Performance Metrics of Heterogeneous Distributed Memory Parallel Computer System Using Recursive Models

O.E. Oguike\textsuperscript{1} and M.N. Agu\textsuperscript{2}
D.U. Ebem\textsuperscript{3} and S.C. Echezona\textsuperscript{4}

\textit{Department of Computer Science}
University of Nigeria
Nsukka, Enugu State
Nigeria

e-mail: \\
\textsuperscript{1} osondu.oguike@unn.edu.ng \\
\textsuperscript{2} monica.agu@unn.edu.ng \\
\textsuperscript{3} deborah.ebem@unn.edu.ng \\
\textsuperscript{4} stevenson.echezona@unn.edu.ng

H.O.D. Longe\textsuperscript{5}
O. Abass\textsuperscript{6}

\textit{Department of Computer Sciences}
University of Lagos
Lagos, Nigeria

e-mail: \\
\textsuperscript{5} hodlonge@yahoo.co.uk \\
\textsuperscript{6} olaabass@unilag.edu.ng

Abstract—In a heterogeneous distributed memory parallel computer system, each of the processors has its own memory. Describing the system using queuing network, each of the processors has its own ready queue. Since the system has many I/O queues and I/O processors, therefore, the various processors with their various ready queues and the various I/O processors and their various I/O queues form the queuing network of such a system. Using a round robin scheduling algorithm for the processor queues, we aim at evaluating the performance of the system from the time a job arrives in the system to the time job departs from the system. This paper uses recursive models to evaluate and assess the accuracy of the performance of the whole queuing network of a heterogeneous distributed memory parallel computer system.

Keywords- heterogeneous parallel computer; distributed memory parallel computer; queuing network; performance metrics.

I. INTRODUCTION

The queuing network of a heterogeneous distributed memory parallel computer consists of parallel processors, parallel processor queues, I/O processors and I/O queues [21]. We assume a round robin scheduling algorithm for the various parallel processor queues. Suppose there are \( n \) different parallel processors per queuing system and \( k \) different I/O queuing systems. A queuing system in this context is defined as a processor with its own queue. We also assume that the various queues are finite \([1, 4,21]\) i.e. there is a limit to the number of jobs that can be admitted into the queues, we assume that \( X_1, X_2, \ldots, X_n, X_{n+1}, X_{n+2}, \ldots, X_{n+k} \) are the maximum number of jobs that can be admitted into the respective queues. We assume that jobs arrive in the various queues according to Poisson distribution, and they are serviced according to exponential distribution \([5, 6]\). Figure 1 illustrates a model of the queuing network of a heterogeneous distributed memory parallel computer system.

Figure 1. Queuing Network of a heterogeneous distributed memory parallel computer system.
II. STATEMENT OF THE PROBLEM

A good way of evaluating the performance of the system is to consider the whole queuing network [15, 19, 21] rather than considering part of the queuing network [18]. The reason for this is because it will give a true picture of the performance of the system from the time a job enters into the system to the time the job leaves the system. Therefore, it is necessary to consider all the queuing systems of the queuing network of heterogeneous parallel computer system. Furthermore, using recursive models to evaluate the performance of the whole queuing network of the parallel computer system will solve the problem of inefficient utilization of the resources of the computer system used to simulate the recursive models, especially when a high performance computer is used to do the simulation [18]. Though recursive models can be argued to have exponential time complexity [9], however, the time complexity of recursive models will be minimized when the models are simulated on high performance computer with high memory capacity, like modern parallel computer systems.

III. LITERATURE REVIEW

Using queuing approach to model the performance of computer system has been discussed extensively in literature. However, this has been done in different ways and for different models of computer systems. In [14], a recursive computation approach was used to solve the steady state equations, thereby modeling the various performance metrics of a multi-terminal system that is subject to breakdown. Furthermore, [24] used a rigorous approach to model the performance of heterogeneous parallel computer system without introducing any constraint on the kind of interconnection between the heterogeneous nodes. Furthermore, in [24], systems with the same interconnection speed were considered when modeling the performance of heterogeneous parallel computer system. The opinion of authors in [25] was on the use of alternative ways of measuring the performance of heterogeneous parallel computer system, by modeling linear speed and linear efficiency using simulation-modeling techniques. In [26], the author discussed that Little’s formulae could be universally applicable, if properly interpreted to take account of state-varying entrance rates, batch arrivals, and multiple customer classes. It was confirmed in [27] that Little's formula could be applied to very general queuing systems (not just M/M/1), even whole networks! The authors in [28] considered a new performance metric, variation of the computing power as a unique performance metric that is ideal for a heterogeneous network of workstations, though a different was used to do this.

Though analytic models have been used to evaluate the performance of the whole queuing network of a heterogeneous distributed memory parallel computer system, as reported in [19], the models developed in [19] are different from the models developed in this paper because this paper uses recursive models to evaluate the performance of the whole queuing network of a heterogeneous distributed memory parallel computer system. Furthermore, though recursive models have been used in [18] to evaluate the performance of a parallel computer system, however the models developed in [18] are different from the models developed in this paper. This is because the model developed in [18] applied to a shared memory parallel computer system, rather than a distributed memory parallel computer system.

IV. METHODOLOGY

We aim at using recursive models to evaluate and assess the accuracy of the performance metrics of the whole queuing network of a heterogeneous distributed memory parallel computer system. We have achieved this by using computer queuing approach [10, 21,22], with finite queues. The parallel processors depict parallel servers, and we have used all the related laws that are based on queuing systems, like Little’s formulae as stated in [7]. We have used statistical method of probability density function to develop the probability density function of the number of processes that join the queuing system, and some probability theory laws [15]. We have used novel method of deriving recursive models that determines the xth terms and important functions of a given mathematical sequence to develop the recursive models [18,21,22]. We have used the recursive models to develop the various performance metrics of a heterogeneous distributed memory parallel computer system [22]. We have used the recursive models to assess the accuracy of the performance metrics of a heterogeneous distributed memory parallel computer system. We have simulated the models using Java programming language and we have used statistical regression/trend line analysis to analyze the results of the simulation [11, 21,22].

V. DEVELOPING THE MODELS

As a result of the use of the above methodologies, the following models have been developed for one queuing system [18].

A. Models Based on a Queuing System

The following models have been developed in [18] based on one queuing system

- Probability Density Function of the Number of Processes in a Queuing System.

Let X denote the maximum number of processes that can be in the i-th finite queuing system at any time [12, 13]. Suppose the arrival rate, \( \lambda_x \) when x processes are in the i-th queuing system of the queuing network be described as:

\[
\lambda_x = \begin{cases} 
\lambda_i, & x = 0,1,2,3,...,X_i - 1 \\
0, & \text{otherwise}
\end{cases}
\]  

Since the various processors are heterogeneous, therefore, it implies that the departure rate will vary, which can be described as:
\[ \mu_{x,i} = \begin{cases} \mu_i, & x = 1, 2, 3, 4, \ldots, X_i \\ 0, & \text{otherwise} \end{cases} \quad (2) \]

Using the steady state probability as stated in [7, 16] the probability that \( x \) processes will be in the \( i \)th queuing system is

\[ P_{x,i} = \begin{cases} \rho_i^x P_{0i}, & x \leq X_i \\ 0, & \text{otherwise} \end{cases} \quad (3) \]

\( P_{0i} \) can be obtained as we sum all the probabilities for the \( i \)th queuing system and equate it to 1. This implies that:

\[ \sum_{x=0}^{X_i} P_{x,i} = 1. \quad (4) \]

From equations (3) and (4), it implies that:

\[ P_{0i} + \rho_i P_{0i} + \rho_i^2 P_{0i} + \rho_i^3 P_{0i} + \cdots + \rho_i^{X_i} P_{0i} = 1. \quad (5) \]

Factorizing equation (5), it implies that

\[ P_{0i} (1 + \rho_i + \rho_i^2 + \rho_i^3 + \cdots + \rho_i^{X_i}) = 1. \quad (6) \]

Using mathematical method of deriving recursive model for the \( x \)th terms of the sequence, \( \rho_i^0, \rho_i^1, \rho_i^2, \rho_i^3, \ldots \), \( \rho_i^{X_i} \), the \( x \)th terms of the sequence is given as

\[ \text{Term1}(X_i, \rho_i) = \begin{cases} 1, & X_i = 0 \\ \rho_i \times \text{Term1}(X_{i-1}, \rho_i), & X_i \neq 0 \end{cases} \quad (7) \]

Therefore, the sum of the first \( x \)th terms of the sequence is given as

\[ \text{Sum1}(X_i, \rho_i) = \begin{cases} 1, & X_i = 0 \\ \text{Term1}(X_i, \rho_i) + \text{Sum1}(X_{i-1}, \rho_i), & X_i \neq 0 \end{cases} \quad (8) \]

It has been derived in [18] that the probability density function of \( x \) processes in the \( i \)th queuing system is:

\[ E(x_i) = \sum_{x=0}^{X_i} x_i P_{x,i}. \quad (10) \]

It has been derived in [18] that the above equation can be simplified as:

\[ E(x_i) = \frac{1}{\text{Sum1}(X_i, \rho_i)} \left( \rho_i + 2\rho_i^2 + 3\rho_i^3 + 4\rho_i^4 + \cdots + X_i\rho_i^{X_i-1} \right) \quad (11) \]

Factorizing equation (11), we obtain

\[ E(x_i) = \frac{1}{\text{Sum}(X_i, \rho_i)} \rho_i^{[1 + 2\rho_i + 3\rho_i^2 + 4\rho_i^3 + \cdots + X_i\rho_i^{X_i-1}]} \quad (12) \]

A recursive model can be used to determine the convergence of this series, \( 1 + 2\rho_i + 3\rho_i^2 + 4\rho_i^3 + \cdots + X_i\rho_i^{X_i-1} \) as seen in equation (12). In order to determine the recursive model, we first determine the \( x \)th terms of the sequence, \( 1, 2\rho_i, 3\rho_i^2, 4\rho_i^3, \ldots, \rho_i^{X_i}, \rho_i^{X_i-1} \), which can be determined by considering the sequence as two sequences. The first sequence is: \( 1, 2, 3, 4, \ldots, X_i \), while the second sequence is: \( \rho_i, \rho_i^1, \rho_i^2, \rho_i^3, \ldots, \rho_i^{X_i-1} \). Therefore the recursive model for the \( x \)th terms of the first sequence is given as follows:

\[ \text{Term2}(X_i) = \begin{cases} 1 + \text{term2}(X_{i-1}), & X_i \neq 1 \\ 1, & X_i = 1 \end{cases} \quad (13) \]

The recursive model for the \( x \)th terms of the second sequence has been developed earlier in equation (7). Therefore, combining the two models in equations (7) and (13), the series, \( 1 + 2\rho_i + 3\rho_i^2 + 4\rho_i^3 + \cdots + X_i\rho_i^{X_i-1} \) converges to this recursive model, \( \text{Sum2}(X_i, \rho_i) \), which is given as:

\[ \text{Term2}(X_i) \times \text{term1}(X_{i-1}, \rho_i) + \text{Sum2}(X_{i-1}, \rho_i), X_i \neq 1 \]

Using equations (8) and (14), it has been derived in [18] that the above equation (12) can be reduced to:

\[ L_{si} = \frac{\rho_i}{\text{Sum1}(X_i, \rho_i)} \] \quad (15)

- Effective Arrival Rate, \( \lambda_{eff} \) for a Queuing System
The effective arrival rate, $\lambda_{eff}$, models the rate of arrival of processes that actually join the $i$th queue. The reason for this is because the queue of the $i$th queuing system is finite; this means that there is a limit to the number of processes that the $i$th queue can admit. Therefore, a process that arrives to join the queue can actually join the queue or be lost on arrival. It has been derived in [18] that the effective arrival rate for the $i$th queuing system is given as:

$$\lambda_{eff} = \lambda_i (1 - P_{iX})$$  \hspace{1cm} (16)

- **Average Waiting Time in a Queuing System.**
  Using Little’s formulae as stated in [7], the average length of the $i$th queuing system is directly proportional to the average waiting time in the $i$th queuing system. This can be expressed as follows,

$$L_i = \alpha W_i$$  \hspace{1cm} (17)

$L_i$ denotes the average length of the $i$th queuing system and $W_i$ denotes the average waiting time in the $i$th queuing system. It has been derived in [18] that the average waiting time in the $i$th queuing system is given as:

$$W_i = \frac{\rho_i}{\lambda_{eff}i} \cdot \text{Sum}_1(X_i, \rho_i)$$  \hspace{1cm} (18)

$\lambda_{eff}$ is the constant of proportionality.

- **Average Waiting Time in a Queue.**
  The average waiting time in the $i$th queuing system, which will be denoted as $W_i$, has been defined in [17], as average time that a process waits in the $i$th queue together with the average service time. This has been derived in [18] as:

$$W_i = \frac{\rho_i}{\lambda_{eff}i} \cdot \text{Sum}_2(X_i, \rho_i) - \frac{1}{\mu_i}$$  \hspace{1cm} (19)

- **Average Number of Processes in a Queue.**
  Using another version of Little’s formulae as stated in [7], the average length of the $i$th queue is directly proportional to the average waiting time in the $i$th queue. This can be expressed as

$$L_{qi} = \alpha W_{qi}$$  \hspace{1cm} (20)

$L_{qi}$ denotes the average length of the $i$th queue and $W_{qi}$ denotes the average waiting time in the $i$th queue. This performance metric has been derived in [18] as:

$$L_{qi} = \frac{\rho_i \cdot \text{Sum}_2(X_i, \rho_i)}{\text{Sum}_1(X_i, \rho_i)} - \frac{\lambda_{eff}}{\mu_i}$$  \hspace{1cm} (21)

**B. Models Based on the Whole Queuing Network.**

Having developed the models for the performance metrics of one queuing system, we proceed to develop the performance metrics of the whole queuing network by considering all the parallel processor queues and the various I/O queues. We define $\delta_i$ as the probability that a process will join the $i$th queue after each cpu burst, and $\delta_0$ as the probability that the execution of a process has been completed. Arrival of processes into the various parallel processor queues can come from the outside world or from the various I/O queues or from the particular parallel processor, at the expiration of the time quantum for that process. Let $\lambda_0$ be the total arrival rate into the various queues, and let $\lambda_i$ be the rate of arrival of processes into the $i$th processor queue, and $\lambda$ be the rate of arrival of processes from the outside world. Under the steady state, the arrival rate into the $i$th queue is the same as the arrival rate from the $i$th queue to the queuing network [15]. Therefore, it means from the above discussion that

$$\lambda_0 = \lambda + \sum_{i=1}^{n+k} \lambda_i$$  \hspace{1cm} (22)

Simplifying equation (22), we obtain

$$\lambda = \lambda_0 - \sum_{i=1}^{n+k} \lambda_i$$  \hspace{1cm} (23)

From discussion above and from definition, $\lambda_i = \lambda_0 \delta_i$.

Using equation (24) in equation (23), we obtain

$$\lambda = \lambda_0 - \sum_{i=1}^{n+k} \lambda_0 \delta_i$$  \hspace{1cm} (24)

Factorizing equation (25), we obtain the following:

$$\lambda = \lambda_0 \left(1 - \sum_{i=1}^{n+k} \delta_i\right)$$  \hspace{1cm} (26)

From probability theory, sum of all probabilities is 1. It implies that

$$\sum_{i=0}^{n+k} \delta_i = 1$$  \hspace{1cm} (27)

Simplifying equation (27), and using it in equation (26), we obtain the following:

$$\lambda = \lambda_0 \delta_0$$  \hspace{1cm} (28)

Simplifying equation (28), we obtain

$$\lambda_0 = \frac{\lambda}{\delta_0}$$  \hspace{1cm} (29)

Using equation (29) in equation (24), we obtain the following:

$$\lambda_i = \frac{\lambda \delta_i}{\delta_0}$$  \hspace{1cm} (30)

Therefore for the queuing network, arrival of processes in the $i$th queue has Poisson distribution with the following arrival rate

$$\lambda_i = \begin{cases} \frac{\lambda}{\delta_0}, & i = 0 \\ \frac{\lambda \delta_i}{\delta_0}, & i \neq 0 \end{cases}$$  \hspace{1cm} (31)
Similarly, the execution time in each of the queues has exponential distribution with parameter \( \mu_i \). Therefore \( \rho_i \), can be defined as:

\[
\rho_i = \begin{cases} 
\frac{\lambda}{\delta_0 \mu_i}, & i = 0 \\
\frac{\lambda \delta_i}{\delta_0 \mu_i}, & i = 1, 2, 3, \ldots, n+k
\end{cases}
\] (32)

- **Probability Density Function of the Number of Processes in the Whole Queuing Network.**
  
  This measures the probability that \( x_1, x_2, x_3, \ldots, x_n \), \( x_{n+1}, x_{n+2}, x_{n+3}, \ldots, x_{n+k} \) are in each of the respective queues. 1, 2, 3, \ldots, \( n+1, n+2, n+3, \ldots, n+k \). Since the random variables are independent random variables, therefore, the various probability density functions of the queues are independent. Since the independent probability density function of \( x_i \) processes is in the \( i \)th queuing system has been defined in (9), it follows from the product law of probability theory [15] that the above probability density function can be defined as follows:

\[
P_{x_i} = \prod_{i=1}^{n+k} \frac{\rho_i^{x_i} \sum_{1}^{x_i} (X_i, \rho_i)}{1}, \quad x_i \leq X_i
\] (33)

- **Number of Processes in the Whole Queuing Network.**

  Two performance metrics can be defined based on the number of processes in the whole queuing network, they are: sum of the average number of processes in all the queuing systems of the queuing network and average number of processes in the queuing systems of the queuing network. Suppose \( x_i \) is a random variable that denotes the number of processes in the \( i \)th queuing system. We can define another random variable, \( Y \) as:

\[
Y = \sum_{i=1}^{n+k} x_i
\] (34)

The random variable, \( Y \), will denote the total number of processes in all the queuing networks. Taking the statistical expectation of \( Y \), we obtain:

\[
E(Y) = E \left( \sum_{i=1}^{n+k} x_i \right)
\] (35)

Using the probability theory law of expectation, equation (35) becomes:

\[
L_s = E(Y) = \sum_{i=1}^{n+k} E(x_i)
\] (36)

Using the model developed in equation (15), which is the average number of processes in the \( i \)th queuing system, the performance metric under consideration can be modeled as follows:

\[
L_s = \sum_{i=1}^{n+k} L_{s_i}
\] (37)

The performance metric in equation (37) can be called the sum of the average number of processes in the whole queuing network. Furthermore, suppose we define another random variable, \( A \), which will denote the average number of processes in the various queuing systems of the whole queuing network. Therefore, the random variable, \( A \), can be written as:

\[
A = \sum_{i=1}^{n+k} x_i
\] (38)

Taking the statistical expectation of \( A \), we obtain:

\[
E(A) = E \left( \sum_{i=1}^{n+k} x_i \right)
\] (39)

Using probability theory law, equation (39) becomes:

\[
E(A) = \left( \frac{1}{n+k} \sum_{i=1}^{n+k} E(x_i) \right)
\] (40)

Using equation (15) in equation (40), we obtain:

\[
E(A) = \left( \frac{1}{n+k} \sum_{i=1}^{n+k} L_{s_i} \right)
\] (41)

Equation (40) models the average number of processes in queuing systems of the queuing network.

- **Waiting Time in the Whole Queuing Network.**

  This performance metric models the waiting time of one process in the entire queuing network. Following the same argument that was used to develop the performance metric that models the average waiting time in one of the queuing systems in equation (18) and extending Little’s formulae to consider all the queuing systems of the queuing network, this performance metric can be modeled as follows:

\[
W_s = \frac{L_s}{\sum_{i=1}^{n+k} \lambda_{e_i} \frac{X_i}{n+k}}
\] (42)

\[\left( \frac{\sum_{i=1}^{n+k} \lambda_{e_i}}{n+k} \right)\]

is the chosen constant of proportionality

- **Waiting Time in the Whole Queue.**

  This performance metric models the waiting time in all the queues of the queuing network. Following the same
argument that was used to derive equation (19), which is the performance metric that models the average waiting time in one queue, and using equation (42), this performance metric can be modeled as follows:

$$W_q = W_s - \sum_{i=1}^{n+k} \frac{1}{\mu_i}$$  \hspace{1cm} (43)

- Number of Processes in the Whole Queue.

This performance metric models the number of processes in all the queues of the queuing network. Following the same argument that was used to derive the model in equation (21), which models the average number of processes in one of the queues of the queuing network, this performance metric can be modeled as follows:

$$L_q = W_q \times \frac{\sum_{i=1}^{n+k} \lambda_i \cdot \rho_i}{n + k}$$  \hspace{1cm} (44)

VI. ESTIMATING THE STANDARD ERROR OF THE MODELS

One of the ways of assessing the accuracy of probabilistic, statistical models is to estimate the standard error of the models. Suppose $x_i$ is the random variable that denote the number of processes in the $i$th queuing system of the queuing network, we can define the random variable $Y$, to represent the total number of processes in all the queues of the queuing network as follows:

$$Y = \sum_{i=1}^{n+k} x_i$$  \hspace{1cm} (45)

Statistically, the standard error of the random variable $Y$ in equation (45) can be defined as:

$$SE(Y) = \sqrt{\text{VAR}\left(\sum_{i=1}^{n+k} x_i\right)}$$  \hspace{1cm} (46)

Using probability theory law as stated in [20], equation (46) could be reduced to:

$$SE(Y) = \sqrt{\sum_{i=1}^{n+k} \text{VAR}(x_i)}$$  \hspace{1cm} (47)

Using the definition of variance of $x_i$ as stated in [20], we obtain:

$$\text{VAR}(x_i) = \text{E}(x_i^2) - \left(\text{E}(x_i]\right)^2$$  \hspace{1cm} (48)

$E(x_i^2)$ has been developed in [22] as

$$E(x_i^2) = \frac{1}{\text{Sum1}_{i}(X_i, \rho_i)} \cdot \rho_i \cdot \text{Sum3}_{i}(X_i, \rho_i)$$  \hspace{1cm} (49)

$\text{Sum3}_{i}(X_i, \rho_i)$ has also been developed in [22] as:

$$\text{Sum3}_{i}(X_i, \rho_i) = \frac{1}{\text{Sum1}_{i}(X_i, \rho_i)} \cdot \rho_i \cdot \text{Sum3}_{i}(X_i, \rho_i)$$

Similarly, $\text{Term3}_{i}(X_i)$ has been developed in [22] as

$$\text{Term3}_{i}(X_i) = \begin{cases} 1, X_i = 1 \\ (2X_i-1) + \text{Term3}_{i}(X_i-1), X_i \neq 1 \end{cases}$$  \hspace{1cm} (51)

Therefore, using equations (49) and (15) in equation (48), we obtain $\text{VAR}(x_i)$, which is given below in equation (52) as:

$$\text{VAR}(x_i) = \left(\frac{\rho_i \cdot \text{Sum1}_{i}(X_i, \rho_i)}{\text{Sum1}_{i}(X_i, \rho_i)} - \left(\frac{\rho_i \cdot \text{Sum1}_{i}(X_i, \rho_i)}{\text{Sum1}_{i}(X_i, \rho_i)}\right)^2\right)$$  \hspace{1cm} (52)

Therefore, using equation (52) in equation (46), the standard error of the random variable $Y$ can be written as:

$$\sqrt{\sum_{i=1}^{n+k} \left(\frac{\rho_i \cdot \text{Sum1}_{i}(X_i, \rho_i)}{\text{Sum1}_{i}(X_i, \rho_i)} - \left(\frac{\rho_i \cdot \text{Sum1}_{i}(X_i, \rho_i)}{\text{Sum1}_{i}(X_i, \rho_i)}\right)^2\right)}$$  \hspace{1cm} (53)

VII. ANALYSIS OF THE PERFORMANCE METRICS

The analysis of the performance metrics will help us to determine how parameter changes will affect the performance metrics [10]. The models were simulated on the computer using Java programming language, and the results of the simulated models were analyzed using statistical trend line analysis. Suppose the arrival rate from the outside world into the network is 18 and the probability that a process leaves the system after a processor burst is 0.2, figure 2a shows the results of the simulated model when we vary the total number of processes in the network in order to determine the waiting time in the system or response time. Similarly, for the same arrival rate from the outside world into the network and suppose the probability that a process leaves the network after a processor execution is 0.6, figure 2b shows the result of the simulated model when we vary the total number of processes in the network in order to determine the waiting time or response time in the system. When you compare the trend lines in figure 2a and 2b, you discover that for a specific number of processes in the queuing network, the waiting time or response time in the system in figure 2a is higher than the waiting time or response time in figure 2b.

This is because the probability of a job leaving the queuing system is smaller in figure 2a than in figure 2b. The
interpretation of the above result is that small processes will spend a smaller time in the queuing network than large processes. Figure 2c shows that as the execution rate of all the processors increases, the response time or waiting time of the whole system decreases. The interpretation of the above result is that if most of the processors in the queuing network are fast processors, it will imply low system waiting time or system response time; on the other hand, if most of the processors in the queuing network are slow, it will imply high system waiting or response time. Similarly, in figure 2d, for non compute intensive applications of each of the queues of the queuing network, i.e. $\rho_i$ is less than 1, the standard error of the queuing network decreases to a minimum value as the overall departure rates of each of the processors increase. The interpretation of this result is that as heterogeneous fast processors are used in the parallel computer for non-compute intensive applications, the standard error decreases to a minimum value.

VIII. SUMMARY AND CONCLUSION

This paper has been able to use recursive models to evaluate the performance of a heterogeneous distributed memory parallel computer system. It has been able to establish when the standard error of the model assumes a minimum value. This was accomplished by considering the whole queuing network of a heterogeneous distributed memory parallel computer system. The models were simulated using Java programming language and the results of the simulation were analyzed using regression/trend line analysis. We conclude by saying that efficient recursive models have been used to evaluate and assess the accuracy of the performance metrics of a heterogeneous distributed memory parallel computer system.

ACKNOWLEDGMENT

We thank the Academic Board and the Postgraduate Board of the Department of Computer Science, University of Nigeria, Nsukka for giving us the opportunity to lecture the following courses, Queuing Theory, Operating Systems, Algorithms, Structured Programming and Computer Performance Evaluation. The lectures that we delivered on the above courses at both undergraduate and postgraduate levels contributed to the development of this paper. We also thank the Postgraduate Board of the Department of Computer Sciences, University of Lagos, Nigeria for giving us the opportunity to carry out research in the area of parallel computing.

REFERENCES

[6] Ivan Stojmenovic; Recursive Algorithms in Computer Science Courses ; Fibonacci Numbers and Binomial Coefficients; IEEE Transactions on Education; Vol. 48, No. 3. 
[21] E. Post, H.E. Goosen; Evaluating the Parallel Performance of a Heterogeneous System
[24] Xiaodong Zhang; Yong Yan; Modeling and Characterizing Parallel Computing Performance on Heterogeneous Network of workstations; Proceedings of the 7th IEEE Symposium on Parallel and Distributed Processing (SPDP '95) 1063-6374/95 $10.00 © 1995 IEEE.

IJSST, Vol. 12, No.4 24 ISSN: 1473-804x online, 1473-8031 print
Figure 2. Results of the simulated models