Query Processing in Distributed Data Warehouse using Scheduling Algorithms

S. Krishnaveni
Research Scholar, Dept. Computer Science
Karpagam University, Coimbatore, India

M. Hemalatha
Head, Dept. Computer Science
Karpagam University, Coimbatore, India

ABSTRACT

Data warehouse is a centralized repository for analyzing and storing huge amount of data. In distributed data warehouse, data can be shared across multiple data repositories which belong to one or more organizations. Query sorting is one of the issues for formatting the number of queries that can be selected together. Reducing the usual completion period of a random order is a common concern. In this paper, we are dealing three scheduling algorithms for query scheduling and the performance report based on processing time and memory size is also evaluated. The algorithms discussed are Optimal Resource Constraints (ORC), Grouping based Fine-grained Job Scheduling (GFJS) and Heuristic Algorithm (HA). ORC allocates queries according to their processor's capabilities. GFJS is based on resource characteristics. HA selects some possible schedules that are having the shortest sum of completion time and this set contains the optimal one.

Keywords: Data Warehouse, Optimal Resource Constraints (ORC), Grouping based Fine-grained Job Scheduling (GFJS), Heuristic Algorithm (HA)

1. INTRODUCTION

A data warehouse is an electronic storage of the huge amount of data which are uploaded from the operational systems in the data warehouse. The crucial mechanisms of a data warehousing system are,

- To retrieve and analyze the data
- To extract, transform and load data
- To manage data dictionary

The main applications are personal productivity, data query and reporting and planning and analysis. From these applications the data are shared with the set of user requirements.

Modern trends in distributed data warehouse have improved the significance of query selecting. In distribution logistics some of the large query quantities are being exchanged to number of small queries, which have to be practiced in extremely rigid time gaps.

Query selecting is the process of salvage items from their storage locations to fill customer orders, is known as the most time consuming and laborious component of the warehousing activities [6]. So the query selecting operation is a strong candidate for productivity improvement studies. Performance and competence of the query selecting operations are inclined by four vital factors, like warehouse layout, map-reading and sorting procedure, storage policy and grouping method[2].

Task scheduling is explained as the designing of tasks or queries to specific physical resources to reduce the cost function processed by the client. This is an NP-complete problem and different heuristics may be used to reach an optimal or near optimal solution[8]. Effective computation and task scheduling are rapidly becoming one of the main challenges in various computing systems and is seen as being vital for its success.

2. LITERATURE REVIEW

The simple allocation schemes such as First Fit back fills (FF) are used in practice[12]. In any transactions First in First out (FIFO) algorithm does not prioritize and transactions are performed based on their arrival time. Some recent algorithms support various resource allocations for a task and run to complete the scheduling. Scheduling procedures are based on First-Come-First-Serve (FCFS) algorithm[3] which allocates the resources for tasks based on their arrival time. The benefit of FCFS grants the level of determinism on the waiting time of each task[1].

In Resource Co-Allocation for Scheduling Tasks with Dependencies (RCSTD)[4] algorithm, each step combines the clusters based on the dependencies between the combined clusters. Therefore these clusters are combined if any dependencies exist between current and former clusters. The aim of this algorithm is to enhance the load balancing efficiency and minimum time for the execution of tasks. This algorithm minimizes the task execution time and it has a dynamic nature as a result of within a cluster the tasks are allocated to the appropriate resource on that it can be scheduled at the earliest time. This RCSTD algorithm found a sensible load balancing for all the resources for a set of tasks scheduled on each resource in the system. Cluster communication overhead and unspecified task requirements are the main drawback for this algorithm.

Optimal Resource Constraint (ORC) Scheduling[11] allocates tasks according to their processor's capabilities. ORC applies Round Robin (RR) scheduling and a Best Fit algorithm to distribute the tasks for available processors. It gives better performance than FCFS, SJF and RR. It reduces the average waiting time, turnaround time and minimizes the process allocation complexity. High communication overhead is the drawback for this algorithm.

Grouping-based Fine-grained Job Scheduling (GFJS) algorithm[7,14] is based on resource characteristics. This algorithm integrated with Greedy and FCFS algorithms improve the Fine-grained jobs. Then the coarse-grained tasks are formed by the grouping of fine-grained jobs. These coarse-grained are allocated to the available resources according to their capability (MIPS) and bandwidth (Mb/s). GFJS maximizes the resource utilization, reduces the task execution time, processing time and network latency. High pre-processing time and memory size constraint inconsiderable are the main drawbacks for this algorithm. The grouping strategy considers the processing power, memory size and bandwidth

requirements of each task realize the real grid system[13]. Provides a real grid computing environment and reduces the waiting time of the grouped tasks.

Heuristic algorithm[5] is used in experience based learning (EBL) for calculating the processing time. Heuristic algorithm is not considering all the possible schedules. It selects some possible schedules that are having the shortest sum of completion time and this set contains the optimal one. Patient scheduling is done by the use of this algorithm. This shows the patient's minimal waiting time in hospital and minimizes the total completion time.

The query grouping issue is essential for operating manual picker-to-parts query or task selecting systems in distribution warehouses efficiently. The proposed meta-heuristics[10] are related to the different capacities of selecting devices, antithetic routing policies and required scenarios. They suggest the researchers to focus the minimization of overall query or task selecting time for issues involving due dates. Another model for determining the optimal layout for minimizing the throughput time of a data warehouse is proposed[9]. Here the yields are randomly stored.

In this paper, we are implementing three scheduling algorithms and compare their performance in terms of time and memory size according to the number of queries.

3. SCHEDULING ALGORITHMS

Task scheduling is described as the designing of tasks or queries to specific physical resources to minimize the cost function and reduce the overall completion time processed by the client. It is one of the main challenges in distributed data warehouse system. Some of the scheduling algorithms are working on this paper.

A. Optimal Resource Constraint (ORC) Scheduling algorithm

ORC allocates tasks according to their processor's capabilities.

Steps

- Put all incoming queries into the queue pool
- Searches the all queries in the queue and check the resources and its capability
- The scheduler will allocate the queries to the resources based on its capability and tasks size
- The tasks put it in Round Robin queue and again search the resource's capability to process the queries
- After the completion of allotted queries by the resource, the queries are allotted to the best fit free resources
- This process will be continued until all the queries are completed.

B. Grouping-Based Fine-Grained Job Scheduling (GFJS) Algorithm

Based on the resource status, lightweight tasks are grouped as coarse-grained tasks consistent with processing capabilities (in MIPS) and also the bandwidth (in Mb/s) of the available resources.

Steps

- The scheduler receives the queries and gets resources status
- According to the size of queries, query_list is sorted in descending order
- Based on the resource status, small queries can be grouped as coarse-grained tasks according to processing capabilities and the bandwidth of the available resources
- Processing time of the coarse-grained task should not exceed the expected time
- Here only the processing capacity and bandwidth are used to constrain the sizes of coarse-grained
- If any new tasks come, it will be allocated to an appropriate resource without grouping, if a coarse-grained task running in that resource
- Then the fine-grained task can be grouped as several new tasks and this group size should be less than the capacity of available resources

C. Heuristic Algorithm (HA)

The query scheduling problem can be effectively solved by using this heuristic method. Thus the optimal schedule is having the minimum total weighted completion time and total tardiness is obtained.

Steps

- Scheduler collects the details of the number of queries, number of resources and the number of queries already assigned to each resource
- Calculate the weighted sum of total completion time and total delay of each resource then tries to minimize the weighted sum by rearranging list of queries
- Generate all possibilities for rearranging queries and resources to minimize the overall weighted sum
- Among these generated the correct possible scheduling sequences select one with the minimum weighted sum of total completion time and total delay
- Every iteration the weighted sum is calculated based on previous total completion time of query and delay of each resource.

4. IMPLEMENTATION AND RESULTS

We are using five computers with 512 MB RAM capacity for this work. The user can submit queries from different systems. Submitted queries split according to scheduling algorithms which are allocated into the different inter processors. After query processing is completed the results are collected from the inter processors and sent to the corresponding client.

The tasks or queries are randomly generated by the client or user in the distributed data warehouse environment. By the use of scheduling algorithms the client receives the appropriate query results in minimal processing time (in seconds) and also reduced levels of memory size (in bytes/1000). By considering various algorithms Heuristic Algorithm performs well.

A. Data set Description

In this work we are using food-mart data set. It contains twenty four relevant tables that are stored in MS Access database. These tables are randomly distributed into three various sites.

B. Performance Analysis

1) *Comparison of Scheduling Algorithms with Processing Time*: Table(I) shows the processing time (in seconds) and different number of queries for Optimal Resource Constraints

Scheduling Algorithm, Grouping-Based Fine-Grained Job Scheduling Algorithm and Heuristic Algorithm.

Processing Time of ORC, GFJS and HA scheduling algorithms with a number of queries are studied in Fig.1. This chart shows that the Heuristic Algorithm outperforms than Optimal Resource Constraint Scheduling algorithm and Grouping-Based Fine-Grained Job scheduling algorithm. In comparison to other algorithms, HA performs well with the total number of queries increases the processing time is also less.

Table I. Processing Time of number of queries

Number of Queries	Time (seconds)		
	ORC	***GFJS***	***HA***
10	15	10	6
20	20	18	11
30	21	18	13
40	21	19	19
50	25	24	24
60	40	33	30
70	40	38	32
80	48	44	38
90	63	50	44

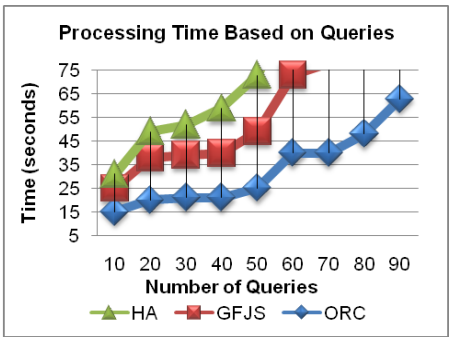

Fig.1. Performance of Scheduling Algorithms with Processing Time

2) *Comparison of Scheduling Algorithms with Memory Size:* Table(II) shows the memory size (in bytes/1000) and different number of queries for Optimal Resource Constraints Scheduling Algorithm, Grouping-Based Fine-Grained Job Scheduling Algorithm and Heuristic Algorithm.

Table II. Memory Size of number of queries

Number of Queries	Memory Size (bytes/1000)		
	ORC	***GFJS***	***HA***
10	150	90	70
20	280	190	150
30	430	280	150
40	600	370	220
50	750	460	280
60	750	570	340
70	880	680	420
80	1050	770	480
90	1180	870	560

The memory size of ORC, GFJS and HA scheduling algorithms with a number of queries are studied in Fig. 2. This chart shows that the Heuristic Algorithm outperforms than ORC and GFJS Scheduling algorithms. In comparison to other algorithms, HA performs well with the total number of queries increases the memory allocation is also less.

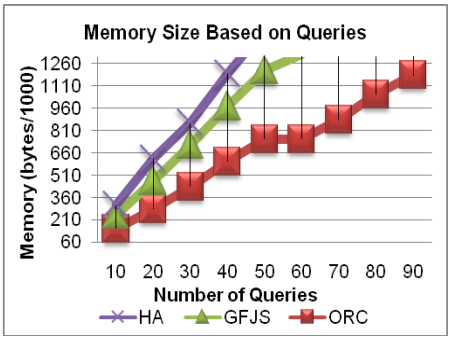

Fig.2 Performance of Scheduling Algorithms with Processing Time

5. CONCLUSION

In this paper, we surveyed various scheduling algorithms which are used in data warehousing as well as grid computing. We have used food-mart data set with relevant tables that are selected randomly with which it is distributed to various sites (inter processors). From the result we conclude that the processing time (in seconds) and memory size (in bytes/1000) with respect to the number of queries is reduced in using HA. In the above performance analysis we found that the Heuristic Algorithm shows better performance than the other two algorithms. Our future work will be focused to develop a new scheduling algorithm for minimizing the processing time and memory size better than heuristic algorithm.

6. ACKNOWLEDGMENT

We thank the Karpagam University for the Motivation and Encouragement to make this work as successful one.

7. REFERENCES

[1] Carsten Ernemann, Volker Hamscher, Uwe Schwiegelshohn and Ramin Yahyapour, "On Advantageous of Grid Computing for Parallel Job Scheduling," 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 39-46, 2002.

[2] C.G. Petersen, "An Evaluation of Order Picking Routing Policies," International Journal of Operations & Production Management, vol. 17, Iss. 11, pp. 1098–1111, 1997.

[3] Claus Bitten, Joern Gehring, Uwe Schwiegelshohn and Ramin Yahyapour, "The NRW-Metacomputer-Building Block for a Worldwide Computational Grid," 9th Heterogeneous Computing Workshop, pp. 31-40, 2000.

[4] Diana Moise, Izabela Moise, Florin Pop and Valentin Cristea, "Resource CoAllocation for Scheduling Tasks with Dependencies in Grid," International Workshop on High Performance in Grid Middleware, pp. 41-48, 2008.

[5] E. Grace Mary Kanaga, M.L. Valarmathi and Juliet A Murali, "Agent Based Patient Scheduling Using Heuristic Algorithm," International Journal on Computer Science and Engineering vol. 2, pp. 69-75, 2010.

[6] J.A. Tompkins, J.A. White, Y.A. Bozer and J.M.A.T. Tanchoco, "Facilities Planning," John Wiley and Sons, New York: chap. 7, pp. 432-444, 2003.

[7] Quan Liu and Yeqing Liao, "Grouping-Based Fine-grained Job Scheduling in Grid Computing," 1st IEEE International Workshop on Education Technology and Computer Science, pp. 556-559, 2009.

[8] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra and Prachet Bhuyan, "A Survey of Job Scheduling and Resource Management in Grid Computing," World Academy of Science, Engineering and Technology, vol. 64, pp. 461-466, 2010.

[9] Roodbergen, K.J., De Koster, R., 2001. Routing Methods for Warehouses with Multiple Cross Aisles. International Journal of Production Research 39 (9), 1865–1883.

[10] Sebastian Henn and Gerhard Wäscher, "Tabu Search Heuristics for the Order Batching Problem in Manual Order Picking Systems," European Journal of Operational Research, Accepted manuscript, pp. 1-31, 2012.

[11] Somasundaram, K. and S. Radhakrishnan, "Node Allocation in Grid Computing using Optimal Resource Constraint (ORC) Scheduling," International Journal of Computer Science and Network Security, vol. 8, Iss. 6, pp. 309-313, 2008.

[12] Vijay Subramani, Rajkumar Kettimuthu, Srividya Srinivasan, Sadayappan, P., 2002. Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests. 11th IEEE International Symposium on High Performance Distributed Computing, 359-366.

[13] Vishnu Kant Soni, Raksha Sharma and Manoj Kumar Mishra, "Grouping-Based Job Scheduling Model in Grid Computing," World Academy of Science, Engineering and Technology, vol. 65, pp. 781-784, 2010.

[14] Yeqing Liao and Quan Liu, "Research on Fine-grained Job Scheduling in Grid Computing," International Journal of Information Engineering and Electronic Business, pp. 9-16, 2009.

8. AUTHOR'S PROFILE

Dr. M. Hemalatha completed M.Sc., M.C.A., M. Phil., Ph.D (Ph.D, Mother Terasa women's University, Kodaikanal). She is Professor & Head and guiding Ph.D Scholars in Department of Computer Science at Karpagam University, Coimbatore. Twelve years of experience in teaching and published more than hundred papers in International Journals and also presented more than eighty papers in various national and international conferences. She received best researcher award in the year 2012 from Karpagam University. Her research areas include Data Mining, Image Processing, Computer Networks, Cloud Computing, Software Engineering, Bioinformatics and Neural Network. She is a reviewer in several National and International Journals.

S. Krishnaveni completed M.C.A., M.Phil. and currently pursuing Ph.D in computer science at Karpagam University under the guidance of Dr.M.Hemalatha, Professor and Head, Dept. of Software System, Karpagam University, Coimbatore. Published five papers in International Journals and presented one paper in national conference and one paper in the international conference. Area of research is Data Mining, Data Warehouse, Computer Networks and Grid Computing.