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 \(C_1\) consists of computers with a faster mean speed than the computers in cluster \(C_2\), does this imply that cluster \(C_1\) is more productive than cluster \(C_2\)? What if the computers in cluster \(C_1\) have the same mean speed as the computers in cluster \(C_2\): 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 \(C_1\) is faster by at least a threshold amount than the mean speed of computers in cluster \(C_2\), then \(C_1\) is more productive than \(C_2\). (2) If the computers in clusters \(C_1\) and \(C_2\) have the same mean speed, then \(C_1\) is more productive than \(C_2\) when the variance in speed of computers in cluster \(C_1\) is higher by at least a threshold amount than the variance in speed of computers in cluster \(C_2\).

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 \(C_1\) outperforms cluster \(C_2\) if cluster \(C_1\) completes more work than cluster \(C_2\) 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 \(C_1\) consists of computers with a faster mean speed than the computers in cluster \(C_2\), does this imply that cluster \(C_1\) is more productive than cluster \(C_2\)?
- If the computers in cluster \(C_1\) have the same mean speed as the computers in cluster \(C_2\), 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 \(C_1\) is faster by at least a threshold amount than the mean speed of computers in cluster \(C_2\), then \(C_1\) completes more work than \(C_2\) in the same amount of time. (2) If the computers in clusters \(C_1\) and \(C_2\) have the same mean speed, then \(C_1\) is more productive than \(C_2\) when the variance in speed of computers in cluster \(C_1\) is higher by at least a threshold amount than the variance in speed of computers in cluster \(C_2\).
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 $\rho_i$ time units. That is, faster computers have smaller $\rho$-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 $W = \sum_{i=1}^{n} w_i$. In our simulation, we normalize $\rho$-values so that the $\rho$-value of the slowest computer over all clusters that are being considered is 1.0. We assume each unit of work produces $\delta$ 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.](image)

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 $\pi_i$ time units per work unit. The network has a uniform transmitting rate of $\tau$ time units per work unit. When $C_j$ receives the data, it has to unpack the data at a rate of $\pi_j$ 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, $\pi_i$ is faster than $\pi_j$ by the factor of $\rho_j/\rho_i$. Table I presents sample values of the architectural parameters that we later use in simulations.

**TABLE I:** Sample values of architectural parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Wall-Clock Time/Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>transit rate: $\tau$</td>
<td>1 usec per work unit</td>
</tr>
<tr>
<td>(un)packaging rate: $\pi_0$</td>
<td>10 µsec per work unit</td>
</tr>
<tr>
<td>computing rate: $\rho_0$</td>
<td>1 sec per work unit</td>
</tr>
</tbody>
</table>

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.

---

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

\[
\rho = \frac{\pi_0}{B - \left(1 - \pi_0 X(P)\right)^{1/n}} - \frac{A}{B} \quad (3)
\]

A homogeneous cluster with faster \( \rho \)-values always completes more work than another homogeneous cluster with slower \( \rho \)-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 \( X(P) \) 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 \( C_1 \)'s computer is faster than cluster \( C_2 \)'s, then \( C_1 \) completes more work than \( C_2 \) 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 \( C_1 \) still more productive than cluster \( C_2 \) when the computers in cluster \( C_1 \) have a faster mean speed than the computers in cluster \( C_2 \)? 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 \( C_1 \), with profile \( P_1 \), and cluster \( C_2 \), with profile \( P_2 \), share the same mean speed. When \( C_1 \) and \( C_2 \) each has two computers, \( C_1 \) outperforms \( C_2 \) if and only if \( \text{VAR}(P_1) > \text{VAR}(P_2) \), where \( \text{VAR}(P) \) 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 $C_1$ have the same mean speed as the computers in cluster $C_2$, 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 $\rho$-values. That is, the possible $\rho$-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 $C_1$ have a faster mean speed than the computers in cluster $C_2$. The percentage of failed predictions when one predicts that cluster $C_1$ is more productive than cluster $C_2$ 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 $C_1$ with profile $P_1$ has a faster mean speed $\bar{\rho}_1$ than cluster $C_2$ 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 $C_1$ always outperforms $C_2$. 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 $C_1$ and $C_2$ with profiles $P_1 = \langle 0.01, 0.95, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 \rangle$ and $P_2 = \langle 0.08, 0.08, 0.08, 0.08, 0.08, 0.08, 0.08, 0.08 \rangle$, 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. 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 $\rho$-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 heterogeneous profiles that have the same mean speed but different variances in speed.

Assume that the computers in cluster $C_1$ and cluster $C_2$ share the same mean speed, but the computers in $C_1$ have a higher variance in speed than the computers in $C_2$. The percentage of failures when $C_2$ is more productive than $C_1$ 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 $C_1$ with the maximum variance $v_1$ at the mean $\bar{\rho}$.
   b) Generate profile $P_2$ of cluster $C_2$ 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 $C_1$’s profile $P_1$ and $C_2$’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 $\rho_i$ and $\rho_j$ from profile $P_1$, where $\rho_j - \rho_i > 0.01$.
      ii) Increase $\rho_i$ and decrease $\rho_j$ 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 $\rho_i$ and $\rho_j$ from profile $P_2$, where $\rho_i > 0.01$ and $\rho_j < 1.0$.
      ii) Decrease $\rho_i$ and increase $\rho_j$ 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 $f(x) = 0.0927 \cdot (1 - e^{-x}) + 0.0582$.

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 $C_1$ with profile $P_1$ has mean speed $\bar{\rho}$ and variance in speed $v_1$, and cluster $C_2$ 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 $C_1$ does not always outperform cluster $C_2$. We would like to find a threshold $T_{var}$ such that if $v_1 - v_2 > T_{var}$, then $C_1$ always outperforms $C_2$ 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 $C_1$ always outperforms $C_2$ in this case. The value of $T_{var}$ grows rapidly but appears to reach an asymptotic value of 0.16.
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 \( T_{\text{var}} \) and \( \bar{\rho} \). Fig. 11 shows \( T_{\text{var}} \) 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.
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 increases as cluster size grows, and is bounded at 16% in our simulations.

Second, let the computers in cluster $C_1$ have the same mean speed as the computers in cluster $C_2$ and the computers in cluster $C_1$ have a higher variance in speed than the computers in cluster $C_2$. The percentage of failed predictions is 0% when one predicts that $C_1$ completes more work than $C_2$ 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 $\bar{p} = 0.1$, and then decreases almost linearly to zero when $\bar{p} = 1$.

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**


**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.