Journal of Theoretical and Applied Information Technology
31st January 2013. Vol. 47 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

QUERY MANAGEMENT IN DATA WAREHOUSE USING VIRTUAL MACHINE FAULT TOLERANT RESOURCE SCHEDULING ALGORITHM

[1]S. KRISHNAVENI and [2*]M. HEMALATHA
1, 2* Department of Computer Science, Karpagam University, Coimbatore, India
E-mail: [1]sss.veni@gmail.com , [2*]csresearchhema@gmail.com

ABSTRACT

Multi-stage processing occurred in distributed data warehouse. Many joints and splits are required at the time of submitting queries. Scheduling algorithms are used to resolve these issues. Reducing the processing time and cost are common concerns in scheduling. By this way the available resources are used to accomplish the task with quick manner. In this paper, we propose Virtual Machine Fault Tolerance Resource Scheduling (VMFTRS) Algorithm for scheduling the resources. Processing time, memory size and Replication Metric are taken as performance metrics and each one calculated depending upon the given query. Performance results show that the proposed VMFTRS algorithm gives better result than other existing algorithms. It is also adaptable for all distributed data warehouse systems.

Keywords: *Data Warehouse, Resource Scheduling (RS), Research On Novel Dynamic Resource Management (RNDRM), New Resource Mechanism With Negotiate Solution (NRMNS), Virtual Computing Grid Using Resource Pooling (VCGRP), Virtual Machine Fault Tolerant Resource Scheduling (VMFTRS)*

1. INTRODUCTION

Data warehouse technology is fruitfully implemented in various commercial projects. Stored data are uploaded from the operational systems in the data warehouse, where the data can go through an operational data for supplementary operations before it is used for reporting. It also concentrates the exploitation of the available computing power and catering for the intermittent demands of large computational exercises. Main data warehousing applications are

- Personal productivity
- Data query and reporting
- Planning and analysis

Statistical packages, spreadsheets and graphics tools are personal productivity applications that are used in individual computers for manipulating and presenting data. Small amount of warehouse data required to develop a standalone environment. Distributive warehouse data are accessed by data query and reporting applications which are listed-oriented queries and they provide an overview of historical data. The planning analysis applications share a set of user requirements. They cannot be met by applying query tools against the historical data maintained in the warehouse repository.

Some of the major challenges in a distributed environment are site autonomy, heterogeneous substrate, policy extensibility, co-allocation and online control. Modern trends in distributed data warehouse have improved the significance of query selecting. In distribution logistics some of the large quantity queries are being exchanged by a number of small queries, which have to be practiced in extremely rigid time gaps. Distributed data warehouse containing more than two local data warehouses at each collection point and coordinator site. If the client gives queries in distributed system, query processing performed at local sites [1]. Skalla system is designed for distributed data warehouse to evaluate OLAP queries. Skalla translates OLAP queries to reduce the amount of data that needs to be shipped among sites.

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 [2]. 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 [3].

Organizing the resources between various tasks is called as scheduling. Scheduling is classified into two types. They are task or job or operation scheduling and resource scheduling. Task scheduling is 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 [4].

The resource scheduling is the process of harmonizing a query for resources based on required characteristics. The resource scheduling process is critical, when the client or user try to construct information as quickly and reliably [5-9]. In general resource scheduling in large-scale grids can be very challenging because the resources are not centrally controlled as well as they can enter and leave from the system at any time.

This paper is structured as follows. The next section deals the related works. Section 3 gives problem definition. Section 4 presents the proposed Virtual Machine Fault Tolerant Resource Scheduling (VMFTRS) algorithm. Section 5 deals about dataset. Section 6 mention the performance measures mentioned in section 6. Section 7 shows an experimental analysis of all algorithms. Finally, in section 8 we conclude our work and also mentioned future work.

2. RELATED WORKS

We survey various scheduling algorithms that focuses resource scheduling (RS), query selecting, time and cost minimizations, etc. Performance has exposed their number of jobs with respect to processing time, memory size and replication metrics. The simple allocation schemes such as First Fit back fills (FF) are used in practice [10]. 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 [11] which allocates the resources for tasks based on their arrival time. The benefit of FCFS grants the level of determinism for the waiting time of each task [12]. Demerit of FCFS shows, the tasks in the ready queue cannot be scheduled immediately due to lacking of resources but the tasks in the queue would be able to execute given the currently accessible resources. These latter tasks are blocked from executing while the system resources are remaining idle [13].

In distribution logistics few-but-large quantity orders are being replaced by many-but-small orders. The optimal routing policy shown in [14] for a warehouse with multiple cross aisles that can be found by using dynamic programming. Roodbergen proposed a model for determining the optimal layout for minimizing the throughput time of a data warehouse [15]; here the yields are randomly stored. Construction Management in Decision Support System (CMDSS) can provide exact and timely information to support project managers in construction decision-making [16].

Gademann et al., [17] think about the issue of grouping queries to reduce the entire travel time for a multiple-aisle selecter-to-part warehouse. They illustrate the problem is still NP-hard in the strong sense when the quantity of orders per batch is better than two. A branch-and-price algorithm is intended to solve instances of modest size to optimality. For larger instances, it is instructed to use an iterated descent approximation algorithm.

An approach for determining the optimal selecting batch size to order-pickers in a typical 2-block warehouse found by [18]. It is a simple but efficient approach also supports the average waiting time of a random order is a convex perform of the group size. It is difficult to capture the impact of aisle blockage, composite-Poisson arrivals or other storage methods and various layouts.

More than 300 papers are surveyed by [19]. They classified the literature on setup time consistent with store environments, group and non-group setup times, sequence-dependent and independent setup times (costs), and task and group availability models. Also they mentioned the issues of resource-dependent task and setup constraints, task and set-up corrosion, and task or group transportation. They suggest that future researches have to concentrate a specific solution method.

A client order scheduling problem deals with [20], wherever every order contains a set of tasks that must be shipped as one group at the same time. They proposed a new Minimum Flow Time Variation (MFV) dispatching rule for client order scheduling in a normal task shop to minimize the total completion time of all tasks within the same order. This rule will efficiently minimize the finished goods' storage level and controls the waiting time before they can be shipped but it does not concentrate the finished time.

Research on Novel Dynamic Resource Management (RNDRM) scheduling algorithm is a model of agent based dynamic resource management system proposed by [21]. It provides a good paradigm for integrating agent technology with grid computing based applications. RNDRM is Heap Sort Tree (HST) based algorithm. HST is constructed with the use of two layers. They are Autonomy Representation Agent (ARA) and Node State Monitoring Agent (NSMA). The combination of both ARA and NSMA called as Grid Resource Management Agents (GRMA). ARA activated on very high computational availability resource. NSMA deployed and activated on all resources. They examined this model by submitting sample tasks and concluded that this algorithm is feasible, rational, dynamic, robust, scalable, efficient, good load balance and high performance. So we used this algorithm in the distributed data warehouse environment for minimizing the processing time and reduce the cost.

Another agent based resource management system model is New Resource Mechanism with Negotiate Solution (NRMNS) algorithm innovated by [22]. NRMNS is the combination of Grid Architecture for Computational Economy (GRACE) and Resource Pricing Fluctuation Manager (RPFM) within the Agent Based Resource Management with Alternate Solution (ABRMAS). They evaluate the success rate between with negotiation and without negotiation. Their final result shows that the negotiation method increases 10% success rate of resource discovery. It gave an alternate solution at the time of a resource discovery malfunction. This algorithm is highly dynamic in grid environment. For this reason we worked with this algorithm to improve the resources success rate.

Virtual Computing Grid using Resource Pooling (VCGRP) is based on the loosely coupled model [23]. The system can select a resource and distribute tasks to it. Virtual Computing Grid Portal (VCGP) and Virtual Computing Grid Monitor (VCGM) are used in this system to minimize the complexity. Major work is to exploit the ideal computing power with less effort. So we are using the VCGRP algorithm in the data warehouse for reducing the resource complexity.

The query grouping issue deals with [24]. That is essential for operating manual picker-to-parts query or task selecting systems in distribution warehouses efficiently. The proposed meta-heuristics 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.

3. PROBLEM DEFINITION

When distributed data warehouse has been using, resources are available for data collection. At the time of collecting information, few demerits were occurring such as lack of error information notification, high processing time and high cost. To address these conflicts we proposed Virtual Machine Fault Tolerant Resource Scheduling (VMFTRS) algorithm. The main aim of this algorithm is recycling the virtual machines. i.e., when the primary virtual machine might be failing, the secondary virtual machine is created by the use of Reconfigurable Virtual Machine (RVM) or Physical Resources (PRs).

4. PROPOSED VIRTUAL MACHINE FAULT TOLERANT RESOURCE SCHEDULING (VMFTRS) ALGORITHM

To start with, initialize the virtual machines (VMs) and physical resources (PRs). Then allocate queries by using VCGRP algorithm and check the status of virtual machines. If any virtual machine is failed then sort all other virtual machines based on MIPS (Million Instruction Per Seconds) availability and check query size. If the query size is greater than the first virtual machine then queries are allocated for the particular virtual machine. Else compare all other virtual machines with query size. If the state then unsatisfied checks all possible combinations of virtual machines. If the condition is satisfied the queries are allocated otherwise create the new virtual machine from physical resource or combination of physical resources until all the queries are allocated. The architecture for the proposed VMFTRS algorithm is shown in Fig 1.

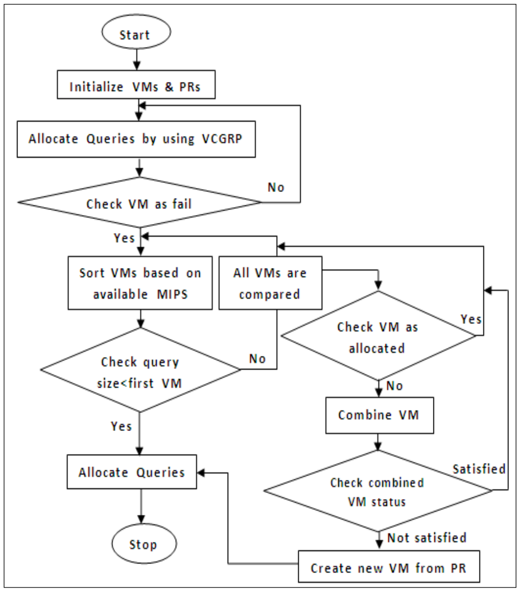

Figure 1. Architecture of Proposed VMFTRS Algorithm

4.1 Proposed VMFTRS Algorithm Steps

1. Initialize the number of VM reconfigurable, number of physical machines
2. Allocate queries as per VCGRP algorithm
3. If any VM fails, check available virtual machines and sort them according to the available MIPS (MIPS are free after their own allocation) and stored in an array
4. Available MIPS of the first Virtual machine in the array is greater than the query size the query is allocated to that virtual machine
5. Check is any other virtual machine failed. If any check availability of MIPs from the available VMs in the array. Then allocate VMs if satisfied with query size
6. If available MIPs from current VMs are not meeting query requirement, the available MIPS of two virtual machines combined and check the requirement for query processing. If it is not satisfied then combined free MIPS of three Virtual machines and so on
7. If the MIPS of reconfigurable virtual machines is not meeting the requirement and then the MIPS from combination of virtual machines not meeting the requirement then the VMs are created from available physical machines.
8. The VMs may be created from a single physical machine alone or from a combination of physical machines which is based on availability of MIPs on physical machines
9. Repeat step 3 until meets the requirement and for new queries

5. DATASET DESCRIPTION

The dataset for the work contains food mart data of 2010 and 2011. It contains twenty four relevant tables. The table names are Account, Category, Currency, Customer, Days-check, Department, Employee, Expense_fact, Position, Inventory_fact_2010, Inventory_fact_2011, Product, Product_class, Promotion, Region, Reserve_employee, Salary, Sales_fact_2010, Store, Time_by_day, Sales_fact_2010, Sales_fact_dec_2011, Warehouse and Warehouse_class. Total number of records are 3,20,835. In our work, these tables are randomly distributed into different sites.

6. PERFORMANCE MEASURES

For evaluating our proposed VMFTRS algorithm with other existing RS algorithms, we are using three performance measures. They are processing time, memory size and replication metric. Calculation of each metric is given below,

6.1 Processing Time

The difference between query completion time and query initialization time is mentioned as processing time.

Processing Time = Q_c - Q_i

Q_c - Query completion time

Q_i - Query initialization time

6.2 Memory Size

The variation between total memory utilization and pre memory utilization at the time of query processing is mentioned as memory size.

Memory Size = M_t - M_p

M_t – Total memory utilization

M_p– Pre memory utilization

6.3 Replication Metric

Occasionally same queries are repeated in various sites or resources. The count of same query repetition is declared as replication metric.

Replication Metric = $\sum_{j=1}^{n}\sum_{i=1}^{n}(Q_i \in R_i)\cap(Q_j \in R_j)$

Q_i = number of queries given

R_i = number of resources used

7. RESULTS AND DISCUSSION

Queries are randomly generated by the client in a distributed data warehouse environment. Submitted queries are allocated into different inter-processors according to scheduling algorithms. After query processing is completed the results are collected from inter-processors by the server and sent to the corresponding client. This work deals with performance of four RS algorithms namely Research on Novel Dynamic Resource Management (RNDRM), New Resource Mechanism with Negotiate Solution (NRMNS), Virtual Computing Grid using Resource Pooling (VCGRP) and proposed Virtual Machine Fault Tolerant Resource Scheduling (VMFTRS) algorithms. Results are compared based on processing time, memory size and replication metric.

7.1 Results of Proposed VMFTRS Algorithm

Table 1. illustrates the processing time, memory size and replication metric values for different number of queries such as 10, 20, 30, 40 and 50 for the proposed VMFTRS algorithm.

Table 1. Performance Values For Proposed VMFTRS Algorithm

Performances of Proposed VMFTRS Algorithm			
Number of Queries	*Processing Time (seconds)*	*Memory Size (Kb)*	*Replication Metric*
10	1	39.1	0
20	2	78.1	0
30	4	107.4	1
40	6	136.7	1
50	8	175.8	1

7.2 Comparative Analysis of Various Scheduling Algorithms with Processing Time

Table 2. shows the processing time values for different number of queries such as 10, 20, 30, 40 and 50 for RNDRM, NRMNS, VCGRP and proposed VMFTRS resource scheduling algorithms. A unit of the processing time is taken as seconds.

Table 2. Processing Time Of Various RS Algorithms

Number of Queries	Processing Time (seconds)			
	RNDRM	*NRMNS*	*VCGRP*	*Proposed VMFTRS*
10	5	3	2	1
20	10	6	4	2
30	13	10	7	4
40	18	14	10	6
50	21	18	12	8

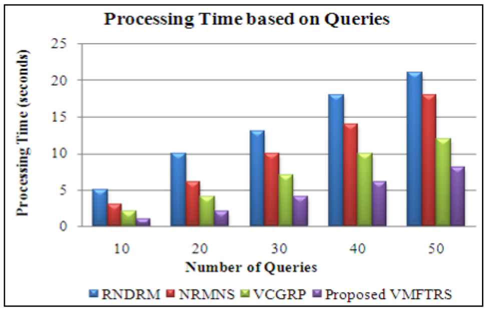

Figure 2. Performance of RS Algorithms with Processing Time

Fig. 2 shows the performance chart of processing time of RNDRM, NRMNS, VCGRP and proposed VMFTRS algorithms. The proposed VMFTRS algorithm takes 1 second for 10 queries, 2 seconds for 20 queries, 4 seconds for 30 queries, 6 seconds for 40 queries and 8 seconds for 50 queries. While the total number of queries increases the processing time is less in the proposed VMFTRS algorithm than other existing resource scheduling algorithms. Finally, it concludes that the proposed VMFTRS algorithm outperforms than others.

7.3 Comparative Analysis of Various Scheduling Algorithms with Memory Size

Table 3. depicts the memory size values for different number of queries such as 10, 20, 30, 40 and 50 for RNDRM, NRMNS, VCGRP and proposed VMFTRS resource scheduling algorithms. A unit of the memory size is taken as kilobytes (Kb).

Table 3. Memory Size Of Various RS Algorithms

Number of Queries	Memory Size (Kb)			
	RNDRM	*NRMN S*	*VCGRP*	*Proposed VMFTRS*
10	58.6	48.8	39.1	39.1
20	117.2	97.7	87.9	78.1
30	185.5	156.3	127	107.4
40	244.1	205.1	166	136.7
50	302.7	253.9	214.8	175.8

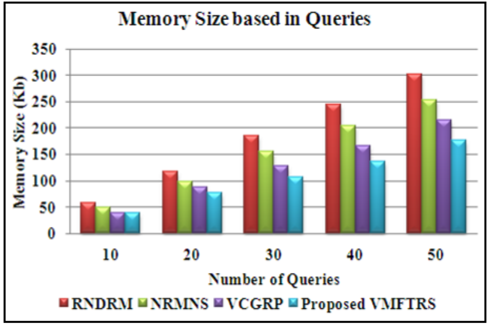

Figure 3. Performance Of RS Algorithms With Memory Size

Fig. 3 shows the performance chart of memory size for RNDRM, NRMNS, VCGRP and proposed VMFTRS algorithms. The proposed VMFTRS algorithm takes 39.1 Kb for 10 queries, 78.1 Kb for 20 queries, 107.4 Kb for 30 queries, 136.7 Kb for 40 queries and 175.8 Kb for 50 queries. While the total number of queries increases the memory size low in the proposed VMFTRS algorithm than other existing resource scheduling algorithms. Finally, it concludes that the proposed VMFTRS algorithm outperforms than others.

7.4 Comparative Analysis of Various Scheduling Algorithms with Replication Metric

Table 4. shows the replication metric values for RNDRM, NRMNS, VCGRP and proposed VMFTRS resource scheduling algorithms when we gave 25 queries.

Table 4. Replication Metric of various RS Algorithms

Algorithms	*Replication Metric*
RNDRM	3
NRMNS	3
VCGRP	2
Proposed VMFTRS	1

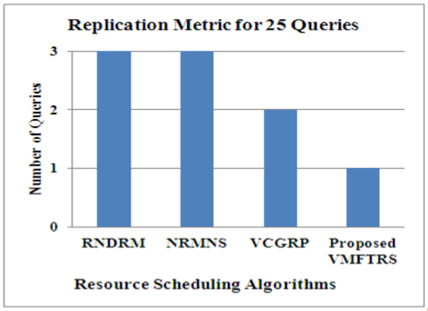

Figure 4. Performance of Replication Metric for 25 Queries

Fig. 4 shows the performance chart of replication metric values for RNDRM, NRMNS, VCGRP and proposed VMFTRS algorithms. The proposed VMFTRS algorithm gives 1 query when we are checking with 25 queries. The replication metric is minimized in the proposed VMFTRS algorithm than other existing resource scheduling algorithms. Finally, it concludes that the proposed VMFTRS algorithm outperforms than others.

8. CONCLUSION AND FUTURE WORK

Based on our work, we have concluded that the existing grid based resource scheduling algorithms are working fine in a distributed data warehouse environment. Even though there are few demerits in existing algorithms such as lack of error information at the time of job submission, high processing time, error in resource discovery and high cost. Our proposed Virtual Machine Fault Tolerant Resource Scheduling (VMFTRS) algorithm overcomes these issues and reduce the of query repetition, minimize the processing time and minimize the memory utilization. From the above performance analysis, we found that the proposed VMFTRS algorithm outperforms than other algorithms because of the virtual machine recycling process. Our future work will be based on the integration of resource scheduling algorithm with job scheduling to minimize the cost, increase the scalability and success rate.

ACKNOWLEDGEMENT

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

REFERENCES:

[1] M.O. Akinde, M.H. Bhlen, T. Johnson, L.V.S. Lakshmanan and D. Srivastava, “Efficient OLAP Query Processing in Distributed Data Warehouses”, *Information Systems*, Vol. 28, 2003, pp. 111-135.

[2] J.A. Tompkins, J.A. White, Y.A. Bozer and J.M.A.T. Tanchoco, “Warehouse Operations”, *Facilities Planning*, 4th Edition, Pub. John Wiley & Sons, Chapter 7, 2003, pp. 383-448.

[3] C.G. Petersen, “An Evaluation of Order Picking Routing Policies”, *International Journal of Operations and Production Management*, Vol. 17, No. 11, 1997, pp. 1098–1111.

[4] 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, 2010, pp. 461-466.

[5] K. Ranganathan and I. Foster, “Simulation studies of computation and data scheduling algorithms for data grids”, *Journal of Grid Computing*, Vol. 1, No. 1, 2003, pp. 53-62.

[6] Ian Foster and Carl Kesselman, “The Grid: Blueprint for a New Computing Infrastructure”, *Elsevier Inc.*, 2nd Edition, 2003.

[7] M. Caramia, S. Giordani and A. Iovanella, “Grid Scheduling by on-line Rectangle Packing”, *International Journal of Networks (Wiley Periodicals)*, Vol. 44, No. 2, 2004, pp. 106-119.

[8] Y. Gao, H. Rong and J. Huang, "Adaptive Grid Job Scheduling with Genetic Algorithms", *Future Generation Computer Systems,* Vol. 21, 2005. pp. 151-161.

[9] C. Weng and X. Lu, "Heuristic scheduling for bag-of-tasks application in combination with QoS in the computational grid", *Future Generation Computer Systems,* Vol. 21, 2005. pp. 271-280.

[10] Vijay Subramani, Rajkumar Kettimuthu, Srividya Srinivasan and S. Sadayappan, "Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests", *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*, IEEE Xplore Press, Nov. 7, 2002, pp. 359-366.

[11] Claus Bitten, Joern Gehring, Uwe Schwiegelshohn and Ramin Yahyapour, "The NRW-Metacomputer-Building Block for a World-wide Computational Grid", *Proceedings of the 9th Heterogeneous Computing Workshop*, IEEE Xplore Press, Cancun, Aug. 6, 2000, pp. 31-40.

[12] Carsten Ernemann, Volker Hamscher, Uwe Schwiegelshohn and Ramin Yahyapour, "On Advantageous of Grid Computing for Parallel Job Scheduling", *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, IEEE Xplore Press, May 21-24, 2002, pp. 39-46.

[13] Hongzhang Shan, Leonid Oliker and Rupak Biswas, "Job Superscheduler Architecture and Performance in Computational Grid Environments", *Proceedings of the ACM/IEEE Conference on Supercomputing*, IEEE Xplore Press, Nov. 15-21, 2003, pp. 44-58.

[14] K.J. Roodbergen and R. De Koster, "Routing Methods for Warehouses with Multiple Cross Aisles", *International Journal of Production Research,* Vol. 39, No. 9, 2001, pp. 1865-1883.

[15] K.J. Roodbergen, "Layout and Routing Methods for Warehouses", *Ph.D. Thesis. Erasmus Research Institute of Management (ERIM)*, Erasmus University Rotterdam, The Netherlands, 2001.

[16] K.W. Chau, Ying Cao, M. Anson and Jianping Zhang, "Application of Data Warehouse and Decision Support System in Construction Management", *Automation in Construction*, Vol. 12, No. 2, 2002, pp. 213-224.

[17] N. Gademann and S. Van de Velde, "Order Batching to Minimize Total Travel Time in a Parallel-Aisle Warehouse", *IIE Transactions,* Vol. 37, 2005, pp. 63-75.

[18] Tho Le-Duc and Rene M.B.M. de Koster, "Travel Time Estimation and Order Batching in a 2-block Warehouse", *European Journal of Operational Research,* Vol. 176, No. 1, 2007. pp. 374–388.

[19] Ali Allahverdi, C.T. Ng, T.C.E. Cheng and Mikhail Y. Kovalyov, "A Survey of Scheduling Problems with Setup Times or Costs", *European Journal of Operational Research*, Vol. 187, 2008, pp. 985-1032.

[20] Sheng Yuan Hsu and C.H. Liu, "Improving the Delivery Efficiency of the Customer Order Scheduling Problem in a Job Shop", *Computers and Industrial Engineering,* Vol. 57, 2009, pp. 856–866.

[21] Fufang Li, Deyu Qi, Limin Zhang, Xianguang Zhang and Zhili Zhang, "Research on Novel Dynamic Resource Management and Job Scheduling in Grid Computing", *Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences,* IEEE Xplore Press, Jun. 20-24, Vol. 1, 2006, pp. 709-713.

[22] Junyan Wang, Yuebin Xu, Guanfeng Liu, Zhenkuan Pan and Yongsheng Hao, "New Resource Discovery Mechanism with Negotiate Solution Based on Agent in Grid Environments", *Proceedings of the 3rd International Conference on Grid and Pervasive Computing Workshops*, IEEE Xplore Press, 2008, pp. 23-28.

[23] Alpana Rajan, Anil Rawat and Rajesh Kumar Verma, "Virtual Computing Grid using Resource Pooling", *Proceedings of the International Conference on Information Technology,* IEEE Xplore Press, Bhubaneswar, Dec. 17-20, 2008, pp: 59-64. pp. 59-64.

[24] Sebastian Henn and Gerhard Wascher, "Tabu Search Heuristics for the Order Batching Problem in Manual Order Picking Systems", *European Journal of Operational Research,* Vol. 222, No. 3, 2012, pp. 484-494.