A Comparative Study of Task Scheduling and Load Balancing Techniques with MCT using ETC on Computational Grids

S.Sheikh and A. Nagaraju

Department of Computer Science, Central University of Rajasthan, Ajmer– 305817, Rajasthan,India;
sophiya.sheikh@gmail.com, nagaraju@curaj.ac.in

Abstract

Objectives: In this paper various task scheduling algorithm along with load distribution techniques investigated to ensure efficient mapping of tasks to resources and for coherent resource utilization in heterogeneous environment. **Statistical Analysis:** A detailed comparative approach is conferred for different job scheduling and load balancing algorithm available in the literature. Considered parameters for comparisons are scheduling approaches, techniques, findings, benefits, pros and cons. This study facilitates to identify the efficient scheduling and load balancing approach which optimizes resource use, minimizes completion time as well as task waiting time and balanced load among resources. **Findings:** Make span time and flow time are two important parameters to compute and minimize job execution time. This paper explores an intense survey of minimization of task execution time with minimum make span time and flow time. Apart from this a joint review on job scheduling, load balancing and minimizing job completion time using Expected Time to Compute (ETC) matrix is presented. At the end of this paper a research model is proposed which is a novel and hybrid approach with a combination of the efficiency of scheduling and load balancing. **Application:** The Load Balancing (LB) is an important feature of the grid environment. The efficient LB algorithm assures cost effective resource usage by the provisioning of resources to users on demand schedule in pay-as-you-say-manner.

Keywords: Computational Grid, ETC Matrix, Grid Computing, Job Scheduling, Load Balancing, Task Execution Time

1. Introduction

As technology rapidly increases, the numbers of users are also increasing simultaneously to access resources in the heterogeneous and dynamic environment. The Grid computing is the collection of resources resides on geographically dispersed location to achieve common objectives. Workload among resources will not interact with each other, although they work in parallel to reach to a common objective. Grid infrastructure is beneficial because its main emphasis is on resource sharing among different organizations as well as platforms. Grid computers are more heterogeneous and geographically dispersed than cluster computers. To manage, such as high intensive geographically dispersed environment a resource alloca-tion should be not only scalable but also highly reliable. In this paper different load balancing and task scheduling algorithm has been analyzed with their limitations and benefit.

Computational gird is most widely used for dynamic as well as a computational problem solving. It is a collec-tion of highly computational intensive tasks executed on individual machines[1]. In view of proper usages of under-utilized resources, allocation technique must be efficient enough to process large number of computational jobs. This paper describes that how computational jobs can be executed with minimum completion time. For proper utilization and load balancing of resources efficient scheduling technique is a necessary requirement[2]. This paper illustrates that how these complex tasks would be

mapped to the resources presented in a distributed environment. Task's execution will be done in parallel, then result will combine return to the originate system. Since the computational grid provides us a way to use various devices in combination therefore it is necessary to ensure proper load balancing in such environment. A number of algorithms reported in the literature to defines efficiently load balancing among resources[3].

Another problem with computational grid is that compute intensive jobs cannot be executed at the user's end. The task might be dependent tasks or independent in nature. If the task is independent then it can be easily executed in a parallel manner in grid system. Such tasks are called *Metatasks* and extensively used in metatask scheduling by Expected Time to Compute (ETC) matrix. ETC[t][m] matrix denotes expected time required by a task to execute on a particular resource. This paper presents a detailed survey of approaches using the ETC model to minimize task computation time.

Rest of the paper organized as follows: Section II defines related work reported in the literature with a comparative study between the various approaches. Section 3 defines proposed hybrid approach to achieve minimum completion time and balanced load among available resources. The paper ends with Section IV as conclusion with proposed future research work.

2. Research Methodology

In this section we discuss a number of job scheduling, load balancing approach along with techniques that has been based upon minimization of job execution time with an efficient task to resource mapping using expected time to compute matrix on computational grid.

2.1 Job Scheduling in Computational Grid

In[2] reported Agent based job scheduling algorithm implemented for efficient and effective execution of user jobs. Various parameters were considered in performance analysis of job scheduling like waiting time for job, turnaround time and response time of resources. Other considered parameters were total completion time of task, bounded slowdown time and stretch time. It is an efficient, high performance and scalable job scheduling algorithm on a computational grid using real workload traces.

In[4] introduced a local heuristic search technique, implemented upon swarm intelligence. The approach is inspired by a meta heuristic approach called ant colony optimization. Scheduling in grid always faces issues of heterogeneity for both job and resource level. Author introduced four phases for grid scheduling like resource discovery, selection, monitoring, and scheduling. Author proposed improved Ant Colony Optimization (ACO) based on mutation concept, for the computational grid scheduling problem. A 3-opt mutation operation on the parent's solution is designed to create child solution. Therefore, a search optimal solution in the search space can be obtained. Results are compared with traditional ant colony optimization and Min-Min algorithm and results shows decreasing in make span in hybrid ACO. Although authors presented improved ACO with mutation and 3-opt local search capability yet it's not suitable for heuristic approaches. Another disadvantage of this algorithm is that it does not deal with resource utilization and load balancing over grid resources. A resource can be used in an optimum way, if it utilizes the underutilized computing power.

In[5] implemented Precedence Based Distributed Job Scheduling for Computational Grid. The objective of the proposed model was to minimize turnaround time of the submitted jobs by searches of the best cluster and the best nodes for the job modules with respect to the execution time. The implemented scheduling strategy is not only schedules the job with the suitable grid resources as per the job's requirements, but also preserving the precedence constraints within the job. Proposed strategy designed for multi cluster organization where each cluster is having multiple nodes called machines. Each node has its own processor, local scheduler, and cluster also has its own cluster scheduler with its scheduling policy. Authors prepare a cluster table having track of current status and various other attribute information. As the job modules increase turnaround time and efficiency also increase. But as the number of nodes increase turnaround time and efficiency decrease. Although it can deal with minimum completion time and turnaround time, but cannot deal with minimizing makespan time. Another disadvantage is its efficiency decreases when the number of nodes increases.

In[6] have proposed a task scheduling policy based on memory availability and introduced a memory based task partition and scheduling. They discussed three task partition policies based on Central Processing Unit (CPU),

memory and combined partitioned of CPU-memory combined.

Gravitational Search Algorithm (GSA) is a Meta heuristic approach to find minimum make span and flow time[7]. Authors generated task machine matrix called ETC matrix. This approach defines the interaction of particles of the universe and gravitational force applied between these particles which are based upon Newton's second law of gravitation. Every particle has two specifications, mass and position. The agent works upon search space based on the force applied to it. Author calculated fitness value, mass based on fitness function and acceleration matrix of agents. Agents are updated during the generation of velocity matrix. The algorithm is tested for consistent ETC matrix. As compared with PSO and GA and GSA has better performance. Results show that GSA perform better fitness calculation and provides minimum make span time and flow time for different grid environment. GSA has a good deal with the speed of convergence and also obtains feasible solution.

The grid is geographically dispersed administrative domains therefore it may have some problem like irregular task receiving patterns and uneven computational powers. To fully utilize grid resource an effective resource management is essential. Author enhances the efficiency of already proposed Nearest Deadline First Served (NDFS) for load balancing on computational grid by parallelism Operation[8]. In this approach resource broker is responsible to collect job requirements from the clients and runtime status information about resources. Thus, it ensures suitable allocation of resources to each client's request.

Scheduling in a grid environment is an NP complete problem. To optimize the turnaround time and response time in computational grid a Level based Batch scheduling Strategy with Idle slot Reduction (LBSIR) has been implemented[9]. Proposed approach considers inter module communication within the modules of the jobs represented using Direct Acyclic Graph (DAG). The model worked in two phase allocation phase and idol slot reduction phase. In allocation phase dividing the batch into a number of partitions as per the depth level, then assigning of sub jobs or module from the partitions to the best fit node in terms of execution time offered for all the partitions. While in next phase idle slots generated during allocation level in each depth level, then reduced by inserting the best fit module into these slots after all allocation of modules from higher depth level. Results

show that the level zed allocation of resources will minimize the response time for user intensive application. It is a centralized level of batch scheduling. Proposed strategy optimized turnaround time and response time within a batch of job consisting sub jobs for a computational grid. Proposed approach compared with LBSIR (LMS) and LBSIR (SMS) with MinMin, MaxMin, Suffrage, LJFR-SJFR, HEFT-1, HEFT-2, CPOP and DLS by varying number of nodes as well as the jobs and consistency of the grid system using static ETC simulation benchmark followed by observing the effect of the communication requirements.

Multi-objective Variable Neighborhood Search (MVNS) algorithm has been implemented for scheduling independent jobs on computational grid[10]. Performance is better in the comparison of any other Meta-heuristic algorithm like Min–Min algorithm, Simulated Annealing (SA) and Greedy Randomized Adaptive Search Procedure (GRASP) algorithm. VNS is approached for minimizing make span and flow time.

In[11] presented dynamic job scheduling algorithm for geographically distributed computing resources. Author suggested that no single scheduling method was sufficient for scheduling. The proposed approach mimics the combination of three approaches Genetic Algorithm, Simulating Annealing, and Tabu Search and applied in computational grid to ensure better performance. Instead of pure genetic algorithm search if GA is combined with other scheduling algorithm, the results are the GA-SA algorithm has a better convergence then GA-TS algorithm. This will also improve the efficiency of the GA. Genetic algorithm deals with population of solutions, while TS and SA deal with only one solution at a time. A hybrid heuristic algorithm provides better performance than any other classical approach.

In[12] proposed a model based upon genetic algorithm to provide efficient load balancing on computational grid. In this algorithm load variation is minimized as much possible. Author calculated average workload by total workload and computational load, then derived the objective function. Like any genetic algorithm first it takes chromosome, then perform crossover, mutation and selection. This algorithm took best possible schedule to minimize the load variation. Authors tested the load variation for varying number of nodes and calculated the load variation and distribution for varying number of jobs and also tested the approach for six different environments. First three cases were with fixed load and

varying nodes for both fine grain and coarse grain module. In Fourth case authors considered a varying number of load and node. While in the fifth case fixed node is considered with varying load. And the last case dealt with the coarser grain modules having fixed load and varying nodes. In fine grain module if the number of nodes were fixed, then load distribution was efficient in comparison of the coarser grain module.

In[13] proposed an enhanced ant colony optimization for efficient resource utilization has been proposed. It gave a better throughput and control cost as price factors included in it. A dynamic load balancing strategy introduced, increased resource utilization that cuts down processing cost and processing time. This approach is tested for 40 users and resources. Results show that it provides better processing cost and processing speed in the comparison of traditional ant colony algorithm and cost controlled algorithm.

A novel approach, multi-objective evolutionary algorithm for execution of multiple objectives based upon pare to dominance has been introduced[13,14]. This is a combination of evolutionary computation and the theoretical frameworks of multi-criteria decision making. EMA is designed for minimizing make span time and flow time. Author first developed pare to sets, then minimize the distance for generating solution and gives maximized diversity for them. Author tested the approach for multiple jobs and machine. In proposed approach authors used binary tournament selection which is used as an operator for mutation and crossover operation. Proposed approach was compared with the Genetic Algorithm (GA), Simulated Annealing (SA) and Particle Swarm Optimization (PSO). The result showed that Multi Objective Evolutionary Algorithm (MOEA) gives better make span time in comparison of GA, SA and PSO.

A comparative approach has been presented in Table 1 of various job scheduling algorithms reported in literature so far. Compared parameters for job scheduling are considered approach, techniques and findings of implementations with merit and demerits.

2.2 Load Balancing on Computational Grid

In[1] proposed NSGA II with Fuzzy Adaptive Mutation Operator for task scheduling and load balancing. It is to be observed that in a distributed environment Load balancing is difficult to handle and resource management is quite complicated In a view of effective job schedul-ing and allocation of appropriate resources by scheduler we author proposed Non-dominated Sorting Genetic Algorithm(NSGA) II with Fuzzy Adaptive Mutation Operator. An efficient task scheduling algorithm should be capable enough to reduce the turnaround time, cost of job execution and improve load balancing. In load balancing scenario load is distributed among computational resources for optimal resource utilization. For this purpose NSGA II with Fuzzy Adaptive Mutation Operator is implemented to deal with independent task assignments problems in parallel distributed computing systems.

Proposed Ant Based Heuristic Approach to Scheduling and workload distribution (AHSWDG) for efficient load balancing among the available resources[15]. Here ants represent the submitted jobs, however the ant's pheromone trail represents the computational capacity of the grid resources. The main problem with computing capacity of resources is their frequent nature. This scheduling approach, allocated task to the most optimal grid resources for faster task allocation. Once the job is allocated to resources, then the computational capacity will be updated. This strategy adapts itself according to the varying situations of the entire environment and types of jobs. Job allocation will be based upon transition probability as a resource having maximum transition probability will ready for allocation of job. If allocation fails, then the job will move to failed queue which will be further allocated to other resource. Job always allocated to grid resource having the best computational capacity and then the capacity will be updated. The algorithm is best to provide resource allocation and load balancing, but has disadvantages too like specially design for resource utilization but could not work with process utilization and minimizing job execution time.

Some problems exist with EGDC like grid let executes below the lower range of normal loaded range of resources and no precisely the description that how grid let scheduling will take place henceforth improved Enhanced grids with Deadline Control proposed[16]. The novel approach improves the execution time of grid let. The Scheduling mechanism depends upon resource bandwidth and capacity of resources. Results show that the improved algorithm provides faster bandwidth and higher capacity of resources. The resources are divided into three categories such as; overloaded, normal loaded and under loaded and assign to the respective resource queue then grid lets transfer from overloaded resources to under loaded resources. A resource which has fastest

Table 1. Comparison of job scheduling techniques in computational grid

Approach	Model	Compared Approach	Technique	Findings	Benefit	Drawback
Hybrid ACO	Meta Heuristic	Min Min and traditional ACO	ACO with Mutation and 3-opt local search capability	ACO with Mutation and 3-opt local search capability	Decreasing makes pan	Not suitable For Heuristic Approach And For multi cluster Environment
PDJSS	Multi Cluster Organization	Compared for varying number of module and load	Minimizing TAT of the Submitted jobs by searching the best cluster and the best nodes for the job	Cluster table Designed for keeping track of jobs and efficiency	Minimizing turnaround time and makespan time, Increased hardware and software parallelism	Turnaround Time and Efficiency Only increase When software parallelism is very high as compared to hardware parallelism
AHS	Dynamic and Distributed approach	First Come First Served (FCFS), Round Robin (RR), Proportional Local Round Robin(PLRR), Shortest Process Next(SPN),Priority(P) and Longest Job First (LJF)	Statistical Analysis of The LCG1 workload trace to study the dynamic nature of grid jobs	Minimizing Waiting time, turnaround time, response time, total completion time, bounded slowdown time and stretch time	Optimal performance, efficiency And Scalability	Load distribution is not even and efficient
GSA	Meta Heuristic	GA and PSO	Based on Newton's Second law of gravitation, Measured fitness of agents	Measured makespan time, flow time, acceleration, mass, updated mass, dimension of system and updated dimension and velocity	Minimize Makespan and flow time, benefits in speed of convergence and provides feasible solution.	No Pre-emption, not a feasible way for multiple task and single machine
NDFS	Distributed approach	Min Min, Max Min, LBEGS, FPLTF, WLB	Broker collects Job requirements from the Client and runtime status of resources then prepare service quality agreement, for job scheduling	Minimum communication overhead and minimum job execution time of gridlets	Enhance efficiency of existing NDFS load balancing algorithm and provide effective utilization of resources.	Increased overhead in scheduling and resource allocation

MVNS	Scheduling independent jobs on computational grid	Min Min, SA, GRASP	Fitness Function calculation ,minimise expected compute time of task by ETC	Efficient job scheduling approach also provides computation efficiency	Minimizing Makespan and flow time	Scheduling in cluster so Not efficient for decentralized or distribute Architecture.
Load balancing in computational grid using GA	Parallel and distributed model based upon genetic algorithm	Tested in six different environment with varying no of load and nodes	Objective is to measure workload, Average workload And load variation	Author tested Approach for different environment with varying no of load and nodes	Work Efficiently for Fine grain and coarse grain jobs	Proper load balancing approach but increases execution time of task
LBSIR	Inter module communi-cation within the module using DAG	LBSIR(LMS) and LBSIR(SMS)with Min-Min, Max-Min, Sufferage, LJFR-SJFR, HEFT-1,HEFT-2, CPOP and DLS	Two phase working allocation phase and idol slot reduction phase, Levelized allocation of resources.	Approach is tested for varying heterogeneity level of jobs and nodes	Optimized Flow-time, turnaround time and response within the job of a batch also provides efficient resource utilization	Approach is designed only for batch scheduling not for dynamic or dependent task scheduling
Enhanced ACO	Dynamic Load balancing strategy	Compared With traditional ACO and cost controlled algorithm	Tested for 40 Users and resources	Mere traditional ant colony and cost control algorithm	Better throughput and cut down resource cost	Maximizing average utilization and task completion time
Multi-objective evolutionary algorithm	A novel multi-objective approach	GA,SA,PSO	Based upon pareto sets, Combination Of evolutionary computation and theoretical frameworks of multi-criteria decision making	Minimizing the distance for generated solution and maximise diversity	minimizing makespan time and flow time	Decreasing resource utilization

bandwidth, highest capacity will be allotted to the grid lets. At last all unfinished gridlets will be assigned to the under loaded resources for maximizing the job execution. Proposed approach works for constant jobs and constant grid lets.

In[17] presented a scheduling algorithm for minimizing completion time of jobs and maximize resource utilization. Dynamic load balancing is an ideal for efficient load balancing. Author proposed schedule_DLB to balance workload among different clusters presented in a grid system. First, it searches idle cluster, then assign resources

by considering CPU utilization, CPU speed, expected completion time and queue length. Proposed approach transfers jobs from overloaded to under loaded cluster to maximize the resource utilization. First, it works within the cluster then with the overall cluster environment.

Another load balancing approach has been introduced by author in[18] proposed a dynamic load balancing scheduling algorithm for the Homogeneous Multiprocessor System to balanced load and minimize job completion time.

Xiao et al proposed approach for Quality of Serice (QOS) based resource allocation for grid infrastruc-

ture with constraints like budget and deadline[19]. Virtual resource agents are introduced to solve this purpose. The proposed approach is designed for multi-cluster computational grid model. Each cluster has computing element which has Virtual Resource Agents (VRA) and Meta scheduler. Once the job is ready for execution, it select suitable VRA based on budget and deadline. All VRA buy resources from the system and sell it to the client. VRA can also adjust task dynamically. Utility functions designed for the VRA to maximize resource utilization. The relationship between VRA and Client can be cooperative or non cooperative. Performance compared with *Round Robin Policy*, *Capability-based Random Policy*, *Cluster Minimized Policy* and *Virtual Resource Agent Policy*. Results show that proposed approach provides better turnaround time, average resource utilization and balance workload among various computing nodes present in a cluster.

Proposed a load balancing approach deals with both system and individual optimal objectives[21] Load balancing scheme either be cooperative or non cooperative. Non cooperative load balancing only considered individual response time while cooperative considered the entire system's response time. In this approach a competitive equilibrium approach is proposed for the given load balancing problem. Node processing rate and job arrival rate are taken as input which will be divided into fractions. At some prices author compute load to maximize utility functions and obtain market clearing error, then adjust the prices as per aggregate demands till market clearing error is less than error tolerance. The global optimal solution provided better mean response time, but it's only for individuals not for all user jobs while Nash equilibrium solution achieved better fairness but provided poor mean response time. The competitive equilibrium solution provides minimum average response time not for individual, but also for the system.

In[21] scheduling algorithm for hierarchical load balancing has been presented for dividing the computational load in an efficient way. The algorithm addresses two important aspects, one is Multiple Parameters based Load Balancing (MPLB) and another is a job scheduling at Logical Hierarchical Levels (LHL). Reddy and Roy presented a load balancing technique over a tree based grid model. Results show that Hierarchical Job Scheduling (HJS) approach is more efficient as compared to Flat Structure Job Scheduling (FJS).

Another load balancing scheme states that regional grids which are clusters around broker sites considered network transfer delay and organized in fully decentralized fashion[22] Henceforth to achieve static and dynamic load distribution and redistribution in efficient way author proposed hybrid policy for job scheduling. This would minimize the job completion time and average response time. The proposed technique is suitable for large scale distributed grid environment. The hybrid policy also provides a balance between the inherent efficiency of centralized approach, load balancing among the resources and fault tolerant features offered by distributed approach.

In[23] multi-objective genetic algorithm is known as NSGA-II using fuzzy operators has been improved for better quality and performance of task scheduling in heterogeneous grid environment. In a view of task scheduling the main objective of multi objective optimization is load balancing, minimizing make span time and cost. The purpose of this efficient load balancing scheme is to decrease computation and increase decision making for the users. Proposed task scheduling strategy indirectly uses the fuzzy system without implementing the third objective function and uses the fuzzy operators and Pareto-optimal solutions with better quality and variety.

Lee introduced a new decentralized scheduling scheme. This scheme is not only useful for local scheduling, but also for queue balancing. The main emphasis is on backfilling across multiple nodes. In this approach author compares decentralized approach improved load balance and throughput time of resources with online centralized scheduler. In[24] proposed a new approach called Minimum Completion Time (MCT) which is the combination of Opportunistic Load Balancing (OLB) algorithm and Minimum Execution Time (MET) algorithm. As OLB provides poor make span time and MET is not good in consideration of availability and load on the machine. Henceforth combination of both provides best execution time and also considered the availability of machine with balanced load. This approach selects machine that has minimum completion time for task execution. Results show that it is an efficient load balancing approach presents an artificial life technique such as genetic algorithm and tabu search for load balancing[25]. The proposed approach is optimized and can handle complexity of heterogeneous grid environment. On computational grid, each computing node has a task to be executed. After

Table 2. Comparison of load balancing techniques in computational grid

Approach	Model	Compared approach	Technique	Findings	Benefit	Drawback
AHSWDG	Heuristic	Random Approach and non AHSWDG	Ant represents submitted jobs while ant's Pheromone trail Represents the Computational capacity	Initial computational capacity, Maximum transition probability	Even workload distribution and better resource utilization	Average waiting time and throughput value is high.
IEGDC	Dynamic and distributed load balancing technique	EDGC, LBEGS	Considered criteria for scheduling is resource bandwidth and capacity of different resources	Resource State table generated Under loaded Resources have Fastest bandwidth And high Capacity of job execution	Enhance resource utilization and prevent Resources with overloading, Provides better response time, resubmitted time for finished and unfinished Gridlets	Gridlet executes Only below the lower range of normal loaded range of resources and not useful for real time grid system
Schedule_DLB	Dynamic load balancing approach	Random approaches	Find ideal cluster and assign to the resources	Job submitted to the cluster to enhance resource utilization	Minimum completion time and maximise resource allocation	Increased throughput of resources
QOS based cost allocation model	Multicluster computational grid	Round Robin Policy, Capability-based Random Policy, Cluster Minimized Policy and Virtual Resource Agent Policy	Select suitable VRA after arrival of job based on budget and deadline	Dynamic task allocation and efficient resource utilization based on utility function	Maximize resource Utilization for given deadline And budget	Increased job execution time, uneven load distribution
competitive equilibrium solution	Centralized Architecture for nodes	Nash equilibrium Solution(NES) and Global optimal Solution (GOS)	Node processing rate and job arrival rate taken as input then load divided into fractions.	Balance load to maximize utility function and obtain market clearing error	Minimizing average response time for user and system	Static load balancing scheme so resources allocation cost may be high for sometime

Hierarchical Job Scheduling (HJS)	Load balancing approach designed for tree based structure	Flat Structure Job Scheduling (FJS)	By heap memory processors utilized for efficient load	Combination of two aspects Multiple Parameters based Load Balancing (MPLB) and Logical Hierarchical Levels (LHL)	Dividing computation load in efficient way	Specially designed for hierarchical or tree based LAN topology
Decentralized load balancer model	Meta Scheduler Approach	Random Load balancer model	Load Balancing algorithm using Designed using Threshold approach and Boundary value approach	Cluster contains head node responsible for scheduling and computing node for execution of user's	Decrease elapse time, increase idle Resource utilization .	Increase job execution time
NSGA-II using fuzzy operators	Task scheduling in heterogeneous grid environment	Updated NSGA II for fuzzy adaptive Operators.	Fuzzy Adaptive Mutation Operator Is implemented to deal with independent task assignments problems	Load distribution among resources with fuzzy adaptive operator for optimal resource utilization	Minimum job execution time And cost for job and efficient Load balancing	Increased average utilization of resources
Artificial life Technique	Combining genetic algorithm and tabu search	best fit, Min-Min, Max-Min and sufferage	After each task execution acknowledgement will be send to processor than send next task for execution	communication path between two sites worked upon transmission delays and data transmission rate	Optimized Approach with efficient load balancing	Extra storage and processing requirement with extra overhead specially in decreasing cost and processing power.
ELISA and LBA	Decentralized, scalable, adaptive and distributive approach	MELISA	Communication delay and processing delays are constraints	Network created for resources heterogeneity	Balance load when job arrives and Minimize the Execution time and Arrival time	Not suited for large scale grid environment

completion of a task node sends an acknowledgement to scheduler then scheduler sends another task for execution. The task could be data intensive or computational intensive. In the proposed system model, there exists a communication path between two sites working upon transmission delays and data transmission rate. Genetic algorithm implemented for load balancing with selection, crossover, and mutation operations. In tabu search author used adaptive memory for solving problems and considered parameters are tabu list, search intensification, search diversification. Proposed work compared with best fit, Min-Min, Max-Min and sufferage and the result shows that GA and TS perform better in term of make span time and gives better results if the scheduling period increased.

Shah et al proposed a decentralized, scalable, adaptive and distributive approach for balancing load among

resources[26]. They proposed two load balancing algorithm modified Estimated Load Information Scheduling Algorithm (ELISA) and load balancing on arrival. These are job migration algorithm deals with load execution time and arrival time of jobs. Results show that MELISA is 44.9% efficient than ELISA and 18% better performance then ELISA. LBA is designed to balance the load when a job arrives and minimize the execution time of the jobs proposed a Load balanced Min-Min algorithm for task scheduling[27]. Firstly the algorithm schedules the tasks, then remaining task which have not executed yet will be rescheduled. Henceforth minimum execution time for rescheduling can be achieved. This algorithm calculates the maximum completion time for each job, then assigns it to that resource has minimum make span time. Disadvantage of the algorithm is that it could not deal with machine heterogeneity.

A comparative approach has been presented in Table 2 of various load balancing algorithms reported in literature so far. Compared parameters for load balancing are considered approach, techniques and findings of implementations with merit and demerits.

2.3 Minimizing Job Completion Time using ETC on Computational Grid

A novel model for secure grid architecture adapted Artificial Neural Network introduced[28]. The Artificial Neural Network module provides a secure task machine mapping to overcome the problems related to machine failure. It also supports genetic-based schedulers through the integration of the sub-optimal schedules generated by the neural network. Author evaluated the performance for all kinds of grids presents in the grid infrastructure. Both risky and secure, security scheduling modes have considered for implementation. The scheduling criteria considered by the author are makespan time and flow time. Proposed approach also considered running times of schedulers and grid resource failure probability. Results show that it is an efficient approach to get minimum completion time of task.

Estimated time to compute matrix has been used to achieve robustness in computational task[29]. Robustness is an important feature in the heterogeneous computational grid. It can be achieved by work completion latency. In proposed strategy a high-performing scheduling algorithm implemented which is found to be relatively robust against uncertainties in estimated task completion times.

Author presents three scheduling algorithm and implemented them with different workload. First, it investigates the performance of all these algorithms that are related to each other, and then investigate the effects of variation in work completion times on a specific scheduler.

In immediate mode scheduling[30] when a job arrives in the system, it has to be allocated to the resources. For efficient memory allocation author measured some system parameters like make span time, flow time, matching proximity and resource utilization. In proposed algorithm expected time to compute matrix is generated which consist no of jobs and machine, workload on each machines, computing capacity of each machine, and ready time of the machine. In this paper five immediate mode method Opportunistic Load Balancing (OLB), Minimum Completion Time (MCT), Minimum Execution Time (MET), Switching Algorithm (SA) and k-Percent Best (kPB) are implemented and empirically evaluated. Proposed approach designed for the job as well as machine heterogeneity and results calculated for consistent, inconsistent and semi consistent ETC matrices has defined a simulation model to generate ETC matrix. They have generated ETC matrix for all types like inconsistent, semi consistent and consistent.

A comparative approach has been presented in Table 3 of various models that minimized job completion time using ETC reported in literature so far. Compared parameters for MCT are considered approach, techniques and findings of implementations with merit and demerits.

3. Proposed Hybrid Approach including MCT and Load Balancing

It is necessary to achieve systems related and user related objectives in grid environment. The proposed model is constructed for efficient resource utilization from the user's perspective by calculating the execution time of job. The main aim of this model is to achieve "Efficient resource utilization and load balancing on computational grid in a distributed environment." In order to perform optimum utilization of grid resources, make span time should be minimized that leads to scheduler assigns the task an efficient way to the resources. The proposed approach is designed to divide the task in a manner so that each task has minimum make span time[11]. Apart from make span time flow time is also considered which is the response time from task submission to task comple-

Table 3. Comparison of MCT using ETC techniques on computational grid

Approach	Model	Compared approach	Technique	Findings	Benefit	Drawback
ANN +module	Secure task machine mapping approach	Random approach	6 genetic-based Scheduler technique Through the Integration sub-optimal schedules generated	Performance Evaluation in all kind of grid environment and both risky and secure scheduling modes are considered	Minimizing makespan, flowtime and execution time	Cost of average utilization of resources is high
EMGEN Tool	Meta task scheduling approach	Random Approach	ETC consist rows And columns Represented task And resources Respectively then generate memory Table	As per memory requirement and availability MR and MA Table generated Approach tested for consistent, inconsistent and partial consistent matrix	Ensuring robustness In High Performance computing and minimise execution time	Increased value of Makespan and Flowtime. .
Immediate mode scheduling	Memory allocation and scheduling technique	OLB, MCT, MET, Switching Algorithm(SA) and k-Percent Best (kPB)	As job arriving In the system it is allocated to resources considering four parameters makespan time, flowtime, matching proximity and average resource utilization cost	Expected time to compute matrix compute workload for each resources for each job find computing capacity and ready time for each machine	Worked well for task and machine heterogeneity, calculated for all type of ETC matrix	Not efficient for WAN or other distributed environment
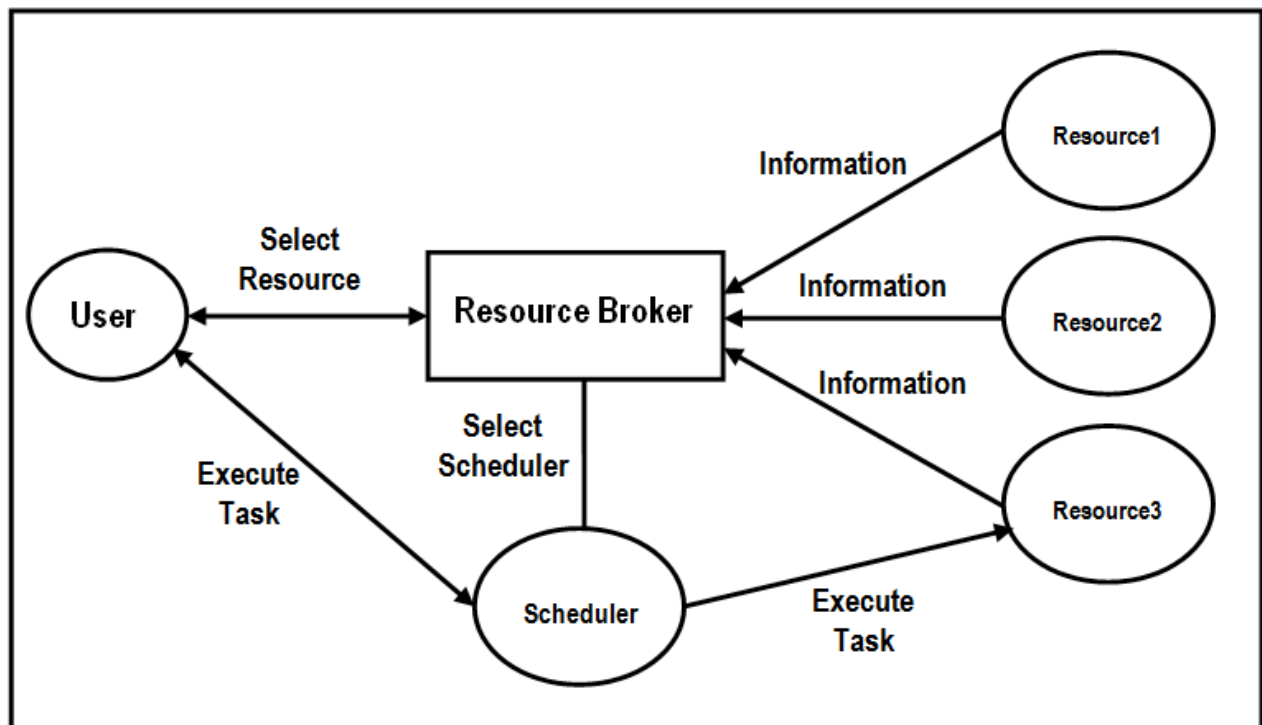
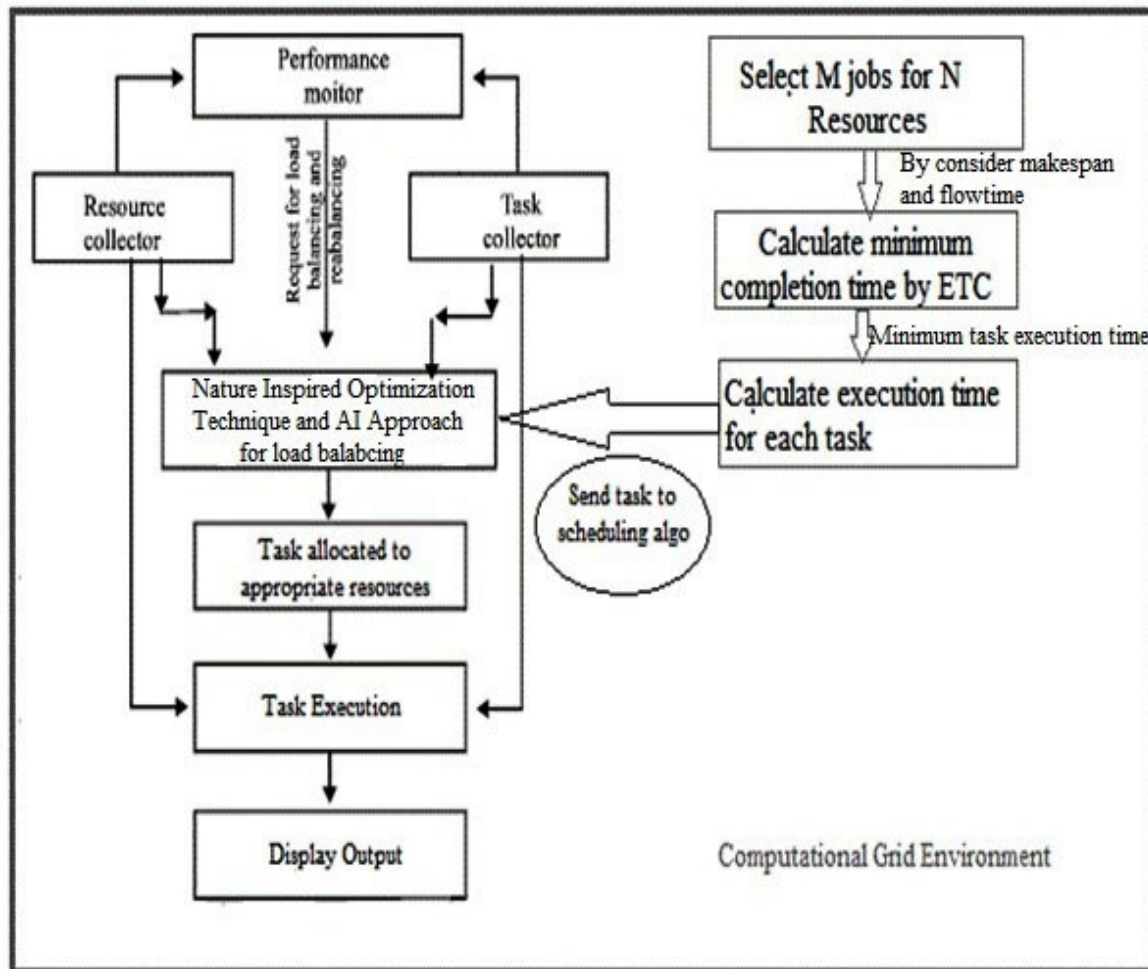
Figure 1. Resource allocation in Computational Grid

Figure 2. A proposed strategy combining minimum task execution time and load balanceFigure

tion. By minimization of flow time, average response time of grid system can be reduced while by minimizing Make span time jobs will be executed in a faster way. Proposed model performs scheduling of n jobs to m resources which is an NP complete problem.

ETC model denotes the expected (estimated) time needed for the completion of the task on machine. The proposed functions of ETC matrix can provide a way to minimizing completion time of job as shown in Figure 1. Minimum completion time could be achieved by minimization of make span and flow time.

The functioning of the proposed research model is illustrated in Figure 2. Firstly an approach will be designed to calculate the number of tasks and corresponding machine then n task will be allocated to m machine for execution. Total computational time for task execu-

tion could be fined by ETC matrix. An approach will be designed in a manner so that task execution time could be minimized by minimizing make span time and flow time. Resultant all tasks executed with minimum completion time thereafter as per the second phase of Figure 2 the task will be assigned to the scheduling algorithm. The scheduling algorithm assigns the task to resources through proper scheduler to balanced load among resources. The Load balancing algorithm will be implemented by any meta heuristic techniques or any AI approach. The scheduling algorithm will be designed in a manner so that it properly utilizes the resources. The proposed model is a novel and hybrid approach which is specifically designed to combine the advantages of minimum completion time and efficient resource utilization.

The proposed work will be done using distributed scheduling and load balancing among resources in the computational grid environment. In this work, the main challenge is after calculating completion time of task, further it performs load balancing among resources to make resource utilization more effective as shown in Figure 2. One or a combination of AI approaches with Meta heuristics may also be adapted for load balancing purpose. The above proposed work is beneficial because minimum completion time with proper load balancing is essential for efficient resource utilization in computational grid. Proposed approach will be implemented using Gridsim.

4. Conclusion and Future Work

Computational grid plays a vital role in resource sharing and load balancing among resources. It has become more common and widely adapted techniques to solve computational problems occurred in scientific and technical areas. Load balancing and job scheduling is an essential need of today's grid environment. A scheduling algorithm could not found to be good an optimum until it doesn't provide good and efficient timings. This paper illustrated various load balancing algorithms in grid environment. Furthermore, a comparative approach between these algorithms like there advantages, disadvantages, findings of authors etc. have also been proposed. ETC is a task to machine mapping matrix that works upon various functions to minimize makes pan time and flow time by which machine completion time can be minimized. Our future work is to develop a novel and hybrid to achieve minimization of job scheduling and load balancing technique on computational grid. The proposed approach is a hybrid approach that contains tasks with minimum completion time and further assign to resources so that no resource will be over utilized and no will be underutilized.

5. References

1. Salimi R, Motameni H, Omranpour H. Task scheduling with Load balancing for computational grid using NSGA II with fuzzy mutation. 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC), Delhi, India. 2012 Dec. p. 79–84. Crossref
2. Shah SN, Zakaria MN, Mahmood AK, Pal AJ, Haron N. Agent based priority heuristic for job scheduling on computational grids. Procedia Computer Science. 2012 Jan; 9:479–88. Crossref
3. Lu K, Zomaya AY. A hybrid policy for job scheduling and load balancing in heterogeneous computational grids. 6th IEEE International Symposium on Parallel and Distributed Computing, California, USA. 2007 Jul. p.19. Crossref
4. Kumar ES, Sumathi A, Zubar HA. A hybrid ant colony optimization algorithm for job scheduling in computational grids. Journal of Scientific and Industrial Research. 2015 July; 74:377–80.
5. Shahid M, Raza Z. A precedence based distributed job scheduling for computational grid. IEEE International Conference on Advance Computing (IACC), Gurgoan, India. 2014 Feb. p.702–7. Crossref
6. Wu M, Sun XH. Memory conscious task partition and scheduling in grid environments. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing. Pittsburg, USA. 2004 Nov. p.138–45.
7. Zarrabi A, Samsudin K. Task scheduling on computational grids using gravitational search algorithm. Cluster computing. 2014 Sep; 17(3): 1001–11. Crossref
8. Goswami S, Das A. Deadline stringency based job scheduling in computational grid environment. 2nd International Conference on IEEE Computing for Sustainable Global Development (INDIACom), New Delhi, India. 2015 Mar. p. 531–6.
9. Shahid M, Raza Z, Sajid M. Level based batch scheduling strategy with idle slot reduction under DAG constraints for computational grid. Journal of Systems and Software. 2015 Oct; 108:110–33. Crossref
10. Selvi S, Manimegalai D. Multiobjective Variable Neighborhood Search algorithm for scheduling independent jobs on computational grid. Egyptian Informatics Journal. 2015 Jul 31; 16(2):199–212. Crossref
11. Abraham A, Buyya R, Nath B. Nature's heuristics for scheduling jobs on computational grids. The 8th IEEE international conference on advanced computing and communications (ADCOM). 2000 Dec. p.45–52.
12. Patel DK, Tripathy C. An improved approach for load balancing among heterogeneous resources in computational grids. Engineering with Computers. 2015 Oct; 31(4): 825–39. Crossref
13. Sathish K, Reddy AR. Enhanced ant algorithm based load balanced task scheduling in grid computing. IJCSNS. 2008 Oct; 8(10): 219.
14. Grosan C, Abraham A, Helvik B. Multiobjective evolutionary algorithms for scheduling jobs on computational grids. International Conference on Applied Computing, Spain. 2007 Jan. p.459–63.
15. Saxena R, Kumar A, Kumar A, Saxena S. AHSWDG: An Ant Based Heuristic Approach to Scheduling and Workload Distribution in Computational Grids. IEEE International Conference on Computational Intelligence

and Communication Technology (CICT), Ghaziabad, India. 2015 Feb. p. 569–74.

16. Hao Y, Liu G, Wen N. An enhanced load balancing mechanism based on deadline control on GridSim. Future Generation Computer Systems. 2012 Apr; 28 (4):657–65. Crossref

17. Rajavel R. De-centralized load balancing for the computational grid environment. IEEE International Conference on Communication and Computational Intelligence (INCOCCI), Tamil Nadu, India. 2010 Dec. p. 419–24.

18. Alam M, Varshney AK. A New Approach of Dynamic Load Balancing Scheduling Algorithm for Homogeneous Multiprocessor System. International Journal of Applied Evolutionary Computation (IJAEC). 2016 Apr 1; 7(2): 61–75. Crossref

19. Xiao P, Hu Z. A novel QoS-based co-allocation model in computational grid. IEEE Global Telecommunications Conference (GLOBECOM 2008), New Orleans, USA. 2008 Nov. p. 1-5. Crossref

20. Chatrapati KS, Rekha JU, Babu AV. Competitive equilibrium approach for load balancing a computational grid with communication delays. Journal of theoretical and applied Information Technology. 2010; 19(2):126–33.

21. Reddy KH, Roy DS. A hierarchical load balancing algorithm for efficient job scheduling in a computational grid testbed. IEEE 1st International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India. 2012 Mar. p.363–8. Crossref

22. Keerthika P, Kasthuri N. A hybrid scheduling algorithm with load balancing for computational grid. International Journal of Advanced Science and Technology. 2013; 58:13–28. Crossref

23. Salimi R, Motameni H, Omranpour H. Task scheduling using NSGA II with fuzzy adaptive operators for computational grids. Journal of Parallel and Distributed Computing. 2014 May; 74(5): 2333–50. Crossref

24. Braun TD, Siegel HJ, Beck N, Bölöni LL, Maheswaran M, Reuther AI, Robertson JP, Theys MD, Yao B, Hensgen D, Freund RF. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed computing. 2001 Jun; 61(6): 810–37. Crossref

25. Subrata R, Zomaya AY, Landfeldt B. Artificial life techniques for load balancing in computational grids. Journal of Computer and System Sciences. 2007 Dec; 73(8): 1176–90. Crossref

26. Shah R, Veeravalli B, Misra M. On the design of adaptive and decentralized load balancing algorithms with load estimation for computational grid environments. IEEE Transactions on parallel and distributed systems. 2007 Dec; 18(12):1675–86. Crossref

27. Kokilavani T, Amalarethinam DG. Load balanced Min-Min algorithm for static meta-task scheduling in grid computing. International Journal of Computer Applications. 2011 Apr; 20(2):43–9.

28. Grzonka D, Kołodziej J, Tao J, Khan SU. Artificial Neural Network support to monitoring of the evolutionary driven security aware scheduling in computational distributed environments. Future Generation Computer Systems. 2015 Oct; 51:72–86. Crossref

29. Kokilavani T, Amalarethinam DG. EMGEN – A Tool To Create ETC matrix Memory Characteristics For Meta Task Scheduling In Grid Environment. The International Journal of Computer Science and Applications. 2013 May; 2(3).

30. Xhafa F, Barolli L. Immediate mode scheduling of independent jobs in computational grids. 21st International Conference on IEEE Advanced Information Networking and Applications (AINA), Canada. 2007 May. p.970–7. Crossref