*International Journal of Information Technology and Knowledge Management*
*January June 2009, Volume 2, No. 1, pp. 199-202*

# A FRAMEWORK FOR ACHIEVING BETTER LOAD BALANCING AND JOB SCHEDULING IN GRID ENVIRONMENT

**K. Mukherjee[*] & G. Sahoo[**]**

Traditional multiprocessors Super computer systems do have certain constraints viz computing elements, storage space etc. These constraints can be minimized in Grid environment provided a proper job scheduling and load balancing algorithm is available. The existing job scheduling and load balancing are either in fully centralized or fully decentralized in nature. In this paper, we propose a scheme that focuses on both Local Area Grid and Wide Area Grid efficiently. If Local Area Grid is overloaded then only jobs are transferred to wide Area Grid and hence a well job monitoring is carried out to achieve better parallelism in Grid Environment that saves valuable clock cycles needed for job execution. The proposed algorithm has been studied by conducting simulation using Remote Method Invocation (RMI) paradigm.

***Keywords:*** Grid, Hierarchical Agent Framework, Load Balancing.

## 1. Introduction

A Grid is a distributed collection of computing and data resources, which is shared by entities in a virtual organization. Grid computing is an emerging computing paradigm wherein authorized users can access and use distributed computing recourses by submitting jobs to the Grid and getting result back from the Grid. Grid System coordinates recourses that are not subject to centralized controlling and uses open general purpose protocols and infrastructures. It is of-course, a type of parallel and distributed system that enables sharing, exchange, selection and aggregation of geographically distributed autonomous and diverse recourses, owned by diverse organization or owners. Computational Grid addresses computationally problems on relatively small data sets. Data Grid addresses data intensive application applications that deal with the evolution and mining of large amounts of data in the terabyte and beta byte range. Computational Grid [1,6] are typically a conglomeration of various resources with different owners but make it possible for users to develop complex applications that access remote sites (nodes). Each of these sites could be a uni-processor machine, a symmetric multiprocessor clusters, a distributed memory processor system or a massively parallel super computer. Thus , any Grid system has enormous potential in view of computations, Data storages etc. with the multitude of resources of Grid, a proper scheduling and efficient load balancing across the Grid can lead to improve overall system performance and a lower turn-around time for individual jobs. Hence, an efficient job scheduling algorithm is at most needed to address the diverse resources of Grid viz CPU, shared memory, large disk frame, distinct I/O channels and software licenses that can be independently allocate different jobs. The Organization of this paper is as follows: Related work is described in 2, section 3 gives details of existing Grid System model; section 4 gives our proposed Grid Framework, proposed algorithm and details of the experimental setup and result would be given at the next issue of this journal. Section 5 concludes with a direction of the future work.

## 2. Related Work

Scheduling in Grid has been extensively studied in different contexts e.g. scheduling for computational Grid [9][10], decoupled computational and data scheduling [4] and scheduling based on economic models [16]. Economic models e.g. scheduling model used in Nimrod-G [17] has used factors like resources cost, user priority, job deadline, user preference, resource architecture etc in making scheduling decisions. Ranganathan *et al*. [4] have proposed a synchronous job scheduling . They have emphasized high performance computing. Takefasa *et al.* [16] has studied deadline scheduling for client-server system on the computational Grid. Casahora *et al*. [17] has described heuristics for scheduling parameter sweep applications in computational Grids. Park *et al.* [5] has described cost models to schedule jobs on the Grid to minimize response for the user and again they emphasize user response time optimization. Stockinger *et al.*[6] has described a cost model for distribution and replicated data sources. Min *et al*. [21] and Simit *et al*.[18] have discussed advance reservation of resources in their scheduling algorithm to satisfy the QoS requirements of the user. Subrami *et al.* [20] has evaluated some centralized and distributed

[*] Department of Computer Science & Engineering., B. I. T Mesra Extension Centre Lalpur, Ranchi, Jharkhnad, India, *E-mail: kmukherjee@bitmesra.ac.in*

[**] Dept. of IT, BIT Mesra, Ranchi, Jharkhnad, India. *E-mail: gsahoo@bitmesra.ac.in*

scheduling decisions for a computational Grid and has proposed a scheme which uses redundant distribution of a job to different sites to reduce average job slow down and turn-around time. Manish Arora, Sajal K. Das and Rupak Biswas [13] have proposed a decentralized scheduling and load balancing algorithm for heterogeneous Grid environment.

### 3. Existing System Model of Grid

The existing system model of Grid, consists of sites which have both computing and data storage resources. The different sites of Grid are connected by a wide area network (WAN) and Local area network (LAN). The time for data transfer within the local network (Intrusive Data Transfer) are negligible in comparison to both job execution time and interstice data transfer time. Again, there are two important aspect of any wide area network Grid, first its transfer [4, 5] and second its location [8, 9] polices. The transfer policy decides if there is a need to initiate load balancing across the system, is typically threshold balancing[13]. The location policy selects a partner node for a job transfer transaction and location policy locates complementary nodes to or from which a node can send or receive workload to improve overall system performance [13]. Similarly, local area Grid is built on PC-based clusters to form a Grid computing environment and built over multiple Linux PC clusters by using Globus Toolkit and Sun Grid Engine [21]. Each such cluster has a master node and in case of Sun Grid Engine (SGE), the master node executes SGE master daemon that manage and monitor incoming jobs. Each slave node executes SGE slave daemon to execute incoming job only. These PC-based clusters are located in different locations of premises. In the entire software structure of this type of system, the Grid environment available is a basic environment construction, where it is worth nothing that Globus Toolkit serves as a middleware to take care of message communication and information exchange in the Grid environment. Generally, the graphic user interface supported by java, which is highlighted with its inter-platform characteristics, the debuted interface suits all systems for successful operation.

### 4. Framework of Proposed Grid

A Grid consists of a local Grid (simple composite elements (scm) or clusters as they are commonly called) and wide area Grid [21]. The local Grids are richly connected by relatively homogeneous collection of base system elements (i.e. computing elements, memory, computation and storage). They are often–housed within single administrator domains and in many cases are thought of as a single system [21]. Local Grids are building blocks for wide-area national and international Grid[21]. Our computational Grid system is shown in fig. 1.

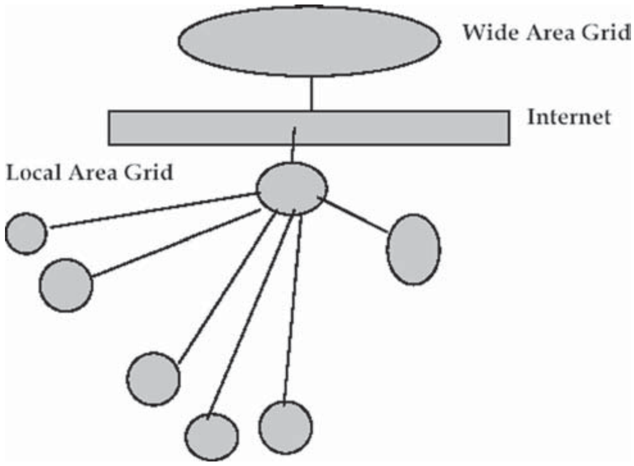


**Figure 1: Computational Grid System**

**Wide Area Grid**

The local Grid can use resources and software to implement the external properties of the composite elements affecting its utility or integration into larger Grid. The local Grid can use resources and software to implement the external properties of the composite elements affecting its utility or integration into larger Grid. Local Grids form the basis for larger Grids. Thus their evolution is an integral part of the challenges in building larger Grid. The frame work of our proposed Grid is shown in figure 2. Here, we have focused on both the local Grid as well as the wide area Grid. But the load balancing algorithm is written in view of local Grid only. In this paper we have focused only on parallel processing aspect of local Grid (cluster based).The framework of our proposed cluster based parallel processing be shown in Fig. 2.

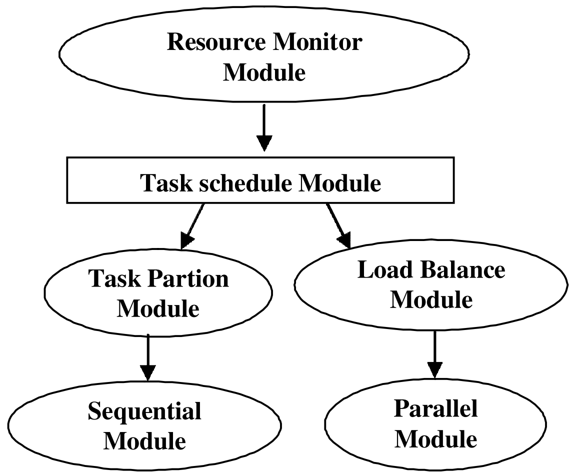


**Figure 2: Framework of Our Proposed Cluster Based Parallel Processing**

When execution of a job starts, Node Agent, as shown in Fig. 4, collects monitoring data, on the basis of this data and the performance property specification (defines the condition, confidence and the suavity for each property), it evaluates the serviettes of specified performance details report (shown in figure 3). A Turning Agent (TA) which is invoked by the NA, is responsible for performing any local tuning action, so that any performance bottleneck or performance problems can be removed. The TA accepts the performance details report from the NA and decides what actions to be taken after consulting the performance property specification and the performance tuning process specification. Performance tuning process specification stores recommendation regarding action, which may be taken for every performance problem depending upon their serviette. The process specification is basically an expert knowledge base, which may be created and stored on a particular recourse provider. The TA generates a performance tuning report, shown in Fig 3. We create a subclass of the process specification for every subclass of performance property. Thus in our current implementation, InsufficientParallelismProcessSpecificatio is a subclass of the class PerformancePropertyProcessSpecification, which specifies the analysis process of the Insufficient parallelism property and LoadImbalanceProcessSpecification is another subclass, which specifies the analysis process of the LoadImbalanceproperty. Similarly, InsufficientParallelismTuningProcess specification is a subclass of the class property. Insufficient parallelism and Load Imbalance Tuning Process Specification is another subclass of the same class that specifies the tuning process for the property LoadImbalance. A class diagram containing the agent class and specification is shown in Fig. 5. In the current implementation, we have implemented the

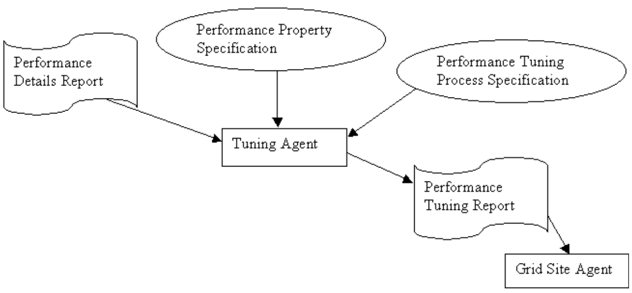


**Figure 3: Tuning Agent**

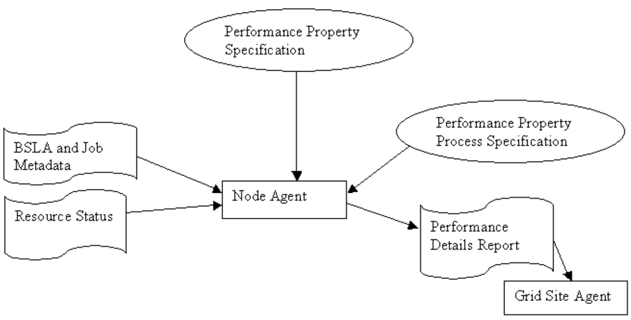


**Figure 4: Node Agent**

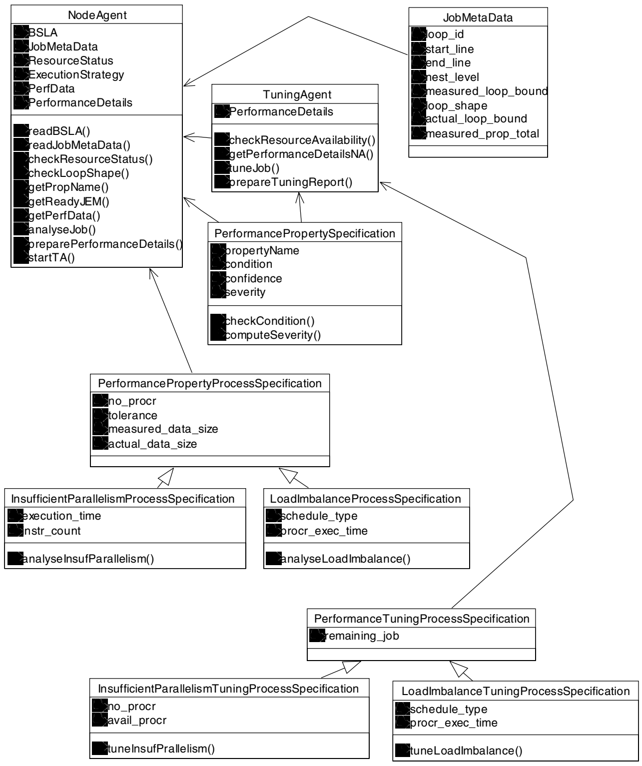


**Figure 5: Class Diagram for Node Agent and Tuning Agent**

specification, computations and tuning related to InsufficientParallelism property. Analysis of this property and tuning the job to avoid a performance related to this property has been shown in agent interaction sequence diagram (Fig. 6)

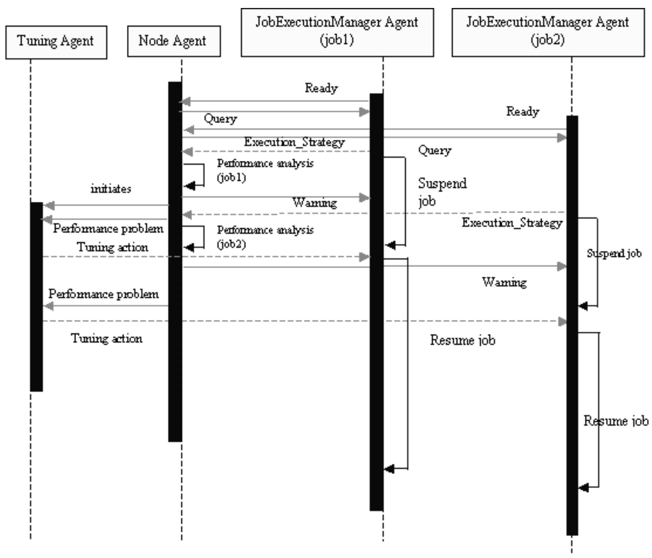


**Figure 6: Agent Interaction**

## 5. CONCLUSION

In this paper, we have presented a better load balancing and job scheduling in Local Area Grid only. It highlights the interaction and exchange of information among the agents for collecting data, analyzing and improving the performance by application of local tuning actions in local grid Area. It also discusses the implementation of some of them. This paper highlights the effectiveness of the framework by demonstrating the performance improvement through tuning and showing that the agent control overheads are negligible even when multiple jobs are submitted concurrently onto the same resource. Proposed algorithm and details of the experimental setup and result would be given at the next issue of this journal In future work, we will extend our Grid framework from Local Area Grid to Wide Area Grid. The job scheduling algorithm, load balancing algorithm would be in view of wide area Grid and we will introduce Load_balance_Wide_Area_Grid procedure in our future work. In future work, we will do more implementation of complex parallel algorithm.

## REFERENCES


[1] M. Livney and R. Raman, "High Throughput Resource Management", in *Proceedings of 4th International Conference on High-Performance Computing*, Los Alamitos, California, IEEE Computer Society, (December 1994), 407–414.

[2] R. F. Freund and T. D. Braun, "Production Throughput as a High-Performance Computing Meta-task", The *2002 International Conference on Parallel and Distributed Processing Technique and Applications (PDPTA 02)*, (January 2000), 23–29.

[3] E. G. Coffman, Jr (ed), "Computer and Job-shop Scheduling Theory", John Willey and Sons, New York, NY, (1976).

[4] K. Ranganthan and I. Foster, "Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications" in *11th IEEE International Symposium on High Performance Distributed Computing* (HPDC-11), (2002), 92–100.

[5] S. Park and J. Kim, Chameleon "A Resource Scheduler in a Data Grid Environment", the *Proceeding of the 3rd IEEE ACM International Symposium on Cluster Computing and Grid (CCGRID03)*, (2003), 39–45.

[6] Balakrishnan, S., S. K. Nandy, A. J. C. Van Germund, "Modeling Multi-threaded Architectures in PAMELA for Real-time High-Performance Applications", in *Proceedings of 4th International Conference on High-Performance Computing*, Los Alamitos, California, IEEE Computer Society, (December 1997), 407–414.

[7] Bull J. M., M. E. Kambities, "JOMP - an OpenMP -like Interface for Java", *Proceedings of the ACM2000 Java Grande Conference*, (June 2000), 44–53.

[8] De Rose, L., Y. Zhang, D. A. Reed, "SvPablo: A Multi-Language Performance Analysis System", in *Proceedings of 10th Internal Conference on Computer Performance*, (Spain, 1998), 352–355.

[9] Fahringer, T., Gerndt, M., Riley, G. D., and Traiff, J. L., "Formalizing OpenMP Performance Properties with ASL", In *Proceedings of the Third International Symposium on High Performance Computing, ISHPC* (October 16-18, 2000), 428–439.

[10] Foster I., C. Kesselman, J. Nick, and S. Tuecke, "Grid Services for Distributed System Integration", *IEEE Computer*, (June 2002), 37–46.

[11] Furlinger K., "Scalable Automated Online Performance Analysis of Applications using Performance Properties", PhD. Thesis, 2006 in Technical University of Munich, Germany.

[12] Furlinger K., M. Gerndt, "A Lightweight Dynamic Application Monitor for SMP Clusters", *HLRB and KONWIHR Joint Reviewing Workshop 2004*, (March 2004).

[13] Manish Arora and Sajal K. Das and Rupak Biswas"A De-centralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid-Environments" *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW02),* (2005), 20–23.

[14] M. Pinedo. "Scheduling: Theory, Algorithm and System". Prentice Hall, Englewood Cliffs, NJ, (1995).

[15] R. Buyya, D. Abramson, J. Giddy and H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing", *Journal of Concurrency and Computation: Practise and Experience (CCPE)*, (2002), 113–124.

[16] A. Takefusa, H. Casanova, S. Matsuoka and F. Berman, "A Study of Deadline Scheduling for Clint Server System on Computational Grid", *HPDC*, (2001), 12–14.

[17] H. Cansanova , A. Legrand, D. Zagorodnov and F. Berman, "Heuristics for Scheduling Parameter Sweep Applications in Grid Environments", *Heterogeneous Computing Workshop (HCW 2000)* , (2000), 13–15.

[18] W. Smith, I. Foster and V. Taylor, "Scheduling with Advanced Reservation, International Parallel and Distributed Processing Symposium (IPDPS00)", (2007), 11–16.

[19] R. Min and M. Maheswaran, "Scheduling Advance Reservation with Priorities in Grid Computing System" *Thirteen IASTED International Conference on Parallel and Distributed Processing (PDP01)*, (2001), 21–27.

[ 20] V. Subramani, R. Kettimuthu, S. Srinivasan and P. Sadayappan, "Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests", *Proceeding of 11th IEEE Symposium on High Performance Distributed Computing (HPDC02)*, (2002), 45–48.

[21] R. Min and M. Maheswarn, "Scheduling Advance Reservation with Priorities in Grid Computing Systems". *Thirteenth IASTED International Conference on Parallel and Distributed Computing Systems* (PDCS 2001), 21–25, (2001).

[22] Ian Foster and Carl Kesselman "The Grid2 –Blueprint for a New Computing Infrastructure", Morgn Kaufmann Publishers, (2006), 45–48.

[23] DeRose, L., Y. Zhang, D. A. Reed, "SvPablo: A Multi-Language Performance Analysis system", in Proceedings of *10th Internal Conference on Computer Performance,* Spain, (1998), 352–355.

[26] Furlinger K., "Scalable Automated Online Performance Analysis of Applications using Performance Properties", PhD. Thesis, (2006) in Technical University of Munich, Germany.