International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-2 Issue-6, January 2013

Modeling Variation of Waiting Time of Distributed Memory Heterogeneous Parallel Computer System using Recursive Models

Oguike, O.E., Agu, M.N., Echezona, S.C.

Abstract— In a heterogeneous parallel computer system, the computational power of each of the processors differs from one another. Furthermore, with distributed memory, the capacity of the memory, which is distributed to each of the processors, differs from one another. Using queuing system to describe a distributed memory heterogeneous parallel computer system, each of the heterogeneous processors will have its own heterogeneous queue. The variation of waiting time of heterogeneous parallel computer system with distributed memory needs to be modeled because it will help designers of parallel computer system to determine the extent of variation of the waiting time. It will also help users to know when to realize minimum variation of the waiting time. This paper models the variation of the waiting time of distributed memory heterogeneous parallel computer system using recursive models. It also uses the statistical method of Z-Transform to verify and validate the recursive model.

Keywords— distributed memory, heterogeneous parallel computer, parallel computer system, queuing network, recursive models, variation, waiting time, Z-Transform.

I. INTRODUCTION

A heterogeneous parallel computer system is one in which the computational power of each of the processors differs from one another. With distributed memory, it means that each of the heterogeneous processors has its own memory. Describing the system using queuing network, each of the processors has its own queue. With a round robin scheduling algorithm, processes can be scheduled to the various parallel processors, whenever a process needs to perform an I/O operation, it joins the appropriate I/O queue. Therefore, the queuing network of a heterogeneous parallel computer system consists of parallel processors, parallel processor queues, I/O processors and I/O queues. Suppose there are n different parallel processor queuing systems and k different I/O queuing systems. A queuing system in this context is defined as a processor, together with its own queue.

We assume that the various queues are finite [1], [2], [3], [4] i.e. there is a limit to the number of jobs that can be admitted into the queues, and negligible communication overhead. Suppose X_1, X_2, X_3, …, X_n , X_{n+1}, X_{n+2}, X_{n+3}, …, X_{n+K} are the maximum number of processes that can be admitted into the respective queues. We assume that processes arrive at 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 parallel computer system with distributed memory.

There are different performance metrics of a parallel computer system that can be modeled, however, for distributed memory heterogeneous parallel computer system, variation of waiting time is an important performance metric that needs to be modeled. This is because the various computational resources and processes are heterogeneous, therefore there is need to measure the extent of variation between the heterogeneous computational resources and processes.

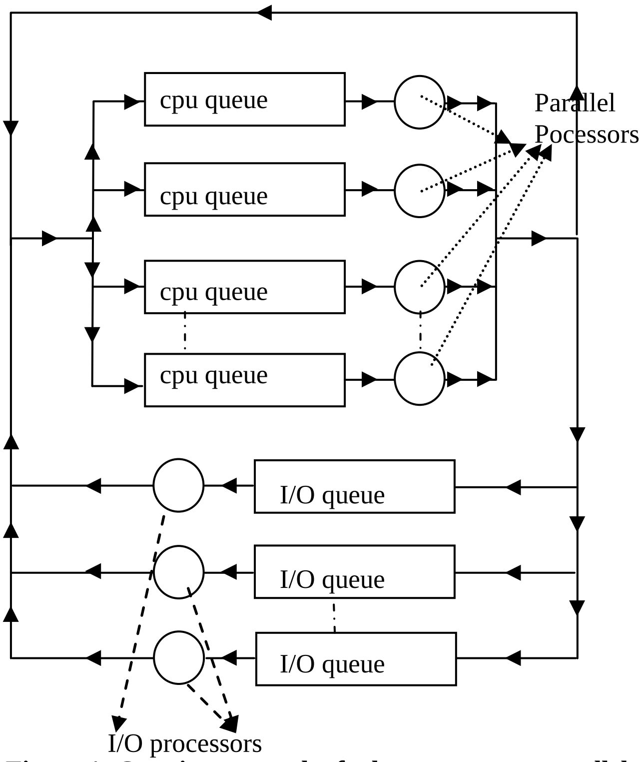

Figure 1: Queuing network of a heterogeneous parallel computer system with distributed memory

II. LITERATURE REVIEW AND LIMITATION OF CURRENT TECHNIQUE

Queuing approach has been used extensively in the literature to model the performance of computer systems. However, this has been done in different ways and for different models of computer systems. In [20], the authors used a recursive computation approach to solve the steady state equations, thereby leading to the modeling of the various performance metrics of a multi-terminal system that is subject to breakdown. Furthermore, the author in [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],

Manuscript received on January, 2013.
Oguike, O. E., Department of Computer Science, University of Nigeria, Nsukka, Enugu State, Nigeria.
Agu, M.N., Department of Computer Science, University of Nigeria, Nsukka, Enugu State, Nigeria.
Echezona, S.C.., Department of Computer Science, University of Nigeria, Nsukka, Enugu State, Nigeria.

systems with the same interconnection speed were considered when modeling the performance of heterogeneous parallel computer system. The authors in [25] looked at 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 showed 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. In [27], the author confirmed 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 an approach different from queuing approach was used to do this. In [29], analytic models were used to model the performance of computer intensive applications of parallel computers, while [30] used recursive models only to evaluate the performance of compute intensive application of a parallel computer system. In [31], recursive models were used to evaluate various performance metrics of heterogeneous parallel computer system with distributed memory; however, variation was not part of the performance metric modeled. In [33], the authors used recursive model to model the variation of the average number of processes in the system, though the developed recursive models were not validated.

Though analytic queuing method has been used in literature [29], [32] to model the performance metrics of various computer queuing models, however, one limitation of the analytic method is its inability to efficiently determine the exact convergence of some mathematical series that are used in modeling variation of waiting time of distributed memory, heterogeneous parallel computer system. Therefore, there is the need for another modeling approach, rather than analytic modeling approach. The use of efficient linear recursive model [9] can efficiently model the variation of waiting time of a distributed memory, heterogeneous parallel computer system. Therefore, recursive models can be used to efficiently determine the exact convergence of any series used in modeling the variation of waiting time of a distributed memory parallel computer system.

III. DEVELOPING THE RECURSIVE MODELS

The recursive model was developed for one queuing system; afterwards, it was generalized to consider all the queuing systems of the queuing network. As a result, the following models have been developed for one queuing system and for all the queuing systems of the queuing network.

A. *Models Based on a Queuing System*

The following models have been developed for one queuing system

1. Recursive Probability Density Function of the Number of Processes in One Queuing System.

Let X_i denotes the maximum number of processes that can be in the ith finite queuing system at any time [12], [13], [14]. Suppose the arrival rate, λ_{xi} when x_i processes are in the ith queuing system of the queuing network be described as:

$$\lambda_{xi} = \begin{cases} \lambda_i, & x_i = 0,1,2,,3,\ldots X_i - 1 \\ 0, & otherwise \end{cases} \qquad (1)$$

Since the various processors are heterogeneous, therefore, it implies that the departure rate will vary, which can be described as:

$$\mu_{xi} = \begin{cases} \mu_i, & x_i = 1,2,3,4,\ldots, X_i \\ 0, & otherwise \end{cases} \qquad (2)$$

Using the steady state probability as stated in [7], [16] the probability that x_i processes will be in the ith queuing system is

$$P_{xi} = \begin{cases} \rho_i^{\,x} P_{0i}, & x \le X_i \\ 0, & otherwise \end{cases} \qquad (3)$$

The utilization factor for the ith queuing system, ρ_i is defined as: $\frac{\lambda_i}{\mu_i}$. To obtain the value of P_{0i} in equation (3), we sum all the probabilities for the ith queuing system and equate it to 1. This implies that:

$$\sum_{x_i=0}^{X_i} P_{xi} = 1. \qquad (4)$$

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

$$P_{0i}+\rho_i P_{0i}+\rho_i^2 P_{0i} + \rho_i^3 P_{0i} + \rho_i^4 P_{0i} + \ldots + \rho_i^{X}{}_i P_{0i} = 1. \qquad (5)$$

Factorizing equation (5), it implies that

$$P_{0i}\,(1 + \rho_i + \rho_i^2 + \rho_i^3 + \ldots + \rho_i^{X}{}_i) = 1. \qquad (6)$$

Recursive model can be used to show where the series in (6) converges. The recursive model is given below as:

$$\text{Sum1}(X, \rho) = \begin{cases} 1, & X = 0 \\ \text{Term1}(X, \rho) + \text{Sum1}(X-1, \rho), & X \neq 0 \end{cases} \qquad (7)$$

The recursive algorithm that can be used to implement the recursive model in equation (7) is given below as:

Double sum1 (int X, double ρ)

1. Request data
   1.1 Request X
   1.2 Request ρ
2. Determine Sum1
   2.1 Sum1 = 1 if X = 0 else
   Sum1=Term1(X, ρ)+Sum1(X-1, ρ)
3. Display Sum1

Term1(X, ρ) is the recursive model that determines the xth term of the series in (6), it is given as:

$$\text{Term1}_i(X_i, \rho_i) = \begin{cases} 1, & X_i = 0 \\ \rho_i * \text{Term1}_i(X_i-1, \rho_i), & X_i \neq 0 \end{cases} \qquad (8)$$

The recursive algorithm that can be used to implement the recursive model in equation (8) is given below as:

Double Term1(int X, double ρ)

1. Request data
   1.1 Request X
   1.2 Request ρ
2. Determine Term1
   2.1 Term1 = 1 if X = 0 else
   Term1= ρ *Term1$_i$(X$_i$-1, ρ_i)
3. Display Term1

Using equation (7) in equation (6), we obtain the following:

$$\text{Poi Sum1}(X_i, \rho_i) = 1 \quad (9)$$

Solving for Poi in equation (9), we obtain the following:

$$\text{Poi} = \frac{1}{\text{Sum1}(X_i, \rho_i)} \quad (10)$$

Using equation (10) in equation (3), we have the following:

$$P_{i x_i} = \begin{cases} \dfrac{\rho_i^{x_i}}{\text{Sum1}(X_i, \rho_i)}, & x \le X_i \\ 0, Otherwise \end{cases} \quad (11)$$

Equation (11) is the probability density function that models the probability that x_i processes will be admitted in the ith queuing system.

2. Average Number of Processes in One Queuing System. Furthermore, the average number of processes in the ith queuing system (i.e the queue and the processor) can be described statistically as expectation of x_i, where x_i is the random variable that denotes the number of processes in the ith queuing system. This can be written as

$$E(x_i) = \sum_{x_i=0}^{X_i} x_i P_{i x_i}\ . \quad (12)$$

Using equation (11) in equation (12), we obtain the following:

$$E(x_i) = \sum_{x_i=1}^{X_i} \left(\frac{x_i \rho_i^{x_i}}{\text{Sum1}(X_i, \rho_i)} \right) \quad (13)$$

Equation (13) can be simplified as:

$$E(x_i) = \left(\frac{1}{\text{Sum1}(X_i, \rho_i)} \right) \rho_i \left(1 + 2\rho_i + 3\rho_i^2 + 4\rho_i^3 + \ldots + X_i \rho_i^{X_i - 1}\right) \quad (14)$$

A recursive model has been used in [30], [31] to determine the convergence of the series in equation (14). The recursive model is called Sum2$_i$(X$_i$, ρ_i), and it is given as:

$$\begin{cases} 1, & X_i = 1 \\ \text{Term2}_i(X_i)*\text{term1}_i(X_i-1, \rho_i) + \text{Sum2}_i(X_i-1, \rho_i), & X_i \neq 1 \end{cases} \quad (15)$$

The recursive algorithm that can be used to implement the recursive model in equation (15) is given below as:

Double Sum2(int X, double ρ)

1. Request data
   1.1 Request X
   1.2 Request ρ
2. Determine Sum2
   2.1 Sum2 = 1 if X = 1 else
   Sum2= Term2$_i$(X$_i$)*term1$_i$(X$_i$-1, ρ_i) +
   Sum2$_i$(X$_i$-1, ρ_i)
3. Display Sum2

Term2$_i$(X$_i$) is given as:

$$\text{Term2}_i(X_i) = \begin{cases} 1, & X_i = 1 \\ 1 + \text{term2}_i(X_i-1), & X_i \neq 1 \end{cases} \quad (16)$$

The recursive algorithm that can be used to implement the recursive model in equation (16) is given below as:

Double Term2(int X)

1. Request data
   1.1 Request X
2. Determine Term2
   2.1 Term2 = 1 if X = 1 else
   Term2= 1 + term2$_i$(X$_i$-1)
3. Display Term2

term1$_i$(X$_i$, ρ_i) is the recursive model in equation (8).

Therefore, using equation (15) in equation (14), we obtain:

$$E(x_i) = \left(\frac{\rho Sum2_i(X_i, \rho_i)}{Sum1_i(X_i, \rho_i)} \right) \quad (17)$$

B. Models Based on The Whole Queuing Network.

Having developed the models for the performance metrics of one queuing system, these models can be extended to the whole queuing systems of the queuing network of a heterogeneous parallel computer system. It is necessary to define δ_i as the probability that a process will join the ith queue after each cpu burst, and δ_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 λ_i be the rate of arrival of processes into the ith queuing system, and λ, the rate of arrival of processes from the outside world.. Under the steady state, when we consider the queuing network, the overall utilization factor has been defined in [31] as:

$$\rho_i = \frac{\lambda \delta_i}{\delta_0 \mu_i}, \quad i = 1,2,3,\ldots, n+k \quad (18)$$

1. Variation of Average Waiting Time in all the Queuing Systems of the Queuing Network

Suppose x_i is the random variable that denotes the number of processes in the ith queuing system. Therefore, the average or mean processes in all the queuing systems of the queuing network can be defined as

$$Y = \frac{\sum_{i=1}^{n+k} x_i}{n+k} \quad (19)$$

Little's formula can be used to related equation (19) to the average waiting time in all the queuing systems of the queuing network. Therefore, using the constant of proportionality of Little's formulae [7], we can establish a relationship between the average number of processes in all the queuing systems and the average waiting time in all the queuing systems of the queuing network., as shown in equation (20).

$$Ws = \left(\frac{\sum_{i=1}^{n+k} x_i}{n+k} \right) \frac{1}{\lambda_{e\,ff}} \quad (20)$$

The constant of proportionality,

$$\lambda_{e\,ff} = \frac{\sum_{i=1}^{n+k} \lambda_{e\,ff_i}}{n+k} \quad (21)$$

Using equation (20), we can take the variance of the average waiting time as:

$$VAR(Ws) = VAR\left(\left(\frac{\sum_{i=1}^{n+k} x_i}{n+k} \right) \frac{1}{\lambda_{e\,ff}} \right) \quad (22)$$

Using one of probability theory laws in [23], we obtain:

$$VAR(Ws) = \frac{1}{(n+k)^2 \lambda_{e\,ff}^{\;2}} \sum_{i=1}^{n+k} VAR(x_i) \quad (23)$$

From [23], the variance can be defined statistically as:

$$VAR(x_i) = E(x_i^{\;2}) - (E(x_i))^2 \quad (24)$$

Simplifying equation (24) further, we obtain:

$$E(x_i^{\;2}) = \sum x_i^{\;2} P_{i\,x_i} \quad (25)$$

Using equation (11) in equation (25), we obtain:

$$E(x_i^{\;2}) = \sum_{x_i=1}^{X_i} \left(\frac{x_i^{\;2} \rho_i^{\;x_i}}{Sum1_i(X_i, \rho_i)} \right) \quad (26)$$

Simplifying equation (25), we obtain:

$$E(x_i^{\;2}) = \left(\frac{1}{Sum1_i(X_i, \rho_i)} \right) \left(1^2 \rho_i + 2^2 \rho_i^{\;2} + 3^2 \rho_i^{\;3} + 4^2 \rho_i^{\;4} + \ldots + X^2{}_i \rho_i^{\;X_i} \right) \quad (27)$$

Simplifying equation (27) further, we obtain:

$$E(x_i^{\;2}) = \left(\frac{1}{Sum1_i(X_i, \rho_i)} \right) \left(1\;\rho_i + 4\rho_i^{\;2} + 9\rho_i^{\;3} + 16\rho_i^{\;4} + \ldots + X^2{}_i \rho_i^{\;X_i} \right) \quad (28)$$

Factorising equation (28), we obtain:

$$E(x_i^{\;2}) = \left(\frac{1}{Sum1_i(X_i, \rho_i)} \right) \rho_i \left(1 + 4\rho_i + 9\rho_i^{\;2} + 16\rho_i^{\;3} + \ldots + X^2{}_i \rho_i^{\;X_i - 1} \right) \quad (29)$$

The convergence of the series may not be efficiently determined analytically; therefore we seek for its convergence using recursive models. The same approach used earlier can be used to determine the convergence of the series, $\left(1 + 4\rho_i + 9\rho_i^{\;2} + 16\rho_i^{\;3} + \ldots + X^2{}_i \rho_i^{\;X_i-1}\right)$. The series can be considered as two sequences, which are: sequence1 = 1, 4, 9, 16, …, X^2, while the other sequence is: sequence2 = $\left(1, \rho_i, \rho_i^{\;2}, \rho_i^{\;3}, \ldots, \rho_i^{\;X_i-1}\right)$. The recursive model that can be used to determine the xth terms of sequence1 can be obtained by adding 2X-1, which is the common difference between the xth term and the (x-1)th term, to the (x-1)th term of the sequence. The recursive model can be represented as shown below in equation (30), as:

$$Term3_i(X_i) = \begin{cases} 1, & X_i = 1 \\ (2*X_i - 1) + Term3_i(X_i - 1), & X_i \neq 1 \end{cases} \quad (30)$$

The recursive algorithm that can be used to implement the recursive model in equation (29) is given below as:

Double Term3(int X)

1. Request data
   1.1 Request X
2. Determine Term3
   2.1 Term3 = 1 if X = 1 else
   Term3= (2*X_i-1) + $Term3_i(X_i-1)$
3. Display Term3

The recursive model that determines the xth terms of sequence2 has been developed in equation (8). Therefore, combining equation (30) and equation (8), the series in equation (29) converges to this recursive model, called $Sum3_i(X_i, \rho_i)$, which is shown below as:

$$Sum3_i(X_i, \rho_i) = \begin{cases} 1, & X_i = 1 \\ Term1_i(X_i - 1, \rho_i) * Term3_i(X_i) + Sum3_i(X_i - 1, \rho_i), & X_i \neq 1 \end{cases} \quad (31)$$

The recursive algorithm that can be used to implement the recursive model in equation (31) is given below as:

Double Sum3(int X; Double ρ_i)

1. Request data
   1.1 Request X
   1.2 Request ρ
2. Determine Sum3
   2.1 Sum3 = 1 if X = 1 else
   Sum3= $Term1_i(X_i-1, \rho_i)$*$Term3_i(X_i)$ + $Sum3_i(X_i-1, \rho_i)$
3. Display Sum3

Therefore, using equation (31) in equation (29), we obtain:

$$E(x_i^{\;2}) = \left(\frac{\rho_i Sum3(X_i, \rho_i)}{Sum1(X_i, \rho_i)} \right) \quad (32)$$

Using equations (17) and (32) in equation (24), we obtain:

$$VAR(x_i) = \left(\frac{\rho_i Sum3(X_i, \rho_i)}{Sum1_i(X_i, \rho_i)} \right) - \left(\frac{\rho_i Sum2(X_i, \rho_i)}{Sum1_i(X_i, \rho_i)} \right)^2 \quad (33)$$

Therefore, using equation (33) in equation (23), we obtain:

$$VAR(Ws) = \frac{1}{(n+k)^2 \lambda_{e\,ff}^{\;2}} \left(\sum_{i=1}^{n+k} \left(\left(\frac{\rho_i Sum3(X_i, \rho_i)}{Sum1_i(X_i, \rho_i)} \right) - \left(\frac{\rho_i Sum2(X_i, \rho_i)}{Sum1_i(X_i, \rho_i)} \right)^2 \right) \right) \quad (34)$$

Equation (34) models the variation of the average waiting time in all the queuing systems of the queuing network.

IV. VERIFYING AND VALIDATING THE MODELS USING Z-TRANSFORM

Model verification and model validation are essential parts of model development that will help to assess the quality of the developed models. If a model is not verified and validated it cannot be assured of quality, therefore, it can be sent back to the drawing board. Model verification is done in order to ensure that the simulation algorithm i.e. algorithms used to implement the models on the computer are correct and the simulation programs i.e. model implementation programs are correctly programmed. Model verification eliminates every error that may occur when implementing the models on the computer. On the other hand, model validation aims at making the model address the right problem, address accurate information about the system being modeled. Model validation compares the results of the simulated models with the results of a real system. Therefore, model validation tries to establish if the model is an accurate representation of the real system. However, due to one reason or the other, it may not be easy sometime to obtain results of the real system, in such a situation, expert knowledge can be used to determine if the qualitative data from the simulated model is valid or invalid [35]. The authors in [34] argued that though quantitative comparison will provide the basis for validation, however, it can miss the qualitative discrepancies or agreements that human are capable of detecting. One of the ways they suggested that can be used to detect such discrepancies or agreements is through visualization. Visualization, according to them helps to map numerical data into graphical structure that human can more readily understand. This graphical display of the results of the simulated model or the system behavior will help us to determine if the model is valid or invalid. Furthermore, [34] pointed out that quantitative comparison is needed to make finer distinctions between behaviors that agree in their basic form, but qualitative comparison can help to eliminate models that are not in the right ballpark [34]. Sometimes, a validated model can be used to validate another model by comparing qualitative and quantitative data of the two models.

The statistical method of Z-transform can be used to validate the recursive models. The Z-transform for the ith queuing system, using the statistical generating function is given as:

$$G_{Xi}(z) = E(z^{X}) \qquad (35)$$

Therefore, using equation (35), the variance of the ith queuing system can be expressed in terms of the z-transform as in [25]:

$$VAR(x_i) = \left.\frac{\delta^2 G_{Xi}(z)}{(\delta z)^2}\right|_{z=1} + \left.\left(\frac{\delta G_{Xi}(z)}{(\delta z)}\right)\right|_{z=1} - \left.\left(\frac{\delta G_{Xi}(z)}{(\delta z)}\right)^2\right|_{z=1} \qquad (36)$$

Simplifying equation (36) further, using the analytic model for the probability density function for $\rho_i \neq 1$, as stated in [33], we obtain the following:

$$G_{Xi}(z) = \sum_{x_i=0}^{X_i} z^{x_i}\left(\frac{\rho_i^{x_i}(1-\rho_i)}{1-\rho_i^{X_i+1}}\right) \qquad (37)$$

Simplifying further, we obtain the following:

$$G_{Xi}(z) = \left(\frac{(1-\rho_i)}{1-\rho_i^{X_i+1}}\right)\sum_{x_i=0}^{X_i} z^{x_i}\rho_i^{x_i} \qquad (38)$$

Simplifying further, we obtain the following:

$$G_{Xi}(z) = \left(\frac{(1-\rho_i)}{1-\rho_i^{X_i+1}}\right)\sum_{x_i=0}^{X_i}(z\rho_i)^{x_i} \qquad (39)$$

Simplifying further, we obtain the following:

$$G_{Xi}(z) = \left(\frac{(1-\rho_i)}{1-\rho_i^{X_i+1}}\right)\left(\frac{\left(1-(z\rho_i)^{X_i+1}\right)}{(1-(z\rho_i))}\right) \qquad (40)$$

Therefore, taking the first derivative of equation (40), with respect to z, and initializing z to 1, we obtain the following:

$$\frac{\delta G_{Xi}(z)}{\delta z} = \left(\frac{\rho(1-\rho_i)}{1-\rho_i^{X_i+1}}\right)\left(\frac{1+X\rho^{X+1}-\rho^{X}(X+1)}{(1-\rho)^2}\right) \qquad (41)$$

Simplifying further, equation (41) reduces to:

$$\frac{\delta G_{Xi}(z)}{\delta z} = \left(\frac{\rho}{1-\rho_i^{X_i+1}}\right)\left(\frac{1+X\rho^{X+1}-\rho^{X}(X+1)}{(1-\rho)}\right) \qquad (42)$$

Furthermore, taking the second derivative of equation (40), with respect to z and initializing z to 1, we obtain the following:

$$\frac{\delta^2 G_{Xi}(z)}{(\delta z)^2} = \left(\frac{(1-\rho_i)}{1-\rho_i^{X_i+1}}\right)\left(\frac{u+v}{(1-\rho)^4}\right) \qquad (43)$$

Simplifying further, u and v are given as:

$$u = (1-\rho_i)^2\left(X_i(X_i+1)\rho_i^{X_i+2} - X_i^2\rho_i^{X_i+1} - X_i\rho_i^{X_i+1}\right) \qquad (44)$$

$$v = 2\rho_i(1-\rho_i)\left(X\rho_i^{X_i+2} - X_i\rho_i^{X_i+1} - \rho_i^{X_i+1} + \rho_i\right) \qquad (45)$$

Therefore, using equations (44) and (45) in equation (43), and using equations (43) and (42) in equation (36), we obtain the z-transform model for the variation of waiting time in the ith queuing system. Furthermore, the z-transform can be used to obtain variation of the average waiting time in all the queuing systems of the queuing network, as shown in equation (46) below.

$$VAR(Ws) = \frac{1}{(n+k)^2\lambda_{eff}^{\,2}}\left(\sum_{i=1}^{n+k} VAR(x_i)\right) \qquad (46)$$

However, the z-transform cannot be used to effectively validate the recursive model for the isolated case when $\rho_i = 1$.

V. METHODOLOGY

This paper has used recursive models to model the variation of waiting time of distributed memory, heterogeneous parallel computer system. A queuing approach, with finite queues has been used to achieve the above aim, with parallel processors depicting parallel servers. The statistical method of probability density function and other probability theory concepts have used [15], [23]. A novel method of deriving the recursive model that determines the xth terms and the convergence of important mathematical series have been used to develop the recursive models.

The simulation of the models on the computer has been done using Java programming language and the statistical regression/trend line analysis has been used to analyze the results of the simulation [11]. The simulated recursive models have been validated using statistical method of Z-Transform.

VI. RESULTS OF THE SIMULATION

The results of the simulation have been analyzed to determine how variation of the waiting time changes as a particular parameter varies, while other parameters remain constant [10]. Table 1 and figure 2 show the result of the simulation, suppose the probability of a process leaving the system is known to be 0.2 and the probabilities that a process will join the first and second queues are 0.775 and 0.025, respectively. Suppose the first processor is a high-speed processor with high departure rate of 30, while the second processor is a low speed processor with a low departure rate of 10. Suppose the maximum number of processes to be allowed into first queue is 20, while maximum number of processes to be allowed into the second queue is 5. The experimental trials were carried out several times, in each trial, the arrival rate was changed, and the corresponding variation was obtained as the result of the simulation.

Table 1: Result of VARIATION AGAINST Arrival Rate

AR	V.From Model	V.From Z-Transform
3	0.00186	0.00186
4	0.00221	0.00221
5	0.00323	0.00323
6	0.00578	0.00578
7	0.00978	0.00978
8	0.00999	0.00999
9	0.00623	0.00623
10	0.00341	0.00341
11	0.002015	0.002015
12	0.001332	0.001332

Key to the Table:
AR: Arrival Rate.
V.from Model: Waiting Time Variation, using Recursive models.
V.from Z-Transform: Waiting Time Variation, using Z-Transform.

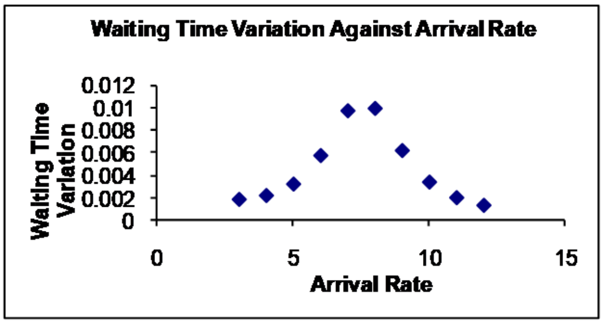

Figure 2: Variation Against Arrival Rate

The undulating nature of the result shows the various points where minimum variations and maximum variation can be realized.

Furthermore, table 2 and figure 3 show the simulation results as we keep the following input parameters constant, the probability that a process will leave the network is 0.2, the probabilities that a process will join queue 1 and 2 are 0.775 and 0.025, respectively, while the departure rates for processor 1 and 2 are 30 and 10, respectively, and the maximum number of processes in queue 1 and 2 (degree of multiprogramming for the two queues) are 20 and 5, respectively, and the arrival rate from the outside world is 4 (for non-compute intensive applications) and 30 (for compute intensive applications). By changing the degree of multiprogramming (maximum number of processes in the system) for the two queues of a two-processor parallel computer system, we obtain the corresponding variations shown in table 2 and figure 3 for non-compute intensive applications.

Table 2: Result Of Variation Against Degree Of Multiprogramming

TMP	V. from Model	V. from Z-Transform
8	0.0010662	0.0010662
13	0.0015138	0.0015138
18	0.0020149	0.0020149
23	0.0021338	0.0021338
28	0.0021958	0.0021958
33	0.0022076	0.0022076
38	0.0022127	0.0022127
43	0.0022135	0.0022135
48	0.0022138	0.0022138

Key to the Table:
TMP: Total Maximum Number of Processes.
V.from Model: Waiting Time Variation from Model.
V.from Z-Transform: Waiting TimeVariation from Z-Transform.

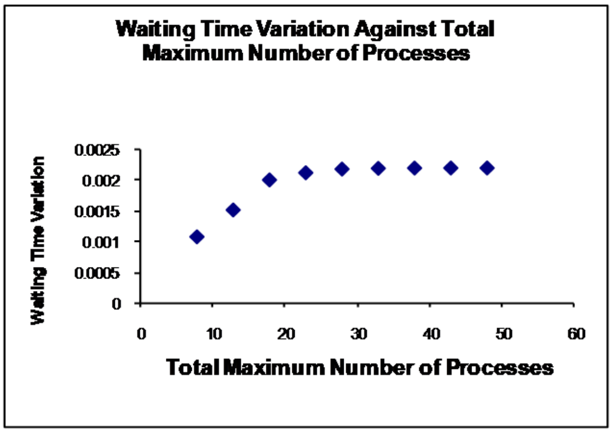

Figure 3: Variation Against the Degree of Multiprogramming

From the results in table 2 and figure 3, it can be seen that for non-compute intensive applications, where overall utilization factor is less than 1, as the total maximum number of processes in all the queues increases, the waiting time variation increases, but afterwards, it remains constant.

TABLE 3: Result Of Variation Against Degree Of Multiprogramming

TMP	V. from Model	V. from Z-Transform
8	2.82E-04	2.82E-04
13	3.09E-04	3.09E-04
18	3.13E-04	3.13E-04
23	3.14E-04	3.14E-04
28	3.14E-04	3.14E-04
33	3.14E-04	3.14E-04
38	3.14E-04	3.14E-04
43	3.14E-04	3.14E-04
48	3.14E-04	3.14E-04

Table 3 and figure 4 show the results of the waiting time variation against the total maximum number of processes for compute intensive applications, i.e. when the overall utilization factor is greater than 1. The behavior of the waiting time variation is the same for both compute and non-compute intensive applications.

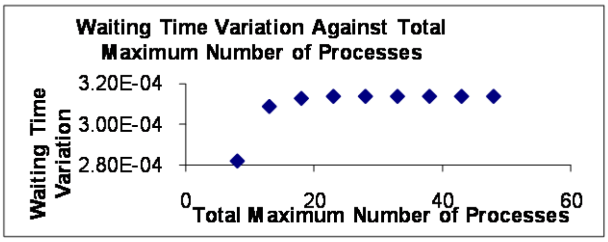

Figure 4: Variation Against the Degree of Multiprogramming

In a similar manner, as we keep the following input parameters constant, probability of a process leaving the network is 0.2, while the probability of a process going to queue 1 and 2 is 0.4, the arrival rate from the outside world is 5. The maximum number of processes that can be in queue 1 and 2 are 15 and 14, respectively. By changing the departure rates of the two processors, we obtain the corresponding variations of the waiting time, as shown in table 4 and figure 5. The result shows the behavior of the waiting time variation for compute intensive applications, i.e. when the overall utilization factor is greater than 1, is different from the behavior of the waiting time variation for non-compute intensive applications, i.e. when the overall utilization factor is less than 1. From the results in table 3 and figure 4, increasing the speed of the processors for compute intensive applications will lead to a corresponding increase in the waiting time variation. On the other hand, increasing the speed of the processors for non-compute intensive applications will lead to a corresponding decrease in the waiting time variation.

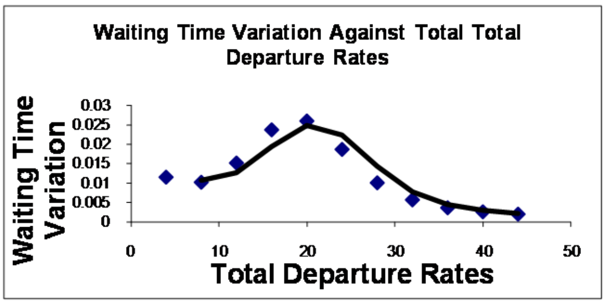

Figure 5: Variation Against Total Departure Rate

TABLE 4: Result Of Variation Against Departure Rate

TDR1	TDR2	WTVM	WTVZT
1	3	0.011495	0.011495
3	5	0.010178	0.010178
5	7	0.015133	0.015133
7	9	0.023701	0.023701
9	11	0.025995	0.025995
11	13	0.018658	0.018658
13	15	0.010034	0.010034
15	17	0.005631	0.005631
17	19	0.003593	0.003593
19	21	0.002543	0.002543
21	23	0.001934	0.001934

Key to the table:
DRP1 Departure Rate for Processor 1
DRP2: Departure Rate for Processor 2
WTVM: Waiting Time Variation from Model
WTVZT: Waiting Time Variation from Z-Transform

VI. SUMMARY AND CONCLUSION

This paper has been able to model the variation of a waiting time of heterogeneous parallel computer, using recursive models and queuing approach. The models have been simulated on the computer using Java programming language and validated using statistical Z-Transform method, the results of the simulation have been analyzed in order to determine when to realize minimum variation.

REFERENCES

1. Henry H. Liu and Pat V. Crain, An Analytic Model for Predicting the Performance of SOA-Based Enterprise Software Applications, Proc. International Conference of Computer Measurement Group, (2004).
2. S. Balsamo et al, A Review of Queueing Network Models with Finite Capacity Queues for Software Architecture Performance Prediction, (2002).
3. Catalina M. Liado et al, A Performance Model Web Service, Proc. International Conference of Computer Measurement Group, (2005).
4. Rosselio, J et al, A Web Service for Solving Queueing Network Models Using PMIF. www.perfeng.com/paperndx.htm, (2005).
5. Cathy H. Xia, Zhen Liu., Queueing systems with long-range dependent input process and subexponential service time. Proc. ACM SIGMETRICS international conference on Measurement and modeling of computer systems,(2003).
6. Shanti Subramanyam, Performance Modelling of a J2EE Application to meet Service Level s, Agreement, Proc. International Conference of Computer Measurement Group, (2005)
7. Hamdy A. T.,. Operation Research: An Introduction, Prentice-Hall of India, (1999).
8. Ivan Stojmenovic; Recursive Algorithms in Computer Science Courses : Fibonacci Numbers and Binomial Coefficients; IEEE Transactions on Education; Vol. 48, No. 3
9. Arjan J.C. van Gemund; Performance Modelling of Parallel Systems: An Introduction.
10. Justyna Berlinska, The Statistical models of parallel applications, Annales UMCS Informatica, (2005).
11. Arranchenkov, K.E., Vilchersky, N.O., Shevlyakor, G.L Priority queueing with finite buffer size and randomized push-out; mechanism. Proc. of ACM SIGMETRICS international conference on measurement and modeling of computer systems.; (2003).
12. Abunday, B.D., and Khorram, E. The finite source queueing model for multiprogrammed computer systems with different CPU times and different I/O times. Acta Cybern. 8, 4 , (1998)
13. J. Sztrik; Finite-Source Queueing Systems and their Applications: A Biliography;
14. Trivedi K. Shridharbhai, Probability and Statistics with Reliability, Queuing and Computer Science Applications, John Wiley & Sons Inc., (2002).

15. Per Brinch Hansen. Operating System Principles. Prentice-Hall of India Private Limited, (1990).
16. J. Sztrik[a] and T. Gál A recursive solution of a queueing model for a multi-terminal system subject to breakdowns; Performance Evaluation Volume 11, Issue 1, Published by Elsevier, (1990).
17. Robert V. Hogg and Allen T. Craig; Introduction to Mathematical Statistics; Macmillan Publishing Co. Inc.; (1978).
18. Andrea Clemantis, Angelo Corana; Modelling Performance of Heterogeneous Parallel Computer System; Journal of Parallel Computing, Volume 12, Issue 9, Elsevier; pages 1131-1145; (1999).
19. E. Post, H.E. Goosen; Evaluating the Parallel Performance of a Heterogeneous System
20. Beutler, F; Mean sojourn times in markov queuing network: Little's formula revisited; IEEE Transaction on Information Theory; Volume 29, Issue 2, page 233-241; (2003).
21. Ken Vastola; http://networks.ecse.rpi.edu/~vastola/pslinks/perf/node46.html
22. Xiaodong Zhang, Yong Yan; Modeling and Characterizing Parallel Computing Performance on Heterogeneous Network of workstations; Proceedings of the 7th IEEE Symposium on Parallel and Distributeed Processing (SPDP '95) 1063-6374/95 $10.00 © 1995 IEEE
23. O.E. Oguike et al; Modelling the Performance of Computer Intensive Applications of Parallel Computer System; Proc. Of IEEE 2nd International Conference on Computational Intelligence, Modeling and Simulation; (2010).
24. O.E. Oguike et al; Evaluating the Performance of Parallel Computer System Using Recursive Models; Proc. Of IEEE 4th UKSim European Modeling Symposium; (2010).
25. O.E. Oguike et al; Evaluating the Performance of Heterogeneous Distributed Memory Parallel Computer System Using Recursive Models; 2nd IEEE International Conference on Intelligent Systems, Modeling and Simulation; (2011).
26. Leonard Kleinrock, Queueing Systems Volume 1 and 2, John Wiley & Sons, (1975).
27. O.E. Oguike et al; Modelling Variation of a Performance Metric of Distributed Memory Heterogeneous Parallel Computer System, Using Recursive Models; In proc. of 3rd IEEE International Conference on Computational Intelligence Modeling and Simulation; (2011).
28. Bernard P. Zeigler et al; Theory of Modelling and Simulation; Elsevier; (2000)
29. Cor van Dijkum et al; Validation of Simulated Models; Siswo Publication 403, Amsterdam, (1999)

AUTHORS FROFILE

Oguike, Osondu Everestus, is a Senior Lecturer in the Department of Computer Science, University of Nigeria, Nsukka, Enugu State, Nigeria. He obtained his B.Sc degree from the University of Lagos, Nigeria. His postgraduate Diploma and M.Sc degree were obtained from Queen Mary and Westfield College, University of London, United Kingdom. He has received many academic prizes and scholarships as a result of his outstanding academic performance. His research interest is in performance modeling of parallel computer systems.

Dr Monica N. Agu, is of Department of Computer Science, University of Nigeria, Nsukka, in the faculty of Physical Sciences. She obtained her first degree (B.Sc. Honors) in Computer Science, from University of Ibadan, Ibadan, Nigeria. Her M.Sc. degree was obtained from the University of Nigeria, Nsukka, and her Ph.D was obtained from Ebonyi State University, Abakaliki, Nigeria. She has co-authored some books, published some journal articles. She is a member of Nigeria Computer Society and Ccomputer Professional of Nigeria. Her research has focused on using Information and Communication Technology on Poverty Alleviation and Modelling the performance of computer systems.

Echezona Stephenson C. is a faculty member, as well as, on a PhD of the Computer Science Department, University of Nigeria, Nsukka. He obtained his B.Sc and M.Sc degree from the University of Nigeria, Nsukka. His research has focused on using Z-Transform to corroborate the Recursive Performance Models of Parallel Computer Systems.