IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

Scheduling Algorithms in Distributed Data Warehouse Environment: A Survey

Krishnaveni Sakkarapani[1], Sathish Ayyaswamy[2]

Assistant Professor, Department of Computer Applications, Pioneer College of Arts & Science, Coimbatore, India[1]

Assistant Professor, Department of Computer Science, Maharaja Arts and Science College, Coimbatore, India[2]

Abstract: The data are actuality distributed across many data warehouses in various systems at different sites are collected by the distributed data warehouse system and transformed into a single view which support the decision makers to create queries, perform analysis and make reports. In a spread environment multi-stage processing is performed which involves many joint and fragmented operations to be performed when processing the queries thereby increasing the processing time. Scheduling algorithms are used to determination these issues. The query sorting technique is commonly used for formatting the number of queries and grouped together. This idea describes prior art and some important work done in the research areas of data warehouse, information collected, content-based data retrieval and query processing. It borrows techniques from these fields and the purpose is to allow the reader to place their contributions in the context of previous work done in these areas and also makes a comparative analysis of existing solution strategies. Also this paper discuss about the usage of grid based algorithms in data warehouse environment at the time of scheduling tasks as well as resources.

Keywords: Data warehouse, scheduling algorithms, grid computing.

I. INTRODUCTION

A. Data warehouse

This A data warehouse is a repository or store house of integrated information, available for queries and analysis. Data and information are extracted using data gathering techniques from heterogeneous sources as they are generated. This makes it much easier and more efficient to display queries over data that initially came from different sources [15].

According to Bill Inmon (Father of data warehouse), a data warehouse is a "subject-oriented, integrated, time varying, non-volatile collection of data that is used primarily in organizational decision making".

Data warehousing applications have exploded in the 1990s and the gratitude of the need for these applications is not at all new. The need for data warehousing came up in the mid-to-late 1980s with the prime recognition that information systems must be renewed into operational and information systems [5].

Many data warehouse solutions are comprehensive and their support can be involved in many of these categories such as, Decision Support Systems (DSS), On-Line Analytical Processing (OLAP), Executive Information Systems (EIS), Managed Reporting Environments (MRE), Managed Query Environments (MQE), Data Mining and Data Visualization.

B. Data Warehouse Architecture

Data warehouse consists of tools which are used to extract data from multiple operational databases and external sources for altering, cleaning and integrating this data for loading data into the data warehouse and for rarely revitalizing the warehouse to reflect updates at the sources and to eliminate data from the warehouse, perhaps onto slower archival storage. Fig. 1 illustrates the overall architecture of the data warehouse.

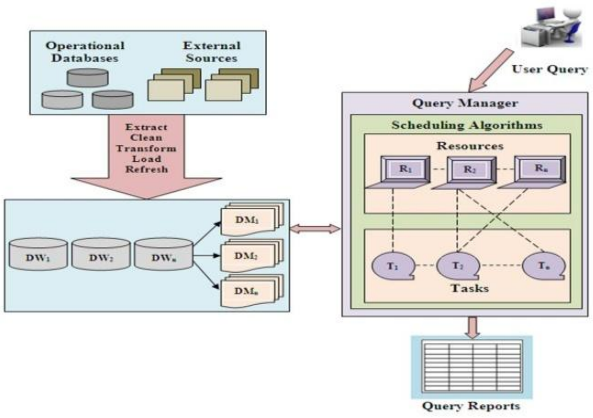

Fig. 1. Data Warehouse Architecture

The sharing of data can be implied in the common format. When queries are posed to the data warehouse environment, the query manager directs queries to the suitable tables. These queries are then mapped and sent to processors which display the query reports by the use of task and resource scheduling algorithms. Finally, the repository is mainly used for storing and dealing metadata and also contains tools for monitoring and administering the warehousing system.

Consistency and recoverability of the database are critical and maximizing transaction throughput. As a result, the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

database is measured as duplicate and the operational semantics of known applications leads to minimize concurrency conflicts.

C. Data Warehouse Components

A data warehouse has four main components such as operational systems of record, the data staging area, the data presentation area and data access tools [24]. Every component of the data warehouse serves a single function in preparing data for manipulation and examination. Fig. 2 defines the back-end process of the data warehouse.

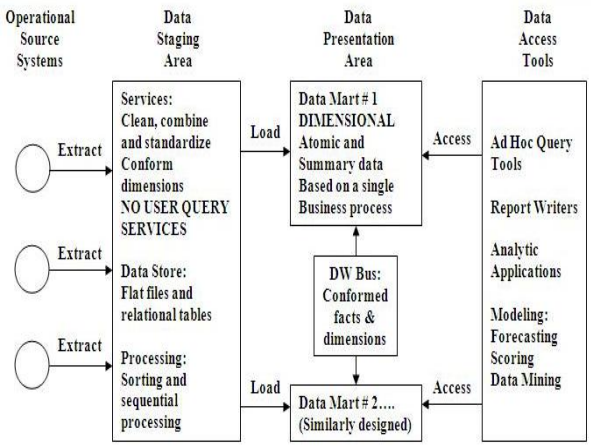

Fig. 2. Basic Data Warehouse Elements

D. Data Warehouse Information

Data warehouse information must be integrated, accurate, easily accessible, credible and timely. Information must be enterprise absorbed and support all possible Business Intelligence (BI) analyses in all technologies. It should be intended to load massive amounts of data in very short amounts of time and for optimal data extraction. Information must be dependable, flexible, stable and non-redundant.

E. Distributed Data Warehouse

The warehouse stimulates load balancing, scalability and higher convenience. The metadata repository is replicated with fragments of the warehouse and the entire warehouse is administered centrally. An alternative architecture implemented for convenience, can be used in times of high costs to construct a single logically integrated enterprise warehouse.

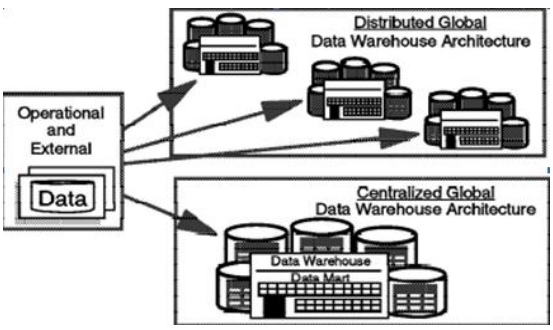

Fig. 3. Global Data Warehouse Architecture

The distributed data warehouse supports the decision makers by providing a single view of data even though that data is physically distributed across multiple data warehouses in multiple systems at different branches. Currently, the field of distributed data warehouse in terms of architecture and design is considered a significant research problem that needs investigation. Fig. 3 shows the centralized global and spread global data warehouse architectures [3] which are the two primary architecture methods.

The Inmon's approach assumes that being of both local and global data warehouses, with data stored in each being mutually exclusive. In this approach, data create in any local data warehouse are not stored in the global data warehouse and vice versa, thereby guaranteeing no redundancy between them. Three types of distributed data warehouses are as follows:

- **Local Data Warehouse & Global Data Warehouse:** Business is distributed geographically or over multiple, differing product lines. The data warehouse current locally represents data and processing at a remote site and the global data warehouse represents part of the business combined across the business.
- **Technologically Distributed Data Warehouse:** The volume of data will be distributed over multiple processors. Logically there is a single data warehouse, but physically there are many data warehouses which are all strongly related but reside on separate processors.
- **Independently Evolving Distributed Data Warehouse:** The data warehouse environment grows up in an uncoordinated manner, first one data warehouse appears, then another. The lack of management in the growth of the different data warehouses results in political and organizational differences.

F. Grid Computing

Grid computing coordinates the resources of the heterogeneous distributed environment. The resources can be categorized as computational resources and storage resources. The example for computational resource is CPU and an example of storage resources is all storage devices like hard disc and drives. Based on the need the resources can be scheduled. The management and scheduling of the resource is difficult to solve. Fig. 4 describes the fundamental grid model. There are four basic building blocks in a grid model; the user, resource broker, the Grid Information Service (GIS) and resources[23]. As soon as the user submits the job to the broker, the broker splits the job into various sub tasks and allocates to the appropriate resources based on the user specification. The broker gets the resource information from the GIS to update the information regarding the available resources.

The grid computing allows the users to share the heterogeneous resources which are distributed

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

geographically. The main benefit of the grid is to make use of the unused resources and resource utilization implements better scheduling.

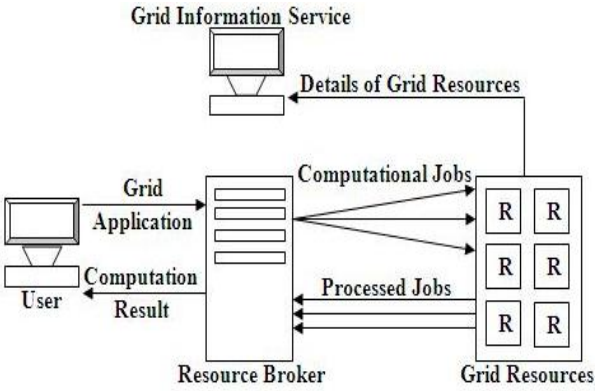

Fig. 4. Basic Grid Model

G. Query and Reporting

Query and reporting analysis is the process of posing a question to be answered, retrieving relevant data from the data warehouse, transforming it into the appropriate context, and displaying it in a readable format. It is driven by analysis which must create those questions to receive an answer. Fig. 5 describes the steps involved in query[3] and reporting process.

A distributed architecture of the data warehouse with Efficient Priority Allocation Mechanism Layer (EPAML) has been introduced[22]. It involves users with priorities based on their importance. The users with a higher priority level placed first to avoid waiting. A low complexity query is preferred based on the priority database. The set of users that belong to the different hierarchal background are also considered. Thus the submitted priority balancing mechanism retrieves data from the warehouse depends on the significance of the users.

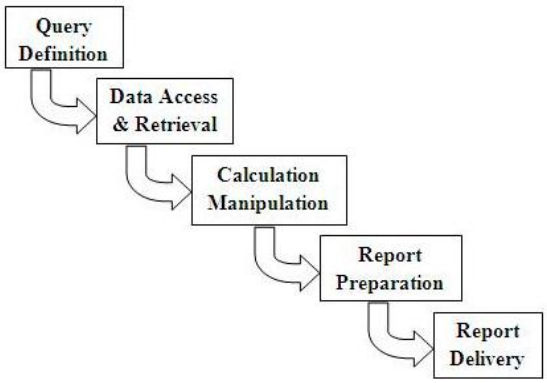

Fig. 5 Steps of Query and Reporting Analysis

Query definition is the process of analyzing a business question/hypothesis and translating it into a query format that can be used by a particular decision support tool. When the query is processed, the tool develops an appropriate language command to access and retrieve the requested data, which in turn is called as answer set. The answer sets are to achieve the desired results to fit into a display or report template for ease of understanding by the end user. This template consists of combinations of text, graphic images, video and audio files. Finally, the report is delivered to the end user in the desired output form, which can be printed on paper and visualized on a computer display device or presented audibly. This process continues until the desired results are reached.

II. SURVEY OF JOB SCHEDULING ALGORITHMS

Job scheduling is the process in which jobs are mapped to the specific physical resources, thereby minimizing the cost function to the user. This is a NP-complete problem and different heuristics may be used to reach an optimal or near optimal solution [23]. Effective computation and job scheduling are rapidly becoming one of the main challenges in grid computing and is considered as being important for its success.

The simple allocation schemes such as First Fit (FF) back fills are used in practice [30]. In any transaction First in First out (FIFO) algorithm does not prioritize and transactions are performed based on their arrival time. Scheduling procedures are based on First Come First Serve (FCFS) algorithm [6] 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]. 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 [10].

Two sets of experiments were conducted using a simulation model under various workloads. The experiments indicated that when uniform gang sizes are considered, both intended approaches [27] produce almost the same results in low to medium workloads, with approach 2 resulting in better performance for high workloads. However, when power-of-2 gang sizes are considered, approach 2 is the best solution in all workloads, since it completes a higher percentage of gangs. The results show that even when there is an error in the predictions, backfilling is an excellent gang scheduling technique for this particular system. Even though predicting a job's runtime with absolute accuracy is not realistic, the error does not significantly affect the overall system performance.

Hierarchical Job Scheduling (HJS) model [25] is based on a hierarchical approach using global and local level scheduler. The global scheduler uses a separate queue for different type of tasks and local scheduler uses a single queue for all tasks for scheduling with the FCFS, FF or Shortest Job First (SJF) policy. The global scheduler has more functions, from that anyone is identical to the resources are requested for participating clusters by a task or queries. Others are the best utilization of the available clusters.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

In Scheduling Framework for Bandwidth-Aware Job Grouping-Based scheduling (SFBAJG) algorithm [21], tasks are scheduled in system by the use of bandwidth-aware scheduling and also consider their computational and communication capabilities of the resources. It uses network bandwidth of resources for priority determination of every resource. At the time of information retrieval, task grouping method was used and maximize the resource utilization. After that the grouped tasks are sent to earliest finished resources.

A task scheduling model based on Maximum Processor Utilization and Throughput (MPUT) scheduling algorithm[9] that exploits the CPU utilization, throughput and reduces turnaround time. This Job Schedule Model Based (JSMB) algorithm provides reliability with sensible load balance but it does not contemplate any constraints of tasks and resources.

In Highest Response Next (HRN) Scheduling[28] tasks are allotted to the processors based on their priority as well as processor's capability. It provides high responses with memory, CPU requirement and time. This has effectively completed all the tasks quickly than FCFS and SJF. But it is not suitable for huge number of task allocation as there are considerable amount of CPU and memory wastage is there. HRN's turnaround time is also high.

In Resource Co-Allocation for Scheduling Tasks with Dependencies (RCSTD) algorithm[7], 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 within a cluster the tasks are allocated to the appropriate resource on which 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 for each resource in the system. Cluster communication overhead and unspecified task requirements are the main demerit for this algorithm.

Order picking is often considered as the most crucial warehouse activity. Any inefficiency in order picking can lead to unsatisfactory service and high operational cost of its warehouse, and consequently for the whole supply chain. Order batching is one of the important decision problems that need to be solved in order to control the order picking process efficiency. A simple and efficient approach for determining the optimal picking for the batch size of order-pickers in a typical 2-block warehouse[29] has been presented. In order to do so[2], the single-block warehouses are used to estimate the first and second moment of the service time. Then, these moments are also used to estimate the waiting time of a random order based on the corresponding batch service queuing model. The optimal picking batch size is determined in a straightforward manner. Results from the simulation experiments show that the suggested approach provides a good accuracy level. The average waiting time of a random order involves a convex function of the batch size. As a result, there always exists a unique optimum picking batch size[18]. The optimum batch size is always close to its lower bound. A simple greedy heuristic procedure has been presented to obtain optimum batch size in a negligible computational time. The order picking system is a simple one and can be extended further in any direction easily in multiple order-picking. However, it is rather difficult to capture the effect of aisle congestion, compound-poisson arrivals or other storage strategies and different layouts.

In Customer Order Scheduling (COS) each job is part of a customer order and the make-up of the jobs are pre-specified. Most of the existing research deals with a single machine or in a parallel machine shop for developing an optimal solution. COS is common in job shops, and the more complex the shop, the more complex the scheduling. Existing study has focused on reduction of the completion time of the batch. The focus was made at the time the last job is finished and omits the actual duration of the jobs within the same order. If it takes longer time to complete the job than there is an increase in the stock of finished goods and reduction in efficiency in logistics and supply chain management. Based on this concept, the Minimum Flow time Variation (MFV) method has been advocated[26] for solving the COS problem, and to enhance the control of the variation in finished time among the jobs in the same order. The individual completion times of all jobs of the same customer order will be controlled to improve the shipping performance. Two kinds of normal job shop: the simple shop and the complex shop were tested using a simulation model. Seven job-based rules and six orders-based rules were considered in comparing and benchmarking. In the simulation test and the statistical analysis, the stock level of finished goods under the MFV rule was reduced by more than 70% compared to the others.

Primary-Backup (PB) approach can be achieved by fault-tolerance approach [12] in which two copies of a task run on two different processors. The three basic PB approaches like PB exclusive, PB concurrent and PB overlapping are compared for dynamic scheduling of object-based real-time tasks. The work has been extended with three different PB based fault-tolerant approaches namely, PS-EXCL, CONCUR and OVERLAP, to object-based task model. Fault-tolerant dynamic scheduling involves object-based real-time tasks and utilizes the parallelism due to cloning and ARPC (Asynchronous Remote Procedure Call). The implementation of CONCUR and OVERLAP is harder than the PS-EXCL, as each set of output actions have to be scheduled twice in the former two approaches and only once in the latter.

A problem of online rescheduling has been addressed for new orders that arrive randomly and static orders being picked in warehouse environments. Two real-time incremental static reschedules are promoted: OR1, which

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

is based on a steepest descent insertion scheme and OR2, which is based on a multistage optimization procedure as distance minimization, load-balancing and queuing delay minimization [14]. These results show that the intended online reschedules are competitive against existing online methods when uncertainty in the system is low to moderate. These are the levels of dynamism often encountered in real picking systems that tend to restrict the incorporation of random orders with existing ones for accountability and due-date concerns. Thus, OR1 and OR2 can provide significant improvements in real picking systems that have a mix of static and stochastic inputs.

III. SURVEY OF RESOURCE SCHEDULING ALGORITHMS

The resource scheduling is the process of allocating a query to the resources based on its specific characteristics. The resource scheduling process is dangerous, when the user tries to obtain information as quickly and reliably as possible ([16], [11], [19], [31], [4]). In general, resource scheduling in large-scale grids is considered to be a challenging task because the resources are not centrally controlled as well as they can enter and leave from the system at any time.

Research on Novel Dynamic Resource Management (RNDRM) scheduling algorithm [8] is a model of agent based dynamic resource management system. It provides a good paradigm for integrating agent technology with grid computing based applications. RNDRM is a 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.

Agent Based Resource Management with Alternate Solution (ABRMAS) algorithm provides an alternate solution to the situation [20] when resource discovery fails. This Algorithm identifies an equivalent resource without affecting the performance and it also avoids unnecessary resource discovery. Sometimes resource discovery is done for time bound task when the required resource is unavailable. The alternate solution reduces delay overhead in waiting for the unavailable resource and enhances the system's efficiency. Implementation result shows the system success rate is 30% higher with an alternate solution.

Existing resource prediction models are based on the auto-correlation or cross-correlation with one or more resources. So, a multi-resource prediction model (MModel) has been designed [13] based on auto-correlation and cross-correlation to achieve higher prediction accuracy. Two adaptation techniques have been presented to enable the model to adapt to the changing characteristics of the underlying resources. Experimental results with CPU load prediction in both workstation and grid environment show that on average, the adaptive MModel (called MModel-a) can achieve from 6% to more than 90% reduction in prediction errors compared with the auto-regressive model especially for highly dynamic resources and long term predictions which has previously been shown to work well for CPU load predictions.

The main requirements of distributed systems can be formalized through the definition of a primary set of services which the system should provide. List of services composing the resource management system are,

- Resource discovery
- Access to information about resources
- Monitoring the task status and the environment state
- Resource allocation
- Reservations/limits/contracts handling
- Task execution management
- Accounting and reporting

The existing study on fault-tolerant scheduling based on the active replication schema makes use of $\in+1$ replica for each task to tolerate $\in$ failures. However, it may or may not lead to a higher reliability and involves more replicas which imply more resource consumption and higher economic cost. To address this problem, with the target to satisfy the user's reliability requirement with minimum resources, a new fault-tolerant scheduling algorithm MaxRe has been introduced for heterogeneous systems[17]. MaxRe algorithm incorporates the reliability analysis into the activable replication schema and exploits a dynamic number of replicas for different tasks. Both theoretical analyses satisfy the user's reliability requirement. Compared with the Fault-Tolerant Scheduling Algorithm (FTSA), the MaxRe algorithm is much more scalable especially when more processors, tasks and failures existed in the system. Specifically, the resource usage is almost 70% less than FTSA, and the performance on time is also acceptable and even shorter than FTSA in some experiments.

IV. CONCLUSION

Living data warehouses have to manage continuous flows of updates and queries and must comply with conflicting requirements, such as short response time and data freshness. This paper began with a brief introduction to distributed data warehouse and describes relevant recent research works done using various scheduling concepts, different types of job and resource scheduling techniques which are related to the distributed data warehouse environment very crisply. Some existing techniques related to the grid based environment also addressed.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

REFERENCES

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

[2] Chew Ek Peng and Tang Loon Ching, 1999. Travel Time Analysis for General Item Location Assignment in a Rectangular Warehouse. European Journal of Operational Research, 112: 582–597.

[3] Chuck Ballard, Dirk Herreman, Don Schau, Rhonda Bell, Eunsaeng Kim and Ann Valencic, 1998. Data Modeling Techniques for Data Warehousing. International Technical Support Organization, SG24-2238-00, 9-18.

[4] Chuliang Weng and Xinda Lu, 2005. Heuristic Scheduling for Bag-of-Tasks Applications in Combination with QoS in the Computational Grid. Future Generation Computer Systems - Elsevier, 21(2): 271-280.

[5] Claudia Imhoff, Nicholas Galemmo and Jonathan G. Geiger, 2003. Mastering Data Warehouse Design - Relational and Dimensional Techniques. Wiley Publishing, Inc., 3-13.

[6] Claus Bitten, Joern Gehring, Uwe Schwiegelshohn and Ramin Yahyapour, 2000. The NRW-Metacomputer - Building Blocks for a Worldwide Computational Grid. In: Proceedings of the 9th Heterogeneous Computing Workshop (HCW 2000), 31-40.

[7] Diana Moise, Eliza Moise, Florin Pop and Valentin Cristea, 2008. Resource CoAllocation for Scheduling Tasks with Dependencies in Grid. In: Proceedings of the 2nd International Workshop on High Performance in Grid Middleware (HiPerGRID 2008), 41-48.

[8] Fufang Li, Deyu Qi, Limin Zhang, Xianguang Zhang and Zhili Zhang, 2006. Research on Novel Dynamic Resource Management and Job Scheduling in Grid Computing. In: Proceedings of the IEEE 1st International Multi-Symposiums on Computer and Computational Sciences, (IMSCCS '06), 1: 709-713.

[9] Homer Wu, Chong Yen Lee, Wuu Yee chen and Tsang Lee, 2007. A Job schedule Model Based on Grid Environment. In: Proceedings of the 1st IEEE International Conference on Complex, Intelligent and Software Intensive System (CISIS 2007), 43-52.

[10] Hongzhang Shan, Leonid Oliker and Rupak Biswas, 2003. Job Superscheduler Architecture and Performance in Computational Grid Environments. In: Proceedings of the ACM/IEEE Conference on Supercomputing, 44-58.

[11] Ian Foster and Carl Kesselman, 2003. The Grid 2: Blueprint for a New Computing Infrastructure. 2nd Edition, Morgan Kaufmann Publishers Inc., San Francisco, CA.

[12] Indranil Gupta, G. Manimaran and C. Siva Ram Murthy, 2000. Fault-tolerant Dynamic Scheduling of Object-Based Tasks in Multiprocessor Real-time Systems. Dependable Network Computing (The Springer International Series in Engineering and Computer Science), 538: 433-462

[13] Jin Liang, Klara Nahrstedt and Yuanyuan Zhou, 2004. Adaptive Multi-Resource Prediction in Distributed Resource Sharing Environment, In: Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2004), 293-300.

[14] Jose I.U. Rubrico, Toshimitu Higashi, Hirofumi Tamura and Jun Ota, 2011. Online Rescheduling of Multiple Picking Agents for Warehouse Management. Robotics and Computer-Integrated Manufacturing, 27(1): 62–71.

[15] Jun Yang, 2001. Temporal Data Warehousing. Ph.D. Dissertation, Stanford University, 1-50.

[16] Kavitha Ranganathan and Ian Foster, 2003. Simulation Studies of Computation and Data Scheduling Algorithms for Data Grids. Journal of Grid Computing, 1(1): 53-62.

[17] Laiping Zhao, Yizhi Ren, Yang Xiang and Kouichi Sakurai, 2010. Fault-Tolerant Scheduling with Dynamic Number of Replicas in Heterogeneous Systems. In: Proceedings of the 12th IEEE International Conference on High Performance Computing and Communications, 434-441.

[18] Le-Duc, T. and Rene De Koster, 2002. An Approximation for Determining the Optimal Order Batch Size for Order Pickers in Single-Aisle Warehouses. In: M. Meller, M.K. Ogle, A.P. Brett, G.D. Taylor and J. Usher (Eds.), Progress in Material Handling Research, Charlotte: Material Handling Institute, 267-286.

[19] Massimiliano Caramia, Stefano Giordani and Antonio Iovanella, 2004. Grid Scheduling by On-line Rectangle Packing. An International Journal on Networks - Wiley Periodicals, 44(2): 106-119.

[20] Muthuchelvi, P. and V. Ramachandran, 2007. ABRMAS: Agent Based Resource Management with Alternate Solution. In: Proceedings of the IEEE 6th International Conference on Grid and Cooperative Computing (GCC 2007), 147-153.

[21] Ng Wai Keat, Ang Tan Fong, Ling Teck Chaw and Liew Chee Sun, 2006. Scheduling Framework for Bandwidth-Aware Job Grouping-Based Scheduling in Grid Computing. Malaysian Journal of Computer Science, 19(2): 117-126.

[22] Nouman MaqboolRao, Muheet Ahmed Butt, Majid Zaman and Waseem Jeelani Bakshi, 2013. Distributed Data Warehouse Architecture: An Efficient Priority Allocation Mechanism for Query Formulation. International Journal of Engineering and Innovative Technology, 2(9): 157-159.

[23] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra and Prachet Bhuyan, 2010. A Survey of Job Scheduling and Resource Management in Grid Computing. World Academy of Science, Engineering and Technology, 40: 461-466.

[24] Ralph Kimball and Margy Ross, 2002. The Data Warehouse Toolkit. 2nd Edition, The Complete Guide to Dimensional Modeling, John Wiley and Sons, Inc., 6-8.

[25] Santoso, J., G.D. van Albada, B. A. A. Nazief and P.M.A. Sloot, 2000. Hierarchical Job Scheduling for Clusters of Workstations. In: Proceedings of the 6th Annual Conference of the Advanced School for Computing and Imaging (ASCI 2000), 99-105.

[26] Sheng Yuan Hsu and C.H. Liu, 2009. Improving the Delivery Efficiency of the Customer Order Scheduling Problem in a Job Shop. Computers and Industrial Engineering - Elsevier, 57(3) 856–866.

[27] Sofia K. Dimitriadou and Helen D. Karatza, 2010. Job Scheduling in a Distributed System Using Backfilling with Inaccurate Runtime Computations. In: Proceedings of the IEEE International Conference on Complex, Intelligent and Software Intensive Systems (CISIS '10), 329-336.

[28] Somasundaram, K., S. Radhakrishnan and M. Gomathynayagam, 2007. Efficient Utilization of Computing Resources using Highest Response Next Scheduling in Grid. Asian Journal of Information Technology, 6(5): 544-547.

[29] Tho Le-Duc and Rene M.B.M. de Koster, 2007. Travel Time Estimation and Order Batching in a 2-block Warehouse. European Journal of Operational Research, 176: 374–388.

[30] Vijay Subramani, Rajkumar Kettimuthu, Srividya Srinivasan and P. Sadayappan, 2002. Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests. In: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11 2002), 359-366.

[31] Yang Gao, Hongqiang Rong and Joshua Zhexue Huang, 2005. Adaptive Grid Job Scheduling with Genetic Algorithms. Future Generation Computer Systems - Elsevier, 21: 151-161.

BIOGRAPHIES

Dr. S. Krishnaveni completed MCA, M.Phil., Ph.D., in Computer Science and currently working as an Assistant Professor, Dept. of Computer Applications in Pioneer College of Arts and Science. Three years of experience in teaching and published twelve papers in International Journals and also presented seven papers in various National and International conferences. Research areas includes Data mining and warehousing, Grid computing, Bioinformatics and Computer Network.

A. Sathish completed MCA, M.Phil, in Computer Science and currently working as an Assistant Professor, Dept. Computer Science in Maharaja College of Arts and Science. Area of research is Data mining.