Performance Evaluation of a Mobile Wireless Computational Grid Model

Adeyelu Adekunle Adedotun[1], Ge Tivlumun Cosmas[2], Egahi, Musa[3]

1, 2, 3 Department of Mathematics and Computer Science, Benue State University, Makurdi, Nigeria

Abstract

This work developed and simulated a mathematical model for a mobile wireless computational Grid architecture using networks of queuing theory. This was in order to evaluate the performance of the load-balancing three tier hierarchical configuration. The throughput and resource utilization metrics were measured and the results discussed using descriptive statistics.

Key words*: wireless computational grid, queuing theory, load-balancing, hierarchical, descriptive statitistics*

1.0 **Introduction**

Grid computing is an emerging computing idea [1], [2], that encompasses the combination of network of connected computers to form a large scale, distributed system for coordinated problem solving and resource sharing. Grid users can use the benefit of these enormous storage, computational and bandwidth resources that would otherwise only be found only on multiprocessor supercomputers.

Computational grids as a type of grid represent a transparent aggregation of many devices on a network that enhances sharing of these distributed and accessible resources [3]. They are typically categorized in the region of sharing processing power of many independent computers interconnected by a wired network. Most grid applications are focused on high performance computing mostly supporting applications of scientific research. Thus computing devices employed for such implementations usually consist of collections of similar computing assets which are rich in different resources up to the class of servers [4].

Wireless computational grids expand the capacity of grid computing to wireless computing devices. It extends the wired grid thus simplifying the exchange of information and collaboration between dissimilar wireless devices [5]. Its continuous growth is facilitated by the ever increasing developments in wireless and grid computing technologies. The number of cell phones, Personal Digital Assistants (PDA), users of laptops, palmtops, notebooks and other wireless devices is ever increasing; thus leading to more networked wireless devices, and creating an infinite amount of connected potential of untapped resources [6]. Wireless computational grid computing supports sharing of these resources and wireless devices within the organizations.

The topological change of the mobile nature of the limited computational and battery powered devices poses a great challenge in the way of frequent disruptions in connections and thus increases computation time for processing jobs for a mobile computational grid. These challenges were addressed by developing a fault tolerant coordination paradigm with self configuring and self-administering capability that allows dynamic changes for this architecture [7]. In this work, the architecture [7] is being mathematically modeled, simulated and the throughput and resource utilization performances are measured.

2.0 Materials And Methods

The architecture [7] is a distributed computer system model that consists of **m** heterogeneous computational mobile nodes (resources) providing independent exponential service shared by **n** level coordinators. Each coordinator i ($i \leq n$), contain the tasks at average Poisson arrival rate λ_i sent by the chief coordinator and dispatches tasks to the mobile nodes. The level coordinator decides which mobile modes will process the tasks and assigns tasks to those mobile nodes and finally gathers results. There is a simple communication link between each level coordinator and all of the mobile nodes independent of others with predefined capacity. The goal of the scheme is to minimize the total expected execution time (response time). Thus the problem is formulated as a non-cooperative game among level coordinators under the assumption that each level coordinator attempts to minimize the expected response time of their own tasks by assigning the designated fraction of them to each mobile resource. Each level coordinator has a queue of processes to be executed by the mobile nodes as first-come-first serve. Depending on the computational power provided by the mobile nodes, each mobile node executes processes at an average rate of μ_j cycles/second. For stability, it is assumed that the process arrival at mobile node j; σj must be less than the execution rate of mobile node j; μ_j i.e. $\sigma j \leq \mu_{j.}$

In formulating the model, the following assumptions were taken:

i. Computation power request rate at any node is less than servicing rate of the resource.

ii. Mobile resources are readily available.

iii. Scheduling of jobs, tasks or processes is First Come First Served.

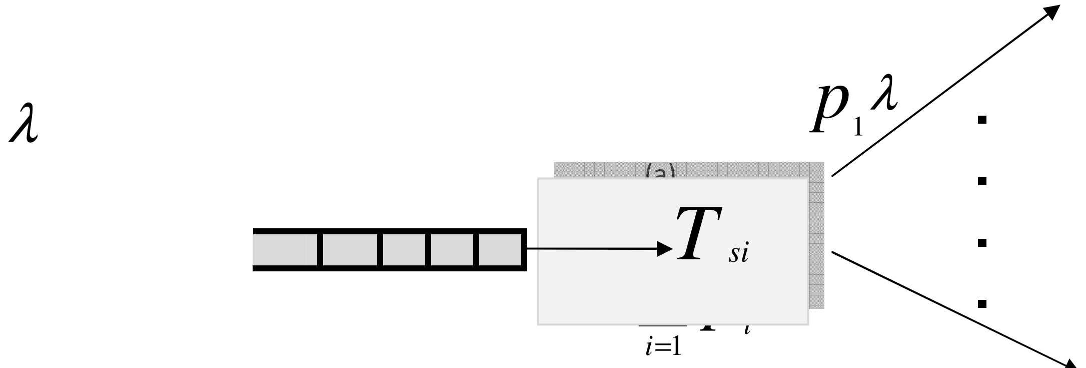
iv. Communication delay from level coordinator to mobile node is negligible.

v. The Chief Coordinator and Level Coordinators are fault tolerant.

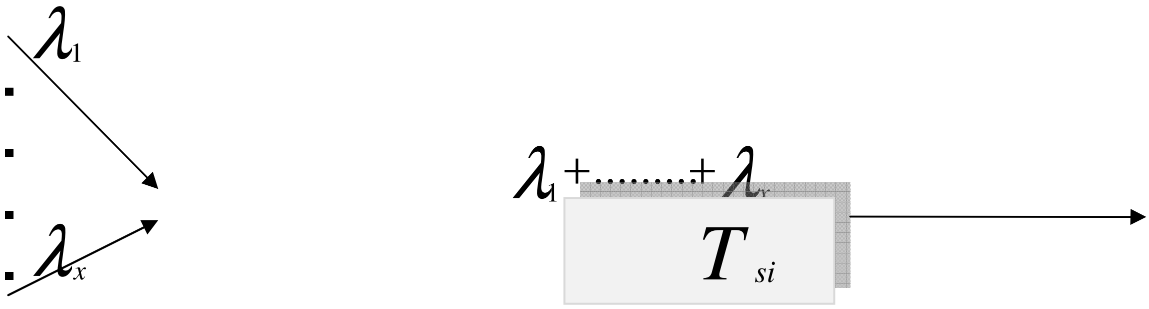
The system is modelled as a network of interconnected queues. The services rendered by the system are in two phases: job dispatch and assembly. The first phase is modelled as a network of job partitioning, where a job is broken down into tasks by the chief coordinator and each task broken down into processes for mobile nodes by the level coordinator (Figure 1a). The second phase is modelled as a network of merging of traffic where the results of processes are collated at the level coordinators and results sent as tasks results to the chief coordinator for further compilation for final submission to the owner of the job (Figure 1b).

The queuing network can be analyzed using Jackson's theorem [8] [9]. Applying this theorem, each node is treated as an independent queuing system with a Poisson input determined by the principles of partitioning, merging or Tandem queuing. For both queuing networks (job despatch and compilation) an M/M/1 model is applied (Figures 2 and 3).

For an M/M/1 queue, let
λ= arrival rate
μ= Service rate
$\rho = {}^{\lambda}/_{\mu}$, Traffic intensity



(a) Traffic Partitioning



(b) Traffic Merging

$\lambda, \lambda_1, \ldots\ldots \lambda_x$ = arrival rates of an object into the queuing system

$P_1 \ldots\ldots\ldots P_i$ = probabilities such that $P_1 + \ldots\ldots\ldots\ldots\ldots + P_i = 1$

T_{si} = Servicing mechanism

Figure 1: **Network of Queues Adapted from [8]**

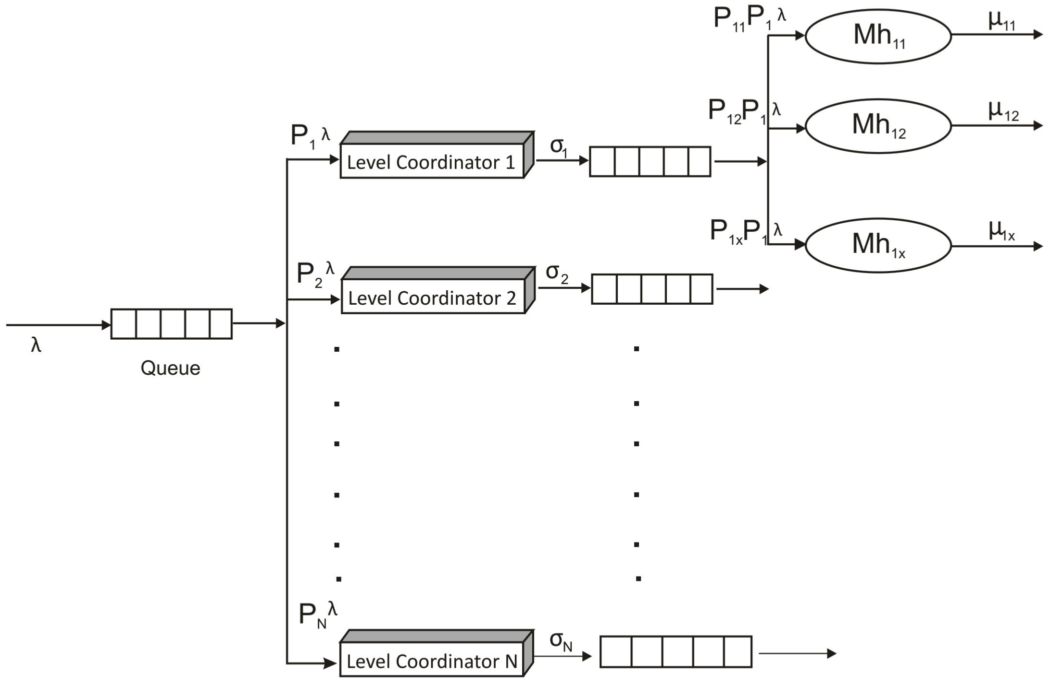
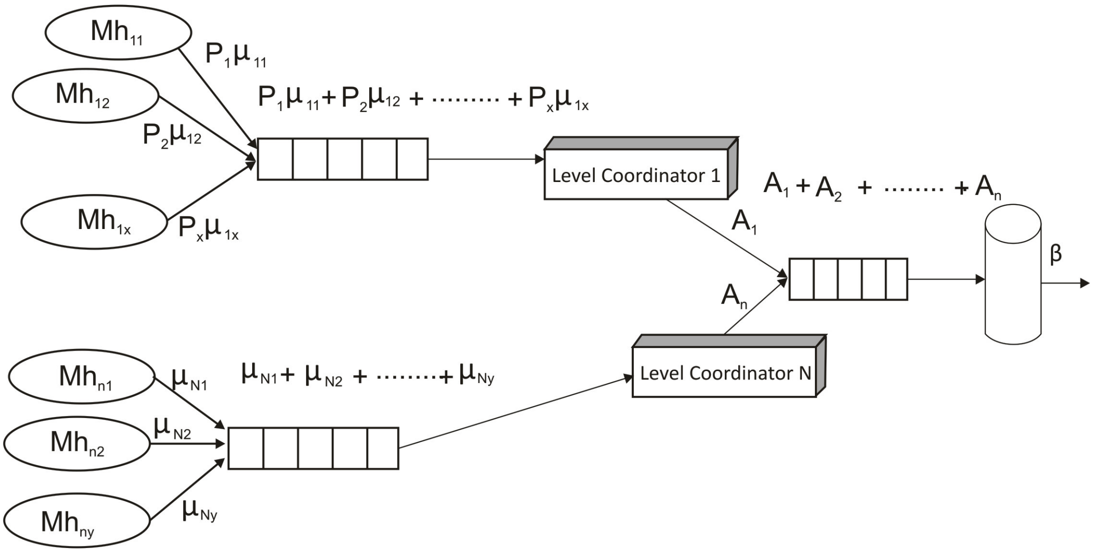


Figure 2: Job despatch mathematical model



A_n= available dispatching power of level coordinator n
β= available processing power of chief coordinator

Figure3: Job compilation mathematical model

Definition:

number of jobs in the system = $\frac{\rho}{1-\rho}$ = $\frac{\lambda}{\mu-\lambda}$ (1.1)

mean response time = $\frac{1/\mu}{1-\rho}$ = $\frac{1}{\mu-\lambda}$ (1.2)

mean waiting time =ρ $\frac{1/\mu}{1-\rho}$ = $\frac{\lambda}{\mu(\mu-\lambda)}$ (1.3)

At the traffic partitioning stage (Job despatch), the job is broken down into tasks. The tasks are broken down into processes:

The rate of departure of tasks at the chief coordinator (equivalent to request rate at each level coordinator) is $P_1\lambda$, $P_2\lambda$ … $P_N\lambda$ where P_1, P_2 … P_N are probabilities.

$$\sum_{i=1}^{N} P_i = 1$$

N= Number of level coordinators

At each level coordinator, for example level coordinator x, Request rate = $P_x\lambda$
Service rate= σ_x
Similarly, at the mobile hosts, for example mobile host **y** on level coordinator **x**; (MH_{xy})
Request rate = $P_{xy}P_x\lambda$

Service rate=μ_{xy}, P_{x1}, P_{x2}, … P_{xy} are probabilities such that

$$\sum_{i=1}^{y} P_{xi} = 1$$

σ_n= Available service rate of the level coordination

μ_{1n}= Available service rate of the mobile hosts.

3.0 Performance Metrics Specification

The three techniques for performance evaluation are analytical modelling, simulation and measurement [10]. In this study analytical modelling and simulation was used to evaluate the performance of the system. The errors were not studied because the model could not be tested on a real life grid. Thus the study was limited to correct operations only. For each successful job done, the time taken to solve the problem, latency experienced, number of resources consumed per load and the throughput was measured. This led to the following performance metrics:

i. Throughput (load per time)
ii. Resource utilization (number of resources used per successful job)

The system parameters that affected the performance of a given job are the following:

i. Speed of the chief coordinator (during despatch and collection)

ii. Speed of the level coordinators (during despatch and collection)
iii. Speed of the mobile resources (speed during despatch and collection)
iv. Migration rate of the mobile resources (due to failures from instability)

The workload parameters that affected the performance are the following:

i. Number of computing cycles requested by the job
ii. Number of available computing cycles from each level coordinator
iii. Number of available computing cycles from mobile resources

The key factors chosen for this study were the following:

i. Instability of mobile nodes due to mobility out of network range
ii. Instability of mobile nodes due to power failure

These factors have been selected based on resource availability and the interest of the grid users and resource providers.

The workload consists of a synthetic program generating the specified type of job requests in cycles per second. This program also monitored the resources consumed per level coordinator and logged the measured results. The results were analysed and plotted as graphs.

3.1.1 Throughput

This Is Defined As The Rate At Which The Requests can be serviced by the system. It is measured in cycles per second.

$$Throughput = \frac{Total\ Load}{response\ time} \quad (4)$$

Traffic partitioning phase

The processes are executed by the mobile hosts:

On the mobile hosts say mobile host 1 on level coordinator 1 with request rate P_{11} $P_1\lambda$ and service rate μ_{11},

$$\text{Load} = \frac{P_{11}\mathrm{P}_1\lambda}{(\mu_{11}-P_{11}\mathrm{P}_1\lambda)} \quad (5)$$

load = load on mobile hosts originally scheduled + load on hosts migrated to

$$\text{LDP} = \sum_{i=1}^{n}\sum_{j=1}^{k}\frac{P_{ij}\mathrm{P_i}\lambda}{(\mu_{ij}-P_{ij}\mathrm{P_i}\lambda)} + \sum_{i=1}^{n}\sum_{d=1}^{k}\frac{P_{id}\mathrm{P_i}\lambda}{(\mu_{id}-P_{id}\mathrm{P_i}\lambda)} \quad (6)$$

n= number of level coordinators
k= value of the highest mobile node on the coordinators

Traffic merging phase

The tasks are executed by the level coordinators, on level coordinator 1 request rate:

$$A_1 = \sum_{i=1}^{x} P_i\mu_{1i} \quad (7)$$

Service rate = σ_1
Load at the level coordinator:

$$\text{L} = \frac{A_1}{\sigma 1 - A_1} \quad (8)$$

Therefore overall load on the level coordinators = $\sum_{i=1}^{N}\frac{A_i}{\sigma_i - A_i}$ (9)

Request rate at the chief coordinator:

$$\text{C} = \sum_{i=1}^{N} A_i \quad (10)$$

service rate = β
Load at the chief coordinator = $\frac{C}{\beta - C}$ (11)

Total Load at the merging stage LDM =

$$\sum_{i=1}^{N}\frac{A_i}{\sigma_i - A_i} + \frac{C}{\beta - C} \quad (12)$$

Total system load: P= LDP +LDM (13)

$$Throughput = \frac{total\ system\ load}{total\ response\ time} = \frac{p}{r}, \quad (14)$$

3.1.2 Resource utilization

This is defined as the average resource used per load.

Resource utilization =

$$\frac{total\ number\ of\ resources\ used}{total\ load} \quad (15)$$

Total number of resources used = total number of mobile resources + total number of fixed wireless resources + total number of eallocated mobile resources

m $=\sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} + \sum_{i=1}^{n} L_i + \sum_{i=1}^{n}\sum_{j=1}^{k} S_{ij}$) 16)

Where , m= total number of resources
n= number of level coordinators

R_{ij}= mobile resource j initially allocated by level coordinator i
L_i= level coordinator i≤ n
S_{ij}= mobile resource j reallocated by level coordinator i.
K= the number of mobile hosts on the level coordinator with the highest number of allocated hosts
j= number of computing cycles required to complete a job

$$\textit{Resource utilization} = \frac{m}{j} \qquad (17)$$

3.2 Simulation Program Development

The simulation program was written in MATLAB 7.10.0. The workload was simulated as a number of computing cycles required to complete a job. Ten workloads were used to test the simulation program. Different instructions were also incorporated into the program to measure the throughput and resource utilization performance metrics for the workload. The processing powers, in form of computing cycles were set between1MHZ and 3GHZ for mobile nodes and between 3GHZ and 100GHZ for the level coordinators using random numbers. The simulation programs were run for different number of level coordinators **n** = (10, 25, 50, 75, 100), and mobile nodes **m**= (1000, 2500, 5000, 7500, 10000). The workload for each run was increased gradually from 10^{20} to 10^{21} computing cycles with a step of 2. Random numbers using the linear multiplicative congruential pseudo random number generator were generated to simulate all the system parameters and workload parameters. The simulation experiments were carried out eight times (arbitrary value) for the same set of jobs. The geometric means of these results were taken because a single extreme value has less of an impact on the geometric mean of a series than on the arithmetic mean [11]. Using the geometric mean makes it harder for a system to achieve a high score on the benchmark suite by achieving good performance on just one of the programs in the suite, making the system’s overall score a better indicator of its performance on most programs. The geometric mean of **n** values is calculated by multiplying the **n** values together and taking the **n***th* root of the product. The results were plotted as graphs. The performances of the model were tested by varying a set of conditions while keeping the others constant. This was done in order to analyse the sensitivity of the model [12] with respect to throughput and resource utilisation performances as the number of mobile nodes per level coordinator and number of level coordinators vary for different workloads.

4.0 Discussion of Results

The performance metrics of throughput and resource utilization were used to specify the performance of the model formulated for coordinating jobs on the wireless computational grid. The metrics were measured relative to the first and lowest workload; 10^{20} cycles, **n**=10 and **m**=1000 as the cases may be.

4.1 Throughput performance results

The purpose of this is to evaluate the performance of the model in order to determine the point at which the system could be operated for optimal throughput performance. Two workloads, 10^{20} and 8x10^{20} cycles were randomly selected to undergo this study. From figure 5 the results showed that the highest throughput was at point **n**=50, **m**=7500 and from figure 6 points **n**=50, **m**=10000 and **n**=50, **m**=7500. The worst performance was when the system was operated at

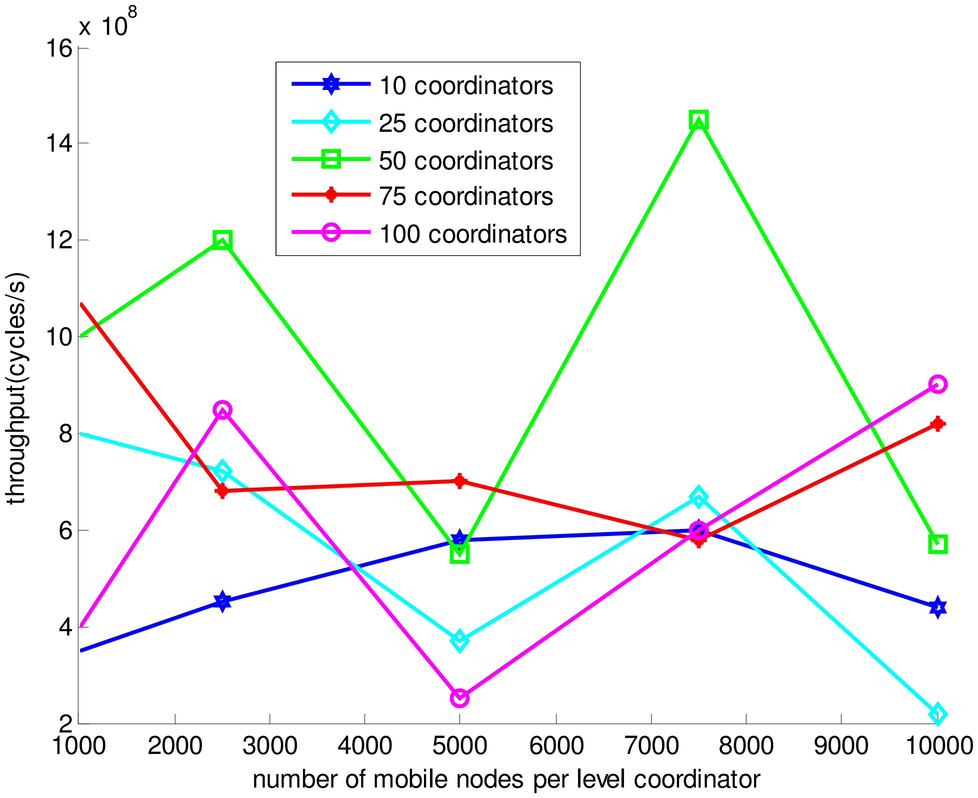


Figure 5: Graph of throughput for 10^{20} cycles

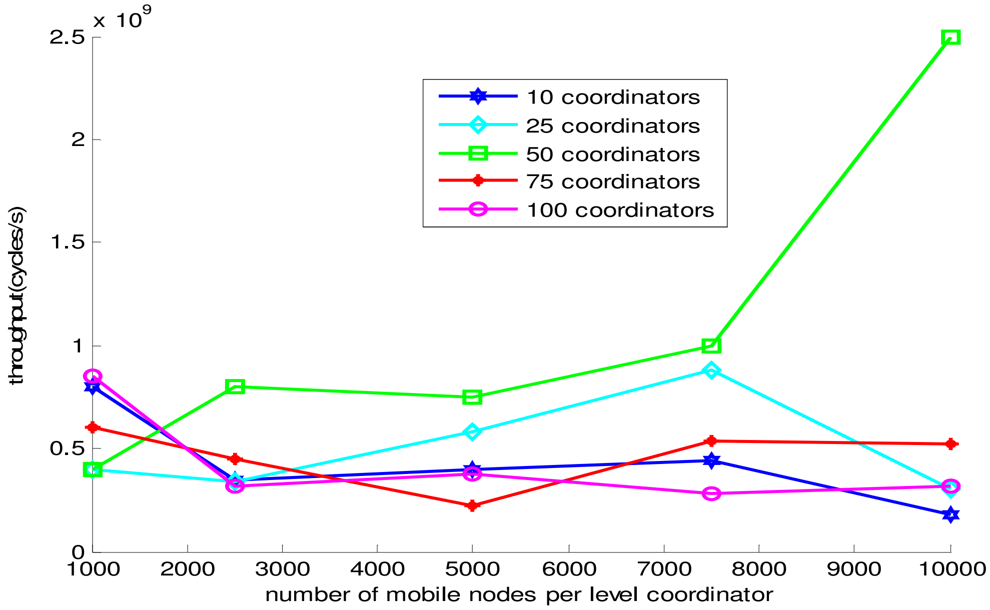


Figure 6: Graph of throughput for $8x10^{20}$ cycles workload

n=25, **m**=10000, and **n**=10, **m**=10000 resulting in throughput of $2x10^8$ and 2.5 $x10^8$ cycles/s for jobs 10^{20} and $8x10^{20}$ cycles respectively. For both workloads the highest throughput were recorded at **n**=50 (see figures 5 and 6). From figure 5, the throughput at **m**=5000 were (5.5, 3.8, 5.5 6.8, 2.8) $x10^8$ cycles/s for **n**= (10, 25, 50, 75 and 100) respectively. These correspond to 30% reduction, 0% increase, 23.64% increase, and 50% reduction in throughput for **n**=10, 25, 50,75,100. Also for $8x20^{10}$ cycles workload, the throughput at **m**=7500 reduced by 46.67%, increased by 2%, increased by 46.67%, reduced by 20%, reduced by 66.67% for **n**=10, 25, 50, 75,100 respectively. This suggested that throughput reduces as **n** exceeds 50. The peak points were at **n**=50, **m**≥7500 for workload 1x10^{20} cycles and **n**=50, **m**≥7500 for 8x10^{20} cycles. These results showed that for all jobs, the system gave the highest throughput when operated at **n** = 50 and **m** ≥ 7500.

4.2 Utilization performance results

Figures 7 and 8 showed respectively the graphs of number of resources consumed on workloads 1x10^{20} and 8x10^{20} cycles for **n**=10, 25, 50, 75 and 100. The purpose was to investigate the percentage of resources utilised in solving a specific job for values of **m**. The results showed that workload 10^{20} cycles at **n**=10 gave utilisation ratios 0.478, 0.460, 0.482, 0.480 and 0.482 for **m**= 1000, 2500, 5000, 7500 and 10000 respectively. Also for workload 8x10^{20} at **n**=25, for example the utilisation ratios 0.544, 0.477, 0.496, 0.549, 0.484 were recorded for **m**= 1000, 2500, 5000, 7500 and 10000 respectively. The results further showed that utilization factor for workload 10^{20}cycles was 0.497 and for 8x10^{20}cycles, 0.503. This suggested that more resources were consumed as the workload increases for **n** and **m**.

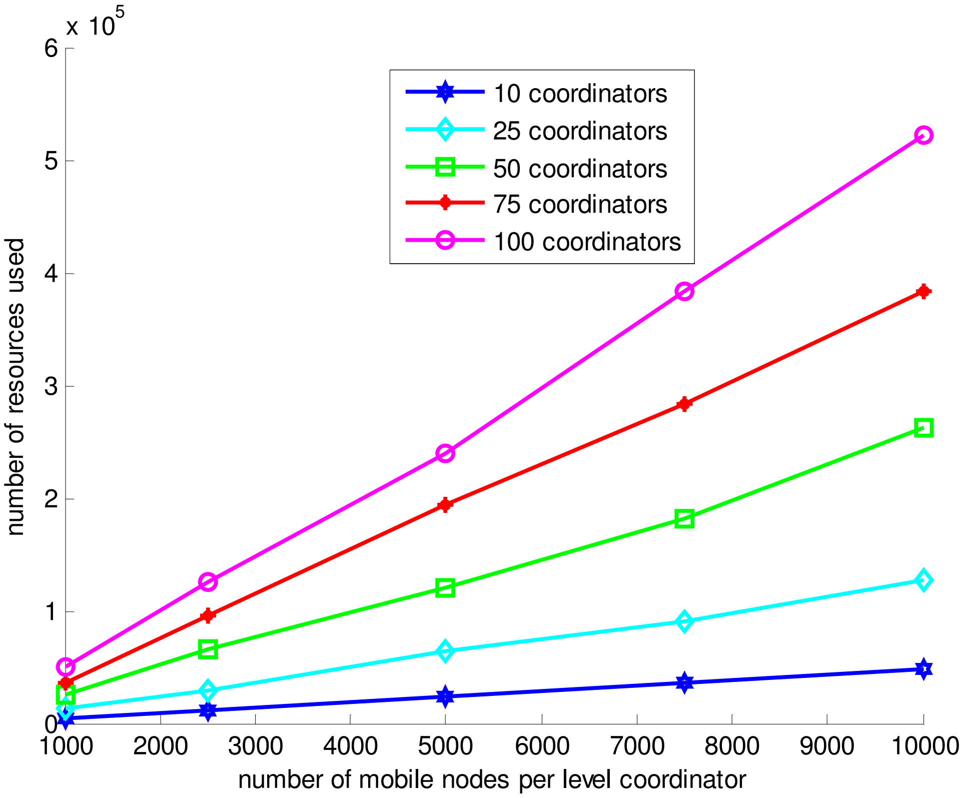


Figure 7: Graph of Utilisation for 10^{20} cycles workload

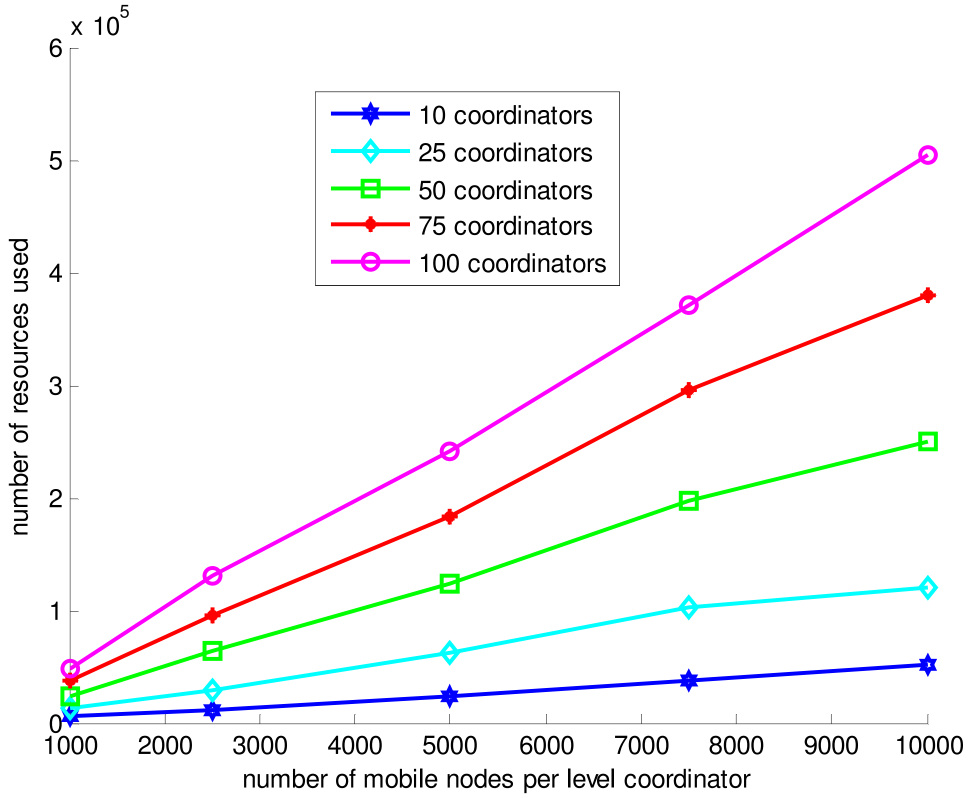


Figure 8: Graph of Utilisation for 8 x 10^{20} cycles workload

5.0 Conclusion

In this study, a mathematical model for a hierarchical load balancing paradigm facilitating the reliability of a mobile wireless computational grid was formulated and simulated. The throughput and utilization performances were evaluated. The results showed that the throughput of the simulated model began to reduce when the number of level coordinators exceeded 50 and number of mobile computational nodes exceeded 7500. Also it was discovered that more resources were consumed as the workload increases for every number of mobile nodes and level coordinators. This is in agreement with the fact that more jobs require more resources. From the above results, it could be recommended that the system should not be operated beyond 50 level coordinators and 7500 mobile computational nodes for maximum efficiency.

.

References

[1] Foster I., Kesselman C., and Tuecke S. 2001 *The anatomy of the grid, enabling scalable virtual organization* , International Journal on Supercomputer Applications, vol. 15, no. 3, 2001.

[2] Foster I., Kesselman C., Nick J. M., and Tuecke S. 2002 *The physiology of the grid, an open grid services architecture for distributed systems integration* , Open Grid Service Infrastructure WG, Global Grid Forum.

[3] Kurkovsky, S., Bhagyavati and Arris, R. 2004. Emerging Issues in Wireless Computational Grids for Mobile devices. In Proceedings of the 8th Multiconference on Systemic, Cybernetics, and Informatics, Seattle, WA

[4] McKnight, L.W. and Howison, J. 2003. Towards a Sharing Protocol for Wireless Grids. Proceedings of International Conference on Computer Communication and Control Technologies (CCCT '03), Orlando FL.

[5] Agarwal, A., Norman, D. O. and Gupta, A. 2004. Wireless Grids: Approaches, Architectures, and Technical Challenges. A working paper 4459-04, MIT Sloan School of Management.

[6] Manvi, S. and Birje, M. 2010. A Review on Wireless Grid Computing. International Journal of Computer and Electrical Engineering, **2**(3): 469-473.

[7] Adeyelu, A., Olajubu E., Aderounmu A., Ge T. 2013. A Model for Coordinating Jobs on Mobile Wireless Computational Grids. International Journal of Computer Applications, 84(13): 17-24.

[8] Stallings, W. (2000). 'Queuing Analysis', [Online[, Available: *williamsStallings.com/ StudentSupport.html* [accessed on 15th May 2014].

[9] Hiller, F.S. and Lieberman, G.J. (1995). 'Introduction to Operations Research', McGrawHill Series in Industrial Engineering and Management Science, sixth Edition: 709-713.

[10] Jain, R. (1991). 'The Art of Computer Systems Performance Analysis, Techniques for Experimental Design, Measurement, Simulation, and Modeling'. John Wiley and Sons Incorporated, USA.

[11] Carter, N. (2002). 'Computer Architecture', Schaums Outline Series, Tata McGraw-Hill, New Delhi: 279-287.

[12] Banks, J. (1998).'Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice, Engineering and Management Press, USA: 23-25.