Analysing the Impact of Heterogeneity with Greedy Resource Allocation Algorithms for Dynamic Load Balancing in Heterogeneous Distributed Computing System

Bibhudatta Sahoo
Department of Computer Sc. and Engineering
NIT, Rourkela, India

Dilip Kumar
Department of Computer Sc. and Engineering
NIT, Jamshedpur, India

Sanjay Kumar Jena
Department of Computer Sc. and Engineering
NIT Rourkela, India

ABSTRACT

Heterogeneous Distributed systems have been an active research area in computer science for the last two decade, task allocation and load balancing have been a major issue associated with such systems. The load-balancing problem, attempts to compute the assignment with smallest possible makespan (i.e. the completion time at the maximum loaded computing node). This paper presents and discusses the dynamic load balancing problem on Heterogeneous Distributed Computing System (HDCS) and analyzes the impact of heterogeneity on computing capability of node on task allocation problem. Since the task assignment problem in NP hard, greedy heuristic algorithms are used to study the impact of heterogeneity on computing resources. The task model is presented as consistent ETC (Expected Time to Compute) matrix in four different heterogeneous computing environments to study the performance of heuristic algorithms to minimize the makespan.

Keywords

Load Balancing, Makespan, Resource Allocation, Heterogeneous Distributed Systems, Expected Time to Compute (ETC).

1. INTRODUCTION

Distributed heterogeneous computing is being widely applied to a variety of large size computational problems. These computational environments are consists of multiple heterogeneous computing modules, these modules interact with each other to solve the problem. Heterogeneous distributed computing system is an integration of distributed computing resources otherwise known as nodes, interconnected with high-speed communication links to meet the computing requirement for a wide verity of tasks [6]. Distributed systems are potential alternative that provides large amount of computing power with only small initial cost alternative to the very expensive massively parallel nodes (supercomputers) [3].

In a Heterogeneous distributed computing system (HDCS), processing loads arrive from many users at random time instants. A proper scheduling policy attempts to assign these loads to available computing nodes so as to complete the processing of all loads in the shortest possible time. The *resource manager* schedules the processes in a distributed system to make use of the system resources in such a manner that resource usage, response time, network congestion, and scheduling overhead are optimized. There are number of techniques and methodologies for scheduling processes of a distributed system. These are *task assignment*, *load-balancing*, *load-sharing* approaches [6, 8, 9].

To improve the utilization of the computing resources or processors, parallel computations require that processes be distributed to processors in such a way that the computational load is spread among the processors. Dynamic load distribution (also called load balancing, load sharing, or load migration) can be applied to restore balance [6]. Dynamic load-balancing algorithms are allocates the task on their arrival to the processor dynamically. These dynamic allocations are possible with the help of four components: *(i)* a *transfer* policy that determines whether a node is in a suitable state to participate in a task transfer, *(ii)* a *selection* policy that determines *which* task should be transferred, *(iii)* a *location* policy that determines to which node a task selected *for* transfer should be sent, and *(iv)* an *information policy* which is responsible *for* triggering the collection of system state information [1, 6, 8, 10].

Due to heterogeneity of computing nodes, tasks encounter different execution times on different processors. Therefore, research should address scheduling in heterogeneous environment. The distributed nature of underlying resources presents problems not present in closely coupled systems, such as communication overheads, or heterogeneity of resources. A poor allocation of tasks to processors could nullify the benefits of using a distributed system by inefficiently utilizing the system resources [2]. Balancing the load of computing resources or nodes is considered as a special case of resource allocation problem in distributed computing system. Resource allocation is very important to achieve load balancing among the computing nodes. The resource allocation is defined as the act of assigning (matching) each task to a node and ordering (scheduling) the execution of the tasks on each node. Resource allocation is generally performed based on estimated computation time of each task on each computing node (processor).The load-balancing problem, aim to compute the assignment with smallest possible makespan (i.e. the completion time at the maximum loaded computing node) The load distribution problem is known to be NP-hard [4], [5] in most cases and therefore intractable with number of tasks and/or the computing node exceeds few units. Due the complexity of dynamic load balancing problem majority of researchers uses heuristic algorithm to obtain near optimal solutions. The integration of heterogeneous computing nodes in distributed computing environment results a considerable amount of computing power to support the demand of large computational problem. Here, the load balancing is resource allocation policies which takes a task as a whole and assign it to a computing node [2].

This paper considers the problem of finding a sub-optimal solution for load balancing in heterogeneous distributed system using heuristic algorithm. The rest of the paper is organized as follows. The next section discusses related work on dynamic load balancing in Heterogeneous Distributed Computing System. *Section 3* presents Heterogeneous distributed computing system (HDCS) structure and the load-balancing problem. *Section 4* describes task model with help of Expected Time to Compute **(**ETC) matrix addressing task heterogeneity. *Section 5* describes the general framework for dynamic load distribution algorithm using greedy paradigm. Behaviours of different greedy load balancing algorithm have been simulated with the simulator developed using Matlab, against four different node heterogeneity. The results of the simulation present the performance of resource allocation algorithms with scalability of tasks on 60 computing nodes and varying tasks arrival in *Section 6.* Finally, conclusions and directions for future research are discussed in *Section 6.*

2. RELATED WORK

Dandamudi [1] addressed dynamic load sharing in distributed systems and established that load sharing improves performance by moving work from heavily loaded nodes to lightly loaded nodes. A general model for heterogeneous distributed/parallel computer system proposed by Li and Kameda [2] and used to formulate the multiclass job load balancing problem as a nonlinear optimization problem. An algorithmic approach to load balancing problem is presented in [4]. Different form of linear programming formulation of the load balancing problem has been discussed along with greedy, randomized and approximation algorithm to produce sub-optimal solutions to the problem. The solution to this intractable problem was discussed under different algorithm paradigm. Modeling of optimal load balancing strategy using queuing theory was proposed by Francois Spies (1996). This is one of the pioneer works reported in the literature that presents an analytical model of dynamic load balancing techniques as M/M/k queue and simulate with fundamental parameters like load, number of nodes, transfer speed and overload rate [18]. Most appropriate queuing model for homogeneous distributed system can be M/M/m/n, has been analyzed in [20]. Queuing-Theoretic models for parallel and distributed system can be found in [17, 19]. General Job scheduling problem of n tasks with m machines, is presented as an optimization problem in [19] to minimize the makespan as NP complete with exponential solution space. Karate and Hilzer simulated load sharing policies for two job class (task heterogeneity) on heterogeneous distributed system with two different type of computing nodes (processors) [5]. A new mapping heuristic is presented by Maheswaran and Siegel in [16] is based on centralized policy to reduce overall execution time in heterogeneous computing system. First time ETC framework for task on for resource allocations in heterogeneous distributed computing systems was addressed by S Ali and et al [15]. The quantification of heterogeneity in DCS is addressed in [14] along with static and dynamic heuristics for mapping tasks to machines in such HC systems. Attiya and Hamam presented a 0-1 integer programming model for load balancing in Heterogeneous Distributed System and proposed a two phase algorithm (simulated annealing and Branch-and-Bound) to minimize the makespan [3]. Zomaya and Teh proposed a dynamic load balancing framework on genetic algorithm that uses central scheduler approach to handle all load balancing decisions [13]. Jong-Chen Chen and *et al.* [25] investigated the contribution made by evolutionary learning on dynamic load balancing problems in distributed computing system. Grosu and Chronopoulos [26] designed an agent based resource allocation protocol to solve static load balancing problem in large scale heterogeneous distributed computing system (Grid). A performance comparison between two static and two dynamic lo ad balancing algorithm on heterogeneous distributed system has been compared in[7] , to show that static load balancing policies have an edge over dynamic load balancing policy considering overhead required by dynamic policy. Bora Ucar and *et al.* have considered the assignment of communicating tasks to heterogeneous processors[8], that uses a task clustering method based upon execution time to allocate the task though the heuristic techniques. Cluster heterogeneity in load balancing algorithm related to (i) the state measurement stage and (ii) to the initiation rule is considered in [9], to study the performance of load balancing algorithm in heterogeneous environments. Tracy Braun et al. examined, 11 different heuristics(Opportunistic Load Balancing, Minimum Execution Time, Minimum Completion Time, Min–min, Max–min, Duplex, Genetic Algorithm, Simulated Annealing, Genetic Simulated Annealing, Tabu, and A*) on Mixed-machine heterogeneous computing (HC) environments to minimize the total execution time of the metatask[29]. Heuristic algorithms are used for solving NP-complete problems[30] to obtain approximate or sub-optimal solutions[22], hence different heuristic algorithms has been used by various researcher to solve load balancing problem in HDCS.

3. HETEROGENEOUS DISTRIBUTED COMPUTING SYSTEM MODEL

Heterogeneous distributed computing system (HDCS) utilizes a distributed suite of different high-performance nodes, interconnected with high-speed links, to perform different computationally intensive applications that have diverse computational requirements [5, 6, 14, 15, and 16]. Distributed computing provides the capability for the utilization of remote computing resources and allows for increased levels of flexibility, reliability, and modularity. In heterogeneous distributed computing system the computational power of the computing entities are possibly different for each processor as shown in *figure 1*[1, 3, 4]. A large heterogeneous distributed computing system (HDCS) consists of potentially millions of heterogeneous computing nodes connected by the global Internet. The applicability and strength of HDCS are derived from their ability to meet computing needs to appropriate resources [2, 3, 9]. Heterogeneity in DCS can be expressed by considering three systems attributes *(i) Processor with computing node*, (ii) *memory*, and *(iii) networking* [3]. The metrics used to quantify the processor or node processing power by means of processing speed and represented with FLOPS (Floating point Operations per Second) and can be measured through LINPACK. Memory attributes are measured as the available memory capacity to support the process. The networking attributes are the link capacity associated with transmission medium, propagation delay and available communication resources [3]. In this paper, simulation has been carried out by considering processing power of the node, which can be represented as Markovian service time distribution [11, 16, 18, 20].

In general, load-balancing algorithms can be broadly categorized as centralized or decentralized, dynamic or static, periodic or non-periodic, and those with thresholds or without thresholds [2, 6, 10]. A centralized load-balancing algorithm framework is preferred because; it imposes fewer overheads on the system than the decentralized algorithm [6, 13]. Centralized load balancing algorithms requires the global information on computing nodes at a single location and the load balancing policy is initiated from the central location. Heterogeneity of architecture and configuration complicates the load balancing problem [6]. Heterogeneity can arise due to the difference in task arrival rate at homogeneous processors or processors having different task processing rates.

All computational tasks are capable of executed on any computing nodes of DCS. A single computing node that acts as a central scheduler or resource manager of the DCS collects the global load information of other computing nodes. Resource management sub systems of the HDCS are designated to schedule the execution of the tasks dynamically as that arrives for the service. HDCS environments are well suited to meet the computational demands of large, diverse groups of tasks. The problem of optimally mapping also defined as matching and scheduling. A basic assumption is that all computing nodes are always available for processing.

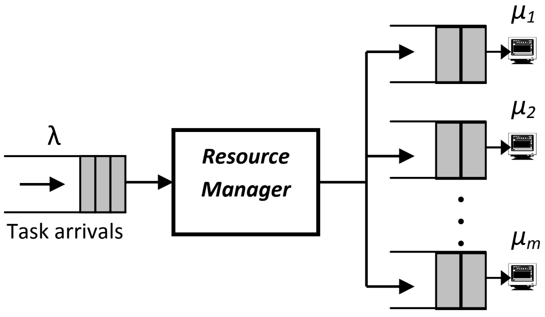

Figure: 1 Heterogeneous Distributed Computing System with central scheduler

This paper uses the characterization model proposed by Shoukat Ali and et al as the basic framework to study the impact of system heterogeneity against different heuristic resource allocation algorithms [15]. A heterogeneous distributed computing system (HDCS) consists of a set of M = {M_1, M_2, … M_m}, m independent heterogeneous, uniquely addressable computing entity (computing nodes). Let there are T = {t_1, t_2, …, t_n} *n* number of tasks with each task t_i has an expected time to compute t_{ij} on node M_j. The entire task has expected time to compute on m nodes of HDCS. Hence the generalized load-balancing problem is to assign each task to one of the node M_j so that the loads placed on all nodes are as "balanced" as possible [4].

Let A(j) be the set of jobs assigned to node M_j; and T_j be the total time machine M_j have to work to finish the entire task in A(j). Hence $T_j = \sum_{t_i \in A(j)} t_{ij}$; for all task in A(j). This is otherwise denoted as L_j and defined as load on node M_j. The basic objective of load balancing is to minimize make span, which is defined as maximum loads on any node (T = $\max_{j:1:m}$ (T_j). Let x_{ij} correspond to each pair (i, j) of node $M_j \in M$ and task $t_i \in T$.

- $x_{ij} = 0$; implies that task *i* not assign to node *j*.
- $x_{ij} = t_{ij}$; will indicate load of task *i* on node *j*.

For each task t_i $\sum_{j=1}^{m} x_{ij} = t_{ij}$; for all task $t_i \in T$

The load on node M_j can be represented as $L_j = \sum_{i=1}^{n} x_{ij}$, where $x_{ij} = 0$ whenever task $t_i \notin A(j)$. The load balancing problem aims to find an assignment that minimizes the maximum load. Let L be the load of a HDCS with m nodes.

Hence the generalized load balancing problem on HDCS can be formulated as

Minimize L

$$\sum_{j=1}^{m} x_{ij} \quad = \quad t_{ij} \text{ , for all } \quad t_i \quad \in \quad T$$

$$\sum_{i=1}^{n} x_{ij} \quad \leq \quad L, \text{ for all } M_j \quad \in \quad M$$

$$x_{ij} \in \{0, \mathrm{t_{ij}}\} \quad , \qquad \text{for all} \quad t_i \quad \in \quad T \text{ and } M_j \quad \in \quad M$$

$$x_{ij} = 0 \quad , \qquad \text{for all} \quad t_i \quad \notin \quad A(j)$$

Feasible assignments are one-to-one correspondence with x satisfying the above constraints [4]. Hence an optimal solution to this problem is the load on a machine (corresponding assignment).

The problem of finding an assignment of minimum makespan is NP-hard [3, 4, and 30]. The problem is therefore untractable with number tasks or computing nodes (processors) exceeds a few units. The solutions to this can be obtained using a dynamic programming algorithm *O(n L^m)*, where L is the minimum makespan. The load balancing problem has been evenly treated, in both the fields of computer science and operation research. The algorithm approaches used for load balancing problem are roughly classified as (i) exact algorithms and (ii) heuristic algorithms [3, 22].

Queuing models are used as the key model for performance analysis and optimization of parallel and distributed system [11, 17]. The HDCS can be modeled as M/M/m/n (Markovian arrivals, Markovian distributed service times, m computing nodes as server, and space for n ≥ m tasks in the system) multi-server queuing system with m servers as computing nodes. However, the heterogeneous multi-server queuing systems are not adequately addressed in research with respect to certain quality of service [17, 21]**.**

The HDCS is modeled as M/M/m/n queuing system with node M_1 is the fastest computing node and M_m is the slowest computing node. Assume that service time follow exponential distribution with service rate so that $\mu_1 > \mu_2 > \dots \mu_m$, where μ_I is the service rate of node M_i. The arrivals of the tasks at the central server or resource manager are modeled as Poisson with arrival rate λ. Each computing nodes can be modeled as shown in figure 2. The tasks that are to be executed at a node are under the control of local scheduler and the scheduling policy of the node is responsible for the execution of the assigned task. First Cum First serve (FCFS) policy is being used at computing nodes, which can be modeled as M/M/1 queuing system [11, 26].

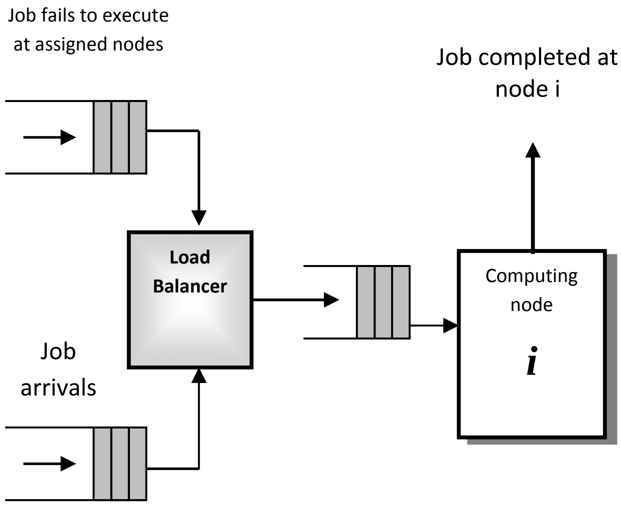

Figure: 2 Job flow at computing node in distributed load balance model

Each task exactly requires the service of exactly one computing node or processor. Resource manager acts as interface between the user and computing resource or nodes , the tasks are in the waiting with resource manager, prior to the assignment of tasks to the computing nodes. Let { X(t), t ∊ R^+} be the number of tasks with the systems at time t, which includes the tasks are with computing nodes and tasks are with resource manager. Let $P_n(t)$ = P(X(t) = n), n=2,3,4, …. Denotes the probability that there is n number of tasks at time t. Let the system state with m computing nodes are represented as $x \;=\; (q, d_1, \cdots, d_m)$, where q is the queue length and d_i denotes the state of the node M_i, d_i = 1 , indicate the active state of node M_i, and d_i = 0 otherwise. The number of busy computing nodes in the HDCS denoted as $d \;=\; \sum_{i=1}^{m} d_i$ [22]. Threshold levels for the node M_i can be denoted as $q^*(x) \;=\; q^*(d_1, \dots, d_{i-1}) \;\; q_i^*$ for each system state x, where d_k =1, for k 1, …, i-1. For $\lambda \leq \mu_k$ threshold levels can be computed analytically as follows:

$$q_i^* \quad = \quad \frac{1}{\mu_i}\sum_{k=1}^{i-1} \mu_k \quad -(i-1), \quad i = 2,3, \cdots, m$$

4. TASK MODEL AND LOAD BALANCING TECHNIQUES

4.1 Task model on HDCS

The tasks are arriving from the different nodes to the resource manager has the probability to be allocated to any of the m computing nodes. Hence the tasks are characterized by expected time to compute (ETC) on all *m* computing nodes, can be represented as follows, In ETC matrix, the elements along a row indicate the execution time of a given task on different nodes[15]

$$\begin{array}{cccccc} & M_1 & M_2 & \cdots & M_J & \cdots & M_m \\ T_1 & t_{11} & t_{12} & \cdots & t_{1j} & \cdots & t_{1m} \\ T_2 & t_{21} & t_{22} & \ldots & t_{2j} & \ldots & t_{2m} \\ T_i & t_{i1} & t_{i2} & & t_{ij} & & t_{im} \\ T_n & t_{n1} & t_{n2} & & t_{nj} & & t_{nm} \end{array}$$

In literature of distributed computing research different task models are used (i) Task graph(TG) or Task interaction graph(TIG)IG, (ii) expected time to compute(ETC) matrix. The task graphs are both directed and undirected weighted graph that represents process or task to be executed. However majority of the models are not representing any mathematical model for quantifying task heterogeneity. The ETC model presented in [15] are characterized by three parameters (i) machine heterogeneity, task heterogeneity and consistency. The task heterogeneity can be represented with two categories (i) consistent and (ii) inconsistent, here a consistent ETC matrix the computing nodes are arranged in the order of their processing capability or may be arranged as decreasing order of FLOPS. In particular a node M_i has a lower execution time than node M_j for task t_k , then $t_{ki}<t_{kj}$. Inconsistent ETC matrix is resulted in practice, when HDCS includes different type of machine architectures. (HPC clusters, Multi-core processor based workstations, parallel computers, and work station with GPU units). In literature most of the task execution times are uniformly distributed. To generate ETC matrix, we have used range base ETC generation technique discussed in [15] and added one component as arrival time of task. The arrival patterns of the task are based on Poisson distribution. The simulation has been carried out using three different λ values to generate the ETC matrix and name the scenario as slow, moderate and, fast arrival of task. The ETC matrices are generated considering four different machine heterogeneity.

4.2 Greedy Load Balancing Algorithms

A dynamic load distribution algorithm must be general, adaptive, stable, fault tolerant and transparent to applications. Load balancing algorithms can be classified as (i) global vs. local, (ii) centralized vs. decentralized, (iii) Non-cooperative vs. cooperative, and (iv) adaptive vs. non-adaptive[6,10]. This paper presents centralized load balancing algorithm, a central node collects the load information from the other computing nodes in HDCS. Central node communicates the assimilated information to all individual computing nodes, so that the nodes get updated about the system state. This updated information enables the nodes to decide whether to send the process to other nodes or accept new process for computation. The computing nodes may depend upon the information available with central node for all allocation decision.

Makespan as load balancing metric for a given distributed computing environment should reasonably predict the performance of the system [14]. This is based on a mathematical model of load balancing problem discussed in algorithm design [4]. This paper follows a simple Greedy-Balance algorithm frame work, discussed by Kleinberg and Tardos to suggest a generalized greedy load balancing algorithm for HDCS. The proposed algorithm uses the ETC matrix along with arrival time for each task for load balancing.

Algorithm 1: **Greedy resource allocation algorithm.**

Input: ETC (MaxTask, MaxNode), AT(MaxTask)

Output: makespan

```
1:    $T_i$ ← 0 for all node $M_i$
2:    $A(i)$ ← ∅ for all node $M_i$
3:    **for** j=1 to MaxTask
4:      **Let $M_i$ be a node with minimum min $T_i$**
5:      Allocate task J to Node $M_i$
6:      $A(i)$ ← $A(i) \cup \{j\}$
6:      $T_i$ ← $T_i + t_{ji}$
7:    **end for**
8:    $T$ ← $max_i T_i$
9:    end
```

Experiments are conducted with three greedy allocation policies namely FCFS, MINMIN and MINMAX along with a random resource allocation algorithm. A randomized resource allocation algorithm is selected along with the heuristic algorithms because the randomness can (probabilistically) guarantee average case behavior as well as it produces efficient approximate solutions to intractable problems.

These greedy allocations of task to the computing nodes can be realized by modifying the step no 4 of algorithm 1. The FCFS algorithm follows the order of arrival time of the task with central scheduler. The *random task allocation algorithm* selects the node randomly from m nodes to allocate task t_j. MINMIN algorithm selects the task with minimum ETC on node to be allocated. MINMAX are realized by selecting the task with maximum ETC on node to be allocated.

5. PERFOMANCE EVALUATION AND COMPARISION

This section presents the performance of different resource allocation algorithms to minimize the makespan. A series of experiments have been conducted through simulation on three different types of Distributed computing system model with n number of computing elements or processors, where n is the sum of computing elements with n processors each having a single processor. In general four different HDCS are considered for this study: (i) all computing processors are homogeneous with service rate μ, (ii) half of the computing processors with service rate μ, and other m/2 processors with service rate as 2μ (iii)m/2 no of processors are homogeneous and other half of the nodes are heterogeneous (iv)all computing processors are heterogeneous so that for any to computing processors $M_{i,}$ and M_j , $\mu_i \neq \mu_j$.

For the analysis of the simulation results through the graph, the ETC matrix uses, the expected completion time of task uniformly distributed {1, 500}time unit or seconds. Central resource manager or load balancing service should be able to effectively control the computing resource for dynamic allocation to the tasks. In first group of experiment uses consistant heterogeneous task matrix along with arrival time of each task and vary the number of nodes up to 100 on four different HDCS with increase in heterogeneity of computing nodes. Results are obtained as makespan, and plotted against vs. no of computing nodes in the figures 3(a)(b)(c)(d). In the experiments in the section, It is observed that 60 computing nodes can be an ideal to carry out the further experiments on four different DCS.

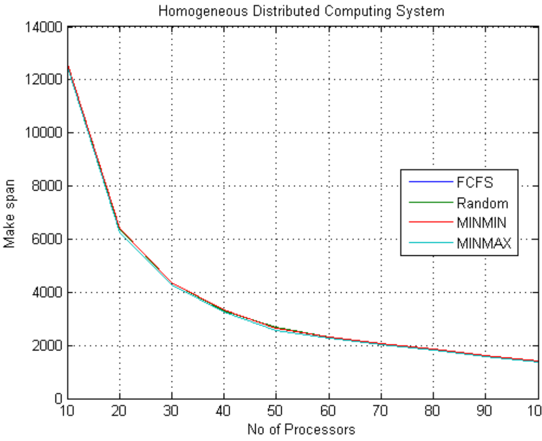

Figure 3(a) Performance of resource allocation algorithms on HDCS (homogeneous computing nodes)

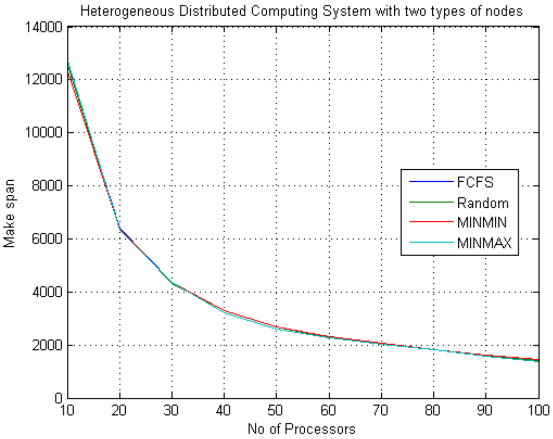

Figure 3(b) Performance of resource allocation algorithms on HDCS (two type of computing nodes)

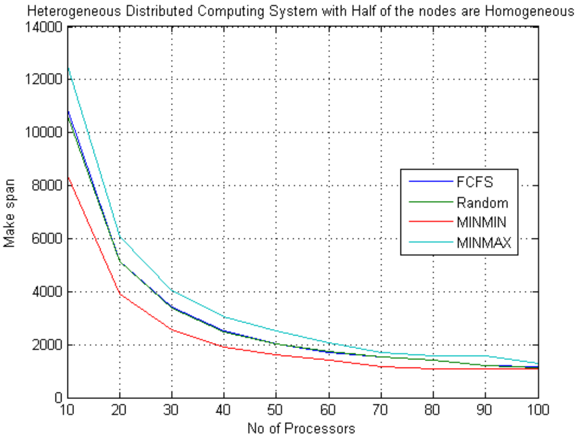

Figure 3(c) Performance of resource allocation algorithms on HDCS (half of the computing nodes are homogeneous)

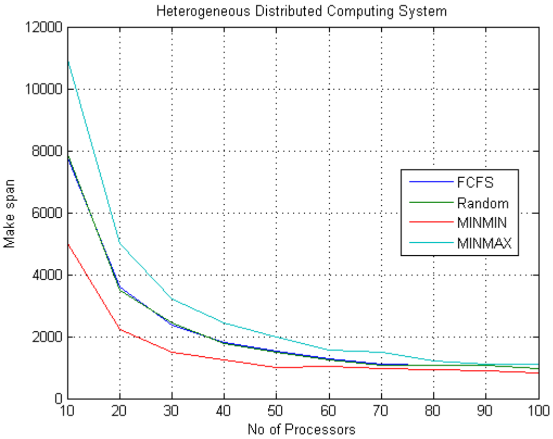

Figure 3(d) Performance of resource allocation algorithms on HDCS (all computing nodes are heterogeneous)

5.1 Minimizing make span with homogeneous computing processors

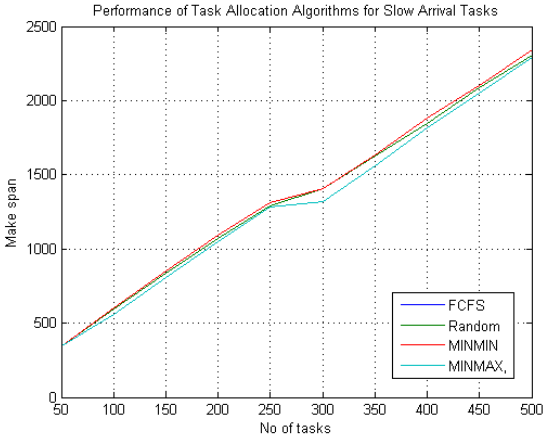

Figure 4(a) Makespan: varied task number for slow arrival of tasks

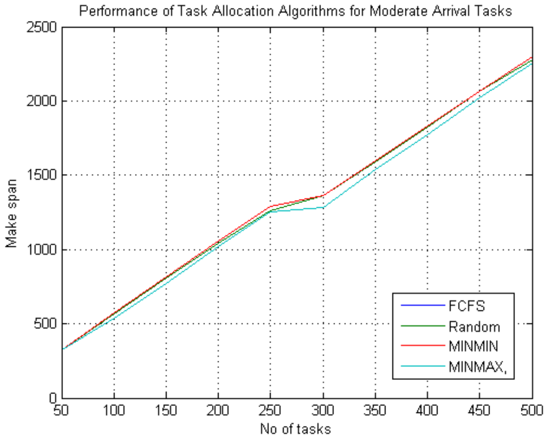

Figure 4(b) Makespan: varied task number for moderate arrival of tasks

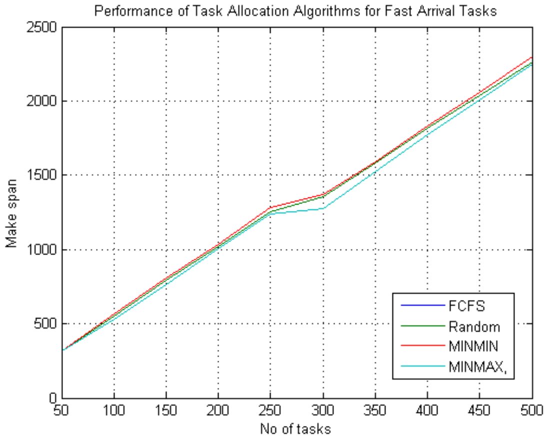

Figure 4(c) Makespan: varied task number for fast arrival of tasks

Figure 4.(a, b, c) shows the performance of heuristic algorithms on HDCS with all computing nodes having identical service rate. The resource allocation heuristics has no significant impact on the task arrival rate when all nodes in the DCS are homogeneous. The plotted makespan value indicates similar performance of all; however their dependency on number of task is significantly indicated from the results.

5.2 Minimizing make-span on DCS with heterogeneous computing nodes with two type of computing processors.

These experiments are based upon the model of distributed computing environment with two type of computing nodes. In this HDCS half of the computing processors with service rate µ, and other m/v2 processors are with service rate as 2µ.

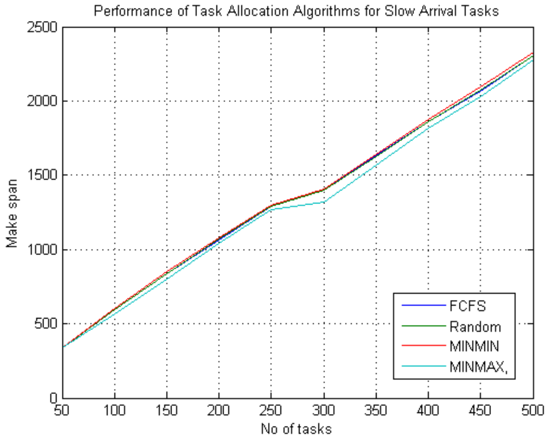

Figure 5(a) Makespan: varied task number for slow arrival of tasks

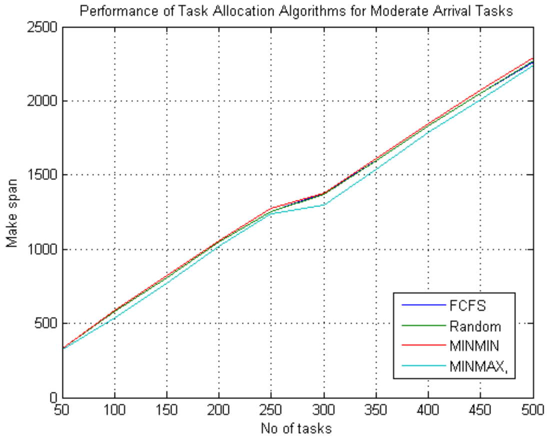

Figure 5(b) Makespan: varied task number for moderate arrival of tasks

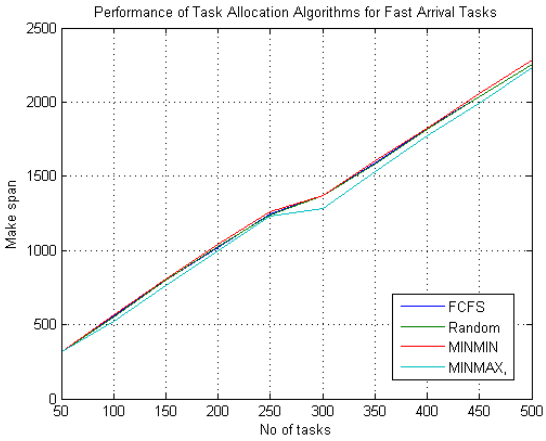

Figure 5(c) Makespan: varied task number for fast arrival of tasks

To study the performance of resource allocation algorithms with uniformly distributed service time of the different task set on ETC matrix along with arrival time. The simulations indicates no significant the dependency of resource algorithm (load balancing algorithms) on makespan; as makespan increases with the number of tasks. It also observed that arrival rate of the task has no significant impact.

5.3 Minimizing make-span on DCS with half of the homogeneous half heterogeneous computing processors

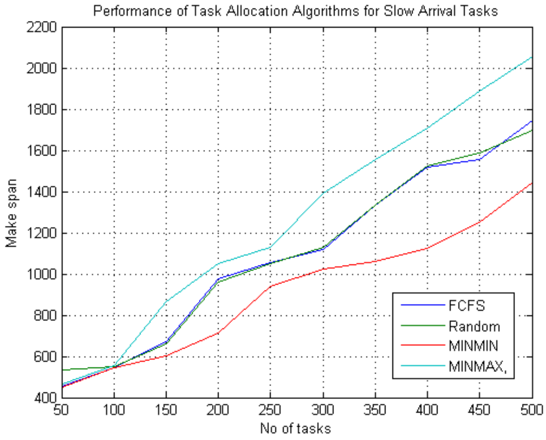

Figure 6(a) Makespan: varied task number for slow arrival of tasks

The dependency of load balancing on *no of the tasks* arrived with Poisson distribution is shown in Figure 6(a, b, c). Moreover, it explains the performance of MINMIN algorithm with different arrival rate of the task to the central scheduler. Also it indicates that FCFS and Randomized algorithms exhibit similar performance in terms of makespan. It also indicates highest makespan, average makespan and minimum makespan value against the varying of task.

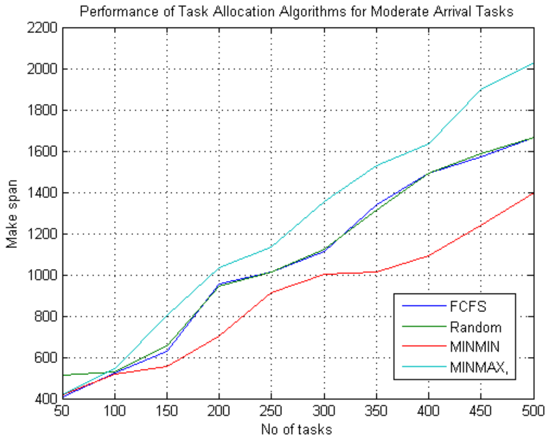

Figure 6(b) Makespan: varied task number for moderate arrival of tasks

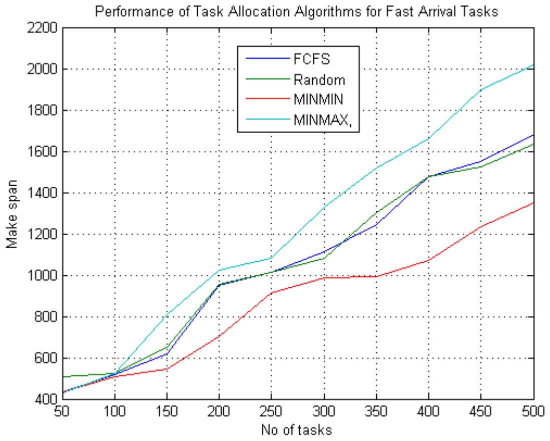

Figure 6(c) Makespan: varied task number for fast arrival of tasks

5.4 Minimizing make-span on DCS with heterogeneous computing processors

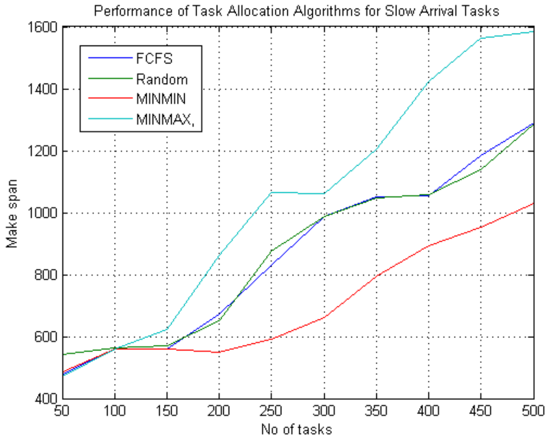

Figure 7(a) Makespan: varied task number for slow arrival of tasks

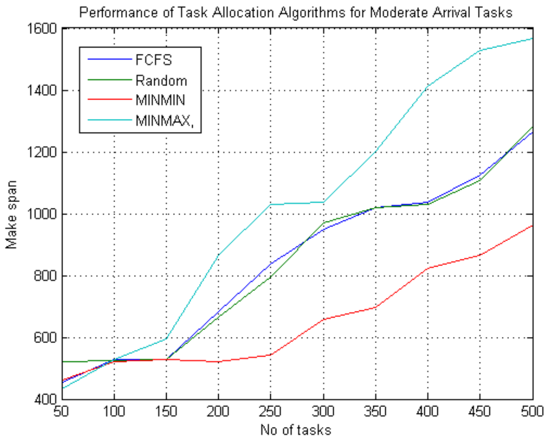

Figure 7(b) Makespan: varied task number for moderate arrival of tasks

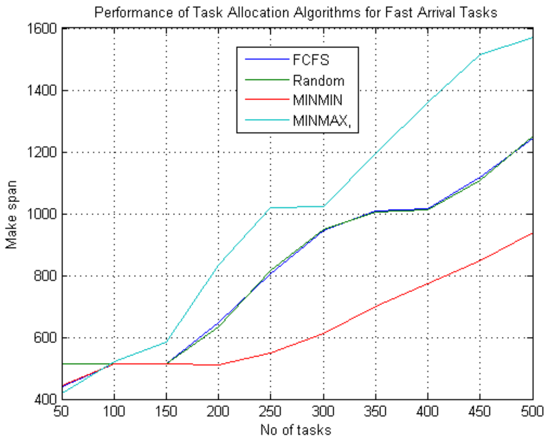

Figure 7(c) Makespan: varied task number for fast arrival of tasks

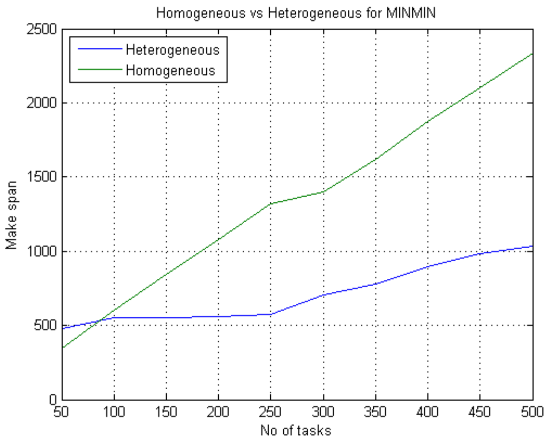

Figure. 8(a) Makespan for 60 processor for moderate arrival of tasks with MINMIN algorithm

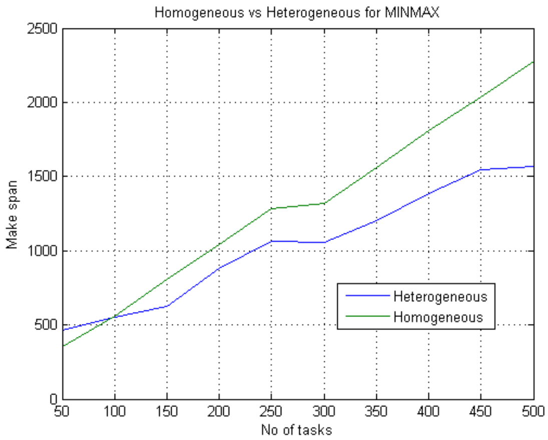

Figure. 8(b) Makespan for 60 processor for moderate arrival of tasks with MINMAX algorithm

Figure 7(a,b,c) shows experimental results of four evaluated algorithms on heterogeneous system with 60 nodes. It is observed that MINMIN significantly minimizes makespan over the three alternatives, where as FCFS and Randomized algorithms exhibit similar performance in terms of makespan. An interesting observation drawn from the figure 8(a) is that MINMIN algorithm shows about 56% less in makespan in process of load balancing on 60 heterogeneous computing nodes. The simulation conducted in varying the number of tasks, a dependency of makespan with no of task can also be observed. From figure 8(b) it has been observed that MINMAX algorithm on heterogeneous computing system produces better results in comparison to homogeneous computing systems with identical computing nodes with approximately 30% less in makespan.

6. CONCLUSION

The load balancing of distributed computing system becomes a major research issue to utilize the ideal computing resources. Load balancing mechanism distributes computing loads by assigning tasks to be executed, by minimizing the makespan. A number of experiments were conducted to examine the performance of greedy resource allocation algorithms against makespan to study the task and node heterogeneity in HDCS with considering arrival time of the task. MINMIN produces optimal performance and able to explore the heterogeneity of computing resources to minimize the makespan. This paper presents experiment result with one class of task model [15], further investigations may be carried out for inconsistent and partially –consistent ETC matrix to study the impact of greedy resource allocation techniques. An extensive simulation with more no of tasks with different node heterogeneity also results similar performance as discussed in section 4

REFERENCES

[1] Sivarama P. Dandamudi, Sensitivity evaluation of dynamic load sharing in distributed systems, *IEEE Concurrency,6*(3), 1998, 62-72.

[2] Jie Li, & Hisao Kameda, Load balancing problems for multiclass tasks in distributed/parallel computer systems, *IEEE Transactions on Computers, 47*(3), 1998, 322-332.

[3] Gamal Attiya & Yskandar Hamam, Two phase algorithm for load balancing in heterogeneous distributed systems, *Proc. 12th IEEE EUROMICRO conference on Parallel, Distributed and Network-based processing,* Coruna, Spain 2004, 434-439.

[4] Jon Kleinberg & Eva Tardos, *Algorithm Design* (Pearson Education Inc. 2006).

[5] Helen D. Karatza, & Ralph C. Hilzer, Load sharing in heterogeneous distributed systems, *Proceedings of the Winter Simulation Conference*, *1*, San Diego California, 2002 Page(s): 2002, 489 – 496.

[6] Jie Wu, *Distributed system design,*(CRC press, 1999)

[7] Y.Zhang, H.Kameda & S.L.Hung, Comparison of dynamic and static load-balancing strategies in heterogeneous distributed systems, *IEE proceedings in Computer and Digital Techniques,144*(2), 1997, 100-106.

[8] Bora Ucar, Cevdet Aykanat, Kamer Kaya, & Murat Ikinci, Task assignment in heterogeneous computing system, *Journal of parallel and Distributed Computing*, *66*, 2006, 32-46.

[9] Marta Beltran, Antonio Guzman, & Jose Luis Bosque, Dealing with heterogeneity in load balancing algorithm, *Proc. 5th IEEE International Symposium on Parallel and Distributed Computing,* Timisoara, Romania**,** 2006, 123-132.

[10] B. A. Shirazi, A. R. Hurson, & K. M. Kavi, Scheduling and load balancing in parallel and distributed systems, CS press, 1995.

[11] K. S. Trivedi, Probability and statistics with reliability, queuing and computer science applications, Prentice Hall of India, 2001.

[12] Bibhudatta Sahoo, S. Mohapatra, S.K. Jena, "Genetic Algorithm Based Dynamic Load Balancing Scheme for Heterogeneous Distributed Systems ",Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2008, Las Vegas, Nevada, USA, July 14-17, 2008, 2 Volumes. CSREA Press 2008, ISBN 1-60132-084-1

[13] A. Y. Zomaya, & Y. H. Teh, Observations on using genetic algorithms for dynamic load-balancing, *IEEE Transactions on Parallel and Distributed Systems, 12*(9), 2001, 899-911.

[14] H. J. Siegel, and S. Ali, Techniques for mapping tasks to nodes in heterogeneous computing, Systems, Journal of Systems Architecture, 46(8), 2000, 627—639.

[15] S. Ali; H.J. Siegel, M. Maheswaran, D. Hensgen; , "Task execution time modeling for heterogeneous computing systems ," *Heterogeneous Computing Workshop, 2000. (HCW 2000) Proceedings. 9th* , vol., no., pp.185-199, 2000.

[16] M. Maheswaran, S. Ali, H.J. Siegel, D. Hensgen, and R. F. Freund, " Dynamic mapping of a class of independent tasks into heterogeneous computing, special Issues on Software Support for Distributed Computing, Vol.59, No.2, pp.107-131, Nov. 1999.

[17] Onno Boxma, Ger Koole, and Zhen Liu,"Queueing-theoretic solution methods for models of parallel and distributed systems", Centrum voor Wiskunde en Informatica, Department of Operations Research, Statistics, and System Theory,1994.

[18] Francois Spies, "Modeling of optimal load balancing strategy using queuing theory" Micro processing and Microprogramming, vol.41, 1996, pp.555-570.

[19] Caffrey, James, and Graham Hitchings. "Makespan distributions in flow shop scheduling." International Journal of Operations & Production Management 15.3 (1995): 50-58.

[20] Rindzevicius, R., D. Poškaitis, and B. Dekeris. "Performance measures analysis of M/M/m/K/N systems with finite customer population." Electr. Electr. Eng 3.67 (2006): 65-70.

[21] Smith, David K. "Calculation of steady-state probabilities of M/M queues: further approaches." Journal of Applied Mathematics and Decision Sciences 6.1 (2002): 43-50.

[22] Thomas V. Christensen, "Heuristic Algorithms for NP-Complete Problems" Project report, Institute of Informatics and mathematical Modelling, Technical University of Denmark.2007.

[23] B Krishna Kumar, S. Pavai Madheswari, and K. S. Venkatakrishnan.. "Transient solution of an M/M/2 Queue with Heterogeneous Server Subject to Catastrophes." Journal of Information and Management Sciences, Vol.(18), no(1) (2007): 63-80.

[24] Gamal Attiya, and Yskandar Hamam, "Task allocation for maximizing reliability of distributed system: A simulated annealing approach", Journal of parallel and Distributed Computing, vol.66, pp. 1259-1266, 2006.

[25] Jong-Chen Chen, Guo-Xun Liao, Jr-Sung Hsie, Cheng-Hua Liao: A study of the contribution made by evolutionary learning on dynamic load-balancing problems in distributed computing systems. Expert Syst. Appl. 34(1): 357-365 (2008)

[26] Daniel Grosu, and Anthony T. Chronopoulos, "Algorithmic Mechanism Design for load balancing in Distributed system", IEEE transaction on system man and cybernetics-Part B: cybernetics, vol.34, No.1, PP. 77-84, 2004.

[27] Chin-Ching Chiu, Chung-Hsien Hsu, and Yi-Shiung Yeh, "A genetic algorithm for reliability-oriented task Assignment with k duplications in distributed systems", IEEE Transaction on reliability, vol.55, No. 1, pp. 105-117, March 2006

[28] A. J. Page and T. J. Naughton, "Framework for Task Scheduling in Heterogeneous Distributed Computing Using Genetic Algorithms," Artificial Intelligence Review, vol. 24, pp. 415–429, November 2005.

[29] Tracy D. Braun, et' al., "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing, vol. 61, pp.810-837, 2001

[30] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, 1979.