Statistical Predictors of Computing Power in Heterogeneous Clusters

Ron C. Chiang[1] Anthony A. Maciejewski[1] Arnold L. Rosenberg[1,2] Howard Jay Siegel[1,2]
[1]Electrical and Computer Engineering Department
[2]Computer Science Department
Colorado State University
Fort Collins, CO 80523, USA
{ron.chiang,aam,rsnbrg,hj}@colostate.edu

***Abstract*—If cluster $\mathcal{C}_1$ consists of computers with a faster mean speed than the computers in cluster $\mathcal{C}_2$, does this imply that cluster $\mathcal{C}_1$ is more productive than cluster $\mathcal{C}_2$? What if the computers in cluster $\mathcal{C}_1$ have the same mean speed as the computers in cluster $\mathcal{C}_2$: is the one with computers that have a higher variance in speed more productive? Simulation experiments are performed to explore the above questions within a formal framework for measuring the performance of a cluster. Simulation results show that both mean speed and variance in speed (when mean speeds are equal) are typically correlated with the performance of a cluster, but not always; these statements are quantified statistically for our simulation environments. In addition, simulation results also show that: (1) If the mean speed of computers in cluster $\mathcal{C}_1$ is faster by at least a threshold amount than the mean speed of computers in cluster $\mathcal{C}_2$, then $\mathcal{C}_1$ is more productive than $\mathcal{C}_2$. (2) If the computers in clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same mean speed, then $\mathcal{C}_1$ is more productive than $\mathcal{C}_2$ when the variance in speed of computers in cluster $\mathcal{C}_1$ is higher by at least a threshold amount than the variance in speed of computers in cluster $\mathcal{C}_2$.**

***Keywords*-cluster computing, heterogeneous computing, scheduling.**

I. Introduction

A *heterogeneous* multicomputer platform comprises computers that may differ in computing power and that are capable of communicating with one another [3], [4], [13]. Heterogeneity pervades almost all modern computing systems such as: Grid computing [10], [15], [16], global computing [12], volunteer computing [17], cloud computing [11], and clusters [5], [19]. The difficulty of scheduling complex computations on heterogeneous platforms greatly complicates the challenge of high performance computing.

There are many studies relating to important scheduling issues associated with heterogeneous platforms (e.g., [1], [2], [6], [7], [8], [14], [20]). The results from [1] propose an environment that exhibits the property where node-heterogeneity among the computers in a cluster is the only factor that influences the performance of a cluster. This current work uses that environment and performs simulations to compare the performance of sample clusters that have different mean speeds, and clusters that have the same mean speed, but different variances in speed among their computers. We say that cluster $\mathcal{C}_1$ outperforms cluster $\mathcal{C}_2$ if cluster $\mathcal{C}_1$ completes more work than cluster $\mathcal{C}_2$ with the same amount of time within the framework of a scheduling problem for clusters called the cluster-exploitation problem.

Our results further extend the work in [21] with respect to understanding the role of statistical moments as predictors of computational power, and answer the following questions about heterogeneity:

- *If cluster $\mathcal{C}_1$ consists of computers with a faster mean speed than the computers in cluster $\mathcal{C}_2$, does this imply that cluster $\mathcal{C}_1$ is more productive than cluster $\mathcal{C}_2$?*
- *If the computers in cluster $\mathcal{C}_1$ have the same mean speed as the computers in cluster $\mathcal{C}_2$, is the one with computers that have a higher variance in speed more productive?*

From our simulation studies, both mean speed and variance in speed (when mean speeds are equal) are typically correlated with the performance of a cluster, but not always; these statements are quantified statistically. Simulation results also show that: (1) If the mean speed of computers in cluster $\mathcal{C}_1$ is faster by at least a threshold amount than the mean speed of computers in cluster $\mathcal{C}_2$, then $\mathcal{C}_1$ completes more work than $\mathcal{C}_2$ in the same amount of time. (2) If the computers in clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same mean speed, then $\mathcal{C}_1$ is more productive than $\mathcal{C}_2$ when the variance in speed of computers in cluster $\mathcal{C}_1$ is higher by at least a threshold amount than the variance in speed of computers in cluster $\mathcal{C}_2$.

In the next section, we introduce the technical background. Section III describes the simulation procedure and results. Section IV is the conclusion.

II. Technical Background

A. The Architectural Model

This work is based on the architectural model from [13]. Let a cluster $\mathcal{C}$ have n computers $C_1, \ldots, C_n$, where each C_i completes one unit of work in ρ_i time units. That is, faster computers have smaller ρ-values. We call the vector $\langle \rho_1, \ldots, \rho_n \rangle$ $\mathcal{C}$'s *heterogeneity profile*. A server C_0 has W units of work consisting of mutually independent tasks of equal sizes and complexities.[1] In addition, C_0 has access to the cluster $\mathcal{C}$, and distributes w_i units of work to each $C_i \in \mathcal{C}$ in a single message, where $W = \Sigma_{i=1}^{n} w_i$. In our simulation, we normalize ρ-values so that the ρ-value of the slowest computer over all clusters that are being considered is 1.0. We assume each unit of work produces δ units of results. Each C_i has to return the results, in a single message, to C_0. Fig. 1 provides an overview of this environment.

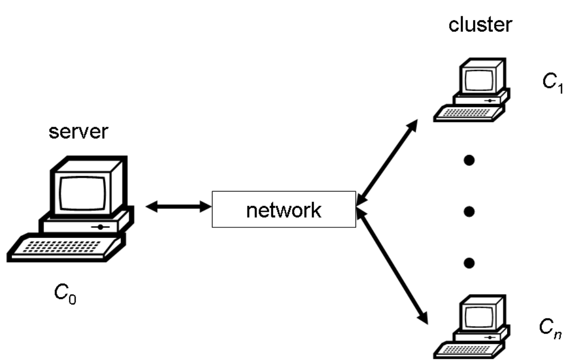

Fig. 1: This graph shows an architectural overview of the CEP.

Consider two computers C_i and C_j (where either C_i or C_j is C_0), which are a sender and a receiver respectively. Before delivering the data through the network, C_i has to package data into a single message at a rate of π_i time units per work unit. The network has a uniform transmitting rate of τ time units per work unit. When C_j receives the data, it has to unpackage the data at a rate of π_j time units per work unit.[2] We assume all computers in this model are architecturally balanced. That is, if C_i is faster than C_j, then all subsystems on C_i are also proportionally faster than those on C_j. For our model, π_i is faster than π_j by the factor of ρ_j/ρ_i. Table I presents sample values of the architectural parameters that we later use in simulations.

[1] "Size" = specification length; "complexity" = computation time.

[2] We equalize packaging and unpackaging rates of the same computer, which reflects the case in most actual architectures.

TABLE I: Sample values of architectural parameters.

Parameter		*Wall-Clock Time/Rate*	
transit rate:	τ	1 μsec	per work unit
(un)packaging rate:	π_0	10 μsec	per work unit
computing rate:	ρ_0	1 sec	per work unit

B. The Cluster-Exploitation Problem and Worksharing Protocols

The Cluster-Exploitation Problem (CEP) is to derive a schedule such that C_0 completes as many units of work as possible on cluster $\mathcal{C}$ within a given lifespan *of L time units.*

We define a *worksharing protocol* as a schedule to solve the CEP. A protocol proceeds as follows:

1) *Transmit work:* For a computer $C_i \in \mathcal{C}$, C_0 packages w_i units of work into a single message, and sends it to C_i. Once C_0 completes sending work to C_i, it starts to prepare and send work to another computer immediately. The server C_0 keeps transmitting work until all computers in $\mathcal{C}$ have their own workloads.
2) *Compute:* After C_i receives its workload, it starts immediately to unpack and process it.
3) *Transmit results:* Once C_i has the results of its work assignment, it immediately packages the results into a single message and returns it to C_0.

Fig. 2 demonstrates how C_0 shares work with a three-computer cluster. Below every action of a computer is the required time of that action. Computer C_i is done only when C_0 receives the entire results of C_i's work. For example, C_1 is the first computer to be done in this case. Then, C_2 is the second and C_3 is the third. The order of starting work is the same as the order of finishing in this example; this is called the *FIFO (First-In-First-Out) worksharing protocol*. This ordering is not true in general for the worksharing protocols [1], however, protocols with this FIFO property are very special within the context of solving the CEP.

Theorem 1 ([1]). *Over any sufficiently long lifespan L, for any heterogeneous cluster $\mathcal{C}$—no matter what its heterogeneity profile:*

1) *FIFO worksharing protocols provide optimal solutions to the CEP.*

C_0	sends work to C_1	sends work to C_2	sends work to C_3			
	$(\pi_0 + \tau)w_1$	$(\pi_0 + \tau)w_2$	$(\pi_0 + \tau)w_3$			
C_1	waits	processes		results		
		$(\pi_1 + \rho_1)w_1$		$(\pi_1 + \tau)\delta w_1$		
C_2	waits		processes		results	
			$(\pi_2 + \rho_2)w_2$		$(\pi_2 + \tau)\delta w_2$	
C_3	waits			processes		results
				$(\pi_3 + \rho_3)w_3$		$(\pi_3 + \tau)\delta w_3$

Fig. 2: C_0 shares work with a three-computer cluster.

2) *$\mathcal{C}$ is equally productive under* every *FIFO protocol, i.e., under all startup indexings.*

Because we know the FIFO worksharing protocol is optimal for *any* cluster and under all startup indexings, the only factor affecting performance of a cluster with respect to the CEP is its heterogeneity profile. We therefore use the framework of [1] to study node-heterogeneity in clusters.

C. Measuring a Cluster's Performance

Within the context of the CEP, the direct method to measure a cluster's performance is to measure its work production. Given a fixed lifespan L, the work production of a heterogeneous cluster under the FIFO worksharing protocol can be calculated asymptotically in Theorem 2. To simplify the expressions in Theorem 2, let $A = \pi_0 + \tau$ and $B = 1 + (1 + \delta)\pi_0$.

Theorem 2 ([1])**.** *Let $\mathcal{C}$ have profile $\mathsf{P} = \langle \rho_1, \ldots, \rho_n \rangle$ and let*

$$X(\mathsf{P}) \;=\; \sum_{i=1}^{n} \frac{1}{A + B\rho_i} \cdot \prod_{j=1}^{i-1} \frac{B\rho_j + \tau}{A + B\rho_j}. \qquad (1)$$

If one uses the FIFO procedure to solve the CEP, then the asymptotic work-production of $\mathcal{C}$ is:

$$W(L; \mathsf{P}) \;=\; \frac{1}{\tau + 1/X(\mathsf{P})} \cdot L. \qquad (2)$$

$X(\mathsf{P})$ is one of the components in calculating $W(L; \mathsf{P})$, and is composed of only fixed system parameters and the heterogeneity profile. For a fixed lifespan L and a fixed τ, $W(L; \mathsf{P})$ is approximately proportional to $X(\mathsf{P})$. That is, $X(\mathsf{P})$ can be used as a measure of work production. Using $X(\mathsf{P})$ as a performance measure is better than using $W(L; \mathsf{P})$ because it allows us to ignore L in our computations.

A related performance measurement called *homogeneous-equivalent computation rate* (HECR) [21] of a *heterogeneous* cluster $\mathcal{C}$ is the ρ-value of a *homogeneous* cluster $\mathcal{C}^{(\rho)}$ with profile $\langle \rho, \ldots, \rho \rangle$, such that $\mathcal{C}^{(\rho)}$ and $\mathcal{C}$ produce the same amount of work in the same amount of time. It is proved in [21] that $\mathcal{C}$'s HECR is

$$\rho \;=\; \frac{\pi_0}{B - \Big(1 - \pi_0 X(\mathsf{P})\Big)^{1/n} B} \;-\; \frac{A}{B}. \qquad (3)$$

A homogeneous cluster with faster ρ-values always completes more work than another homogeneous cluster with slower ρ-values in the same amount of time. We use a cluster's HECR as a measure of its performance in our simulation study because it is a single number that characterizes computing power of a cluster [21].

Both $X(\mathsf{P})$ and the HECR are complicated functions for measuring productivity of a cluster with respect to the CEP. A result from [21] indicates that one can approximate productivity of a cluster by statistical measures of a cluster's heterogeneity profile. For example, if there is only one computer in a cluster and cluster $\mathcal{C}_1$'s computer is faster than cluster $\mathcal{C}_2$'s, then $\mathcal{C}_1$ completes more work than $\mathcal{C}_2$ in the context of the CEP for the same amount of time. But, when there is more than one computer in a cluster, is cluster $\mathcal{C}_1$ still more productive than cluster $\mathcal{C}_2$ when the computers in cluster $\mathcal{C}_1$ have a faster mean speed than the computers in cluster $\mathcal{C}_2$? We explore this question in our simulations.

Theorem 3 indicates that variance in speed is a decisive factor in comparing the performance of two-computer clusters that have the same mean speed.

Theorem 3 ([21])**.** *Assume that cluster $\mathcal{C}_1$, with profile P_1, and cluster $\mathcal{C}_2$, with profile P_2, share the same mean speed. When $\mathcal{C}_1$ and $\mathcal{C}_2$ each has two computers, $\mathcal{C}_1$ outperforms $\mathcal{C}_2$ if and only if $VAR(\mathsf{P}_1) > VAR(\mathsf{P}_2)$, where $VAR(\mathsf{P})$ is the variance of profile P.*

Although Theorem 3 shows that variance in speed definitely determines which cluster completes more work among two-computer clusters that have the same mean speed, we want to know more about its accuracy in predicting relative performance of large clusters with the same mean speed. In other words, if the computers in cluster $\mathcal{C}_1$ have the same mean speed as the computers in cluster $\mathcal{C}_2$, is the one with computers that have a higher variance in speed more productive? We also explore this question via simulations.

III. Procedure and Results

A. Simulation Procedure

We compare representative samples of profile pairs to answer the questions in the previous section. In our first phase of studies, we select a large number of samples uniformly. That is, the sample profiles' mean speeds and variances in speed are equally distributed from the smallest to the largest possible values.

For convenience, let 0.01 be the smallest granularity of ρ-values. That is, the possible ρ-values in our simulations are $0.01, 0.02, 0.03, \ldots, 1.0$. Let the value of d be the factor to control the difference in variances in speed among sample profiles with the same mean speed. In our simulations, $d = 100$. We perform the procedure in Fig. 3 to generate sample profiles for n-computer clusters. By executing the generation procedure in Fig. 3, we sample profiles with different mean speeds and different variances in speed from the smallest to the largest possible values.

We generate sample profiles for each cluster with 2^x computers, where $x \in 1, \ldots, 12$. Because we generate and compare sample profiles in a discrete fashion, we use curve fitting methods to determine a mathematical function to interpolate/extrapolate the results. We apply tests recommended in [9], [18], the Wald-Wolfowitz runs test and Akaike's Information Criterion, to choose the function with the best fit.

B. Results

1) Mean Speed as a Predictor of Performance: We define *cluster size* as the number of computers in a cluster. Assume that the computers in cluster $\mathcal{C}_1$ have a faster mean speed than the computers in cluster $\mathcal{C}_2$. The percentage of failed predictions when one predicts that cluster $\mathcal{C}_1$ is more productive than cluster $\mathcal{C}_2$ is shown in Fig. 4. The percentage of failed predictions is 11.68% when there are two computers per cluster. Then, the percentage of failed predictions increases to approximately 15% when there are eight computers per cluster. For cluster sizes greater than eight, the percentage of failed predictions remains at approximately 15%.

From the results in Fig. 4, the mean speed is not always correlated with performance, but the percentage of failed predictions seems to converge in our simulations. Assume that cluster $\mathcal{C}_1$ with profile P_1 has a faster mean speed $\bar{\rho}_1$ than cluster $\mathcal{C}_2$ with profile P_2 and mean speed $\bar{\rho}_2$. We want to find a threshold $T_{\bar{\rho}}$ such that if $\bar{\rho}_2 - \bar{\rho}_1 > T_{\bar{\rho}}$, then $\mathcal{C}_1$ always outperforms $\mathcal{C}_2$.[3] We apply a binary search to find $T_{\bar{\rho}}$ with the results shown in Fig. 5. $T_{\bar{\rho}}$ is 0.49 time units per work unit at two computers per cluster, and increases to 0.8 time units per work unit at eight computers per cluster. Then, $T_{\bar{\rho}}$ keeps steady around 0.8 time units per work unit when there are more than eight computers in a cluster.

From these results, it appears that mean speed of a cluster is typically a good predictor of performance, i.e., about 85% of the time, however, there are cases where a cluster with significantly lower mean speed will outperform a faster cluster. For example, consider eight-computer clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ with profiles $\mathsf{P}_1 = \langle 0.01, 0.95, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 \rangle$, and $\mathsf{P}_2 = \langle 0.08, 0.08, 0.08, 0.08, 0.08, 0.08, 0.08, 0.08 \rangle$, respectively. P_1's mean speed $\bar{\rho}_1$ is 0.87 time units per work unit and P_2's mean speed $\bar{\rho}_2$ is 0.08 time units per work unit. The difference between $\bar{\rho}_1$ and $\bar{\rho}_2$ is 0.79 time units per work unit. But, P_1's HECR of 0.075 time units per work unit is better than P_2's HECR of 0.08 time units per work unit.[4] In this case, the threshold $T_{\bar{\rho}}$ is 0.8 time units per work unit. In addition, P_1 has the *maximum* variance in speed among all profiles with the mean speed $\bar{\rho}_1$, because any change of the ρ-values in P_1 only decreases its variance in speed if P_1 still keeps the same mean speed $\bar{\rho}_1$; P_2 has the *minimum* variance in speed among all profiles with the mean speed $\bar{\rho}_2$. This leads to the topic of the next section.

2) Variance in Speed as a Predictor of Performance: Although Theorem 3 indicates that variance in speed definitely determines which cluster is more productive among two-computer clusters, we know little from Theorem 3 about the accuracy of variance in speed in predicting the relative performance of large clusters. In this section, we compare the performance of clusters with heterogeneity profiles that have the same mean speed but different variances in speed.

Assume that the computers in cluster $\mathcal{C}_1$ and cluster $\mathcal{C}_2$ share the same mean speed, but the computers in $\mathcal{C}_1$ have a higher variance in speed than the computers in $\mathcal{C}_2$. The percentage of failures when $\mathcal{C}_2$ is more productive than $\mathcal{C}_1$ is shown in Fig. 6. If one predicts that the cluster with

[3] $\bar{\rho}_1$ is smaller than $\bar{\rho}_2$. See Section II-A.

[4] Recall that HECR is a measure of work production, not an encoding of mean speed.

1) For the mean $\bar{\rho} = 0.01$ to 1.0 by 0.01 increment. ($\rho_0 = 1.0$)
2) Given the mean $\bar{\rho}$,
   a) Generate profile P_1 of cluster $\mathcal{C}_1$ with the maximum variance v_1 at the mean $\bar{\rho}$.
   b) Generate profile P_2 of cluster $\mathcal{C}_2$ with the minimum variance v_2 at the mean $\bar{\rho}$.

   Let $\Delta = (v_1 - v_2)/d$.
3) If $v_1 = v_2$, then P_1 is the same as P_2, include P_1 as one sample profile.
4) While $v_1 > v_2$,
   a) Include $\mathcal{C}_1$'s profile P_1 and $\mathcal{C}_2$'s profile P_2 as two sample profiles.
   b) To generate a new sample by reducing v_1, first let $v_1' = v_1$.
   c) While $v_1' - v_1 < \Delta$,
      i) Pick ρ_i and ρ_j from profile P_1, where $\rho_j - \rho_i > 0.01$.
      ii) Increase ρ_i and decrease ρ_j by 0.01.
      iii) Calculate the new variance v_1.
   d) To generate a new sample by enlarging v_2, first let $v_2' = v_2$.
   e) While $v_2 - v_2' < \Delta$,
      i) Pick ρ_i and ρ_j from profile P_2, where $\rho_i > 0.01$ and $\rho_j < 1.0$.
      ii) Decrease ρ_i and increase ρ_j by 0.01.
      iii) Calculate the new variance v_2.

Fig. 3: Procedure for generating sample profiles is presented in pseudocode.

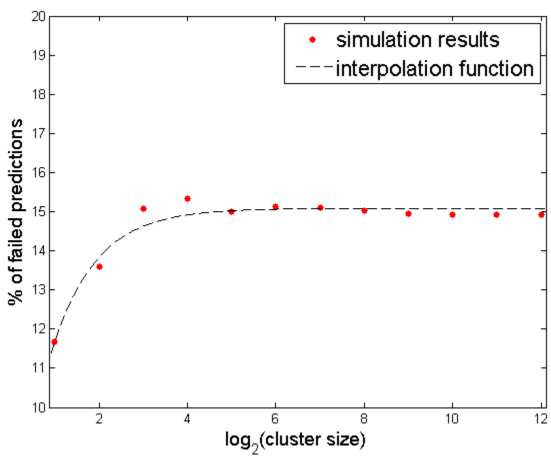

Fig. 4: This graph shows the percentage of failed predictions when using mean speed as a predictor. The interpolation function is $f(x) = 0.0927 \cdot (1 - e^{-x}) + 0.0582$.

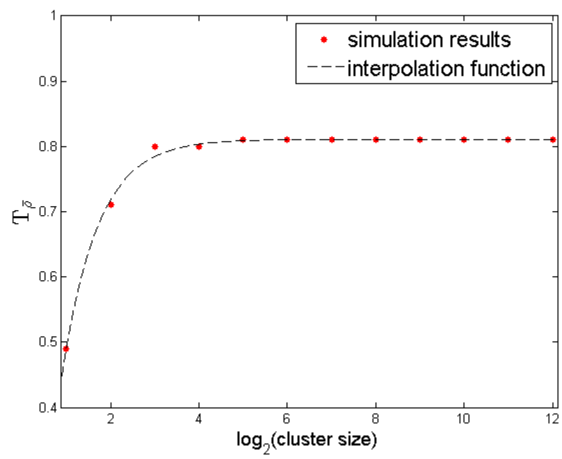

Fig. 5: This graph shows the minimum difference between cluster mean speeds for the faster cluster to outperform the slower cluster, as a function of cluster size. The interpolation function is $f(x) = 1.12 \cdot (1 - e^{-1.25 \cdot x}) - 0.308$.

a higher variance is more productive, then the percentage of failed predictions is 0% for two computers per cluster, which has been shown in Theorem 3. However the percentage quickly climbs up to around 23% at 128 computers per cluster, and keeps steady after that point in our simulations.

Assume that cluster $\mathcal{C}_1$ with profile P_1 has mean speed $\bar{\rho}$ and variance in speed v_1, and cluster $\mathcal{C}_2$ with profile P_2 has the same mean speed $\bar{\rho}$ but different variance in speed $v_2 < v_1$. Fig. 6 indicates that cluster $\mathcal{C}_1$ does not always outperform cluster $\mathcal{C}_2$. We would like to find a threshold T_{var} such that if $v_1 - v_2 > T_{var}$, then $\mathcal{C}_1$ always outperforms $\mathcal{C}_2$ for our environment. A plot of T_{var}, as a function of cluster size is shown in Fig. 7. T_{var} is 0 at two computers per cluster because $\mathcal{C}_1$ always outperforms $\mathcal{C}_2$ in this case. The value of T_{var} grows rapidly but appears to reach an asymptotic value of 0.16

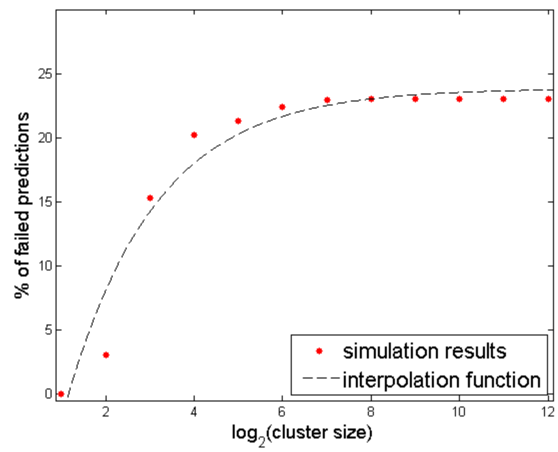

Fig. 6: This graph shows the percentage of failed predictions when using variance in speed as a predictor. The interpolation function is $f(x) = 0.424 \cdot (1 - e^{-0.495 \cdot x}) - 0.185$.

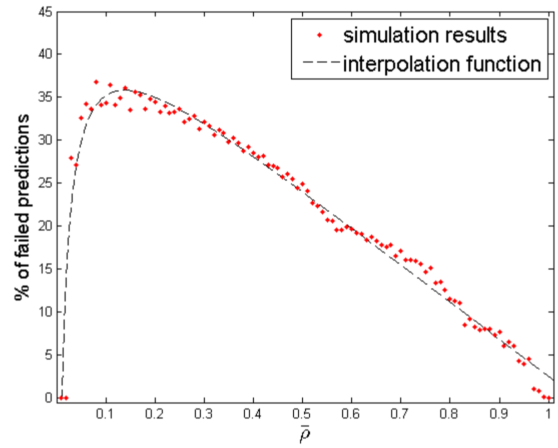

Fig. 8: This graph shows the percentage of failed predictions when using variance in speed as a predictor at different mean speeds $\bar{\rho}$ and 64 computers per cluster. The interpolation function is $f(x) = 0.00419 \cdot \log_2(x)^3 - 0.407 \cdot x + 0.45$.

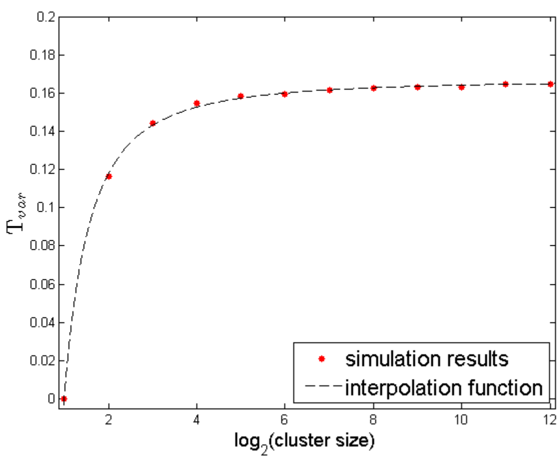

Fig. 7: This graph shows T_{var} among different cluster sizes. The interpolation function is $f(x) = -0.167 \cdot x^{-1.77} + 0.167$.

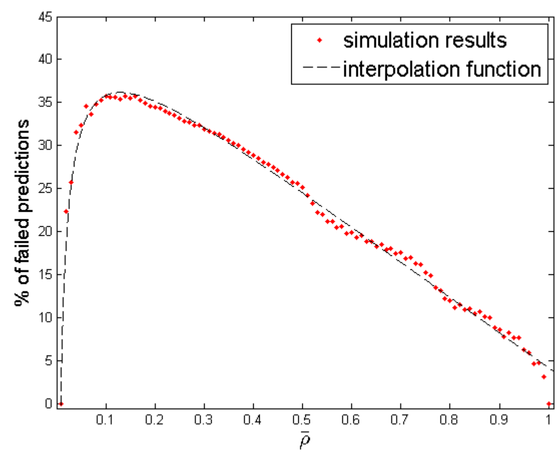

Fig. 9: This graph shows the percentage of failed predictions when using variance in speed as a predictor at different mean speeds $\bar{\rho}$ and 512 computers per cluster.

fairly quickly in our simulations.

We further analyze how often a larger variance fails to predict better performance for different mean speeds. Fig. 8 presents the percentage of failed predictions as a function of mean speed for the case of 64 computers per cluster. The percentage of failed predictions increases rapidly, with a peak value near $\bar{\rho} = 0.1$, and then decreases almost linearly to zero when $\bar{\rho} = 1$. This pattern is similar for other cluster sizes. Fig. 9 and 10 are examples of 512 computers per cluster and 4096 computers per cluster.

Because the percentage of failed predictions changes as $\bar{\rho}$ grows, we therefore explore the relation between T_{var} and $\bar{\rho}$. Fig. 11 shows T_{var} at different mean speeds $\bar{\rho}$ in the 64 computers per cluster case. This pattern also exists in cases of other cluster sizes. Fig. 12 and 13 are examples of 512 computers per cluster and 4096 computers per cluster.

IV. Conclusions

In this work, simulation experiments were performed to generate sample clusters with different mean speeds and different variances in speed, to compare the performance of sample clusters within a formal framework from [1] for measuring the performance of a cluster.

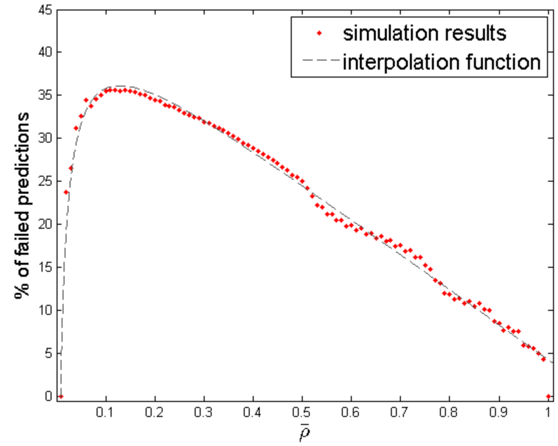

Fig. 10: This graph shows the percentage of failed predictions when using variance in speed as a predictor at different mean speeds $\bar{\rho}$ and 4096 computers per cluster.

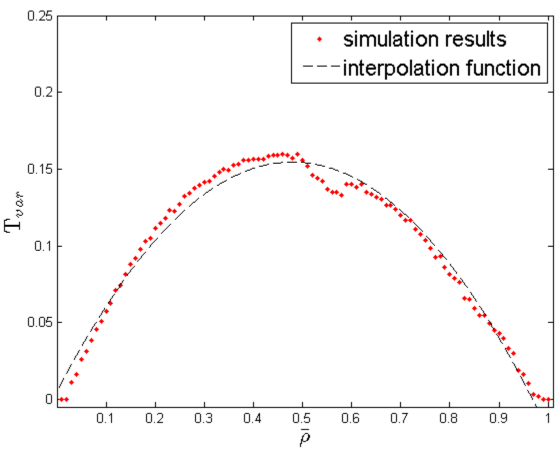

Fig. 11: This graph shows T_{var} at different mean speeds $\bar{\rho}$ and 64 computers per cluster. The interpolation function is $f(x) = -0.6495 \cdot x^2 + 0.6238 \cdot x + 0.004724$.

This work extends the result in [21] with respect to understanding the role of statistical moments as predictors of computational performance, and provides simulation results that indicate heterogeneity influences the performance of a cluster.

Our simulation studies showed that both mean speed and variance in speed (when mean speeds are equal) are typically correlated with the performance of a cluster. We quantify this statement as follows:

First, when using the mean speed of computers in a cluster as a predictor of performance, the percentage of failed predictions is 0% when there is only one computer in a cluster. Then, the percentage of failed predictions

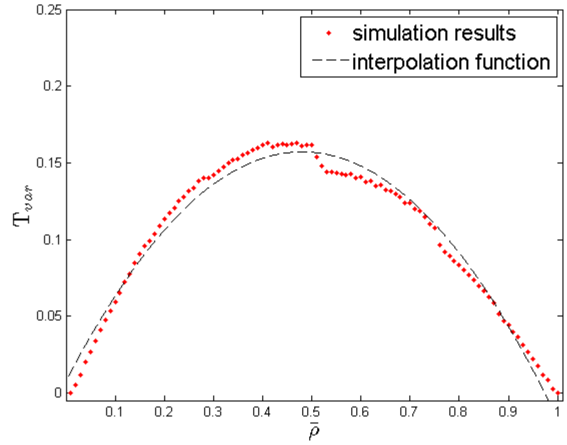

Fig. 12: This graph shows T_{var} at different mean speeds $\bar{\rho}$ and 512 computers per cluster.

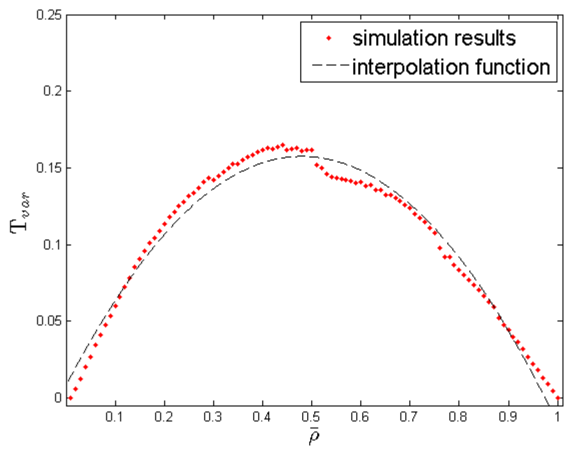

Fig. 13: This graph shows T_{var} at different mean speeds $\bar{\rho}$ and 4096 computers per cluster.

increases as cluster size grows, and is bounded at 16% in our simulations.

Second, let the computers in cluster $\mathcal{C}_1$ have the same mean speed as the computers in cluster $\mathcal{C}_2$ and the computers in cluster $\mathcal{C}_1$ have a higher variance in speed than the computers in cluster $\mathcal{C}_2$. The percentage of failed predictions is 0% when one predicts that $\mathcal{C}_1$ completes more work than $\mathcal{C}_2$ in the same amount of time at two computers per cluster. Then, the percentage of failed predictions increases as cluster size grows, and is bounded at 24% in our simulations. In addition, given a fixed cluster size, the percentage of failed predictions changes as a function of the mean speed $\bar{\rho}$. The percentage of failed predictions increases rapidly, with a peak value near $\bar{\rho} = 0.1$, and then decreases almost linearly to zero when $\bar{\rho} = 1$.

Further study in developing a metric that gives distances between real applications and our model will help to provide a way for the performance assessment of real heterogeneous computing systems.

Acknowledgments: This research was supported in part by NSF Grant CNS-0615170 and CNS-0905399, and by the Colorado State University George T. Abell Endowment. The authors thank Abdulla Al-Qawasmeh and Bhavesh Khemka for their comments.

REFERENCES

[1] M. Adler, Y. Gong, A.L. Rosenberg (2008): On "exploiting" node-heterogeneous clusters optimally. *Theory of Computing Systems*, Vol. 42, No. 4, pp. 465–487.

[2] S. Ali, A.A. Maciejewski, H.J. Siegel (2008): Perspectives on robust resource allocation for heterogeneous parallel systems. *Handbook of Parallel Computing: Models, Algorithms, and Applications*, Chapman & Hall/CRC Press, pp. 41-1–41-30.

[3] A.M. Al-Qawasmeh, A.A. Maciejewski, H.J. Siegel (2010): Characterizing heterogeneous computing environments using singular value decomposition. *Heterogeneity in Computing Workshop (HCW '10)*.

[4] A.M. Al-Qawasmeh, A.A. Maciejewski, H.J. Siegel, J. Smith, J. Potter (2009): Task and Machine Heterogeneities: Higher Moments Matter. *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '09)*.

[5] T.E. Anderson, D.E. Culler, D.A. Patterson, and the NOW Team (1995): A case for NOW (networks of workstations). *IEEE Micro*, Vol. 15, No. 1, pp. 54–64.

[6] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, Y. Robert (2002): Bandwidth-centric allocation of independent tasks on heterogeneous platforms. *IEEE International Parallel and Distributed Processing Symposium (IPDPS '02)*.

[7] O. Beaumont, A. Legrand, Y. Robert (2003): The master-slave paradigm with heterogeneous processors. *IEEE Transactions on Parallel and Distributed Systems*, Vol.14, No. 9, pp.897–908.

[8] O. Beaumont, L. Marchal, Y. Robert (2005): Scheduling divisible loads with return messages on heterogeneous master-worker platforms. *Lecture Notes in Computer Science*, Vol. 3769, Springer, Berlin, pp. 498–507.

[9] K.P. Burnham and D.R. Anderson (2002): *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. New York: Springer.

[10] R. Buyya, D. Abramson, J. Giddy (2001): A case for economy Grid architecture for service oriented Grid computing. *International Heterogeneity in Computing Workshop (HCW '01)*.

[11] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic (2009): Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, Vol. 25, No. 6, pp. 599–616.

[12] W. Cirne and K. Marzullo (1999): The Computational Co-Op: Gathering clusters into a metacomputer. *IEEE International Parallel Processing Symposium (IPPS '99)*, pp. 160–166.

[13] F. Cappello, P. Fraigniaud, B. Mans, A.L. Rosenberg (2005): An algorithmic model for heterogeneous clusters: Rationale and experience. *International Journal of Foundations of Computer Science*, Vol. 16, No. 2, pp. 195–216.

[14] P.-F. Dutot (2003): Master-slave tasking on heterogeneous processors. *IEEE International Parallel and Distributed Processing Symposium (IPDPS '03)*.

[15] I. Foster and C. Kesselman [eds.] (2004): *The Grid: Blueprint for a New Computing Infrastructure (2nd Ed.)*. Morgan-Kaufmann, San Francisco.

[16] I. Foster, C. Kesselman, S. Tuecke (2001): The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, Vol. 15, No. 3, pp. 200–222.

[17] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, M. Lebofsky (2001): SETI@home–Massively distributed computing for SETI. *Computing in Science and Engineering*, Vol. 3, No. 1, pp. 78–83.

[18] H. Motulsky and A. Christopoulos (2004): *Fitting Models to Biological Data Using Linear and Nonlinear Regression: A Practical Guide to Curve Fitting*. Oxford: Oxford University Press.

[19] G.F. Pfister (1995): *In Search of Clusters*. Prentice-Hall.

[20] A.L. Rosenberg (2001): On sharing bags of tasks in heterogeneous networks of workstations: Greedier is not better. *IEEE International Conference on Cluster Computing (CLUSTER '01)*.

[21] A.L. Rosenberg and Ron C. Chiang (2010): Toward understanding heterogeneity in computing. *IEEE International Parallel and Distributed Processing Symposium (IPDPS '10)*.

BIOGRAPHIES

Ron C. Chiang is a Ph.D. student in the Department of Electrical and Computer Engineering at Colorado State University. He received the B.S. degree from the Department of Computer Science and Information Engineering at Tamkang University, and the M.S. degree from the Department of Computer Science and Information Engineering at National Chung Cheng University. His research interest is heterogeneous computing. He is a student member of the ACM and the IEEE Computer Society.

Anthony A. Maciejewski received the B.S., M.S., and Ph.D. degrees in Electrical Engineering in 1982, 1984, and 1987, respectively, all from The Ohio State University. From 1988 to 2001, he was a Professor of Electrical and Computer Engineering at Purdue University. In 2001, he joined Colorado State University where he is currently the Head of the Department of Electrical and Computer Engineering. He is a Fellow of IEEE. A complete vita is available at www.engr.colostate.edu/~aam.

Arnold L. Rosenberg holds the rank of Research Professor in the ECE Department at Colorado State University (with a secondary appointment in Computer Science) and of Distinguished University Professor Emeritus in the Computer Science Department at the University of Massachusetts Amherst. Prior to joining UMass, Rosenberg was a Professor of Computer Science at Duke University from 1981 to 1986, and a Research Staff Member at the IBM Watson Research Center from 1965 to 1981. He has held visiting positions at Yale University and the University of Toronto. He was a Lady Davis Visiting Professor at the Technion (Israel Institute of Technology) in 1994, and a Fulbright Senior Research Scholar at the University of Paris-South in 2000. Rosenberg's research focuses on developing algorithmic models and techniques to exploit the new modalities of "collaborative computing" (wherein multiple computers cooperate to solve a computational problem) that result from emerging technologies, especially Internet-based computing. Rosenberg is the author or coauthor of more than 170 technical papers on these and other topics in theoretical computer science and discrete mathematics. He is the coauthor of the research book "Graph Separators, with Applications" and the author of the textbook “The Pillars of Computation Theory: State, Encoding, Nondeterminism”; additionally, he has served as coeditor of several books. Dr. Rosenberg is a Fellow of the ACM, a Fellow of the IEEE, and a Golden Core member of the IEEE Computer Society. Rosenberg received an A.B. in mathematics at Harvard College and an A.M. and Ph.D. in applied mathematics at Harvard University.

Howard Jay Siegel was appointed the Abell Endowed Chair Distinguished Professor of Electrical and Computer Engineering at Colorado State University (CSU) in 2001, where he is also a Professor of Computer Science. He is the Director of the CSU Information Science and Technology Center (ISTeC), a university-wide organization for promoting, facilitating, and enhancing CSU’s research, education, and outreach activities pertaining to the design and innovative application of computer, communication, and information systems. From 1976 to 2001, he was a professor at Purdue University. Prof. Siegel is a Fellow of the IEEE and a Fellow of the ACM. He received a B.S. degree in electrical engineering and a B.S. degree in management from the Massachusetts Institute of Technology (MIT), and the M.A., M.S.E., and Ph.D. degrees from the Department of Electrical Engineering and Computer Science at Princeton University. He has co-authored over 370 technical papers. His research interests include robust computing systems, resource allocation in computing systems, heterogeneous parallel and distributed computing and communications, parallel algorithms, and parallel machine interconnection networks. He was a Coeditor-in-Chief of the *Journal of Parallel and Distributed Computing*, and was on the Editorial Boards of both the *IEEE Transactions on Parallel and Distributed Systems* and the *IEEE Transactions on Computers*. He was Program Chair/Co-Chair of three major international conferences, General Chair/Co-Chair of seven international conferences, and Chair/Co-Chair of five workshops. He is a member of the Eta Kappa Nu electrical engineering honor society, the Sigma Xi science honor society, and the Upsilon Pi Epsilon computing sciences honor society. He has been an international keynote speaker and tutorial lecturer, and has consulted for industry and government. For more information, please see www.engr.colostate.edu/~hj.