Modeling and Characterizing Parallel Computing Performance on Heterogeneous Networks of Workstations *

XIAODONG ZHANG YONG YAN

High Performance Computing and Software Laboratory
The University of Texas at San Antonio
San Antonio, Texas 78249

Abstract

A heterogeneous network of workstations (NOW) introduces a new performance factor into distributed computing: a large variation of the computing power of the different workstations. This unique factor makes traditional performance models/metrics for homogeneous computing measurement and evaluation not suitable for heterogeneous computing. In this paper, we present models which quantify the heterogeneity of networks and characterize the performance effects. The models consider effects of both the heterogeneity and time-sharing in a nondedicated environment. Speedup, efficiency and scalability are defined. These models are general enough to cover performance evaluation of both homogeneous and heterogeneous computations in dedicated and nondedicated NOW systems. To validate and support performance modeling results, we conducted a collection of experimental measurements for evaluating computing performance and scalability of a group of application programs on a heterogeneous NOW.

1 Introduction

A network of workstations (NOW) is highly cost effective, and widely available. Network computing utilizes a large number of idle cycles to run parallel jobs. A heterogeneous NOW introduces a new performance factor into distributed computing: a large variation of the computing power of the different workstations. This unique factor makes traditional performance models/metrics for homogeneous computing performance measurement and evaluation not suitable for heterogeneous computing. We will study and address this performance issue in this paper.

In practice, people are more interested in two performance results in scientific computing: the execution times when an application program size is fixed, and when it is scaled. In both cases, the system size is scaled. The fixed-size performance studies how fast the application problem of a given size can be processed by increasing the size of the system, while the scaling performance (scalability) measures the ability of a parallel system to improve performance as the size of the application problem and the size of the system increase. In a homogeneous computing environment, metrics for fixed-size performance are the definitions of speedup and the efficiency from Amdahl's law. There are also several performance metrics for parallel computing scalability measurements, such as the latency metric [9]. All these metrics evaluate parallel performance by comparing sequential computation on one processor node as a reference base. In contrast, such an identical reference base does not exist in a heterogeneous computing environment.

Homogeneous computing is considered as a special case of heterogeneous computing. Thus heterogeneous computing performance models/metrics should be general enough to cover performance evaluation of both types. Several discussions about heterogeneous program speedup definitions have been published, e.g. [3, 4, 5, 6]. The definitions in [3, 6] combine the computing features of both types, where the speedup of a heterogeneous computation is defined as the ratio of the time of a program running on the fastest machine to the time of the program running across the heterogeneous network. This definition is suitable for general heterogeneous computing and also is consistent with the standard speedup definition on a homogeneous system. However, network heterogeneity and its effects have not been quantitatively modeled and analyzed. In addition, other related performance definitions, such as superlinear speedup, efficiency and scalability, need to be formally defined for heterogeneous network computing.

Different heterogeneous systems serve different computing purposes. For example, the heterogeneous systems described in [5] exploit different types of parallelisms from different types of multicomputers connected by a network. Our work focuses on performance issues of a heteroge-

---

*This work is supported in part by the National Science Foundation under research grants CCR-9102854 and CCR-9400719, and under instrumentation grant DUE-9250265, by the U.S. Air Force under research agreement FD-204092-64157, and by the Air Force Office of Scientific Research under grant AFOSR-95-1-0215.

Proceedings of the 7th IEEE Symposium on Parallel and Distributeed Processing (SPDP '95)
1063-6374/95 $10.00 © 1995 IEEE

neous NOW. The concepts may also be extended to evaluate other types of heterogeneous systems. In this paper, we present models to quantify the heterogeneity of workstations and characterize the performance. Speedup, efficiency and scalability of heterogeneous computing are defined. To validate and support performance modeling results, we conducted experimental measurements for evaluating parallel computing performance and scalability of a group of scientific application programs on a heterogeneous NOW.

In practice, a heterogeneous NOW system is often a nondedicated system. Hence, performance metrics for heterogeneous computing should consider the effects of both heterogeneity and time-sharing. The organization of this paper is as follows. Section 2 presents basic considerations of heterogeneous computing. Heterogeneous performance evaluation models/metrics are introduced in Section 3. Section 4 describes our heterogeneous NOW environment. In Section 5, we report comprehensive measurement results for heterogeneous computing performance and scalability evaluation. Finally, we summerize the paper in Section 6.

2 Heterogeneous computing models

2.1 Heterogeneous network model

A heterogeneous network (HN) can be abstracted as a connected graph $HN(M, C)$, where

- $M = \{M_1, M_2, ..., M_m\}$ is set of heterogeneous workstations (m is the number of workstations). The computation capacity of each workstation is determined by the power of its CPU, I/O and memory access speed.
- C is a standard interconnection network for workstations, such as an Ethernet or an ATM network, where the communication links between any pair of the workstations have the same bandwidth.

Based on the above definition, if a NOW consists of a set of identical workstations, the system is homogeneous. Moreover, a heterogeneous network can be divided into two classes: a dedicated system where each workstation is dedicated to execute tasks of a parallel job, and a nondedicated system where each workstation executes its normal routines (also called owner workload), and only the idle CPU cycles are used to execute tasks of the parallel job. The term of *owner utilization* of a workstation is used to represent the owner workload rate. Our program execution model assumes that each workstation may execute at most one task of the parallel job. This assumption is consistent with the programming principle in writing a PVM program.

2.2 Heterogeneous programming model

A parallel program, denoted as $A(I)$, where I represents its input-parameter, is assumed to have m tasks $A_1(I)$, $A_2(I), ..., A_m(I)$. Task $A_i(I)$ $(1 \leq i \leq m)$ is assigned and executed on workstation M_i. The size of program $A(I)$ is denoted as $|A(I)|$, which can be defined as the number of required operations to solve $A(I)$ [9]. Thus, $|A(I)| = \sum_{i=1}^{m} |A_i(I)|$. It will simplify notation significantly if we assume the remainder of this paper that the program parameters are included for each case. Thus, when we write A, we really mean $A(I)$.

Let $S_i(A)$ be the speed of workstation M_i to solve A dedicatedly. We add a practical constraint to characterize our target heterogeneous network system: the speed $S_i(A)$ is a constant for a given application program A. Because most operations of many application programs are performed consistently on a class of workstations with different speeds, such as the Sun workstations. We will show experimentally from our case studies on a heterogeneous NOW that the computation speed remains constant in each workstation. The unit of the speed is the average number of operations of different types per second for executing a program.

In order to avoid complicated measurements of the speed, we define a power weight $W_i(A)$ for running program A on workstation W_i as follows:

$$W_i(A) = \frac{S_i(A)}{\max_{j=1}^{m}\{S_j(A)\}} \quad i = 1, ..., m \tag{1}$$

Formula (1) indicates that the power weight of a workstation refers to its computing speed relative to the fastest workstation in a system. The value of the power weight is less or equal to 1. Since the power weight is a relative ratio, it can also be represented by measured execution time. If we define the speed of the fastest workstation in the system as one basic operation per time unit, the power weight of each workstation denotes a relative speed. If $T(A, M_i)$ gives the execution time for executing program A on workstation M_i, the power weight can be calculated by the measured execution times as follows:

$$W_i(A) = \frac{\min_{i=1}^{m}\{T(A, M_i)\}}{T(A, M_i)} \tag{2}$$

3 Performance models/metrics

3.1 Fixed-sized applications

3.1.1 Speedup

Speedup is used to quantify the performance gain from a parallel computation of a fixed-size application over its computation on a single machine in a heterogeneous network system. The speedup of a heterogeneous computation is defined as follows.

Definition 1 *The speedup for executing program A, denoted by $SP(A)$ on a dedicated or a nondedicated heterogeneous network $HN = \{M_1, M_2, \cdots, M_m\}$ is*

$$SP(A) = \frac{\min_{j=1}^{m}\{T(A, M_j)\}}{T(A, HN)}, \tag{1}$$

where $T(A, HN)$ is the total execution time for A on HN, and $T(A, M_j)$ is the execution time for A on workstation M_j $j = 1, ..., m$.

Definition 1 is also used in [3, 6] for a dedicated heterogeneous system.

The network heterogeneity can be quantified by variance of computing powers in a network system. This quantitative expression should be formed carefully because the heterogeneity of a network not only reflects the variance of computing powers but also the dynamic system effects caused by the changes of critical system components, such as the fastest workstation and slowest workstation. The variance of a set of sample data, $d_1, d_2, ..., d_m$, can be either measured by standard deviation,

$$V_{stand} = \sqrt{\frac{\sum_{j=1}^{m}(\bar{d} - d_j)^2}{m}}, \quad (2)$$

or by mean absolute deviation,

$$V_{mean} = \frac{\sum_{j=1}^{m}|\bar{d} - d_j|}{m}, \quad (3)$$

where $\bar{d} = \sum_{j=1}^{m} d_j/m$ is the average of the sample date set. Both methods treat every data observation evenly and use the average value of the data set as the base reference for comparisons to calculate the variance. If the disparity among the workstation power weights is not significant, either deviation method could be a reasonable measure of the heterogeneity. When the disparity is large, for example, a small number of workstations are much faster or slower than the majority of the workstations, the heterogeneity is considered high. Unfortunately, the high heterogeneity in this case could not be well expressed by (2) or (3). Let's take a NOW of 100 nodes as an example, where the fastest workstation has the power weight of 1, and the rest of 99 workstations have the identical power weight of 0.5. The speed of the fastest workstation is then doubled, which makes the power weight of the fastest workstation still 1, but 0.25 for the rest of the 99 workstations. Using the standard deviation or the mean absolute deviation to calculate the variance of the power weights in both the initial network configuration and the final configuration, we obtain less than 1% change between the two variances of the power weights. However, the speedup of a heterogeneous computation would decrease nearly 50% assuming the computation distributions keep unchanged. This is because the sequential execution on the fastest workstation decreases 50% and the distributed heterogeneous computing time would keep roughly the same. Thus, the standard deviation or the mean absolute deviation could not reasonably and precisely capture the critical system change.

Another method of calculating the variance is to set the speed of the slowest workstation as the base reference for comparisons. This does not match our principle of using the speed of the fastest workstation in the network as the base reference for performance evaluation, such as the speedup. Thus, the most suitable way to quantify the heterogeneity is to use the power weight of the fastest workstation as the comparison reference. Using the power weight of the fastest workstation (=1), we define the network heterogeneity as follows.

Definition 2 *The network heterogeneity is defined as:*

$$H = \frac{\sum_{j=1}^{m}(1 - W_j(A))}{m}. \quad (4)$$

Applying Definition 2 to the previous example, we obtain $H = 0.5$ for the initial network configuration, and $H = 0.75$ for the final configuration. The heterogeneity increases 50% after the speed of the fastest workstation is doubled.

Using a single number to present a complex performance factor, such as the heterogeneity in (4), is direct and simple to understand, but it has its limits. For a given heterogeneity, multiple systems with different workstation combinations could be formed. Each system may have a different power distribution although the heterogeneity is the same. However, the heterogeneity presented in (4) does reflect an average balance of the power distributions.

To further observe load balance effects, we define the average parallelism degree of an application program, denoted as P_{deg}, which is the average number of workstations actively executing in parallel in the whole execution phase of the program. This is measured by accumulating the percentage of the active computing time for the parallel program in each workstation:

$$P_{deg}(A) = \frac{\sum_{j=1}^{m} T_j^{act}}{T(A, HN)} \quad (5)$$

where T_j^{act} is the active computing time on each workstation ($j = 1, ..., m$), and $T(A, HN)$ is the total parallel computing time across the network. Figure 1 presents an example to explain the definition of P_{deg}, where $T(A, HN)$ in each workstation is divided into two major parts: the active computing time for the parallel job and the idle time. The idle time is also used for the owner workload computing. Again, to simplify the notation, when we write S_i, W_i, SP and P_{deg}, we really mean $S_i(A)$, $W_i(A)$, $SP(A)$ and $P_{deg}(A)$. We present the following heterogeneous speedup property.

Property 1 *The speedup of a fixed-size application A on a dedicated/nondedicated NOW is:*

$$SP = P_{deg} \times (1 - H), \quad (6)$$

where P_{deg} is the parallelism degree, and H is the network heterogeneity defined in (4).

Proof: The execution time for A on workstation $M_i (1 \leq i \leq m)$ is

$$T(A, M_i) = \frac{|A|}{S_i}, \quad (7)$$

Proceedings of the 7th IEEE Symposium on Parallel and Distributeed Processing (SPDP '95)
1063-6374/95 $10.00 © 1995 IEEE

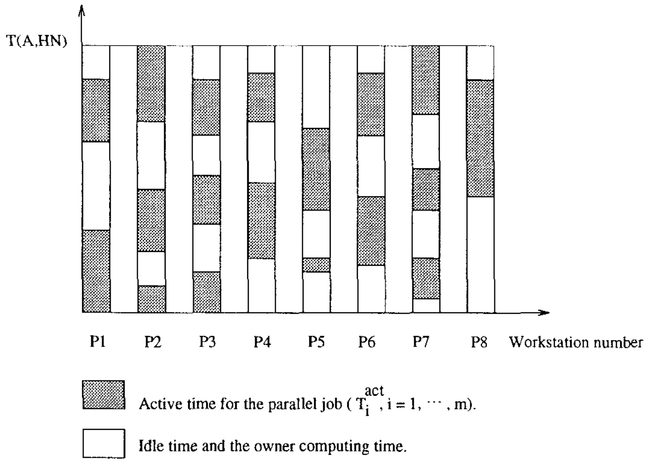

Figure 1: Parallelism degree is measured by accumulating the percentage of the active computing time for the parallel program in each workstation.

where $|A|$ is the size of A, and S_i is the speed of each workstation. We use parallelism degree P_{deg} and the speed of each workstation S_i $(i = 1, ..., m)$, to define an average speed of a heterogeneous NOW:

$$\bar{S} = \frac{P_{deg} \times (S_1 + S_2 + \cdots + S_m)}{m}. \tag{8}$$

Using the average network speed, $\bar{S}$, parallel execution time across the network for solving A can be expressed as

$$T(A, HN) = \frac{|A|}{\bar{S}} = \frac{m|A|}{P_{deg} \sum_{j=1}^{m} S_j}. \tag{9}$$

By Definition 1, (7) and (9), the speedup becomes

$$SP = \frac{\min_{j=1}^{m}\{\frac{|A|}{S_j}\}}{\frac{m|A|}{P_{deg}\sum_{j=1}^{m} S_j}} = \frac{P_{deg}\sum_{j=1}^{m} S_j}{m \times \max_{j=1}^{m}\{S_j\}} = P_{deg} \times (1 - H).$$

End of Proof.

For a given program structure, the program's parallelism degree, P_{deg}, is determined by the network heterogeneity, H. Although the alternative form of the speedup in (6) is in its high level evaluation, it provides several observations for understanding and characterizing performance of a heterogeneous NOW:

- For a given heterogeneous NOW, The speedup of a parallel computation increases if the parallelism degree P_{deg} increases. This observation follows homogeneous computing performance principles.
- The speedup decreases if the heterogeneity H increases assuming the parallelism degree keeps unchanged or decreases.
- The speedup increases if the heterogeneity H decreases, assuming the parallelism degree keeps unchanged or increases.
- The heterogeneity also introduces some uncertain speedup performance predictions which are application dependent. For example, the speedup could either increase or decrease if the parallelism degree of a program increases from increasing the heterogeneity.

3.1.2 Efficiency

Efficiency is a measure of the time percentage for which a machine is usefully employed in parallel computing. In homogeneous computing, it is simply defined as the ratio of speedup to the number of processors. In heterogeneous computing, the efficiency is more complex. A unique feature in a heterogeneous system is that unit times in different types of machines are represented by different computing powers. Thus, to directly use the efficiency definition from homogeneous computations for heterogeneous computations would not precisely characterize how different types of machines are effectively used in the network. It is necessary to average available cycle times using the defined power weight. For an application $A = \{A_1, A_2, \cdots, A_m\}$ in a heterogeneous network $HN = \{M_1, M_2, \cdots, M_m\}$, the effective execution time of A_j $(1 \leq j \leq m)$ on workstation M_j is $T_{eff}(A_j, M_j) = |A_j|/S_j$, where S_j is the speed of M_j. Let T_j^{owner} be the time of machine M_j executing the owner workload, and let *overhead*(j) be the overhead latency time on workstation M_j. For detailed descriptions of obtaining the overhead latency, readers may refer to [9] and Section 3.2 in this paper. We define the efficiency by using the effective and available computing times which are directly related to the heterogeneity of a network system.

Definition 3 *The efficiency of parallel computing of application A on a nondedicated heterogeneous NOW is defined as the ratio of the total effective computing time to the total available cycle time in the system:*

$$E = \frac{\sum_{j=1}^{m}(W_j \times |A_j|/S_j)}{\sum_{j=1}^{m}(T(A, HN) - T_j^{owner})W_j}, \tag{10}$$

where $|A_j|/S_j$ represents the CPU busy time needed to execute A_j on machine M_j. A practical form to represent the total effective computing time is $\sum_{j=1}^{m}\{T(A, HN) - T_j^{owner} - overhead(j)\}W_j$. In order to further give the relationship among speedup, efficiency and system workload, we define

$$\Delta_A = \frac{|A|/\max_{j=1}^{m}\{S_j\}}{\min_{j=1}^{m}\{T(A, M_j)\}}, \tag{11}$$

where $|A|/\max_{j=1}^{m}\{S_j\}$ and $\min_{j=1}^{m}\{T(A, M_j)\}$ represents the fastest execution time of A on a single workstation in a dedicated system and a nondedicated system, respectively. Formula (11) quantifies the effect of workload in a nondedicated system to the single machine execution time for A. In addition, $\Delta_A \leq 1$. By Definition 1 and Definition 3, the relationship among speedup, efficiency and system workload (or owner utilization, defined as $U_i = \frac{T_i^{owner}}{T(A,HN)}$ $(1 \leq i \leq m)$) are derived:

Property 2 *The relationship between the efficiency and the speedup for application program A in a nondedicated heterogeneous NOW system can be expressed as:*

$$E = \frac{SP \times \Delta_A}{\sum_{j=1}^{m}(1-U_j)W_j}. \tag{12}$$

Proof: By (10),

$$\begin{aligned} E &= \frac{\sum_{j=1}^{m}(W_j \times |A_j|/S_j)}{\sum_{j=1}^{m}(T(A,HN)-T_j^o)W_j} \\ &= \frac{|A|/\max_{j=1}^{m}\{S_j\}}{T(A,HN)\sum_{j=1}^{m} W_j - \sum_{j=1}^{m} T_j^o W_j} \\ &= \frac{SP \times (|A|/\max_{j=1}^{m}\{S_j\})}{\min_{j=1}^{m}\{T(A,M_j)\} \times \sum_{j=1}^{m}(1-U_j)W_j} \\ &= \frac{SP}{\frac{\sum_{j=1}^{m}(1-U_j)W_j}{\Delta_A}} \\ &= \frac{SP \times \Delta_A}{\sum_{j=1}^{m}(1-U_j)W_j}. \end{aligned}$$

End of Proof.

Besides describing the relationship between the efficiency and the speedup, Property 2 includes both the power weight and the owner untilization of each machine to characterize the effects of heterogeneity and time-sharing.

In formula (12), $\sum_{j=1}^{m}(1-U_j)W_j$ represents an average amount of available time units in a heterogeneous NOW system, which decreases as the owner workload increases. The efficiency of heterogeneous computing equals to 1 if and only if

$$SP = \frac{\sum_{j=1}^{m}(1-U_j)W_j}{\Delta_A}. \tag{13}$$

The superlinear speedup appears when $SP > \frac{\sum_{j=1}^{m}(1-U_j)W_j}{\Delta_A}$.

Formula (12) is a general form of parallel computing efficiency which considers effects of both heterogeneity and time-sharing. For a dedicated heterogeneous computing system, the relationship in formula (12) is simplified as follows under the condition of $U_j = 0$ $(1 \leq j \leq m)$ and $\Delta_A = 1$:

$$E = \frac{SP}{\sum_{j=1}^{m} W_j}. \tag{14}$$

Formula (14) indicates that if the speedup is larger than $\sum_{j=1}^{m} W_j$, the system computing power, the computation presents a superlinear speedup in a dedicated heterogeneous NOW system. Furthermore, substituting $U_j = 0$ and $W_j = 1$ for a dedicated homogeneous system, the definition of speedup and efficiency returns to the conventional form:

$$E = \frac{SP}{m}.$$

3.2 Scalability evaluation

Scalability measures the ability of a parallel machine to improve performance as there are increases in the size of the application problem and in the size of the system involved. In a heterogeneous network system, the system size can not simply be described by the number of machines as it can in a homogeneous system, because machines have different computing power. We give a practical definition of scalability as follows.

Definition 4 *Scalability is a property which exhibits performance linearly proportional to the computing power of the employed heterogeneous network system.*

Definition 4 indicates that the scalability of an application in a nondedicated system can be measured only if the owner utilization of each machine does not change over the computation time. In our following discussion we assume that each machine in a heterogeneous network system has a constant owner utilization. Here, we extend the latency based scalability metric proposed in [9] for heterogeneous computing scalability evaluation.

The execution time running program A on a heterogeneous network HN, can be divided into three distinct parts on each workstation $M_i, (i = 1, ..., m)$:

1. $T_{\text{eff}}(A_i, M_i)$: the effective time to compute task A_i on M_i excluding the CPU idle time units and the time units spent on work which is not needed in a single machine, such as synchronization time, communication time, and task creation time.
2. T_i^{owner}: the computing time for the owner workload, which includes all the CPU busy and idle times for the same duty.
3. *overhead*(i): overhead latency, which includes all communication time, memory access delay in the heterogeneous computation, and load unbalanced idle time. Formally, for $i = 1, ..., m$, it is expressed as

$$overhead(i) = T(A, HN) - T_i^{owner} - T_{\text{eff}}(A_i, M_i).$$

We quantify average overhead latency by using the power weights as follows:

$$L_{ave}(A, HN) = \sum_{j=1}^{m} overhead(j) W_j. \tag{15}$$

In order to capture the changes in the overhead cost of a parallel computation as a system increases its computing power, we introduce average latency per computational power unit.

Definition 5 *For a parallel program A, the average latency per computational power unit (simplified as average latency unit) in a heterogeneous network system HN is defined as*

$$L(A(I), HN) = \frac{L_{ave}(A(I), HN)}{\sum_{j=1}^{m} W_j} = \frac{\sum_{j=1}^{m} overhead(j) W_j}{\sum_{j=1}^{m} W_j}. \tag{16}$$

By Definition 5, the scalability is defined as follows.

Definition 6 *For a given program $A(I)$, let $L_e(A(I_1), HN_1)$ be the average unit latency for executing the program of size $|A(I_1)|$ on heterogeneous network HN_1, and let $L_e(A(I_2), HN_2)$ be the average unit latency for executing the program of size $|A(I_2)|$ on heterogeneous network HN_2 with larger computing power than HN_1. If the problem size is scaled from $|A(I_1)|$ to $|A(I_2)|$, the system size is scaled from HN_1 to HN_2, and efficiency is kept a constant $e \in [0, 1]$, the scalability is defined as*

$$scale_e(A, (HN_1, HN_2)) = \frac{L_e(A(I_1), HN_1)}{L_e(A(I_2), HN_2)}. \qquad (17)$$

In practice, the value of (17) is less than or equal to 1. A large scalability value of (17) means small increments in latencies inherent in the program and the architecture for efficient utilization of an increasing number of processors, and hence the parallel system is considered highly scalable. On the other hand, a small scalability value means large increments in latency and therefore a poorly scalable system. For a constant efficiency, the relationship among scalability, problem size and system computing power is described by the following property.

Property 3 *Let W^1 and W^2 be the total computing powers of heterogeneous system HN_1 and HN_2 respectively, and S^1 and S^2 be the speeds of the fastest workstations on HN_1 and HN_2 respectively. If the average unit overhead size is measured with respect to the fastest workstation, the scalability defined in (17) has the following alternative form:*

$$scale_e(A, (HN_1, HN_2)) = \frac{|A(I_1)| \times W^2 \times S^2}{|A(I_2)| \times W^1 \times S^1} \times \sigma,$$

where $\sigma = (1 + \frac{T^{owner}_{HN_2} - T^{owner}_{HN_1} \times |A(I_2)|/|A(I_1)|}{L_{ave}(A(I_2), HN_2) \times S^2})$, $|A(I_1)|$ and $|A(I_2)|$ are the problem sizes of A, $|T^{owner}_{HN1}|$ and $|T^{owner}_{HN2}|$ are the owner workload sizes on HN_1 and HN_2 respectively, and $L_{ave}(A(I_2), HN_2)$ is the average overhead latency of $A(I_2)$ executing on HN_2.

Proof: Let e_1 and e_2 be the efficiencies of $A(I_1)$ and $A(I_2)$ executing on HN_1 and HN_2, respectively. By the efficiency definition, we have

$$e_1 = \frac{|A(I_1)|}{|A(I_1)| + L_{ave}(A(I_1), HN_1) \times S^1 + |T^{owner}_{HN_1}|}, \qquad (18)$$

$$e_2 = \frac{|A(I_2)|}{|A(I_2)| + L_{ave}(A(I_2), HN_2) \times S^2 + |T^{owner}_{HN_2}|}. \qquad (19)$$

Substitute $e_1 = e_2$ into equations (18) and (19), we have

$$\frac{L_{ave}(A(I_1), HN_1)}{L_{ave}(A(I_2), HN_2)} = \frac{|A(I_1)| \times S^2}{|A(I_2)| \times S^1} \times \sigma \qquad (20)$$

where $\sigma = (1 + \frac{T^{owner}_{HN_2} - T^{owner}_{HN_1} \times |A(I_2)|/|A(I_1)|}{L_{ave}(A(I_2), HN_2) \times S^2})$. Combining (20) with (17), Property 3 is obtained.

End of Proof.

The above scalability property indicates that the scalability of a computation of two problem sizes using two heterogeneous NOW systems is reversely proportional to the scaled problem size ($|A(I_2|$), and positively proportional to the scaled system size (W^2). The owner workload also negatively affect the scalability of the program running on a heterogeneous NOW.

In practice, the measurement of scalability is dependent on scaling methods. The system size can be scaled in two directions: by increasing the number of workstations (physical scaling), and by upgrading workstation powers (power scaling). In physical scaling, a parallel job tends to increase average overhead latency due to more workstations involved in communications and synchronizations. In power scaling, communication complexity remains constant, but the computation and communication ratio increases, which would reduce the computation efficiency. How to balance the two scaling methods to achieve optimal performance for a parallel job is a rather complex issue.

4 Experimental environment

4.1 A heterogeneous network system

Our heterogeneous network system consists of three SPARC 10-30 workstations, five SPARC 5-70 workstations and five SPARCclassic workstations, connected by an Ethernet of 10Mbps bandwidth. The network has maximal throughput of 1000KB/s and minimal latency on the order of 2 milliseconds.

4.2 Application programs

We select five programs from three application areas. Programs Kernel EP (EP) [1], matrix multiplication (MM) and Cholesky factorization (Cholesky) [2] are selected from numerical applications, edge detection (ED) [8] from image processing, and merge sorting (MS) from basic data structure applications. Each algorithm is parallelized and implemented by PVM [7] on the heterogeneous workstation network.

4.3 The power weight and workload

The computing power weight defined in Section 2 provides an average performance reference to reflect the heterogeneity of network systems. The power weight a combination measurement of a program and a NOW system. We have looked into the effects of memory size and cache size of the NOW on the execution of the selected five applications.

In our experiments, the execution timing results of each application on the network system were measured by using different problem sizes. The performance is directly

Application	SPARC5	SPARC10	SPAECclassic
EP	1	0.91	0.39
EF	1	0.96	0.55
MM	1	0.83	0.48
Cholesky	1	0.91	0.48
MS	1	0.82	0.63

Table 1: Computing power weights of the three types of workstations for the five application programs.

related to the cache size and memory size in each machine. For each application, we divide the problem data domain into six subdomains so that data size in one subdomain corresponds to a data memory size in one of the six memory regions: [0, 2KB], (2KB, 8KB], (8KB, 16KB], (16KB, 8MB], (16MB, 32MB] and (32MB, ∞]. Here, 2KB, 8KB and 16KB are the data cache sizes in SPARCclassic workstations, SPARC 5-70 workstations and SPARC 10-30 workstations. In addition, 16MB is the memory size for SPARCclassic and SPARC 5-70, and 32MB for SPARC 10-30. Then, based on formula (2), the power weights can be measured for each type of workstation by the five applications. All the experiments were repeated 10 times in a dedicated system. Table 2 gives the average computing power weights of the heterogeneous workstations for the five applications when the data size in each application is less than 16MB, which is the memory limit of the SPARCclassic machines and the SPARC 5-70 machines. The standard deviations of multiple measurements are less than 0.05.

We have made the following three observations from these experiments:

1. The computing power weights remained unchanged if the data size of an application was bounded in one of the six memory regions.
2. As the data size increased beyond the cache size of any type of machines in the network, the computing power weights changed slightly.
3. When the data size of an application is larger than the main memory size of a workstation, the execution time was sharply increased, which in turn causes a significant decrease in the computing power weight of the workstation. This is because the dominant computation times were used by the operating system to do context switch and page swapping between main memory and disks. Taking the MM program as an example, when the matrix size is 1500 × 1500, the data size is 27MB which is larger than 16MB of the workstation and less than 32MB. When the MM was executed on one SPARCclassic computer or SPARC 5-70, the execution time is about 17 hours. But on SPARC 10-30, the execution time was only about 30 minutes. So, the relative computing power weight of the SPARCclassic and SPARC 5-70 workstations to SPARC 10-30 is 0.03. When the data size of MM is further increased to exceed 32MB, the execution of MM on the SPARC 10-30 also became unacceptable. In order to provide insight into system effects, we measured the execution time distributions of MM on one dedicated SPARC 5-70 workstation using two different matrix sizes: 512×512 and 1500×1500. The required memory allocation is 12 MB for the 1000×1000 matrix which is within the memory bound, and 27 MB for 512 × 512 which exceeds the memory bound. The execution time distributions were sampled every 10 minutes and averaged. For the matrix of size 1000 × 1000, 95% of CPU time was used for the multiplication. But, for the matrix of size 1500 × 1500, only 10% of CPU time was used for the multiplication, 70% was used by I/O, and 20% was used by the operating system to do context switches. Our experiments indicate that memory access is an important performance bottleneck for large memory-bound applications.

Based on our experimental results, we conclude that an application program should be partitioned and mapped onto a heterogeneous network system in such a way that the memory requirement of a task does not exceed the memory bound of any machine in the system, in order to keep powers weights constant. Our experiments also validated the assumption given in Section 2 on the constant workstation speed for a given application program within the machine memory bound. In all our experiments, we restricted the problem size so that the memory allocation of each task is less than 16MB. Hence, the power weights measured in Table 1 can be used to evaluate the relative computing capacities among the three types of workstations.

In order to analyze the effects of owner workload, we simulate a uniformly distributed computation workload on each workstation in the network by using a system clock interrupt primitive. For a utilization u, $10(1-u)$ seconds in every 10 seconds was used to compute the parallel task and $10u$ seconds was used to simulate the computation of owner workload. The computation of the owner workload did not occupy any main memory.

5 Performance evaluation

5.1 Fixed-size program performance

We fixed the problem size of each program and scaled the system computing power either by increasing the number of workstations (physical scaling), or by upgrading the workstations' power (power scaling). Bounded by the memory limit of $16MB$, the size of the MM program was fixed to a 1000 × 1000 matrix, the MS program to 5×10^5 elements, the EF program to 1500 × 1500 image pixels, the Cholesky program to a 1000 × 1000 matrix, and the EP program to 2^{21} iterations. Using physical scaling, the number of workstations was scaled from 2 up to 12. The

Proceedings of the 7th IEEE Symposium on Parallel and Distributeed Processing (SPDP '95)
1063-6374/95 $10.00 © 1995 IEEE

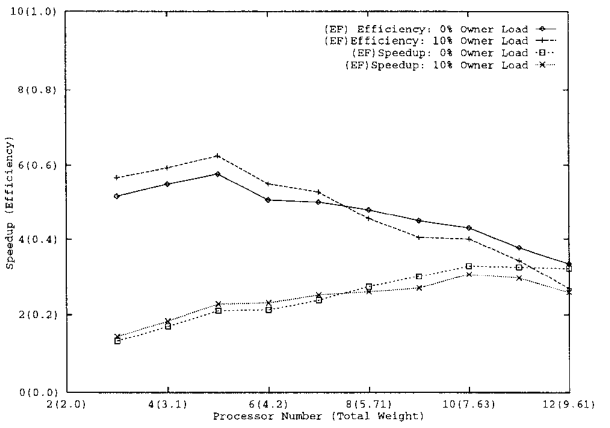

Figure 2: Fixed-size performance for EF.

system initially had two SPARC 5-70 workstations. Then a SPARCclassic was added one by one (5 total), three SPARC 10-30 workstations were added next, and finally two SPARC 5-70 workstations were added. Using power scaling, the number of processors was fixed to 8 in the system, where the combination of workstations was changed in order to adjust the total computing power. The initial system combination had three SPARC 10-30 workstations and five SPARCclassic workstations. The system power was scaled by replacing one slow SPARCclassic from a SPARC 5-70. This power scaling was done until each of the SPARCclassics was replaced by a SPARC 5-70.

Using the two system scaling methods, we measured the speedup and efficiency of each application in a dedicated environment and a nondedicated environment with an owner workload utilization of 10%. The owner task in each workstation had preemptive priority over the heterogeneous tasks.

The performance results are shown in Figures 2 to 6. The performance results of each application are presented from the physical scaling only due to the limited space. The performance results in Figures 2 to 6 plot speedup and efficiency curves in a dedicated environment and a nondedicated environment together. Carefully studying these curves, we obtain some interesting performance results and implications which are described below.

5.1.1 Comparative performance

In comparing the speedup and efficiency curves in the dedicated environment using the physical scaling, Kernel EP had an increasing speedup curve up to about 8 on the 12 workstations which had the computing power of 8.7. The decrease in the efficiency of Kernel EP was less than 20%. In contrast, the speedups of the other four programs were all less than 4. The least efficiency among them decreased more than 20%. The reason for the performance difference mainly comes from the communication pattern

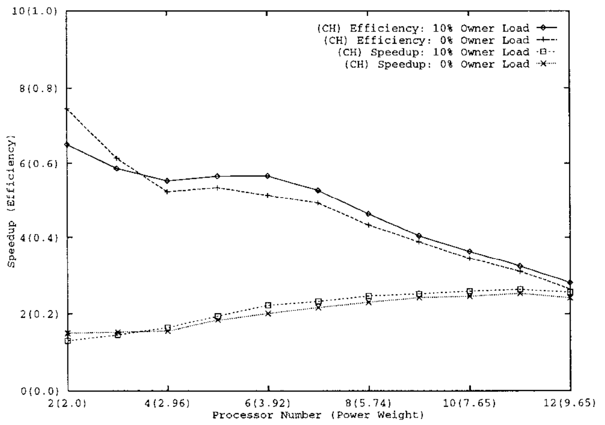

Figure 3: Fixed-size performance for Cholesky Factorization.

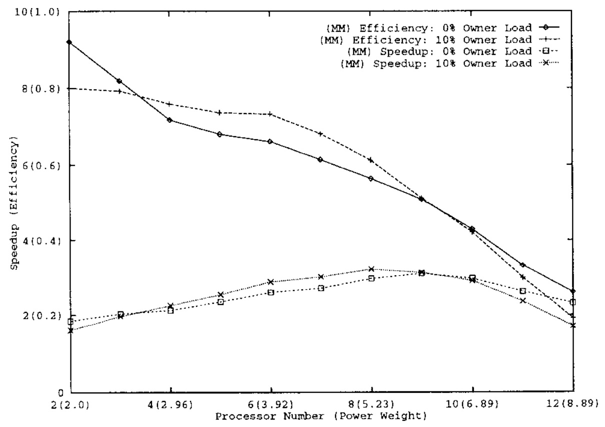

Figure 4: Fixed-size performance for MM.

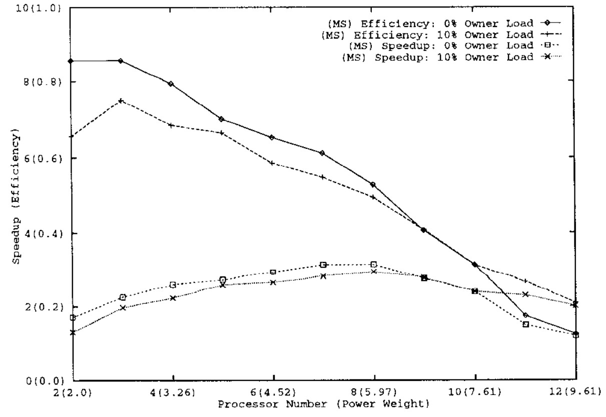

Figure 5: Fixed-size performance for MS.

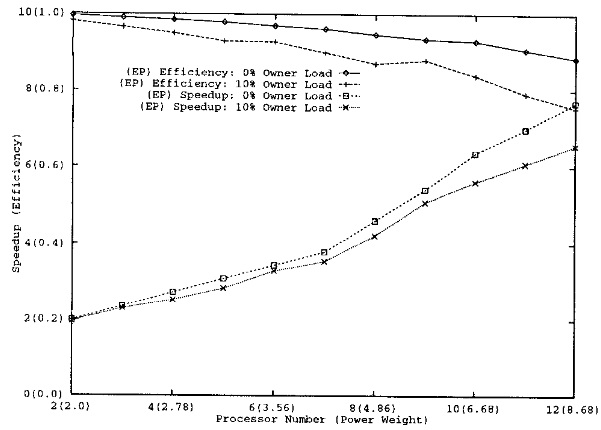

Figure 6: Fixed-size performance for EP.

differences. Only the Kernel EP program has a constant number of communication messages per process, which is independent of task sizes. Further checking the speedup curves of MM on the third row and of MS on the fourth row, we notice that MM reached its maximal speedup on 8 workstations and MS reached its maximal speedup on 7 workstations, which means that execution times reached a minimum. However, using power scaling, the speedup of each program always increased with the increase of the power because this scaling does not cause an increase in communications as does the physical scaling method.

Analytical models in section 3 indicate that parallel computing performance on a heterogeneous NOW is mainly affected by the heterogeneity and the parallelism degree where the workload effects are included. The experiment results are consistent with the modeling results. For example, the EP program had the highest parallelism degree among the five programs, and it presented the highest speedup and the efficiency. The parallelism degree is dynamically changed as the heterogeneity changes in the system. Using the physical scaling and power scaling methods to scale the NOW system, we increased the computing power of the system to a certain level. However, the resulted heterogeneity of each scaled system is different, which in turn causes the same program to present different parallelism degree. Based on our measurements, the parallelism degree of each program using the power scaling method is higher than the one using the physical scaling method. This is because the negative effect of increasing the communication complexity by the physical scaling method was more significant than the negative effect of increasing computation/communication ratio by the power scaling method in our case studies.

5.1.2 Effects of system workload

Comparing the speedup and efficiency curves in a dedicated environment with that in a nondedicated environment of 10% owner workload, we notice that both have similar performance. In the case of Kernel EP, owner workload effects degraded the speedup and efficiency proportionally. In contrast, in the cases of the other four programs, the speedup (efficiency) curves in the dedicated environment intersect the corresponding speedup (efficiency) curves in the owner workload environment. To study the reason behind this difference, we traced the five program executions. Our traces indicate that the owner workload in each workstation could overlap the communication times of the four programs so that the speedups and efficiencies could kept almost as same as that in the dedicated environment. Since communication is light for the Kernel EP program, there was little such overlapping of execution.

Analytical results in section 3 also show some potential performance uncertainty caused by the network heterogeneity. We confirm this by the case studies. The workload affects each program differently by changing its parallelism degree differently. For example, the parallelism degree was reduced more for the EP program than the ones of other four programs when the same owner workload distributions were introduced.

5.2 Scalability performance

The scalability defined in Section 3 measures the average overhead latency unit increment when both the sizes of the problem and the system computing power are adaptively scaled to keep the efficiency constant. In a heterogeneous environment, the system computing power can be adjusted by physical scaling and power scaling. Here, we only measured the scalabilities of the five programs using physical scaling. The scalability of a heterogeneous computation is closely related to the scaling order of the employed machines. We used the same scaling order of workstations as that used in the measurements of fixed-size performance of the programs. By considering the memory bound of each workstation, we chose the constant computing efficiency as 30%. For each program, the scalability results were measured under two workload environments: dedicated and 10% owner workload. From the measurement results, the five programs are ranked by scalability as follows:

$$EP > EF > MS > Cholesky > MM.$$

This order is consistent with the communication complexity orders of the five programs. This shows that communication latency is an important factor affecting the scalability of a program in a heterogeneous network. Comparing the scalabilities in the dedicated environment and the 10% owner workload environment for each program, we notice that the owner workload had the most significant effect to the scalability of the EP program (up to 11% reduction). This is because the overhead latency in the EP program increased most significantly caused by the owner workload.

6 Summary

In this paper, we model and characterize parallel computing performance of heterogeneous NOW by presenting metrics and by conducting experiments. We focus on studying effects of network heterogeneity and owner workload on speedup, efficiency and scalability. Here we summarize performance results and their implications:

- The power weight is an important factor to quantify heterogeneity of a NOW system for defining its speedup, efficiency and scalability. Since the computing power is measured by average execution time of an application program, it does not only depend on a processor's clock rate, but also depends on cache/memory sizes and their access times. The power weight concept may also be applied for general heterogeneous computing performance evaluation.
- In practice, the memory bound of each workstation is an important limit for a program to scale. If the data allocation of a program partition is larger than the memory size in a workstation, the power weight of the workstation may become extremely small. The total execution time would sharply increase and the heterogeneous computation would be negatively effective.
- For a given heterogeneous NOW, parallel computing performance is improved as the parallelism degree of the program increases. However, changes of the network heterogeneity by system scaling and dynamic owner workload distributions would affect the performance differently. We have shown the performance variations through models and experimental case studies.
- The presented models/metrics in this paper are general enough to cover performance evaluation of both homogeneous and heterogeneous computations in dedicated and nondedicated distributed environment.
- The scaling of a heterogeneous system can be done in three directions: physical scaling, power scaling or simultaneous physical and power scalings. Different scaling methods will set different computational power and different communication structures in the system on which scalability of an application program will be different.
- In general, execution performance of a program, such as speedup and efficiency, will degrade proportionally with the increase of owner workload in each workstation. However, if the communication in a program can be overlapped with the workload, the negative workload effect to a parallel job could be reduced.

Since the heterogeneous computing scalability issue involves multi-dimensional factors and effects, we are further studying the complex scaling behaviors. We are also applying the models for practical performance predictions of parallel computing on nondedicated heterogeneous networks of workstations [10].

Acknowledgement: Haixu Yang partially conducted the experiments on the NOW system. We wish to thank the anonymous referees for their insightful comments on the paper. We are grateful to Sandy Dykes and Neal Wagner for their careful reading the paper and their constructive suggestions.

References

[1] D. Bailey et. al., "The NAS parallel benchmarks", *International Journal of Supercomputer Applications*, Vol. 5, No. 3, Fall, 1991, pp. 63-73.

[2] J. Choi, J. J. Dongarra, and D. W. Walker, *The design and implementation of the ScaLAPACK LU, QR, and Cholesky routines*, Technical Report, ORNL/TM-12470, Oak Ridge National Laboratory, September, 1994.

[3] V. Donaldson, F. Berman, and R. Paturi, "Program speedup in a heterogeneous computing network", *Journal of Parallel and Distributed Computing*, Vol. 21, 1994, pp. 316-322.

[4] R. F. Freund, "Optimal selection theory for superconcurrency", *Proceedings of Supercomputing'89*, Nov. 1989. pp. 699-703.

[5] R. W. Moore, "Distributing application across wide area networks," *Proc. Los Alamos Gigabit Testbed Workshop*, January, 1990.

[6] C. R. Mechoso, J. D. Farrara, and J. A. Spahr, "Achieving superlinear speedup on a heterogeneous, distributed system", *IEEE Parallel & Distributed Technology*, Summer Issue, 1994, pp. 57-61.

[7] V. S. Sunderam, "PVM: A framework for parallel distributed computing," *Concurrency: Practice and Experience*, Vol. 2, No. 4, pp. 315-339.

[8] X. Zhang and H. Deng, "Distributed methods and performance of image edge detection", *Proceedings of the Sixth IEEE Symposium on Parallel and Distributed Processing*, IEEE Computer Society Press, October, 1994, pp. 136-143.

[9] X. Zhang, Y. Yan, and K. He, "Latency metric: an experimental method for measuring and evaluating program and architecture scalability", *Journal of Parallel and Distributed Computing*, Vol. 22, No. 3, 1994, pp. 392-410.

[10] X. Zhang and Y. Yan, "A framework of performance prediction of parallel computing on nondedicated heterogeneous networks of workstations", *Proceedings of the 1995 International Conference of Parallel Processing*, CRC Press, Vol. I, August, 1995.