greedy algorithm

resource constraints, provision the same time can be up to several tasks running simultaneously (in parallel) and when the task parallelism, sit does not change the task execution time required. Question: How to schedule tasks in order to shorten the total execution time.

Modeling of task scheduling problem: We proposed the following model to describe the task scheduling problem in a commercial bank off-site audit projects ETL development (Jorg and Deâloch, 2008).

Given:

- An ETL composed of m tasks, the task i is the need to spend time $t_i (1 = i = m)$
- In an ETL process, each task must and can only be performed once
- All tasks can be divided into *n* groups, each between tasks in parallel
- When the task parallelism, it does not change the task execution time
- Grouping k need total execution time of $T_k (1 = k = n)$, T_k is equal to the group of all time and the task execution

Solution:

- How to group tasks in order to make the total execution time T min, i.e. max (T_k) the minimum

Task scheduling problem solving: Task scheduling problem solving is based on the following assumptions:

- Assume that m>n. In fact, if m = n, these m tasks only need to execute parallel, then T = max (t_i) where 1 = i = m
- Without loss of generality, assume the task execution time in descending order, namely $t_1 \geq t_2 \geq \ldots \geq t_i \geq \ldots t_n$.
- Current total execution time of packet k is T_k(1 = k = n)
- Let Su grouping for all the tasks not yet incorporated into the queue, the queue of tasks by the execution time in descending order, i.e., the head of the task queue is always the longest execution time of the task

Grouping greedy algorithm steps are:

- Before the start packet, T_k = 0, where (1 = k = n). Su contains all the tasks
- Let the task in the head of the queue Su incorporated into the current minimum total time packet. Then we removes the task from the queue Su
- Repeat step ②, until the queue Su is empty
- The resulting packet is task scheduling problem solution

Grouping greedy algorithm pseudo-code is as follow:

sort (Su) for k ← 1 to n T_k ← {null} for i ← 1 to m
do min (T_k) ← Su (i)

The process is based on the task execution time in descending order, it is because in descending order of the obtained total execution time is always less than the obtained in ascending order of total execution time. In fact, considering the following tasks queue (value means each task's execution time) and the group number is 3:

10, 5, 3, 2

Then get grouped in descending order as follows:

Group 1:10
Group 2: 5
Group 3: 3, 2s
Total execution time: 10

If you follow in ascending order, get grouped into:

Group 1: 2, 10
Group 2: 3
Group 3: 5
Total execution time: 12

OPERATION RESULTS AND ANALYSIS

In a commercial bank off-site audit project, we designed a greedy algorithm task grouping procedures, In order to verify efficiency and correctness of the ETL task scheduling based on greedy algorithm, we test the research in the DCT system which is developed in this project.

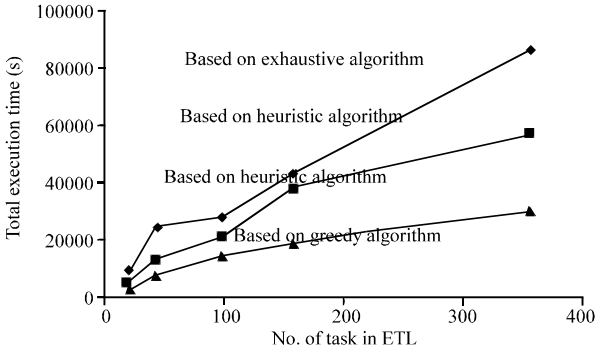

Fig. 3: Comparison of three scheduling algorithm systems' efficiency

Testing environment: First, we select the environment on some virtual machines. After the testing on those virtual machines successfully, we transform the program to physical environment and perform the test cases. Hardware environmen: 2 hosts, the RAM more than 256M and the hard disk spaces more than 100M. We can also use the VMWare virtual machine. Software environment£°the OS is SCO UNIX 5.0.6, the database DBMS is Informix 7.22.

Flowchart realizing task scheduling: The task scheduling flowchart designed basing on the greedy algorithm in the commercial bank off-site audit system ETL as below:

Testing data and analysis: In the factual system testing,we get the sample data of 43 tasks' performing time in some period of the day 2008.02.24. Due to space limitations, we only list 10 of these tasks, the start time of task is:2008.02.24-14:55:4,the over time is: 2008.02.24-16:24:40,the tasks' performing time is shown in the Table 1 as below.

Grouping strategy results got through the greedy algorithm is shown in the Table 2

We compare the operational efficiency through the test using the method of exhaustion, heuristic algorithms, greedy algorithms was verified in the ETL of the commercial bank off-site audit system, testing ETL workflows of different sizes and different ETL task number were selected as 20, 43, 98, 157, 356, the total execution time of three algorithms were got, as shown in Table 3. Actual verification proves that using the greedy algorithm the total execution time of the ETL task scheduling is shortest and the efficiency is the most optimal.

As shown in the Fig. 3, the analysis of the systems' running efficiency is like follows: The total performing

Table 1: Tasks performing time

Task	Task name	Start time	End time	Task with time (s)
T 1	Fully-storing kind dynamic table	14:55:43	14:55:57	15
T 2	Fully-overhead household registration book register	14:56:15	16:24:33	5299
....				
T 6	Fully-ledger-the private demand	15:00:29	17:18:48	8300
T 7	Fully-ledger-of the public demand	15:05:44	16:11:27	3944
T 8	Fully-cards-card files	15:06:37	17:14:59	7703
T 9	Fully-account-accounts	15:18:55	15:19:44	50
T 10	Fully-generalledger-subjects of ledger schedules	15:25:41	16:24:40	3540

Table 2: Scheduling plan

Group	Task	Total time
G1	T6	8300
G2	T8	7703
G3	T2	5299
G4	T7,T4,T5,T1...	5275
G5	T10,T3,T9...	5275s

Table 3:Total execution time of three algorithms

	Based on exhaustive algorithm	Based on heuristic algorithm	Based on greedy algorithm
No. of tasks in ETL	Total execution time (sec)	Total execution time (sec)	Total execution time (sec)
20	9563	5236	4105
43	24800	13210	8300
98	28404	21056	15397
157	43021	38532	19304
356	86549	56942	30453

time based of the exhaustive algorithm is the longest,that based on the Heuristic algorithm is the second,and that based on the greedy algorithm is the shortest and the execution time increase linearly as the number of task in the ETL workflow increases.

During the system running period,some cases can be confronted as follows: In the day T,the execution time is t1;in the day T+1, the execution time is t 2; in the day T+2, the execution time is t 3. And t 3>>t1, t 2, meanwhile t 1, t 2 are equal approximately. If we run group programs based on the last day's results,the group result of T+1 and T+2 may be not so satisfied. In order to avoid such problems,we use the average results of many days before as the reference value,and give tips to some accidental fluctuation like the results of T+1 (for example, the fluctuating time is more than 5 min and the amplitude is more than 5 percent) and compute those groups after processing.

The actual running result shows that, the task scheduling based on the corrected greedy algorithm is more effective and stable.

CONCLUSION

In the ETL of commercial bank off-site audit system, the task scheduling is a complicated optimization problem. Here, basing on the greedy algorithm, with the constraints of task time and concurrency, we propose some thoughts and methods solving such kind of problems. It's concise and efficient and it can improve the efficiency of ETL very effectively when we solve such problems as the optimal task grouping strategy. After actual measurement, it has a very practical value and is worth generalizing.

ACKNOWLEDGMENTS

This study was supported by Natural Science Foundation of China (NSFC50505036). The authors also thank for the support by Science and technology project in Shenzhen (06KJce037).

REFERENCES

El Akkaoui, Z., E. Zimanyi, J.N. Mazon and J. Trujillo, 2013. A BPMN-based design and maintenance framework for ETL processes. Int. J. Data Warehousing Mining, 9: 46-72.

Jorg, T. and S. Deâloch, 2008. Towards generating ETL processes for incremental loading. Proceedings of the International Symposium on Database Engineering and Applications, September 3-5, 2008, Munster, Germany, pp: 101-110.

Karagiannis, A., P. Vassiliadis and A. Simitsis, 2013. Scheduling strategies for efficient ETL execution. Inform. Syst., 38: 927-945.

Luo, H., W. Zhou, D. Ye and J. Yu, 2012. A buffer-based parallel ETL data flow processing framework. Comput. Appl. Software, 1: 88-91.

Song, X.D., X.Z. Zuo, X.B. Liu and X.L. Yan, 2010. Design ETL meta-model in data warehouses. Comput. Simul., 9: 106-108, 119.

Zhang, T., 2003. Sequential greedy approximation for certain convex optimization problems. IEEE Trans. Inform. Theory, 49: 682-691.