Performance Modeling and Prediction of Nondedicated Network Computing

Linguo Gong, Xian-He Sun, *Senior Member*, *IEEE*, and Edward F. Watson

Abstract—The low cost and wide availability of networks of workstations have made them an attractive solution for high performance computing. However, while a network of workstations may be readily available, these workstations may be privately owned and the owners may not want others to interrupt their priority in using the computer. Assuming machine owners have a preemptive priority, in this paper, we study the parallel processing capacity of a privately owned network of workstations. A mathematical model is developed to predict performance for nondedicated network computing. It also considers systems with heterogeneous machine utilization and heterogeneous service distribution. This model separates the influence of machine utilization, sequential job service rate, and parallel task allocation on the parallel completion time. It is simple and valuable for guiding task scheduling in a nondedicated environment.

Index Terms—Network cluster computing, performance modeling and analysis, nondedicated systems, workload distribution.

1 Introduction

The merging of two rapidly advancing technologies, computers and communications, has resulted in a new computing infrastructure, called network of workstations (NOW) [1], [2]. The potential for this new computing infrastructure has attracted recent attention. Many software tools have been developed to support distributed computing over a network of workstations, including such widely used tools as the *Parallel Virtual Machine* (PVM) software and the *Message Passing Interface* (MPI) [4], [8]. A national initiative has been called to build a national information power grid [3]. The popularity of NOW is due to its ability to provide significant cost effective computing, to efficiently support both single processor interactive processing and large batch parallel processing, and to rely on commodity technology.

Depending on the ownership, NOW can be divided into two categories: *dedicated* networks of workstations and *nondedicated* networks of workstations. Dedicated NOW uses a cluster of dedicated workstations collectively to form a cost-effective parallel computer. On the other hand, nondedicated NOW are targeted to utilize the abundant computing cycles available on the network to provide high computing power without, or with little, additional financial investment. In a nondedicated environment, however, workstations are privately owned and likely to be heterogeneous. The "availability" and heterogeneity of nondedicated network computing distinguishes itself from dedicated parallel computing. Though performance modeling of nondedicated computing is essential for the success of next generation network environments (including publicized network meta-computing, ubiquitous supercomputing, world-wide virtual machine environments, and information power grid [3], [5]), there is no widely accepted performance model for nondedicated network computing. In this study, we first introduce an analytical model to predict parallel task completion time in a nondedicated homogeneous environment. Next, similar analysis is extended to systems with heterogeneous machine utilization and heterogeneous service distributions. Based on the analysis, we then propose a task partition procedure that is optimal based on the first two moments in a nondedicated environment. This research results in the separation of the influence of machine utilization, sequential job service rate, and parallel task allocation to the parallel task completion time. It indicates when efficient process migration is critical for the success of nondedicated network computing. It may be used to provide a guideline for process allocation and scheduling in a nondedicated environment.

Performance modeling of distributed network computing has traditionally focused on dedicated systems [12], [15]. Recently, nondedicated network computing has received considerable attention. Much of the recent research is observational in nature. The actual workstation usage patterns are measured by [13], [1]; the performance parameters and their influence on a set of applications are reported by [14], [1] and the issues of dynamic scheduling are addressed in [1]. These results are useful for the development and evaluation of performance models of privately owned NOWs. Other studies focus on modeling the capacity of nondedicated computing for general solutions. Mutka and Livny [13] identified the availability pattern of distributed computing cycles for a cluster of workstations. Based on months of observations, they concluded that the distribution of unavailable time intervals of individual machines could be approximated using hyperexponential distributions. This conclusion seems

- *L. Gong is with the Department of Management Sciences, Rider University, 2083 Lawrenceville Rd., Lawrenceville, NJ 08603. E-mail: lgong@rider.edu.*
- *X.-H. Sun is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616. E-mail: sun@cs.iit.edu.*
- *E.F. Watson is with the Department of Information Systems and Decision Sciences, Louisiana State University, Baton Rouge, LA 70803-4020. E-mail: ewatson@lsu.edu.*

Manuscript received 2 July 1997; revised 9 Mar. 2001; accepted 4 Mar. 2002. For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 105330.

reasonable since this can be interpreted as the coexistence of different user groups in the computer resource. Leutenegger and Sun [10], [11] determined the capacity of nondedicated homogeneous computing environments. They considered a discrete model where the machine owners use their machines with a fixed probability and fixed job length. Kleinrock and Korfhage [9] used Brownian motion to approximate the parallel task completion time in a nondedicated system. Assuming parallel tasks arrive equally during each state of the local sequential processing, they derived analytical expressions of the approximated mean and standard deviation of parallel completion time. While Kleinrock and Korfhage's model might be the most general model available for nondedicated computing, its application in network computing seems elusive. In distributed computing practice, as reported by the NOW group at Berkeley [1], unused machines can always be found on the network if the number of workstations on the network is more than double that of the required parallelism. Parallel tasks are most likely to be started without any waiting. In addition, while Brownian motion is a powerful tool for finding the expected performance of a long process, it does not distinguish the impact of different influential factors.

The purpose of this study is to analyze the nature of nondedicated computing so that we are able to develop a practical approach to performance estimation and so that we can make appropriate decisions for the distribution of parallel tasks. Following [1] and [11], we assume that parallel tasks are only assigned to idle machines. Based on predetermined machine usage patterns, we derive simple formulations to predict the parallel task completion time under different circumstances. The mean, standard deviation and the distribution of parallel task completion time are analyzed. The effects of heterogeneity, machine utilization, number of workstations, task partitioning and allocation, and other parallel considerations are also discussed. Partitioning and scheduling policies for parallel tasks are developed based on the finding from the analytical model developed. Experimental results show that the proposed analytical formulation is reasonably precise, which may provide a practical solution for estimating parallel completion time in a nondedicated environment.

This paper is organized as follows: In Section 2, an analytical model for estimating parallel task completion time in a nondedicated distributed computing environment is developed. The analytical model is carefully examined and evaluated in Section 3 for homogenous systems with homogeneous machine usage patterns. Effects of different factors on the parallel task completion time are examined. The analysis is then extended to systems with heterogeneous machine usage patterns in Section 4. In this section, optimal partitioning of parallel tasks based on the first and second moments are discussed. The main results are summarized in Section 5. Finally, conclusions are given in Section 6.

2 Performance Modeling and Analysis

In this section, we describe our system models, verify the system assumptions, and deduct probabilistic formulas for parallel task completion time. We assume that the parallel task is composed of one single parallel phase with no communication or synchronization requirements other than the final synchronization, which occurs when all of the tasks are completed. Part of the communication delay is implicitly included in the service rate. We assume the computing system is homogeneous. That is, all the machines on the network have the same computation power. This assumption will be lifted in Section 4. The machine owners' local sequential jobs have preemptive priority over processes belonging to parallel tasks. The arrival of the owner's sequential jobs at workstation k is assumed to follow a Poisson distribution with rate λ_k. A newly arriving sequential job must wait if another sequential job is in process. Otherwise, it will start processing immediately by using the unused machine or by preempting a parallel task. We assume that the execution time of the owner jobs at workstation k follows a general distribution with mean $1/\mu_k$ and standard deviation σ_k. μ_k is also called the service rate at workstation k since it directly depends on the computational power of the machine. Based on our assumption, the owner job process is an M/G/1 queuing system. Note that the hyperexponential machine usage pattern observed by Mutka and Livny [13] can be explained by the notion that there are different, independent classes of users and that each class of users assumes the exponential distribution, but with a unique parameter set. The general service assumption in our model is a generalization of the observed usage pattern. As discussed in Section 1, we also assume that the parallel task is only initiated on unused machines. This assumption agrees with the conclusion made by [11] that parallel tasks should be initiated on lightly loaded machines and should be migrated (to other resources) when the load on a machine becomes heavy in a network environment.

We assume that the parallel task requires a total processing time W and is partitioned into m subtasks, $w_1, w_2, \ldots, w_m$, for parallel processing. Subtask w_k is assigned to workstation k and $W = \sum_{k=1}^{m} w_k$. We use T_k to represent the total time required to finish parallel subtask k at workstation k. We list the notation to be used through out this paper in the following:

- W Total demand of the parallel task.
- w_k Demand of the parallel subtask on workstation k.
- m Number of workstations in the system.
- S The number of interruptions encountered.
- λ_k Rate of the job arrival Poisson distribution at workstation k.
- μ Sequential job service rate at workstation k.
- ρ_k Utilization rate at workstation k.
- σ_k Standard deviation of service time on workstation k.
- T_k Parallel task completion time on workstation k.
- $E(.)$ Expectation operator.

2.1 Subtask Completion Time at a Single Workstation

Given a workstation is idle when the parallel task arrives at the workstation, the parallel completion time T_k at workstation k can be expressed as:

TABLE 1
Parameter Set for Distribution of Parallel Subtask Completion Time

W	1	2	4	8	16	32	64	126	252
ρ	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
θ	1.0	2.0	4.0	8.0	16.0				

$$T_k = X_1 + Y_1 + X_2 + Y_2 + \ldots + X_S + Y_S + Z. \quad (1)$$

As defined in Table 1, S is the total number of interruptions that occur during the processing of a parallel task due to the arrival of one or more sequential jobs. X_i, Y_i, $i = 1, \ldots, S$, represents the computing time consumed by the parallel task and the sequential jobs, respectively. Z is the execution time of the last parallel process that finishes the parallel task. We have

$$X_1 + X_2 + \ldots + X_S + Z = w, \quad (2)$$

hence

$$T_k = w + Y_1 + Y_2 + \ldots + Y_S. \quad (3)$$

Let $\Gamma(0) = 0$ and $\Gamma(S) = X_1 + X_2 + \ldots + X_S$, for $S > 0$. Then, from (2), we have random variable $S \in \{0, 1, \ldots, \infty\}$ and $\Gamma(S) < w$, $\Gamma(S+1) \geq w$.

Note the assumption that the owner job arrival process follows a Poisson distribution; X_i is an exponentially distributed random variable. $\Gamma(S)$ is therefore a Gamma distributed random variable for given $S = s > 0$. Using the well-known result in queuing system, we have:

Proposition 1. *Assume that the owner workstation process can be treated as an M/G/1 queuing system with arrival rate λ and service rate μ, then the total number of interruptions S follows a Poisson distribution with parameter λw.*

Hence, the probability function of S satisfies:

$$\begin{aligned} p_s &= \Pr(S = s) = \Pr(\Gamma(s) < w, \Gamma(s+1) \geq w) \\ &= \frac{(\lambda w)^s 2^{-\lambda w}}{s!} S = s > 0. \end{aligned} \quad (4)$$

Note that the job completion time is

$$T_k = w + Y_1 + Y_2 + \ldots + Y_S,$$

where Y_j, $j = 1, 2, \ldots s$, are i.i.d. random variables representing the jth busy period of the machine owner's sequential jobs. Under the assumption that the owner job processing follows $M/G/1$, we have the following first and second moments of Y_j using existing results from queuing theory:[1]

$$\mathrm{E}(Y_j) = \frac{1}{\mu - \lambda} \quad (5)$$

$$\mathrm{E}(Y_j^2) = \frac{\mu(\mu^2\sigma^2 + 1)}{(\mu - \lambda)^3}. \quad (6)$$

1. Readers may refer to Chapter 4, D. Gross and C.M. Harris, for the derivation of the results.

The mean and variance of T_k can be obtained through the following expression:

$$\begin{aligned} \mathrm{E}(T_k) &= \mathrm{E}(\mathrm{E}(T_k \mid S)) = \mathrm{E}(w + Y_1 + Y_2 + \ldots + Y_S \mid S) \\ &= \mathrm{E}(w + SE(Y_1)) = w(1 + \lambda \mathrm{E}(Y_1)) \\ &= \frac{1}{1 - \rho} w, \end{aligned} \quad (7)$$

$$\begin{aligned} \mathrm{V}(T_k) &= \mathrm{E}(\mathrm{V}(T_k \mid S)) + \mathrm{V}(\mathrm{E}(T_k \mid S)) \\ &= \mathrm{E}(V(w + U(S) \mid S)) + \mathrm{V}(\mathrm{E}(w + U(S) \mid S) \\ &= \mathrm{E}(S\mathrm{V}(Y_1)) + \mathrm{V}(w + S\mathrm{E}(Y_1)) \\ &= \lambda w \mathrm{V}(Y_1) + \lambda w \mathrm{E}^2(Y_1) = \lambda w \mathrm{E}(Y_1^2) \\ &= \lambda w \frac{\mu(\mu^2\sigma^2 + 1)}{(\mu - \lambda)^3} = \frac{\rho}{(1 - \rho)^3} \frac{(\theta^2 + 1)}{\mu} w, \end{aligned} \quad (8)$$

where $\rho = \lambda/\mu$ is the workstation utilization rate, $\theta = \sigma\mu$ is the coefficient of variation of service. When the owner system can be approximated by an $M/M/1$ queuing system, then $\sigma = \mu^1$, $\theta = 1$, and

$$\mathrm{V}(T_k) = \frac{2\rho}{(1 - \rho)^3 \mu} w. \quad (9)$$

From (7) and (8), we may conclude the following regarding the parallel subtask completion time in a single nondedicated workstation environment:

- The mean and variance of subtask completion time are proportional to the workload of the subtask. Therefore, the main task in estimating parallel task completion time is to analyze the influence of the owner workstation utilization.
- The mean parallel subtask completion time is independent of service time variation and is the reciprocal of the workstation utilization.
- Further, from (7) and (8), we can easily find that the coefficient of variation of subtask complete time, $[\mathrm{V}(T_k)/\mathrm{E}(T_k)^2]$, goes to 0 as the parallel task time increases. The coefficient of variation is also positively related to the utilization of individual workstations. Further, we know from (8) that the increase in workstation utilization or variability in owner sequential job service will cause more variability in parallel subtask completion time.

The discussion now turns to parallel task completion time.

2.2 Parallel Task Completion Time

Assuming m workstations are used for parallel computing, T can be expressed as:

$$T = \mathrm{Max}\{T_k, k = 1, 2, \ldots, m\}. \quad (10)$$

Assuming the usages of different workstations are independent, the probability that parallel tasks finish within time t is equal to

$$\begin{aligned} \Pr(T \leq t) &= \Pr(\mathrm{Max}_{1 \leq k \leq m}\{T_k, k = 1, \ldots, m\} \leq t) \\ &= \prod_{k=1}^{m} \Pr(T_k \leq t). \end{aligned} \quad (11)$$

From (3),

$$T_k = w_k + Y_{k1} + Y_{k2} + \ldots + Y_{kS_k} = w_k + U(S_k), \quad (12)$$

where Y_{kj} is the jth busy period at workstation k,

$$U(S_k) = \begin{cases} 0, & \text{if } S_k = 0 \\ Y_{k1} + Y_{k2} + \ldots + Y_{ks_k}, & \text{if } S_k > 0. \end{cases} \quad (13)$$

As previously defined, we know that Y_i is the busy period random variable of the queuing system and S_k is a Poisson random variable with rate λw. However, from queuing theory, we know that it is difficult to find the exact distribution of the server busy time. Even for simple *M/M/1* queuing systems, the density function of a busy period can only be obtained through a complicated serial expression [7].

Note that $\Pr(S_k = 0) = e^{-\lambda w_k}$. The distribution of T_k is a combination of random variables.

$$\begin{aligned} \Pr(T_k \le t) &= \Pr(T_k \le t \mid S_k = 0)\Pr(S_k = 0) \\ &\quad + Pr(T_k \le t \mid S_k > 0)\Pr(S_k > 0) \\ &= \begin{cases} e^{-\lambda w_k} + (1 - e^{-\lambda 2_k)}\Pr(U(S_k) \le t - w_k \mid S_k > 0, & \text{if } t \ge w_k \\ 0, & \text{if } t < w_k. \end{cases} \end{aligned} \quad (14)$$

We need only to find the distribution of $U(S_k|S_k > 0)$.

Consequently, the probability that the parallel process finishes within time t is

$$\Pr(T \le t) = \begin{cases} \prod_{k=1}^{m}[e^{-\lambda w_k} + (1 - e^{-\lambda w_K}) \\ \Pr(U(S_k) \le t - w_k \mid S_k > 0)], & \text{if } t \ge w_{max} \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

where $w_{max} = \text{Max}\{w_k\}$. For a special case where the system has a uniform machine usage pattern and equally distributed workload, w, then

$$\Pr(T \le t) = \begin{cases} [e^{-\lambda w} + (1 - e^{-\lambda w})\Pr(U(S_1) \le \tau \mid S_1 > 0)]^m, & \text{if } \tau > 0 \\ 0, & \text{otherwise}, \end{cases} \quad (16)$$

where $\tau = t - w$.

If the distribution probability $\Pr(U(S_k) \le u|S_k > 0)$ can be identified, the distribution of parallel task completion time $\Pr(T \le t)$ can be calculated. The mean and the variance of the parallel task completion time can also be calculated. However, it is difficult, if not impossible, to come up with an explicit expression of $\Pr(U(S_k) \le u|S_k > 0)$ based on the existing result in probability. However, the probability may be approximated if we know the mean and standard deviation of the random variable and if we can approximate its distribution functions by using known ones. By using the following equations:

$$\begin{aligned} \text{E}(T_k) &= \text{E}(T_k|S_k > 0)\Pr(S_k > 0) + \text{E}(T_k|S_k = 0)\Pr(S_k = 0), \\ \text{V}(T_k) &= \text{V}(T_k|S_k > 0)\Pr(S_k > 0) + \text{V}(T_k|S_k = 0)\Pr(S_k = 0), \end{aligned}$$

and, from results $\text{E}(T_k = \frac{1}{1-\rho}w$, $\Pr(S = 0) = e^{-\lambda w}$, $\text{E}(T_k|S = 0) = w$, and $\text{V}(T_k|S = 0) = 0$, we have

$$\begin{aligned} \text{E}(T_k \mid S_k > 0) &= \frac{1}{1 - e^{-\lambda w}}\left[\frac{1}{1-\rho}w - we^{-\lambda w}\right] \\ &= w + \frac{1}{1 - e^{-\lambda w}}\frac{\rho}{1-\rho}w, \\ \text{V}(T_k \mid S_k > 0) &= \frac{1}{1 - e^{-\lambda w}}\text{V}(T_k) \\ &= \frac{1}{1 - e^{-\lambda w}}\frac{\rho}{(1-\rho)^3}\frac{(\theta^2 + 1)}{\mu}w. \end{aligned}$$

Therefore,

$$\text{E}(U(S_k) \mid S_k > 0) = \text{E}(T_k \mid S_k > 0) - w = \frac{1}{1 - e^{-\lambda w}}\frac{\rho w}{1-\rho} \quad (17)$$

and

$$\begin{aligned} \text{V}(U(S_k) \mid S_k > 0) &= \text{V}(T_k \mid S_k > 0) \\ &= \frac{1}{1 - e^{-\lambda w}}\frac{\rho}{(1-\rho)^3}\frac{(\theta^2 + 1)}{\mu}w. \end{aligned} \quad (18)$$

In the following section, we will discuss the possible approximation of the distribution function for $U(S_k)$ given $S_k > 0$.

3 Experimental Analysis of System with Homogeneous Nondedication

From the result of Section 2, we know that the explicit expression for the parallel job completion time (which is a random variable) is difficult to obtain. To approximate the distribution of parallel job completion time, it is important to find the distribution of $U(S_k)$ given $S_k > 0$ for each owner workstation. To achieve the goal of practical usefulness, we determine the unknowns through experimentation. We then integrate these experimentally determined results into our analytical formulations to form a complete performance model. Here, homogeneous nondedication implies that the machines on the network have a uniform machine usage pattern (the machines have the same mean and distribution of utilization and service rate).

This section is divided into two subsections. We first discuss the subtask finishing time for a single workstation in Section 3.1. We focus on the pattern of distribution of the subtask finishing time given different workstation service patterns and different utilization rates. We then discuss the parallel task finishing time in Section 3.2. This discussion generally focuses on the impact of total parallel task demand, the number of workstations, and other parameters of the parallel task completion time. Simulation results will also be compared with the analytical formulation derived in Section 2.

3.1 Simulation to Determine Single Workstation Task Completion Time Distribution

In this section, we use simulation to examine the distribution of $U(S_k)$ given $S_k > 0$ for a single workstation. We omit subscript k whenever there is no confusion. We have simulated a large number of examples with different parameters and sequential task patterns on the owner

workstation. Below, we describe our simulation procedure and report our findings from this analysis.

Without loss of generality, we assume that the arrival rate of the owner local job sequence is $\lambda = 1$. We simulate the distribution of parallel subtask completion time for three different system parameters

$$(\mathrm{W}, \rho, \theta) = (\text{total demand of the parallel task, system utilization, service coefficient of variation}).$$

The parameter set shown in Table 1 is used.

In addition, we consider five different service distributions for sequential tasks: 1) *Exponential* distribution, 2) *Erlang* distribution, 3) *Gamma* distribution, 4) *Log-Normal* distribution, and 5) *Truncated Normal* distribution. Note that the owner system variation is always 1 under the assumption of *M/M/*1. Experiments for different parameters are performed for each service distribution.

Prediction Simulations. One purpose of simulation is to evaluate the mean and standard deviation of the parallel job completion time. In order to generate meaningful simulation results, we use a simulation stopping rule suggested by Ross [15] whereby we can be $(1-\alpha)\%$ certain that the sample mean (the average of the simulated task finishing time) will not differ from the mean of the sampling distribution $\mathrm{E}[X_I]$ by more than 1.96δ (where δ is small). That is, we generate enough simulation runs *k* so that the standard deviation of the average sampling distribution, $\sigma/\sqrt{k}$, is less than the acceptable value δ. This stopping criterion guarantees, with 95 percent confidence, that our estimated answer will not differ from the true value by more than (0.05*(SAMPLE MEAN)) and (0.01*(SAMPLE VARIANCE)). A minimal simulation sample size of $k = 30$ is assumed so that the Sample Standard Deviation *S* can be used as a good approximation of σ. However, we note from the simulation experiments that the number of simulation runs required to achieve the desired precision ranged from over 3,000 to 20,000.

Curve Fitting Simulations. For each set of parameters, a series of simulation runs is conducted in order to fit a distribution for subtask completion time and parallel task completion time. This is essentially an exercise in curve fitting. We use the ARENA simulation system input processor [17] to fit the simulated task completion time to various theoretical distributions. The input processor attempts to fit one of 12 probability distributions to a set of raw data using either maximum likelihood estimators or the method of moments [6]. Information is provided regarding the value of the squared-error as well as various statistical tests: the chi-square test and the Kolmogorov-Smirnov (KS) goodness-of-fit tests expressed in terms of p-values. When generating (simulated) sample points to construct subtask completion time distributions, we ensure that enough observations are collected in each run so that a reasonable distribution fit can be made. Between 5,000 and 10,000 data points were collected to fit each distribution. In addition, the best-fit distribution is evaluated visually by superimposing each theoretical distribution over the raw sample distribution. Visual assessment is viewed by many as providing the best means of evaluation.

Simulation Results. In Fig. 1, a set of simulation outcome examples for task (subtask) completion time of a single workstation (in histograms) and corresponding fitting distributions (in lines) are shown.

A summary of the simulated task completion time results for a single workstation is given below.

- The simulation experiments indicate that Gamma, Lognormal, or Weibull are (almost always) among the best-fit distributions in approximating the distribution of $U(S)|S > 0$. It is difficult to determine which distribution is the best since the decision is input parameter sensitive. Among the simulation experiments we have performed, around 40 percent of them show that the Gamma distribution (which includes the exponential and Erlang) is the best fit; roughly one-third of them show that the Lognormal distribution is the best fit. However, this percentage may change if we select a different set of experimental parameters.
- We find that the owner workstation service distribution may affect subtask completion time distribution, especially when the ratio of subtask time to utilization rate is small and workstation utilization rate is high. For example, Gamma and Weibull distributions are likely the best-fit distribution if an exponential owner service time is assumed. On the other hand, Lognormal may become the best-fit distribution for Lognormal or truncated normal service distributions. However, such a phenomenon of service distribution sensitivity seems to weaken as the ratio of the demand of task time over workstation utilization rate increases. This phenomenon can be explained if we note two facts: 1) Different service distributions will result in different busy cycle distributions at a workstation and 2) the distribution of the job completion time will be effected more significantly if only a small number of the cycles are required.
- Statistically, we may conclude that the Gamma distribution may be a better approximation if the system utilization is low (5-15 percent) and the demand of the parallel subtask is reasonably long. The Weibull may become the best-fit distribution when the owner workstation utilization is medium high (20-50 percent) and the Lognormal distribution also appears frequently in the best-fit list when the owner workstation utilization is high (> 50 percent). However, comparing the distribution patterns of these three distributions, Gamma, Lognormal, or Weibull, we find that the difference among them is very small. Based on such a fact, we will simply use the Gamma distribution as the approximation of subtask completion time at each workstation in the following analytical analysis for parallel task completion time using multiple workstations. In Fig. 1, Gamma distribution fitting is demonstrated in all of our simulation examples.
- Under certain circumstances, such as extremely low workstation utilization, the (simulated) parallel

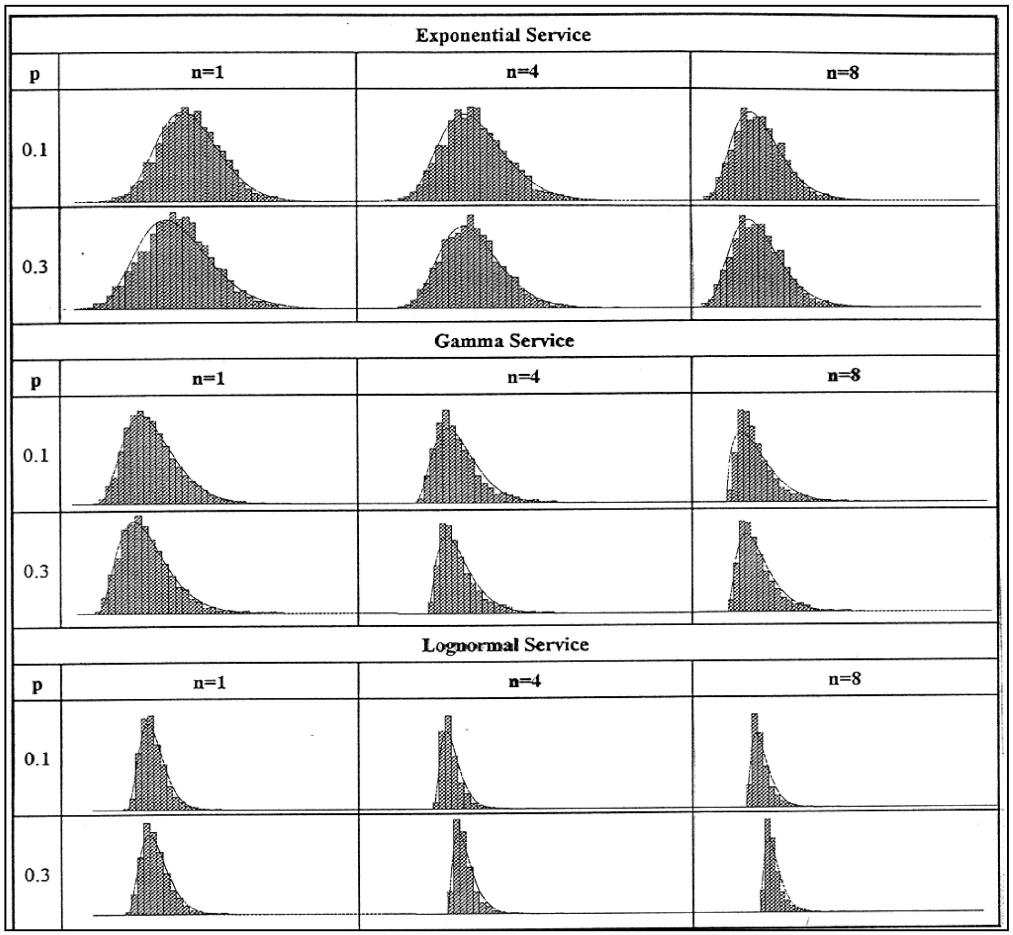

Fig. 1. Simulation for subtask completion time of a single workstation.

subtask completion time may be the same as the (simulated) demand. For instance, when there is no parallel task preempted by local sequential jobs the resulting distribution is hyper-exponential.

The above simulation observations are consistent with the general result of queuing theory where the stationary state job finishing time can be approximately Erlang distribution, which is a special case of Gamma Distribution.

3.2 Completion Time of Parallel Task

We now examine the parallel task completion time T for the multiple workstation case. The purpose is to discuss the effects of different system parameters on parallel task completion time. Using the probability expression (15) in Section 2.2, we are able to evaluate the distribution of parallel task completion time T analytically given that we know the distribution function of $U_k(S)|S_k > 0$. From the discussions in Section 3.1, we know that Gamma can be used to approximate the random variable $U_k(S)|S_k > 0$. In this section, we estimate the mean and standard deviation of parallel task completion time by assuming $U_k(S)|S_k > 0$ is Gamma distributed with mean and standard deviation given by (17) and (18). Correspondingly, we use simulation to verify the analytical results. In the simulation, we assume that the workstations are i.i.d. $M/G/1$ queuing systems with *Lognormal* distributed services times. Four parameters are examined in our computations: total demand of parallel task (W), coefficient of variation at each workstation (θ), the number of workstations (m), and workstation utilization rate (ρ). In order to graphically show the results of the parallel task completion time, two of the four parameters are discussed in each of the following examples (figures). The remaining parameters will be assumed to the values shown in Table 2 if not specified.

Note that the workstation subscriptions for the parameters are omitted since they are the same in a homogenous system.

3.2.1 Mean of Parallel Task Completion Time

Analytical results using (15) are demonstrated in Fig. 2, where $U_k(S)|S_k > 0$ is approximated by the Gamma distribution. The vertical axis in the figures represents the logarithm of the expected parallel task completion time with a base of 2, that is, $\log_2(\mathrm{E}(T))$.

Effect of service station variation. Fig. 2a, Fig. 2b, and Fig. 2c provide the mean of parallel task completion time for different coefficient of variations of service time, different workstation utilization rates, and other parameters. The coefficient of variations of service distribution seems to have very little effect on parallel task completion time T when the workstation service rate is uniformly low. Therefore, the service station coefficient of variation can be ignored when the workstation utilization is low. This conclusion is consistent with the conclusion drawn from

TABLE 2
Parameters

w = 64 hour	m = 8	ρ = 0.2	θ = 4

the research by Kleinrock and Korfhage [9] where Brownian motion is used to approximate the parallel task completion time. However, when the owner workstation utilization is high, Fig. 2a indicates that the CV will have an effect on completion time that should not be ignored.

Effect of Total Demand of Parallel Task: For the given system parameters, we can expect that the parallel task completion time will increase when the total demand of the parallel task increases. The relationship between completion time and total demand of parallel task can be seen in Fig. 2b, Fig. 2d, and Fig. 2f. Our computational analysis clearly indicates that the relation between the expected parallel task completion time T and the total demand of the parallel task (though it seems concave down slightly) can be virtually approximated by a linear function, regardless of system utilization and service distribution. Following this conclusion, given the same environment, we can easily estimate the parallel task completion time as long as we know the total demand of the parallel task. This will help us to make certain decisions such as whether more workstations are necessary to expedite the task completion time or whether migration to another workstation with lower utilization is desirable. This result is also consistent with the findings by Kleinrock and Korfhage's results [9].

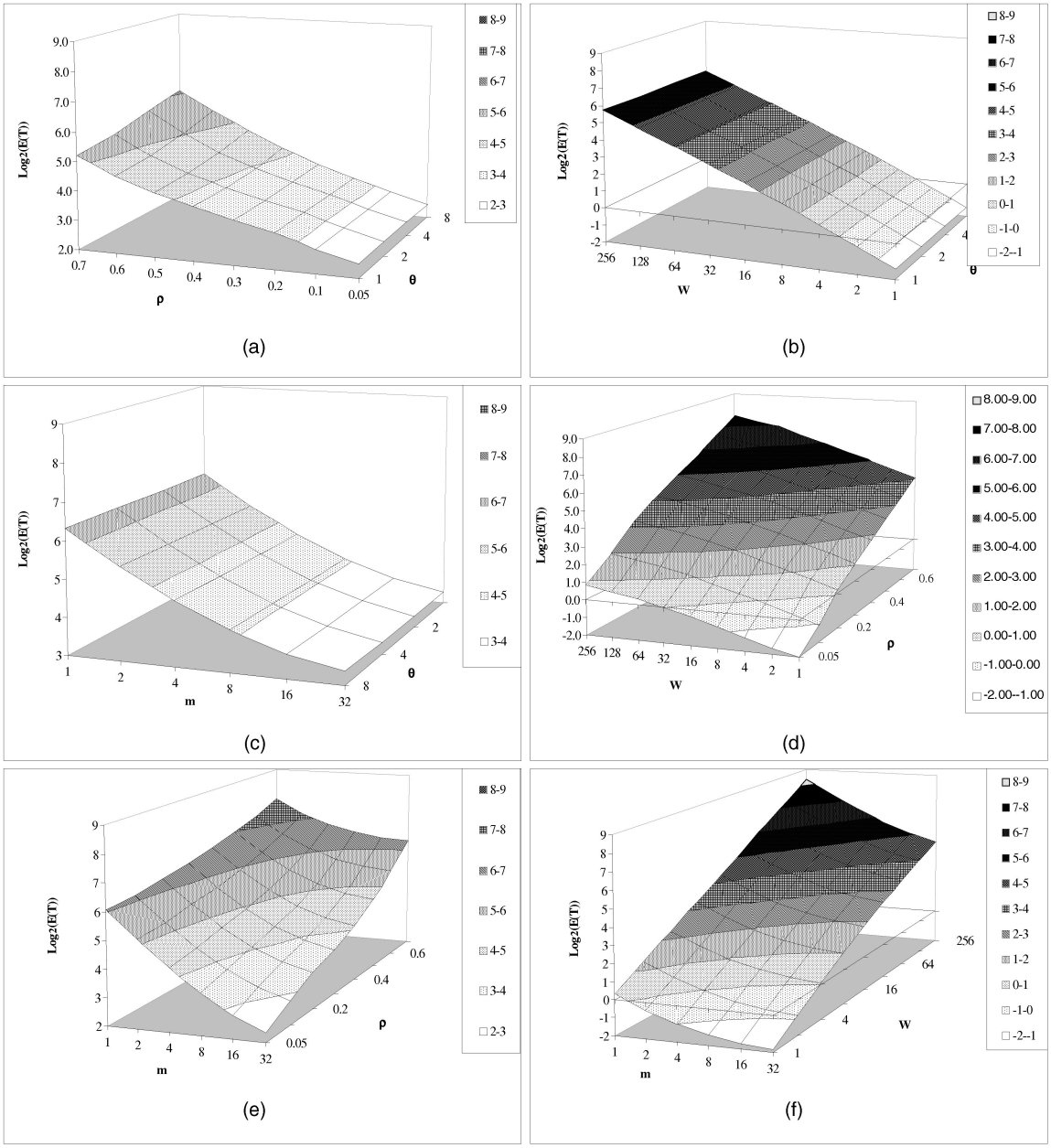

Fig. 2. Expected parallel task completion time. (a) m = 8, W = 64. (b) m = 8, $\rho = 0.2$. (c) W = 64, $\rho = 0.2$. (d) m = 8, $\theta = 4$. (e) W = 64, $\theta = 4$. (f) $\theta = 4$, $\rho = 0.2$.

Effect of the Number of Workstations: The pattern of completion time for a different number of workstations is given in Fig. 2c, Fig. 2e, and Fig. 2f. We can see marginal parallel completion time reduction as the number of workstations increases. Such a phenomenon clearly indicates the importance of choosing the proper number of workstations. In fact, our computational experiment shows that the use of an excessive number of workstations may even contribute negatively to parallel task completion time. This performance decrease is not a direct result of Amdahl's law [1], [16], that is, it is not caused by insufficient parallelisms. It is caused by the growing variation in the combination of subtask completion times resulting from increasing the number of workstations. More specifically, the positive contribution of each additional workstation may reduce as the number of workstations increases, whereas each additional machine may add negative influence on the parallel task completion time due to additional variation to the system. Task ratio, the ratio of the size of a parallel subtask and the mean and variance of the local sequential job service rate, is an important factor in determining the parallel subtask finish time [11].

Effect of Workstation Utilization: Fig. 2a, Fig. 2d, and Fig. 2e illustrate the relationship between the parallel task completion time and workstation utilization. We observe a near linear relationship between $\log_2$ (parallel completion time) and workstation utilization when the utilization is relatively low, less than 0.3 in our examples. In other words, a simple exponential (with base 2) function can be used to approximate the relationship between the parallel task completion time and the workstation utilization rate.

However, it is important to point out that the relationship between workstation utilization and parallel completion time becomes fuzzy when the workstation utilization increases beyond some high level. Our simulation results indicate that the parallel completion time will become less predictable when the workstation utilization rate is high (> 60 percent). Process migration migrates parallel process (subtask) from highly utilized machines to lowly utilized machines. The relation given here confirms that process migration is essential for nondedicated network computing to provide expected high performance.

In order to verify the results from our analytical model, simulations are performed for the examples discussed. Fig. 3 demonstrates the relative errors of the simulated and predicted parallel task completion time. The vertical axis in the figures represents the relative error $(\frac{\text{simulation-analytical}}{\text{simulation}})$. We observe that our analytical approximation and simulation results are reasonably close, especially when the workstation utilization is low and the number of workstations is not very large. Under the circumstances, the predicted parallel completion time tends to be a little higher than that from simulation. However, the result may be reversed as the workstation utilization and the number of workstations increases (see Fig. 3e). It is also interesting to observe that the difference between the simulation and analytical results is not sensitive to total workload (W) when workstation utilization rate is low ($\rho \leq 0.30$) or when service variation is low ($\theta \leq 2.0$). See, for instance, Fig. 3d and Fig. 3b, respectively.

Our simulation results confirm that, with the Gamma distribution as the subtask completion time at each workstation, (16) provides an adequate performance prediction for nondedicated homogeneous computing, when the machine utilization is reasonable (< 60 percent).

3.2.2 STD of Parallel Task Completion Time

Fig. 4 demonstrates the relationship between the standard deviation and system parameters. The vertical axis represents the logarithm of the standard deviation of the parallel task completion time with a base of 2, that is $\log_2(\mathrm{STD}(T))$.

Effect of variation of workstation utilization. Combining Fig. 4a, Fig. 4b, and Fig. 4c, we find that, although the variation of service rate has only a minor effect on the variation of parallel task completion time when the system utilization is low, a very significant effect may result when the workstation utilization rate increases (see Fig. 4a). This again confirms the importance of process migration in nondedicated computing.

Effect of demand of parallel task time. From Fig. 4b, Fig. 4d, and Fig. 4f, we find that the standard deviation of the parallel completion time seems to have a very similar pattern as that of mean parallel completion time. It is close to a linear function of total demand of the parallel task (refer also to Fig. 3b, Fig. 3d, and Fig. 3f). Note that the vertical axis is the logarithm of the standard deviation of the parallel task completion time. This result indicates that the standard deviation of parallel task completion time seems more volatile than the result from Kleinrock and Korfhage [9], where they derived a formula showing that standard deviation of parallel task completion time is proportional to the squared root of the subtask finish time.

Effect of Workstation Utilization. Similar to the mean of parallel task completion time, the workstation utilization has quite a significant influence on the standard deviation of task completion time. Therefore, it is important to make a proper task allocation and migration if workstations have high utilization, especially when we have nonuniform workstation utilization.

Effect of the Number of Workstations. As the number of workstations increases, from Fig. 4b, Fig. 4d, and Fig. 4f, we observe that the standard deviation of the parallel task completion time decreases, although such an influence seems insignificant.

The analytical results for the standard deviation of parallel task completion time are also compared with the simulation results. When we use the same relative error measurement $(\frac{\text{simulation-analytical}}{\text{simulation}})$ as with the mean parallel task completion time, we find that the errors are, in general, less than 20 percent. Our simulation results indicate that our analytical results slightly underestimate the standard deviation of parallel completion time, particularly when the workstation utilization rate and the coefficient of variation are high. This diversion does not influence the applicability of the newly proposed model since parallel processes would be migrated from highly utilized machines to underloaded machines in practical engineering environments.

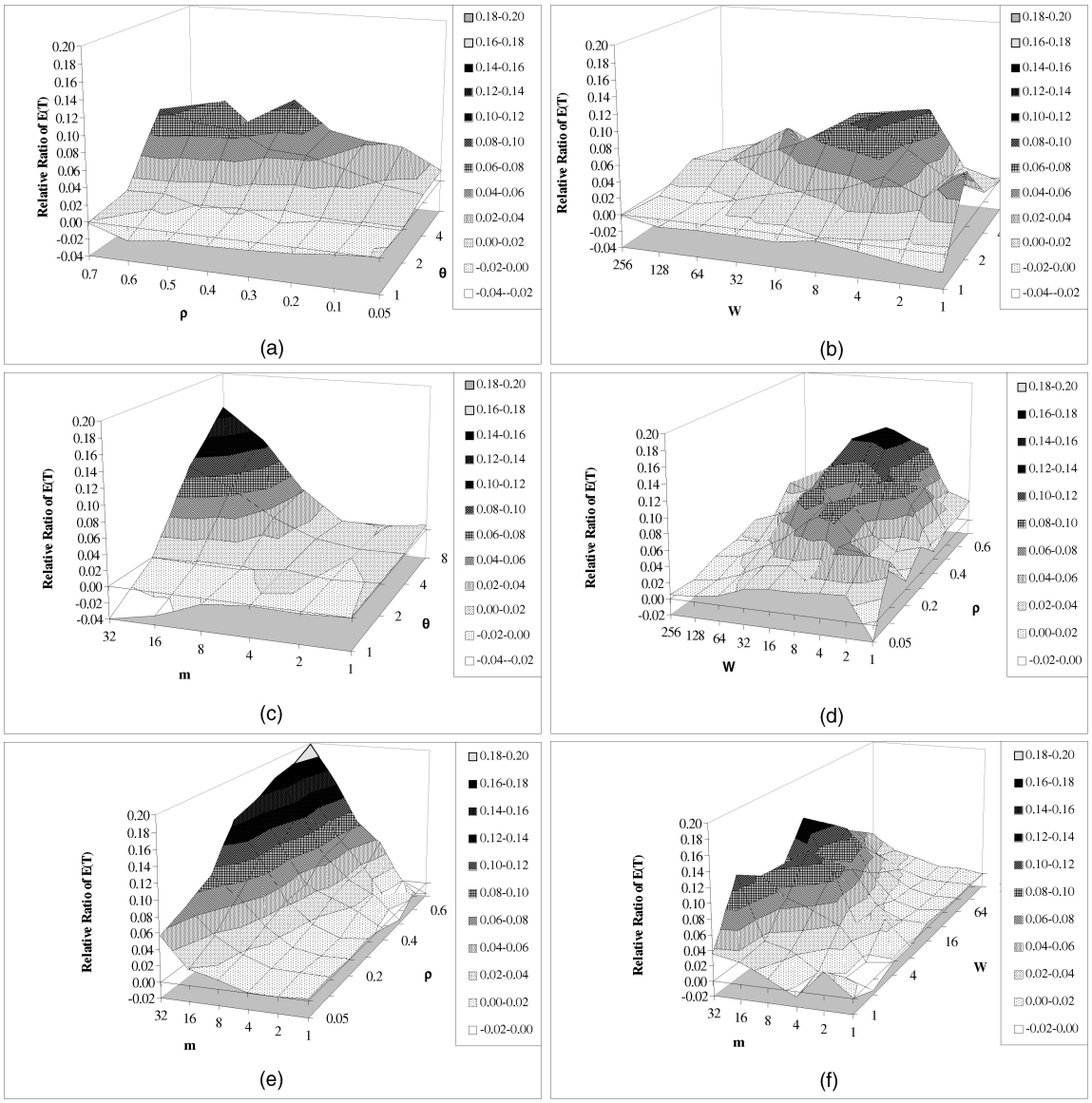

Fig. 3. Relative error of E(T) between analytical and simulation result. (a) m = 8, W = 64. (b) m = 8, $\rho = 0.2$. (c) W = 64, $\rho = 0.2$. (d) m = 8, $\theta = 4$. (e) W = 64, $\theta = 4$. (f) $\theta = 4$, $\rho = 0.4$.

4 Heterogeneous Nondedication System

Optimal parallel task partitioning for homogeneous computing seems straightforward, equal-load partitioning. This will not be true for heterogeneous systems. In fact, a primary concern in a heterogeneous environment is how to partition the parallel tasks and allocate the subtasks to workstations to achieve maximum performance. In this section, we study distributed systems with heterogeneous utilization, service rates, and distribution patterns across different workstations.

For a given parallel subtask with demand w_k, from Section 2, we have the expression of mean and standard deviation $\mathrm{E}(T_k), \mathrm{V}(T_k)$ of the subtask completion time at each workstation *k*. From our discussion of homogenous systems, we know that the mean subtask completion time and workstation utilization have a greater influence on the final parallel completion time than on the variation of service distribution. Hence, it is natural to partition the parallel task W into subtasks with workload w_k for workstation *k* such that the same mean subtask completion time can be reached at different workstations. Let *a* be the value of such a mean of the subtask completion time. We call such a partition approach mean time balancing partition. Note that, from (7), the expected subtask completion time is $\frac{w_k}{1-\rho k}$. The subtask workload w_k for the *k*th workstation is determined by equation

$$\frac{w_k}{1-\rho_k} = a \quad \text{or} \quad w_k = a(1-\rho_k). \tag{19}$$

Note that the total demand of the parallel task is *W* which yields $a = W/(m - \sum_{k=1}^{m} \rho_k)$. Hence,

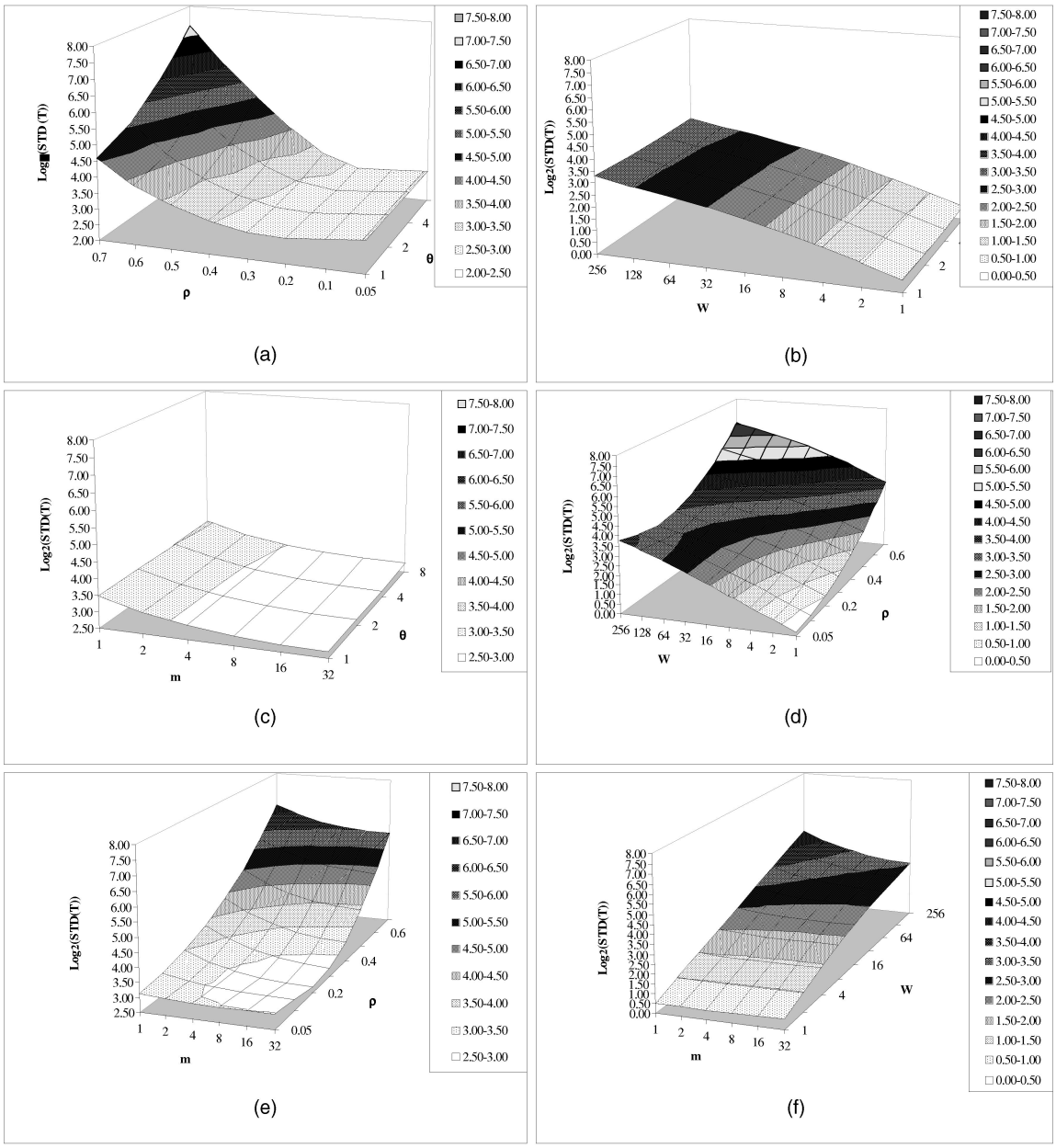

Fig. 4. STD of parallel task completion time. (a) m = 8, W = 64. (b) m = 8, Rho = 0.2. (c) W = 64, $\rho = 0.2$. (d) m = 8, $\theta = 4$. (e) W = 64, $\theta = 4$. (f) $\theta = 4$, $\rho = 0.2$.

$$w_k = \frac{W(1-\rho_k)}{m - \sum_{k=1}^{m} \rho_k} = \frac{W}{m}\frac{1-\rho_k}{1-\bar{\rho}}, \tag{20}$$

where $\bar{\rho} = \frac{1}{m}\sum_{k=1}^{m} \rho_k$, is the average system utilization. The corresponding mean and variance of subtask completion time at workstation k, from (8), is

$$\mathrm{E}(T_k) = \frac{w_k}{1-\rho_k} = \frac{W}{m(1-\bar{\rho})} \tag{21}$$

and

$$\mathrm{V}(T_k) = \frac{\rho_k}{(1-\rho_k)^3}\frac{(\theta_k^2+1)}{\mu_k} w_k = \frac{\rho_k}{(1-\rho_k)^2}\frac{(\theta_k^2+1)}{\mu_k}\frac{W}{m(1-\bar{\rho})}. \tag{22}$$

The parallel task partition rule (20) is focused on balancing the mean subtask completion time at workstations (referred to by the authors as the *mean time balance partition*). Identical average subtask completion time partition seems to be the best partition for dedicated heterogeneous systems. Unfortunately, this partition may not guarantee optimality in nondedicated heterogeneous systems since, from (22), the variation of the subtask complete time may be different at different workstations. However, it is encouraging to see from (22) that the above partition approach not only balances the mean subtask completion time at each workstation but also results in a smaller variation of the subtask completion time than that of the equal-load partition. That is, this partition approach results in a higher variance for workstations with lower than the

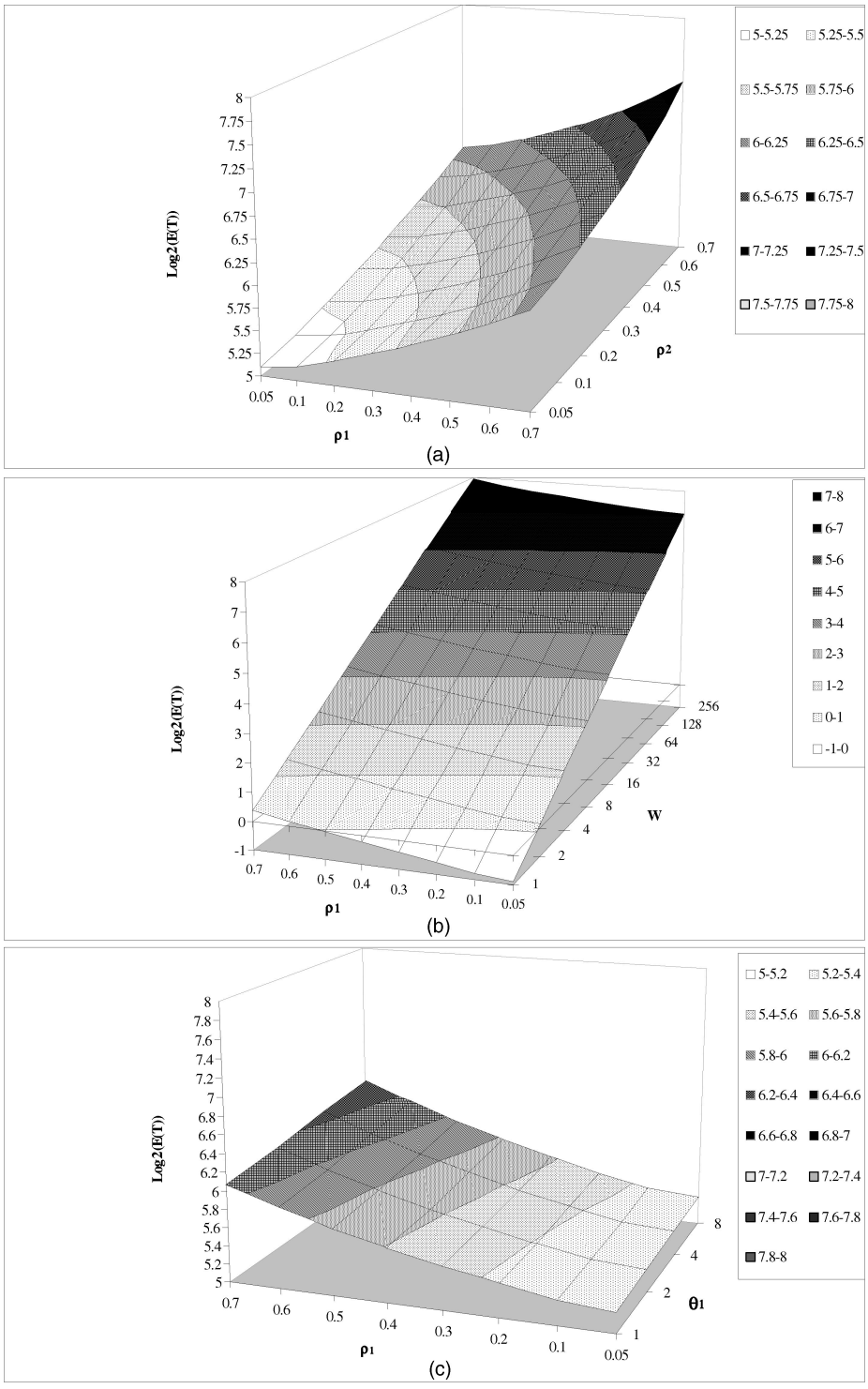

Fig. 5. E(T) of a two-workstation system with the mean-time balance partition. (a) W = 64, $\theta_i = 2$. (b) $\rho 2 = 0.2$, $\theta_i = 2$. (c) $\rho 2 = 0.2$, $\theta_2 = 2$, W = 64.

average utilization rate $\bar{\rho}$ and lower variances for workstations with higher utilization rate than $\bar{\rho}$.

For the mean-time partition approach described above, we simulated the system for the two-workstation case. In the following discussion, we use default values $\rho_i = 0.2$, $\theta_i = 4$, $i = 1, 2$, and $W = 64$. Fig. 5 shows the predicted performance under the mean-time partition. We compare the performance of mean time balance partition with the equal-load partition illustrated in Fig. 6. In the equal-load partition approach, we divide the total demand of the parallel task into m equal subtasks for each workstation. The vertical axis of Fig. 6 is the relative ratio of the parallel task completion time between the two different partitions

$$\frac{\mathrm{E}(T|equal\ partition) - \mathrm{E}(T|mean\ balance\ partition)}{\mathrm{E}(T|equal\ partition)}.$$

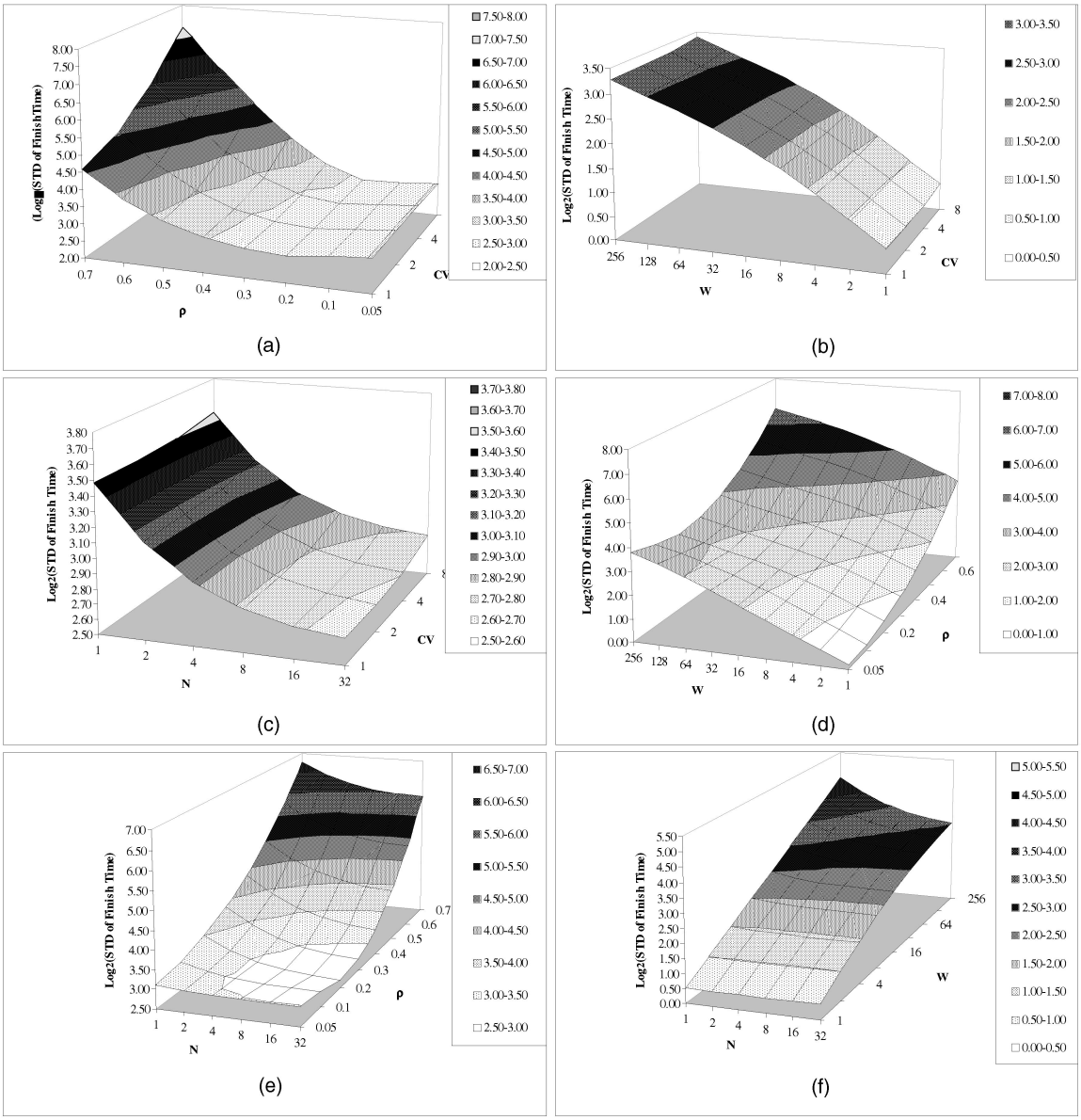

Fig. 6. Relative error of E(T) between mean time balance and equal load partition. (a) W = 64, $\theta_i = 2$. (b) $\rho 2 = 0.2$, $\theta_i = 2$. (c) $\rho 2 = 0.2$, $\theta_2 = 2$, W = 64.

From Fig. 5a, we find that the difference in workstation utilization does not have a significant influence on the parallel task completion time when the mean time balance partition is used. However, as indicated by Fig. 6a, with the equal-load partition, significantly more time is required to complete the parallel task when the difference of the utilization rate between the workstations increases. It is not difficult to predict that the performance difference between these two partition strategies will continue to increase as the number of workstation increases. As shown in Fig. 5b and Fig. 5c, with mean time balance partition, the rate of change of the mean parallel completion time becomes quite moderate as the one of the workstation utilization increases. This is because the increase in the utilization rate of one workstation is actually shared by both workstations. However, from Fig. 6b and Fig. 6c, we observe that such a sharing effect would not exist for the equal-load partition approaches.

Fig. 7 demonstrates the standard deviation of the parallel task completion time when the mean balance partition is used. It seems that the relationship between the base 2 logarithm of standard deviation of the parallel task completion time and the system parameters may be approximated by a linear relationship.

5 Main Results

Based on the analytical and experimental results given in the last three sections, we propose the following procedure to serve as a guideline in choosing the best number of workstations for parallel processing in a heterogeneous environment. Since a homogeneous system is a special case of a heterogeneous system, it can apply to homogeneous systems as well:

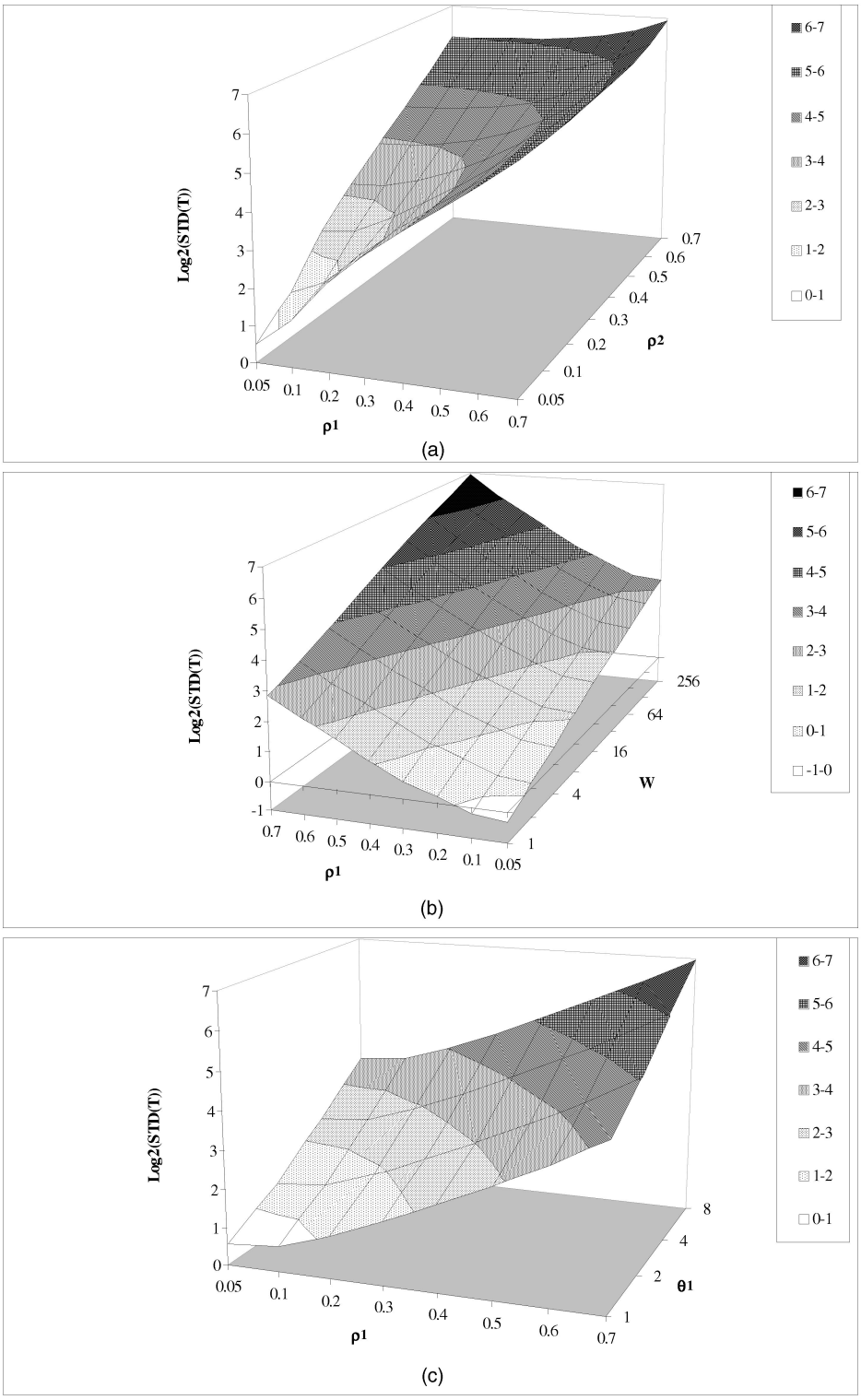

Fig. 7. STD of a two workstation system with the mean-time balance partition. (a) W = 64, $\theta_i = 2$. (b) $\rho 2 = 0.2$, $\theta_i = 2$. (c) $\rho 2 = 0.2$, $\theta_2 = 2$, W = 64.

Procedure *(Degree of Parallelism):*

1. *Start from a list of idle machines that are lightly loaded over an observed time period.*
2. *For the given number of workstations to be used, use the partition strategy equation (20) to partition and allocate subtasks to each workstation.*
3. *Predict the mean and STD of parallel task completion time via (14), where* Gamma function can be used to approximate the random variable $U_k(S)|S_k > 0$.
4. *Repeat Steps 2 and 3 with different numbers of workstations to identify the best number of workstations that should be used under the distributed environment.*

This procedure consists of two parts: workload distribution based on (20) and performance prediction using (14).

Equations (20) and (14) are the two main results of this study. They can be used together to form the above automatic partition procedure. They also can be used separately for load partition and performance prediction, respectively.

It is important to notice that we assume the parallel applications belong to the class of programs that can run efficiently in a dedicated parallel-computing environment. We have not considered the effects of synchronization, communication, process migration, or granularity of parallelism. While, in a heterogeneous environment, some of these overheads can be implicitly included in the service time of the workstations, this study, however, does not intend to provide an accurate performance prediction for a given application. Given that the program executes efficiently in a dedicated system, we wish to provide a guideline for the feasibility and limitation of parallel processing in a nondedicated distributed environment. By considering parallel applications with no parallel overhead, we provide an upper bound on the expected execution time for a given workload. This upper bound is general. It is the upper bound of the distributed system and applicable to any parallel applications. On the other hand, it only serves as an upper bound. The actual performance of a nonperfect parallel application could be far below the upper bound.

The machine utilization and sequential job service rate may vary over time in a nondedicated environment. Some distributed monitor tools may be needed to use (14) and (20) appropriately. Be prepared to migrate parallel processes from highly used machines to lightly loaded machines during runtime when it is necessary.

A tacit assumption of the above procedure is that the parallel task can be partitioned freely into small pieces. If the parallel task has limited parallelism and has inherited function/data dependence, then the load partition will be more involved, which is beyond the scope of this study.

6 Conclusion

In this paper, we present a performance model to study the feasibility and limitation of parallel processing in a nondedicated distributed environment, where parallel task has a lower priority than local sequential jobs. We assume that the parallel application can achieve the perfect speedup in a dedicated environment and assume sufficient idle machines are available in the distributed environment at any given time. Based on this model, we derive a performance prediction formula which provides the performance upper bound and a measure of feasibility of parallel processing of a nondedicated distributed environment. Several interesting findings have been observed. For instance, we find that the variations of sequential job service rate seem to have very little effect on parallel task completion time when the service rate is uniformly low. There is a near linear relation between $\log_2$ (parallel completion time) and workstation utilization when the utilization is relatively low. A procedure is also proposed to identify the number of workstations that should be used for a given parallel workload.

The primary technique used in the model development is a combination of queuing theory and simulation experiments. Stochastic analysis provides a queuing model, whereas the parameters are determined through simulation experiments. Through this combined approach, we attempt to present a simple and effective model.

The analytical approximation for parallel task completion time given in this paper is fairly close to the simulation result from our experiments. While we conclude that machine utilization rates of workstations have a dominant affect on the parallel job completion time, we also find that the variation of machine utilization has a very significant effect on distributed parallel processing. When the variation of machine utilization is very high, the performance of nondedicated systems becomes unpredictable.

Process migration is identified as an essential mechanism for nondedicated systems. Given that process migration is necessary and feasible (and considering the migration cost, live variable analysis, and the variation of machine utilization) determining when and where to migrate is still an unsolved issue. This, we believe, is a topic worthy of further investigation.

Acknowledgments

This research was supported in part by US National Science Foundation grants ASC-9729215 and CCR-9972251 and the Navy PET/Logicon.

References

[1] R. Arpaci, A. Dusseau, A. Vahdat, L. Liu, T. Anderson, and D. Patterson, "The Interaction of Parallel and Sequential Workloads on a Network of Workstations," *Proc. ACM SIGMETRICS/Performance Conf.*, May 1995.

[2] T. Anderson, D. Culler, and D. Patterson, "A Case for Networks of Workstations: NOW," *IEEE Micro,* Feb. 1995.

[3] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann, 1999.

[4] G. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, *PVM: Parallel Virtual Machine—A Users' Guide and Tutorial for Networked Parallel Computing*. The MIT Press, 1994.

[5] A.S. Grimshaw, W.A. Wulf, and the Legion Team, "The Legion Vision of a Worldwide Virtual Computer," *Computer,* vol. 40, no. 1, pp. 39-45, Jan. 1997.

[6] B.S. Gottfried, "Use of Computer Graphics in Fitting Statistical Distribution Functions to Data Representing Random Events," *Simulation,* vol. 60, no. 4, pp. 281-287, 1993.

[7] D. Gross and C.M. Harris, *Fundamentals of Queuing System,* second ed. John Wiley & Sons, 1985.

[8] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* The MIT Press, 1994.

[9] L. Kleinrock and W. Korfhage, "Collecting Unused Processing Capacity: An Analysis of Transient Distributed Systems," *IEEE Trans. Parallel and Distributed Systems,* vol. 4, no. 5, May 1993.

[10] S. Leutenegger and X.H. Sun, "Distributed Computing Feasibility in a Non-Dedicated Homogeneous Distributed system," *Proc. Supercomputing '93,* pp. 143-152, 1993.

[11] S. Leutenegger and X.H. Sun, "Limitations of Cycle Stealing of Parallel Processing on a Network of Homogeneous Workstations," *J. Parallel and Distributed Computing,* pp. 169-178, Oct. 1997.

[12] S. Madala and J.B. Sinclair, "Performance of Synchronous Parallel Algorithms with Regular Structures," *IEEE Trans. Parallel and Distributed Systems,* vol. 1, no. 1, Jan. 1991.

[13] M. Mutka and M. Livny, "The Available Capacity of a Privately Owned Workstation Environment," *Performance Evaluation,* vol. 12, pp. 269-284, 1991.

[14] G. Peterson and R. Chamberlain, "Stealing Cycles: Can We Get Along?" *Proc. 28th Hawaii Conf. System Science,* pp. 422-431, Jan. 1995.

[15] S.M. Ross, *Simulation,* second ed. Academic Press, 1997.

[16] X.H. Sun and J. Gustafson, "Toward a Better Parallel Performance Metric," *Parallel Computing*, vol. 17, pp. 1093-1109, Dec. 1991.
[17] Systems Modeling Corp., "ARENA User's Guide," version 2.0, Sewickley, Pa., 1995.

Linguo Gong received the doctorial degree from the University of Texas at Austin. He is an associate professor in the Department of Management Sciences at Rider University. His research interests are in quantitative modeling and analysis, especially in the areas of supply chain management, quality management, production planning, and control. His publications have appeared in *Management Science*, *Decision Sciences*, *Naval Research Logistic*, IIE transactions, *European Journal of Operations Research*, etc. Prior to joining Rider, he worked at several universities, including work as an associate professor in the Department of ISDS at Louisiana State University and as a visitor in the department of MSIS at Rutgers University.

Xian-He Sun received the PhD degree in computer science from Michigan State University. He was a staff scientist at ICASE, NASA Langley Research Center and was a tenured associate professor in the Computer Science Department at Louisiana State University (LSU). Currently, he is a professor and the director of the Scalable Computing Software Laboratory in the Computer Science Department at the Illinois Institute of Technology (IIT) and a guest faculty member at the Argonne National Laboratory. Dr. Sun's research interests include parallel and distributed processing, software system, performance evaluation, and scientific computing. He has published intensively in the field and his research has been supported by DoD, DoE, NASA, NSF, and other government agencies. He is a senior member of the IEEE, a member of the ACM, New York Academy of Science, Phi Kappa Phi, a partner of the Esprit IV APART (Automatic Performance Analysis: Resources and Tools) working group, and has served and is serving as the chairman or as a member of the program committee for a number of international conferences and workshops. He received the ONR and ASEE Certificate of Recognition award in 1999 and the Best Paper Award from the International Conference on Parallel Processing (ICPP01) in 2001.

Edward F. Watson has industrial engineering degrees from Syracuse (BS) and Pennsylvania State University (MS and PhD). He is an associate professor in the Department of Information Systems and Decision Sciences at Louisiana State University (LSU). His research interests and major publications are in simulation modeling and analysis techniques, process analysis and engineering, and enterprise information systems. His publications appear in *Decision Sciences*, *European Journal of Operations Research*, *Decision Support Systems*, *Interfaces*, and *International Journal of Production Research*. He is a member of DSI, AIS, and ICIS. Prior to joining LSU, he had more than six years work experience with Systems Modeling Corporation and General Motors and has consulted on productivity and capacity issues for Xerox, Whirlpool, PPG, Tennant, Coca-Cola, and Gates Rubber.