ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

Simultaneous Updation in Mass Data Storage by Using Job Separation in Data Warehouse

V. R. Sofia Chandrasekar[1], Dr. J.Jagadeesan[2]

M.Tech (CSE) Student, Department of CSE, SRM University, Ramapuram, Chennai, Tamil Nadu, India[1]

Head Of the Department, Department of CSE, SRM University, Ramapuram, Chennai, Tamil Nadu, India[2]

Abstract: Simultaneous update and dynamic scheduling in a data warehouse which the motto of traditional data warehouses. The main motto is to focus on continuous job into mass data storage as data warehouses for pupation of jobs are to minimize the loss of freshness. The skeleton would not be based on the deadline of the incoming jobs. Heterogeneous sources provides the continuous jobs to update in the corresponding their tables in data warehouse. The skeleton scheduling algorithms for updating in mass data storage in the data warehouse. In this proposed work Dynamic Priority Driven Preemptive Scheduling (DPDPS) algorithm is used to assign different priority level at different jobs. DPDPS job separation algorithm shows the tracks to get the jobs for an updating a job and for improving the performance to meet their job completion on time. Equal job separation method provides the jobs into the clusters according to the similar jobs in each track.

Keywords: Data Warehouse, Scheduling algorithm, Job partitioning, Extraction Transformation Loading (ETL), OLAP.

I. INTRODUCTION

Data warehouse incorporate the information from multiple operational databases to enable complex business analysis. The objective of a data warehouse is to propagate new data across all the relevant tables and views as quickly as possible. Once new data are loaded, the applications and triggers defined on the warehouse can take immediate action. Real-time scheduling is a well-studied topic with a lengthy literature. A typical hard and soft real time system, jobs must be completed before their deadlines a simple metric to understand and to prove results.

In a firm real-time system, jobs can miss their deadlines, and if they do, they are discarded. Heterogeneous sources are external sources which contains multiple sources which are not same with one another. Extraction Transformation Loading (ETL) provides the extraction of the source job, transforming the job and loading the job into data warehouse. Data warehouse provides the larger storage space for having the historical data as well as the present data. It makes the decision support system for real time applications. Online analytical Processing (OLAP) gives multidimensional cubic data storage which provides drill down the job and drill up the job.

On-line analytical processing (OLAP) that is, analysis techniques with functionalities such as summarization, consolidation, and aggregation as well as the ability to view information from different angles. Although OLAP tools support multidimensional analysis and decision making, additional data analysis tools are required for in-depth analysis, such a data classification and the characterization of data changes over time. (Fig.1).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

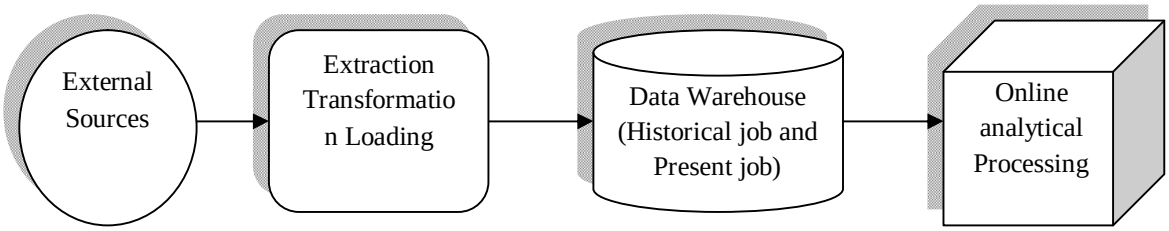

Fig.1 ETL Process Structure

II. RELATED WORK

In [1] an algorithm provides a bound on total staleness; it may still starve some tables. The complexity of scheduling data-loading jobs to minimize the staleness of a real time stream warehouse. To prove that online non preemptive algorithm that is never voluntarily idle achieves a constant competitive ratio with respect to the total staleness of all tables in the warehouse, provided that the processors are sufficiently fast. In [2] author uses a set of novel scheduling techniques that address scheduler overheads by batching approximation, and pre-computation. The train scheduling and super box scheduling help a lot to reduce system overheads. The overheads are affected in a running stream data manager. In particular, these algorithms require tuning parameters like train size and super box traversal methods. To extending the scheduling techniques to distributed environments and other resources in the context of Aurora. In[3] author used an adaptive update policy to balance potentially conflicting miss ratio and Freshness requirements. Initially, all data are updated immediately when their new sensor readings arrive. A novel real-time main memory database architecture called QMF. QMF achieved a significant performance improvement compared to several Baselines including best existing algorithms for miss ratio and freshness trade-off in real-time databases, while supporting the target miss ratio and freshness. As one of the first work on QoS management in real-time databases, the significance of the work will increase as the demand for real-time data services increases.

III. PROPOSED SYSTEM

Our skeleton is giving the solution for non- preemptively scheduling problems in a real time. Our skeleton provides the assigning jobs in the separate track based on their arrival times. Solving the overload problems using short job takes the execution times by cutting the long job into small for some interval.

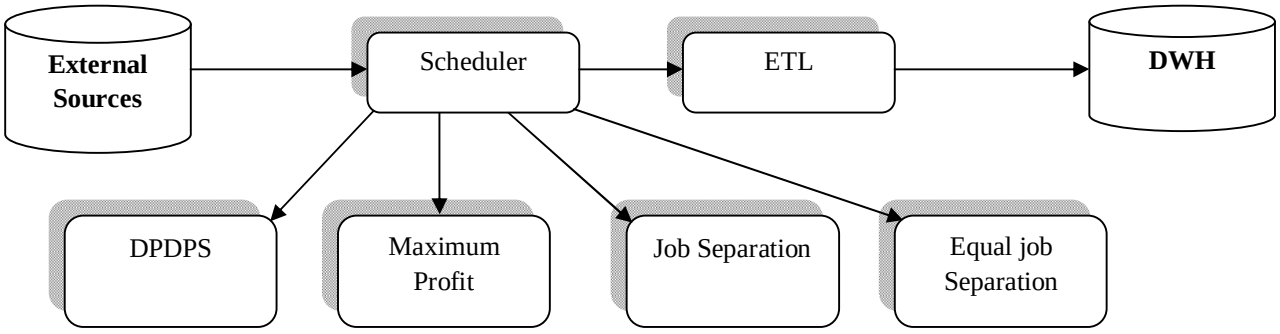

Fig.2 System Architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

The traditional systems are refreshing their updates after closing the program or while doing shutdown but in real time data warehouses are going to be updated whenever job is coming. The current system is not supporting the decision making and not to give the maintenance of the warehouses. The current system does not support the limit the counting of tables to update. High priority jobs are blocked by the low priority due to priority inversion problem the high priority jobs may miss their finishing time. In the proposed system architecture describes the various components discussed as follows:

External Sources: External sources are heterogeneous databases (Flat files, DB2, Sybase etc.) which propagate to the all relevant job tables .heterogeneous databases are streaming warehouses that increasing the fastness of the Extraction Transformation Loading. External sources are pushing the jobs to the data warehouse. If the job scheduling time is not predictable means the user can specify the period for the particular job.

Extraction Transformation Loading (ETL): Extraction Transformation Loading extracts the jobs from the external sources the transforms the jobs for cleansing and then loads into the data warehouse as target. ETL supports the huge job integration and hierarchical transformation. Extraction Transformation Loading satisfies the complex business rules with huge volume of the data. ETL processes are grouping the job which is executed as a batch job (Fig.3).

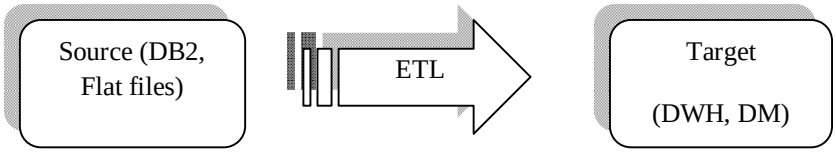

Fig.3 Source files conversion process

Scheduler: Scheduler limits the updating of tables and avoids overloading process .It Schedules job execution based on time. Scheduler is responsible for monitoring jobs and managing jobs in a clustering environment. Capability of a scheduler is to schedule a job to run at a particular date and time. The Scheduler reduces the operating costs. When jobs are coming from the multiple sources existing system does not provide limiting the counting of tables to updates .Number of updates in parallel process take the CPU utilization is very high and provides the poor performance. The solution is to be Scheduler will provide the limiting the job going to update in the tables and provides mechanism for choosing the job to be update further.

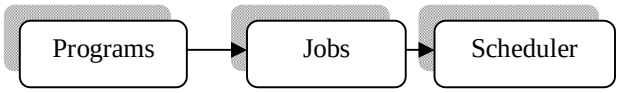

Fig.4 Scheduling formation

Scheduling algorithms ensures the partitioning the job into tracks according to the resources available .A track consists of one or more jobs to run based on their execution time with priority based.

IV .ALGORITHM IMPLEMENTATION

Dynamic Priority Driven Preemptive Scheduling (DPDPS): Dynamic scheduling algorithms are static priority as a Rate Monotonic(RM) scheduling, Dynamic priority as a Earliest deadline First(EDF) scheduling and adaptive scheduling as a feedback control. This skeleton provides the priority allocation done in dynamically. A small job has taken as the higher priority. Job definition will be as follows:

void func (void)_job_num

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

```
void job0 (void)_job_0
{       while (1)
    {     counter 0++;
    } }
```

To find the job J deadline is calculated by arrival time (a) plus period (p) i.e. J=a+p. The CPU utilization will be calculated as the execution time of the job E which is divided by the period (p).The Utilization is defined by $U=\Sigma E/P \leq 1$.

Maximum Profit: Scheduler minimizes the loss of freshness of the table. The profit of the job J is calculated as $p\Delta F$. The overall profit is calculated based on unit of time as $p\Delta F/E(\Delta F)$.Greedy heuristic determines none of the tracks remain idle .Overall Profit do not depend on their period.

Job Separation: Job separation enhances the overall performance .From external sources jobs are pushed from its base table and then it scheduled and separated in the track based on the period of the job. Global scheduling provides many jobs in the single track. Two mechanism for allocating the resources for high priority job EDF separation and equal separation.

DPDPS Job Separation: DPDPS job separation algorithm assigns the jobs in the tracks. Each track has appropriate schedule to decide which job is next to update. All jobs should meet their deadlines and there is no missing job. All jobs are going as a batch jobs. Each job should completed before the next one comes to enter .First sort the job according to their periods then assigns into the tracks. If the particular track is busy means it can preempt to another new track otherwise it can be placed in the own track.

- Utilization of job is calculated as U=E/(P)/P
- Utilization of the track calculated by $Utr=\Sigma U$

From the track of assigning jobs we can find out the maximum jobs and minimum jobs based on the processing times. Track preemption assigns the higher priority jobs in the free track because which do not waste the resources.

Equal Job Separation: Equal job separation gives the jobs to be in clusters which have the similar jobs. So can achieve the equal sharing resources .Ranking the jobs according to the execution time .If the job is smaller than the parameter q then the job will be into cluster one .If not new cluster will be created when the job is larger than the parameter q.

V.PSEUDO CODE IDENTICATION FOR CLUSTERING

Pseudo code for job placing in the tracks:

Step1. Sort the jobs

Step2. If the jobs J own track is free, place the job in the track

Step3. Else search any free track is available

Step4. If yes place the job in the available track

Step5. Else preempt to the another job's own track when there is no pending jobs in the track

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

Step6. Otherwise put the job in waiting mode

Pseudo code for identification of clusters of similar jobs:

Step1: Sort the jobs based on the processing time

Step2: Create a new cluster Cu1

Step3:If the Processing time(E/P) of the job is less than x times

Step4: Then add job J to the cluster Cu1

Step5: Else create a new cluster and add the job J into the new cluster

Step6: Else delay the processing of job J

VI. SIMULATION AND RESULTS

The simulation work gives the simultaneous jobs arrival and checking the usage of the tracks .Fig(5) shows the effect of halt based on the performance and correlative tardiness .When the job is coming with freshness it allotted by the track then the simulator starts the work to find the processing time of jobs.Pr(i)=as+bs* i where i is the interval time of the job loading ,as is the ETL initialization and bs is the the arraival time of the job.If the processing time in the slow up period then it is be multiplied by the slow up constant (O).i.e O*Pr(i).Adjusting the U value to the performance(Ptot).Using this experiments number of tracks won't affect the performance..Fig(6) shows the priorities of classes which takes first class is one and second class is 10 then the prioritized algorithm as maximum profit experience the lower delay.Fig(7) shows comparison of algorithms which gives the equal job separation algorithm is the best from the results with the clustering constant C=1.without job separation the job will provide the larger delay but compare with job separation maximum profit algorithm determines the larger tardiness. Job separation algorithm is good when at less performance of the resources but not good in the higher performance and wastage of resources too.

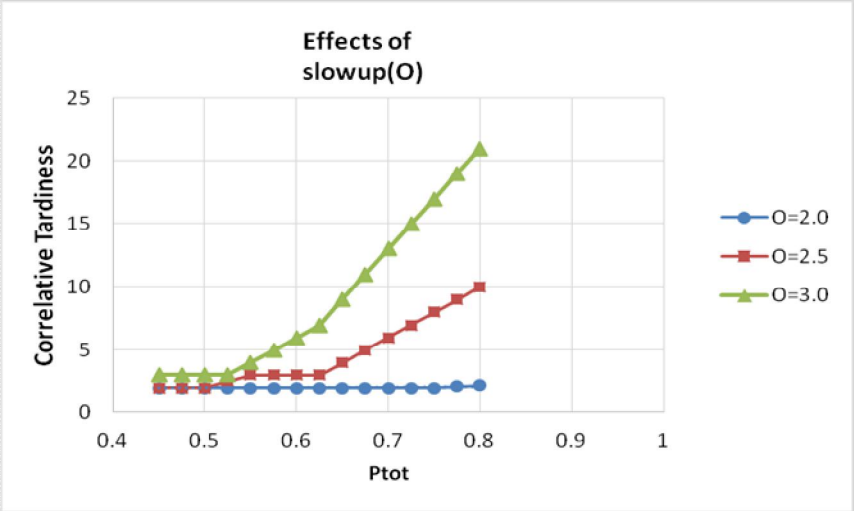

Fig.5 Effects of slowup

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

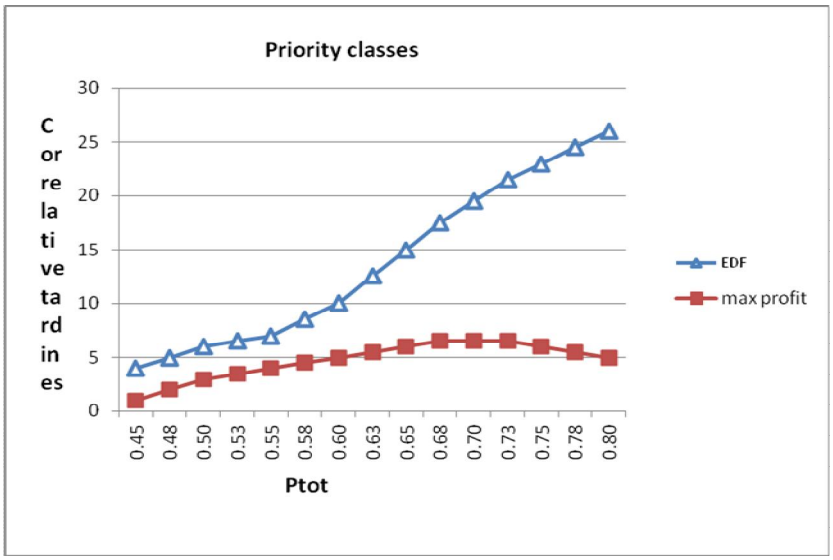

Fig.6 Effects of Priority classes

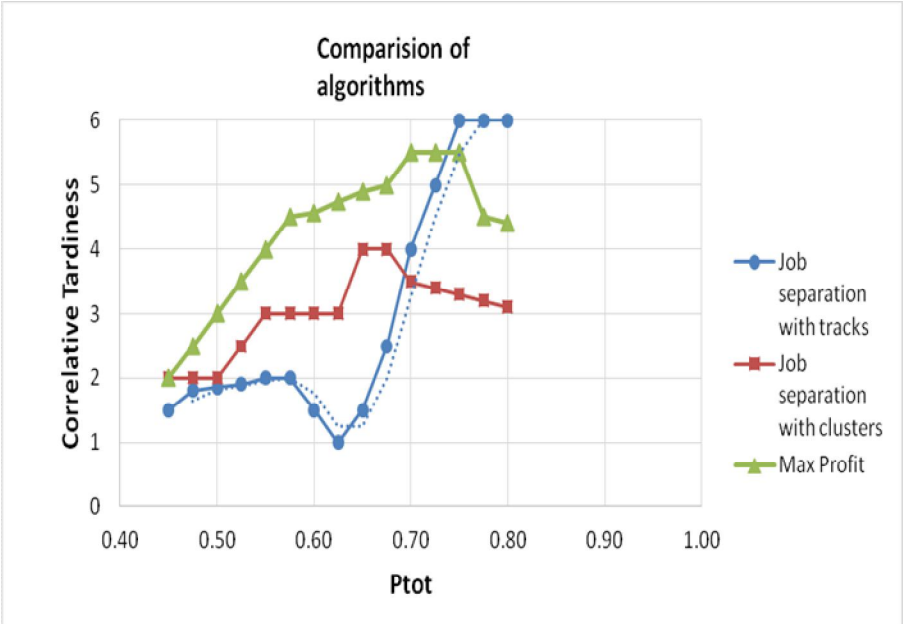

Fig.7 Comparision of Algorithms

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

VII. CONCLUSION

In this paper we motivated, formalized manner by using Dynamic Priority Driven Preemptive Scheduling (DPDPS) and used algorithm of preemptively scheduling updates in real time system in data warehouse. The job of the scheduler is to decide, of all the released tasks having a non-zero freshness delta, which one to execute next on one of the available CPUs. It is known that EDF is an optimal hard real-time scheduling algorithm on a single processor with respect to maximizing the number of tasks that meet their deadlines, if the tasks are pre-emptible scheduling algorithm used and the future work involves dynamic scheduling algorithm to explore in the simultaneous updating in the mass data storage by using job separation in the data warehouse. Equal job separation specifies the best CPU performance and implemented by Data Depot to determine the values for each job which provided the graph for performance utilizations.

VIII. FUTURE ENHANCEMENTS

In future to use round-robin algorithm for scheduling. "round-robin" is a method of choosing a resource for a task from a list of available resources, usually for the purposes of load balancing. This is one of the simplest scheduling algorithms for processes in an operating system. As the term is generally used, time slices are assigned to each process in equal portions and in circular order, handling all processes. In order to schedule processes fairly, a round-robin scheduler generally employs time-sharing, giving each job a time slot or quantum (its allowance of CPU time).The future work involves choosing right scheduling algorithm to explore tradeoffs between update efficiency and effective manner.

REFERENCES

1) M.H.Bateni,L.Golab,M.T.Hajiaghayi,andH.Karloff, “Scheduling to Minimize Staleness and Stretch in Real-time Data Warehouses,”Proc. 21st Ann. Symp. Parallelism in Algorithms and Architectures (SPAA),pp. 29-38, 2009.
2) B. Babcock, S. Babu, M. Datar, and R. Motwani, “Chain: Operator Scheduling for Memory Minimization in Data Stream Systems,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 253-264, 2003.
3) K. D. Kang, S. H. Son, and J. A. Stankovic,” Managing Deadline Miss Ratio and Sensor Data Freshness in Real-Time Databases”, IEEE Transactions on Knowledge and Data Engineering, Volume 16, Number 10, pages 1200-1216. October 2004.
4) A. Labrinidis and N. Roussopoulos, “Update Propagation Strategies for Improving the Quality of Data on the Web,” Proc. 27th Int’l Conf. Very Large Data Bases (VLDB),pp. 391-400, 2001.
5) Y. Oh and S.H. Son, “Tight Performance Bounds of Heuristics for a Real-Time Scheduling Problem,” Technical Report CS-93-24, U. Virginia, 1993.
6) M. Sharaf, P. Chrysanthis, A. Labrinidis, and K. Pruhs, “Algorithms and Metrics for Processing Multiple Heterogeneous Continuous Queries,” ACM Trans. Database Systems, vol. 33, no. 1, pp. 1-44, 2008.
7) C. Thomsen, T.B. Pedersen, and W. Lehner, “RiTE: Providing On-Demand Data for Right-Time Data Warehousing,”Proc. IEEE 24th Int’l Conf. Data Eng. (ICDE), pp. 456-465, 2008.
8) H. Qu and A. Labrinidis, “Preference-Aware Query and Update Scheduling in Web-Databases,” Proc. IEEE 23rd Int’l Conf. Data Eng. (ICDE), pp. 356-365, 2007.
9) Y. Zhuge, J. Wiener, and H. Garcia-Molina, “Multiple View Consistency for Data Warehousing,” Proc. IEEE 13th Int’l Conf. Data Eng. (ICDE), pp. 289-300, 1997.