Toward Understanding Heterogeneity in Computing[*]

Arnold L. Rosenberg and Ron C. Chiang
Dept. of Electrical & Computer Engineering
Colorado State University
Fort Collins, CO 80523, USA
{rsnbrg,ron.chiang}@colostate.edu

Abstract

Heterogeneity complicates the efficient use of multicomputer platforms, but does it enhance their performance? their cost effectiveness? How can one measure the power of a heterogeneous assemblage of computers ("cluster," for short), both in absolute terms (how powerful is this cluster) and relative terms (which cluster is more powerful)? What makes one cluster more powerful than another? Is one better off with a cluster that has one super-fast computer and the rest of "average" speed or with a cluster all of whose computers are "moderately" fast? If you could replace just one computer in your cluster with a faster one, which computer would you choose: the fastest? the slowest? How does one even ask questions such as these in a rigorous, yet tractable manner? A framework is proposed, and some answers are derived, a few rather surprising. Three highlights: (1) If one can replace only one computer in a cluster by a faster one, it is provably (almost) always most advantageous to replace the fastest one. (2) If the computers in two clusters have the same mean speed, then, empirically, the cluster with the larger variance in speed is (almost) always the faster one. (3) Heterogeneity can actually lend power to a cluster!

1 Motivation and Background

Modern multicomputer platforms are *heterogeneous:* their constituent computers vary in computational powers, and they often intercommunicate over layered networks of varying speeds [12]. One observes substantial heterogeneity in modern platforms such as: clusters [2, 22]; modalities of Internet-based computing [21] such as grid computing [9, 15], global computing [11], volunteer computing [17], and cloud computing [10]. The difficulty of scheduling complex computations

[*]A portion of this paper appeared at the *24th IEEE Intl. Parallel and Distributed Processing Symp. (IPDPS), 2010.*

on heterogeneous platforms greatly complicates the challenge of high performance computing in modern environments. In 1994, the first author noted the need for better understanding of the scheduling implications of heterogeneity via rigorous analyses [24]. There has since been an impressive amount of first-rate work on this topic—focusing largely on collective communication [3, 4, 8, 16, 18, 23, 26], but also studying important scheduling issues [1, 5, 6, 7, 14, 19]. That said, sources such as [1] show that there is still much to learn about this important topic—including the questions in the abstract.

1.1 "Understanding" Heterogeneity

Many sources view heterogeneity as a computational encumbrance that arises from negative factors such as hardware failure and obsolescence. Thus viewed, heterogeneity is a phenomenon that must be coped with—but that we would be better off without. In fact, our study illustrates that heterogeneity should often be welcomed! We study heterogeneity within the context of the following formal model.

We have access to $n + 1$ computers: the *server* C_0 and a *cluster* $\mathcal{C}$ comprising n computers, $C_1, \ldots, C_n$, which may differ dramatically in speed. (We call $\mathcal{C}$ a "cluster" for convenience: the C_i may be geographically dispersed and more diverse in power than that term usually connotes.) We have a uniform workload, and each C_i can complete one unit of work in ρ_i time units.[1] The vector $\langle \rho_1, \ldots, \rho_n \rangle$ is $\mathcal{C}$'s (heterogeneity) profile. For convenience:

- We index the C_i in nonincreasing order of power, so that $\rho_1 \geq \cdots \geq \rho_n$.
- We normalize the ρ_i so that the *slowest* computer, C_1, has ρ-value $\rho_1 = 1$. (This "power indexing" only identifies computers, so normalization cannot lead to problems.)

We study heterogeneity within the context of the questions in the abstract. How does one deal with such questions rigorously? When can one say that cluster $\mathcal{C}_1$ "outperforms" (or, is more "powerful" than) cluster $\mathcal{C}_2$? We invoke the framework of a remarkable result from [1] that characterizes all optimal solutions to the cluster-exploitation problem, a simple scheduling problem for node-heterogeneous clusters *solely in terms of clusters' heterogeneity profiles*. We thereby *isolate the heterogeneity* of $\mathcal{C}_1$ and $\mathcal{C}_2$ as the only respect in which they differ: both are performing the same computation optimally, given their respective resources.

Highlighted results. Among our several results, three stand out. (1) If one can replace only one computer in a cluster by a faster one, then it is (almost) always most advantageous to replace the fastest computer. This is always true for "additive" speedups (Theorem 3) and almost always for "multiplicative" ones (Theorem 4). (2) If the computers in two n-computer clusters have the same mean speed, then the cluster with the larger variance in computers' speeds is (almost) always the faster one (Section 4). This is provably always true for 2-computer clusters; empirically, it is true 76% of the time for larger clusters—and (also empirically), true 100% of the time when the

[1] Note that faster computers have smaller ρ-values.

difference in variances is sufficiently large (≥ 0.167). (3) Heterogeneity can actually lend power to a cluster! (Corollary 1).

1.2 The Cluster-Exploitation Problem

C_0 has W units of work consisting of mutually independent tasks of equal sizes and complexities.[2] (Such workloads arise in diverse applications such as data smoothing, pattern matching, ray tracing, Monte-Carlo simulations, chromosome mapping [17, 20, 27].) *The tasks' (common) complexity can be an arbitrary function of their (common) size.* C_0 must distribute a "package" of work to each $C_i \in \mathcal{C}$, in a single message. Each unit of work produces $\delta \leq 1$ units of results; each C_i must return the results from its work, in one "package," to C_0. These activities must be orchestrated so that *at most one intercomputer message is in transit at a time*. Consider the following simple computational problem.

The Cluster-Exploitation Problem (CEP).[3] *C_0 must complete as many units of work as possible on cluster $\mathcal{C}$ within a* lifespan *of L time units.*

A unit of work is "complete" once C_0 has transmitted it to a C_i, and C_i has computed the unit and transmitted its results to C_0. We call a schedule for the CEP a worksharing protocol.

The main focus of this paper is on deriving insights into the nature of heterogeneity in computing, using mathematical analyses, simulations that illustrate and elucidate the analytical results, and simulation-based experiments that study problems that have thus far eluded analysis (especially in Section 4).

2 Worksharing Protocols and Work Production

2.1 The Architectural Model [12]

We assume that $\mathcal{C}$'s computers are *(architecturally) balanced* in the following sense: if $\rho_i < \rho_j$, then every one of C_i's subsystems (memory, I/O, etc.) is faster, by the factor ρ_j/ρ_i, than the corresponding subsystem of C_j. Computers intercommunicate over networks with a uniform *transit rate* of τ time units to send one unit of work from any C_i to any C_j. Before injecting a message $\mathcal{M}$ into the network, C_i *packages* $\mathcal{M}$ (e.g., packetizes, compresses, encodes) at a rate of π_i time units per work unit. When C_j receives $\mathcal{M}$, it *unpackages* it, also at a rate of π_j time units per work unit.[4] We ignore the fixed costs associated with transmitting $\mathcal{M}$—the end-to-end latency of the

[2]"Size" refers to specification length; "complexity" refers to computation time.

[3]It is shown in [1] that an optimal solution to the CEP can be converted efficiently into an optimal solution to the CEP's dual, the Cluster-Rental Problem: *C_0 must complete W units of work on cluster $\mathcal{C}$ in as few time units as possible.*

[4]We equate packaging and unpackaging times; this is consistent with most actual architectures.

first packet and the per-message set-up overhead—because their impacts fade over long lifespans L. Recall that *at most one intercomputer message can be in transit at any moment.*

We thus envisage an environment (workload plus platform) in which several *linear relationships* hold. The cost of transmitting work grows linearly with the total amount of work performed: formally, there are constants κ and κ' such that transmitting w units of work takes κw time units, and receiving the results from that work takes $\kappa' w$ time units. These relationships allow us to *measure both time and message-length in the same units as work.*

Note. *A linear relationship between task-size and task-complexity does* not *limit tasks' (common) complexity as a function of their (common) size: κ is just the ratio of the fixed task size to the complexity of a task of that size.*

Table 1 provides intuition about the sizes of the model's parameters and provides us with definite values for our simulations.

Parameter		*Wall-Clock Time/Rate*	
Transit rate (pipelined):	τ	1 μsec	per work unit
Packaging rate:	π	10 μsec	per work unit
Result-size rate:	δ	1 work unit	per work unit

Table 1: Sample parameter values for perspective (used in simulations).

2.2 Worksharing Protocols [1]

One remote computer. C_0 shares w units of work with a single C_i via the process summarized in the action/time diagram of Fig. 1:

C_0 packages work for C_i	work is in transit	C_i receives the work	C_i computes the work	C_i packages its results	results are in transit	C_0 receives the results
$\pi_0 w$	τw	$\pi_i w$	$\rho_i w$	$\pi_i \delta w$	$\tau \delta w$	$\pi_0 \delta w$

Figure 1: Worksharing with one remote computer (not to scale).

Many remote computers. A pair of ordinal-indexing schemes for $\mathcal{C}$'s computers (to complement the power-indexing) helps us orchestrate communications while solving the CEP. The *startup indexing* specifies the order in which C_0 transmits work within $\mathcal{C}$; it labels the computers $C_{s_1}, \ldots, C_{s_n}$, to indicate that C_{s_i} receives work—hence, begins working—before $C_{s_{i+1}}$. Dually, the *finishing indexing* labels the computers $C_{f_1}, \ldots, C_{f_n}$, to specify the order in which they return their results to C_0. Protocols proceed as follows.

1. *Transmit work.* C_0 prepares and transmits w_{s_1} units of work for C_{s_1}. It immediately prepares and sends w_{s_2} units of work to C_{s_2} via the same process. Continuing thus, C_0 supplies each C_{s_i} with w_{s_i} units of work seriatim—with no intervening gaps: it starts processing $C_{s_{i+1}}$'s work immediately after finishing C_{s_i}'s.
2. *Compute.* As soon as C_i receives its work from C_0, it unpackages and performs the work.
3. *Transmit results.* As soon as C_i completes its work, it packages its results and transmits them to C_0.

We choose work-allocations w_i so that, with no gaps, $\mathcal{C}$'s computers:

- receive work and compute in the startup order $\Sigma = \langle s_1, \ldots, s_n \rangle$;
- complete work and transmit results in the finishing order $\Phi = \langle f_1, \ldots, f_n \rangle$;
- complete all work and communications by time L.

The described protocol is summarized in Fig. 2. Note that in this figure, Σ and Φ *coincide:* $(\forall i)[f_i = s_i]$. This is not true in general—cf. [1]—but protocols that share this coincidence are quite special within the context of 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	waits	processes		results	
			$(\pi_2 + \rho_2)w_2$		$(\pi_2 + \tau)\delta w_2$	
C_3	waits	waits	waits	processes		results
				$(\pi_3 + \rho_3)w_3$		$(\pi_3 + \tau)\delta w_3$

Figure 2: Worksharing with three remote computers (not to scale).

2.3 Solving the CEP Optimally: the FIFO Protocol

The FIFO protocol is defined by coincident startup and finishing indexings ($\Sigma = \Phi$), as in Fig. 2. As long as L is large enough, *FIFO protocols solve the* CEP *optimally* [1].

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

Because FIFO protocols solve the CEP optimally, and because their work production depends *only* on a cluster's heterogeneity profile, *we use these solutions as our vehicle for studying clusters' heterogeneity.*

2.4 Two Ways to Measure a Cluster's Computing Power

The X-measure and work production. The obvious way of using the CEP to measure a cluster $\mathcal{C}$'s computing power is to determine how much work $\mathcal{C}$ completes in L time units. The coda of Theorem 1 in [1] does this via an explicit expression. To simplify expressions, let $A = \pi + \tau$ and $B = 1 + (1+\delta)\pi$; see Table 2.

Sample Values for Perspective		
Quantity	*Wall-Clock Time/Rate*	
$A = \pi + \tau$:	11 μsec	per work unit
$B = 1 + (1+\delta)\pi$	(per-task time) +11 $\times 10^{-6}$ sec	per work unit
B with coarse (1 sec/task) tasks	1.000011 sec	per work unit
B with finer (0.1 sec/task) tasks	0.100011 sec	per work unit

Table 2: Sample parameter values for perspective.

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

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

the asymptotic work completed by $\mathcal{C}$ under the FIFO protocol is $W(L; \mathsf{P}) \ = \ \dfrac{1}{\tau\delta + 1/X(\mathsf{P})} \cdot L.$

Because $X(\mathsf{P})$ "tracks" $W(L; \mathsf{P})$, in that $X(\mathsf{P}_1) \geq X(\mathsf{P}_2)$ if and only if $W(L; \mathsf{P}_1) \geq W(L; \mathsf{P}_2)$, we use $X(\mathsf{P})$ as our primary measure of $\mathcal{C}$'s computing power.

The Homogeneous-Equivalent Computing Rate. $X(\mathsf{P})$ is a viable and tractable measure but not very perspicuous. We therefore employ the following alternative measure for a *heterogeneous* cluster $\mathcal{C}$ with profile $\mathsf{P} = \langle \rho_1, \ldots, \rho_n \rangle$. Consider a *homogeneous* cluster $\mathcal{C}^{(\rho)}$, with profile $\mathsf{P}^{(\rho)} = \langle \rho, \ldots, \rho \rangle$ for some $\rho \leq 1$. $\mathcal{C}$'s homogeneous-equivalent computing rate (HECR), $\rho_{\mathcal{C}}$, is the largest ρ such that $X(\mathsf{P}^{(\rho_{\mathcal{C}})}) \geq X(\mathsf{P})$.[5]

Proposition 1. *The HECR of cluster $\mathcal{C}$ is*

$$\rho_{\mathcal{C}} \;=\; \frac{A - \tau\delta}{B - \Big(1 - (A - \tau\delta)X(\mathsf{P})\Big)^{1/n} B} \;-\; \frac{A}{B}$$

Proof. By (1),

$$X(\mathsf{P}^{(\rho)}) \;=\; \frac{1}{A - \tau\delta}\left(1 \;-\; \left(\frac{B\rho + \tau\delta}{B\rho + A}\right)^n\right). \tag{2}$$

By (2), then,

$$\frac{B\rho + \tau\delta}{B\rho + A} \;=\; \big(1 - (A - \tau\delta)X(\mathsf{P}^{(\rho)})\big)^{1/n}$$

Therefore,

$$B\rho + \tau\delta \;=\; (B\rho + A)\,\big(1 - (A - \tau\delta)X(\mathsf{P}^{(\rho)})\big)^{1/n}$$

so that

$$B\rho\left(1 - \big(1 - (A - \tau\delta)X(\mathsf{P}^{(\rho)})\big)^{1/n}\right) \;=\; A\,\big(1 - (A - \tau\delta)X(\mathsf{P}^{(\rho)})\big)^{1/n} - \tau\delta$$

and

$$\rho \;=\; \frac{1}{B} \cdot \frac{A\,\big(1 - (A - \tau\delta)X(\mathsf{P}^{(\rho)})\big)^{1/n} - \tau\delta}{1 - (1 - (A - \tau\delta)X(\mathsf{P}^{(\rho)}))^{1/n}}$$

Proposition 1 now follows by considering the following symbolic simplification. For all D,

$$\frac{AD - \tau\delta}{1 - D} \;=\; \frac{A - \tau\delta}{1 - D} - A.$$

$\square$

2.5 The HECR Measure "in Action"

We illustrate HECRs as performance measures by focusing on two n-computer heterogeneous clusters, which are identified via their profiles.

For any integer function f, denote the sequence $\langle f(1), \ldots, f(n) \rangle$ by $\langle f(i)|_{i=1}^{n} \rangle$.

[5]Because the value of ρ calibrates a *heterogeneous* cluster's power, we must violate our normalizing convention and allow ρ to assume any value ≤ 1. Recall: *a smaller ρ-value means a faster computer.*

Cluster $\mathcal{C}_1$ has profile $\mathsf{P}_1^{(n)} = \big\langle \big(1-(i-1)/n\big)\big|_{i=1}^{n} \big\rangle$, i.e., each $\rho_i = 1-(i-1)/n$; and cluster $\mathcal{C}_2$ has profile $\mathsf{P}_2^{(n)} = \big\langle \big(1/i\big)\big|_{i=1}^{n} \big\rangle$, i.e., each $\rho_i = 1/i$. The speeds of $\mathcal{C}_1$'s computers are spread evenly in the range $[1/n,\ 1]$, while the speeds of $\mathcal{C}_2$'s computers are weighted in the faster half of the range, namely, $[1/n,\ 1/2]$; e.g., when $n = 8$, $\mathsf{P}_1^{(8)} = \big\langle 1, \frac{7}{8}, \ldots, \frac{1}{8} \big\rangle$, and $\mathsf{P}_2^{(8)} = \big\langle 1, \frac{1}{2}, \ldots, \frac{1}{8} \big\rangle$. Most of $\mathcal{C}_2$'s computers are faster than their counterparts in $\mathcal{C}_1$, a fact that should influence the clusters' HECRs: $\mathcal{C}_1$'s HECR should be larger than $\mathcal{C}_2$'s (Prop. 1). Table 3 presents HECRs for three instantiations of $\mathcal{C}_1$ and $\mathcal{C}_2$: with 8, 16, and 32 computers/cluster. As expected, $\mathcal{C}_1$'s HECR is larger than $\mathcal{C}_2$'s for each cluster size. Additionally, because all but one of $\mathcal{C}_2$'s computers have ρ-values $\le 1/2$, while half of $\mathcal{C}_1$'s computers have ρ-values $> 1/2$, we expect that $\mathcal{C}_2$'s work advantage over $\mathcal{C}_1$ should increase with cluster size. Table 3 demonstrates this trend: the ratio of $\mathcal{C}_2$'s HECR to $\mathcal{C}_1$'s improves from roughly 1.7 for 8 computers/cluster to roughly 2.6 for 16 computers to more than 4 for 32 computers.

Cluster	*Profile*	*Number of Computers*		
		8	16	32
$\mathcal{C}_1$	$\big\langle \big(1-(i-1)/n\big)\big\|_{i=1}^{n} \big\rangle$	0.366	0.298	0.251
$\mathcal{C}_2$	$\big\langle (1/i)\big\|_{i=1}^{n} \big\rangle$	0.216	0.116	0.060

Table 3: HECRs for sample heterogeneous clusters

The remainder of the paper is devoted to studying the following question: *What determines a cluster's power?*

3 Speeding up a Cluster Optimally

We study how to speed up a cluster "optimally." After showing that replacing any of $\mathcal{C}$'s computers by a faster one always enhances $\mathcal{C}$'s power, we consider *which* $C_i \in \mathcal{C}$ is the most advantageous one to replace. We study both *additive* speed-ups, wherein a computer with speed ρ is replaced by one with speed $\rho - \varphi$, and *multiplicative* speed-ups, wherein a computer with speed ρ is replaced by one with speed $\psi\rho$. (Of course, $0 < \varphi < \rho_n$ and $0 < \psi < 1$, so every computer can be "sped up.")

3.1 Faster Clusters Complete More Work

Speedups always enhance work production under the FIFO protocol.

Proposition 2. *FIFO protocols complete more work on faster clusters; i.e., given profiles* $\mathsf{P} = \langle \rho_1, \ldots, \rho_i, \ldots, \rho_n \rangle$ *and* $\mathsf{P}' = \langle \rho_1, \ldots, \rho_i', \ldots, \rho_n \rangle$*: if* $\rho_i' < \rho_i$*, then for all* L*,* $W(L; \mathsf{P}') > W(L; \mathsf{P})$.

Proof. Let profiles P and P' be as in the statement of the proposition. We use a device from [1] to show that $X(\mathsf{P}') > X(\mathsf{P})$, so that $W(L;\mathsf{P}') > (L;\mathsf{P})$ for all L.

We begin by refining the expression (1) for $X(\mathsf{P})$ to make explicit the startup order $\Sigma = \langle s_1, \ldots, s_n\rangle$ used by $\mathcal{C}$. (By Theorem 1.2, this has no impact on $\mathcal{C}$'s work production.) As we write $X(\mathsf{P};\Sigma)$ to announce the use of Σ, the only impact on (1) is that the occurrence of "ρ_i" in the expression becomes "ρ_{s_i}," and the two occurrences of "ρ_j" become "ρ_{s_j}." We next choose any startup order Σ for $\mathcal{C}$, for which $s_n = i$; i.e., Σ has the form $\Sigma = \langle s_1, \ldots, s_{n-1}, i\rangle$. We then form the appropriate versions of (1) that use startup order Σ. For the sake of perspicuity, we write these versions in the following way, which emphasize that $X(\mathsf{P};\Sigma)$ and $X(\mathsf{P}';\Sigma)$ differ only in their first terms.

$$\begin{aligned} X(\mathsf{P};\Sigma) &= \frac{1}{A+B\rho_{s_n}}\prod_{j=1}^{n-1}\frac{B\rho_{s_j}+\tau\delta}{A+B\rho_{s_j}} + \sum_{i=1}^{n-1}\frac{1}{A+B\rho_{s_i}}\prod_{j=1}^{i-1}\frac{B\rho_{s_j}+\tau\delta}{A+B\rho_{s_j}} \\ X(\mathsf{P}';\Sigma) &= \frac{1}{A+B\rho'_{s_n}}\prod_{j=1}^{n-1}\frac{B\rho_{s_j}+\tau\delta}{A+B\rho_{s_i}} + \sum_{i=1}^{n-1}\frac{1}{A+B\rho_{s_i}}\prod_{j=1}^{i-1}\frac{B\rho_{s_j}+\tau\delta}{A+B\rho_{s_j}} \end{aligned}$$

Direct calculation now shows that

$$X(\mathsf{P}';\Sigma) - X(\mathsf{P};\Sigma) = \frac{B(\rho_{s_n}-\rho'_{s_n})}{(A+B\rho'_{s_n})(A+B\rho_{s_n})}\cdot\prod_{j=1}^{n-1}\frac{B\rho_j+\tau\delta}{A+B\rho_j}.$$

This difference is positive because $\rho_{s_n} = \rho_i > \rho'_i = \rho'_{s_n}$. We thus have $X(\mathsf{P}';\Sigma) > X(\mathsf{P};\Sigma)$. $\square$

3.2 Which Computer Should One Speed Up?

Say that one has resources to replace only one of cluster $\mathcal{C}$'s computers by a faster one—or, equivalently, to speed up a single computer. Which computer should one choose? Say that cluster $\mathcal{C}$ has heterogeneity profile $\mathsf{P} = \langle\rho_1, \ldots, \rho_n\rangle$, where each $\rho_k \geq \rho_{k+1}$. Let i and $j > i$ be two of $\mathcal{C}$'s power indices. Is it more beneficial to speed up C_i or C_j? Of course, this question makes sense only when C_i is *strictly* slower than C_j, so that $\rho_i > \rho_j$. We answer this question twice—once for *additive* speedups and once for *multiplicative* ones.

The analyses that embody our comparisons are simplified if we require $\mathcal{C}$ to employ a startup ordering Σ from a specific class—even though part (2) of Theorem 1 assures us that Σ has no impact on $W(L;\mathsf{P})$. Specifically, we have $\mathcal{C}$ employ a startup ordering $\Sigma = \langle s_1, \ldots, s_{n-1}, s_n\rangle$ for which $s_n = i$ and $s_{n-1} = j$. Under such an ordering, we can rewrite expression (1) for X(P) in the following convenient way, using two quantities that are independent of ρ_i and ρ_j and that, importantly, are both *positive*.

$$X(\mathsf{P}) = \frac{A+B(\rho_{s_{n-1}}+\rho_{s_n})+\tau\delta}{A^2+AB(\rho_{s_{n-1}}+\rho_{s_n})+B^2\rho_{s_{n-1}}\rho_{s_n}}\cdot Y(\mathsf{P}) + Z(\mathsf{P}) \qquad (3)$$

where

$$Y(\mathsf{P}) \;=\; \prod_{k=1}^{n-2} \frac{B\rho_{s_k} + \tau\delta}{B\rho_{s_k} + A} \quad \text{and} \quad Z(\mathsf{P}) \;=\; X(\rho_{s_1}, \ldots, \rho_{s_{n-2}})$$

The fact that a faster cluster completes more work than a slower one suggests that we compare competing heterogeneity profiles, P and P', via their *work ratio*, $W(L; \mathsf{P}')/W(L; \mathsf{P})$.

3.2.1 The additive-speedup scenario

We compare two profiles: $\mathsf{P}^{(i)}$ is obtained by speeding up the slower computer (of the two we are focusing on), viz., C_i; $\mathsf{P}^{(j)}$ is obtained by speeding up the faster computer, viz., C_j. Both speedups are by the *additive term* $\varphi < \rho_n$. (This inequality ensures that we can speed up any of $\mathcal{C}$'s computers by the term φ.)

$$\begin{aligned} \mathsf{P}^{(i)} &= \langle \rho_1, \ldots, \rho_i - \varphi, \ldots, \rho_j, \ldots, \rho_n \rangle \\ \mathsf{P}^{(j)} &= \langle \rho_1, \ldots, \rho_i, \ldots, \rho_j - \varphi, \ldots, \rho_n \rangle \end{aligned}$$

Intuitively, the faster computer C is, the more "bang" one gets for one's "buck" by speeding C up additively.

Theorem 3. *Under the additive-speedup scenario, the most advantageous single computer to speed up is $\mathcal{C}$'s* fastest *computer.*

Proof. As we compare $X(\mathsf{P}^{(i)})$ and $X(\mathsf{P}^{(j)})$, we lose no generality by using a startup ordering $\Sigma = \langle s_1, \ldots, s_{n-1}, s_n \rangle$ for $\mathcal{C}$'s computers for which $s_n = i$ and $s_{n-1} = j$. We then obtain the following expressions via (3).

$$\begin{aligned} X(\mathsf{P}^{(i)}) &= \frac{A + B(\rho_i + \rho_j - \varphi) + \tau\delta}{A^2 + AB(\rho_i + \rho_j - \varphi) + B^2(\rho_i - \varphi)\rho_j} \cdot Y(\mathsf{P}) \;+\; Z(\mathsf{P}) \\ X(\mathsf{P}^{(j)}) &= \frac{A + B(\rho_i + \rho_j - \varphi) + \tau\delta}{A^2 + AB(\rho_i + \rho_j - \varphi) + B^2\rho_i(\rho_j - \varphi)} \cdot Y(\mathsf{P}) \;+\; Z(\mathsf{P}) \end{aligned}$$

These expressions differ only in the terms $-B^2\varphi\rho_j$ and $-B^2\varphi\rho_i < -B^2\varphi\rho_j$ in the denominators of the lead fractions of $X(\mathsf{P}^{(i)})$ and $X(\mathsf{P}^{(j)})$, respectively. (The "lead fraction" in both expressions is the fraction that multiplies $Y(\mathsf{P})$.) Because $\rho_i > \rho_j$, it follows that $X(\mathsf{P}^{(j)}) > X(\mathsf{P}^{(i)})$, whence the result. $\square$

Additive speedup "in action." We compare $\mathsf{P}^{(i)}$ and $\mathsf{P}^{(j)}$ via the work ratios $W(L; \mathsf{P}^{(i)})/W(L; \mathsf{P})$ and $W(L; \mathsf{P}^{(j)})/W(L; \mathsf{P})$. Prop. 2 assures us that both ratios exceed 1. We illustrate Theorem 3 "in action" by considering the optimal sequence of additive speedups when we begin with the 4-computer heterogeneous cluster $\mathcal{C}$ whose profile is $\mathsf{P} = \langle 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \rangle$ and the (additive) speedup term

$\varphi = \frac{1}{16}$. (Recall: C_1 is $\mathcal{C}$'s slowest computer, and C_4 is its fastest.) Table 4 presents the work ratios obtained by speeding up each of $\mathcal{C}$'s computers in turn by the additive term φ. Table entries are computed using expression (1) with the appropriate profile P. The table shows that one enhances

i	Profile $\mathsf{P}^{(i)}$	Work ratio $W(L; \mathsf{P}^{(i)}) \div W(L; \mathsf{P})$
1	$\langle 15/16,\ 1/2,\ 1/3,\ 1/4 \rangle$	1.008
2	$\langle 1,\ 7/16,\ 1/3,\ 1/4 \rangle$	1.014
3	$\langle 1,\ 1/2,\ 13/48,\ 1/4 \rangle$	1.034
4	$\langle 1,\ 1/2,\ 1/3,\ 3/16 \rangle$	1.159

Table 4: The work ratios as each of $\mathcal{C}$'s 4 computers is sped up additively.

$\mathcal{C}$'s work production: by 0.8% if one speeds up the slowest computer, C_1, by 1.4% if one speeds up the second slowest computer, C_2, by 3.4% if one speeds up the second fastest computer, C_3, and by 15.9% if one speeds up the fastest computer, C_4. Qualitatively similar results are observed with other clusters $\mathcal{C}$ and other speedup terms φ.

3.2.2 The multiplicative-speedup scenario

We compare two profiles: $\mathsf{P}^{[i]}$ is obtained by speeding up the slower computer (of the two we are focusing on), viz., C_i; $\mathsf{P}^{[j]}$ is obtained by speeding up the faster one, viz., C_j; both speedups are by the *multiplicative factor* $\psi < 1$.

$$\begin{aligned} \mathsf{P}^{[i]} &= \langle \rho_1, \ldots \psi\rho_i, \ldots, \rho_j, \ldots, \rho_n \rangle \\ \mathsf{P}^{[j]} &= \langle \rho_1, \ldots, \rho_i, \ldots, \psi\rho_j, \ldots, \rho_n \rangle \end{aligned}$$

The question of which computer to speed up has a more complicated answer with multiplicative speedups than with additive ones. Informally, it is more advantageous to speed up the *faster* computer multiplicatively—thereby (intuitively) getting more "bang" for one's "buck"—unless *either* this computer is already "very fast" *or* the speedup factor ψ is "very small."

Theorem 4. *Let $\mathcal{C}$ contain computers C_i and C_j, with respective ρ-values ρ_i and $\rho_j < \rho_i$. Under the multiplicative-speedup scenario with speedup factor ψ:*

1. *If $\psi\rho_i\rho_j > A\tau\delta/B^2$, then speeding up C_j (the faster computer) allows one to complete more work than does speeding up C_i.*
2. *If $\psi\rho_i\rho_j < A\tau\delta/B^2$, then speeding up C_i (the slower computer) allows one to complete more work than does speeding up C_j.*

Proof. We have $\mathcal{C}$ employ the same startup order Σ as we compare $X(\mathsf{P}^{[i]})$ and $X(\mathsf{P}^{[j]})$ as we did when we compared $X(\mathsf{P}^{(i)})$ and $X(\mathsf{P}^{(j)})$ (in Theorem 3); hence, $s_n = i$ and $s_{n-1} = j$. Specializing (3) therefore yields

$$\begin{aligned} X(\mathsf{P}^{[i]}) &= \frac{A + B(\psi\rho_i + \rho_j) + \tau\delta}{A^2 + AB(\psi\rho_i + \rho_j) + B^2\psi\rho_i\rho_j} \cdot Y(\mathsf{P}) \ + \ Z(\mathsf{P}) \\ X(\mathsf{P}^{[j]}) &= \frac{A + B(\rho_i + \psi\rho_j) + \tau\delta}{A^2 + AB(\rho_i + \psi\rho_j) + B^2\psi\rho_i\rho_j} \cdot Y(\mathsf{P}) \ + \ Z(\mathsf{P}) \end{aligned}$$

Clearly, then, we have $X(\mathsf{P}^{[i]}) > X(\mathsf{P}^{[j]})$ (resp., $X(\mathsf{P}^{[j]}) > X(\mathsf{P}^{[i]})$) if, and only if,

$$\Upsilon^{[i]} \overset{\text{def}}{=} \frac{A + B(\psi\rho_i + \rho_j) + \tau\delta}{A^2 + AB(\psi\rho_i + \rho_j) + B^2\psi\rho_i\rho_j} \ > \ \Upsilon^{[j]} \overset{\text{def}}{=} \frac{A + B(\rho_i + \psi\rho_j) + \tau\delta}{A^2 + AB(\rho_i + \psi\rho_j) + B^2\psi\rho_i\rho_j}$$

(resp., $\Upsilon^{[j]} > \Upsilon^{[i]}$). By "cross-multiplying" to eliminate the fractions, we note finally that $\Upsilon^{[i]} > \Upsilon^{[j]}$ (resp., $\Upsilon^{[j]} > \Upsilon^{[i]}$) if, and only if, $\Xi^{[i]} > \Xi^{[j]}$ (resp., $\Xi^{[j]} > \Xi^{[i]}$) where

$$\begin{aligned} \Xi^{[i]} &= A^3 + A^2B(\psi\rho_i + \rho_j) + A^2\tau\delta \\ &\quad + A^2B(\rho_i + \psi\rho_j) + AB^2(\psi\rho_i + \rho_j)(\rho_i + \psi\rho_j) + AB(\rho_i + \psi\rho_j)\tau\delta \\ &\quad + AB^2\psi\rho_i\rho_j + B^3\psi\rho_i\rho_j(\psi\rho_i + \rho_j) + B^2\psi\rho_i\rho_j\tau\delta \\ \Xi^{[j]} &= A^3 + A^2B(\rho_i + \psi\rho_j) + A^2\tau\delta \\ &\quad + A^2B(\psi\rho_i + \rho_j) + AB^2(\psi\rho_i + \rho_j)(\rho_i + \psi\rho_j) + AB(\psi\rho_i + \rho_j)\tau\delta \\ &\quad + AB^2\psi\rho_i\rho_j + B^3\psi\rho_i\rho_j(\rho_i + \psi\rho_j) + B^2\psi\rho_i\rho_j\tau\delta \end{aligned}$$

Because $\psi < 1$ and $\rho_i > \rho_j$, the result follows by considering when the difference

$$\Xi^{[j]} - \Xi^{[i]} \ = \ [(B^2\psi\rho_i\rho_j \ - \ A\tau\delta)B][(1-\psi)(\rho_i - \rho_j)]$$

is positive and when it is negative. $\square$

Theorem 4 specifies the boundary values of "very fast" and "very small" in terms of the relation between the quantity $\psi\rho_i\rho_j$, which depends on the (present and anticipated) speeds of cluster $\mathcal{C}$'s computers, and the quantity $A\tau\delta/B^2$, which depends on characteristics of the computational environment: the output-to-input ration δ, the network transit rate τ, and the message-packaging rate π. For perspective, with the values from Table 2, $A\tau\delta/B^2 \ \approx \ 1.1 \times 10^{-5}$. Hence, *we expect that speeding up the faster computer will usually be the better option* in the multiplicative scenario, as it is in the additive one.

Multiplicative speedup "in action." The (simulation-based) experiment that illustrates multiplicative speedup "in action" is quite different from the one that illustrates additive speedup. The current experiment begins with a 4-computer homogeneous cluster $\mathcal{C}$ whose profile is $\mathsf{P} = \langle 1, 1, 1, 1 \rangle$. It iteratively optimally speeds $\mathcal{C}$ up via the factor $\psi = 1/2$. We observe the two conditions of Theorem 4 "in action" via a sequence of "snapshots" that depict the successive profiles of cluster $\mathcal{C}$ after each speedup. Each snapshot is depicted via a bar-graph (cf. Figs. 3 and 4) that represents the then-current profile of $\mathcal{C}$ after one round of the experiment. Specifically, when the four bars in a graph have heights ρ_1, ρ_2, ρ_3, ρ_4 from left to right, this means that $\mathcal{C}$'s profile at that round is $\langle \rho_1, \rho_2, \rho_3, \rho_4 \rangle$.

The experiment proceeds as follows. Say that $\mathcal{C}$ has profile P_i after round i of the experiment. At round $i+1$, we consider four potential successors to profile P_i, call them $\mathsf{P}_i^{[1]}$, $\mathsf{P}_i^{[2]}$, $\mathsf{P}_i^{[3]}$, and $\mathsf{P}_i^{[4]}$. Each profile $\mathsf{P}_i^{[j]}$ is obtained by speeding up only computer C_j of $\mathcal{C}$, by the (multiplicative) factor $\psi = 1/2$. We compare the work-productions of the four potential successor profiles, by comparing the profiles' X-values, $X(\mathsf{P}_i^{[1]})$, $X(\mathsf{P}_i^{[2]})$, $X(\mathsf{P}_i^{[3]})$, $X(\mathsf{P}_i^{[4]})$; and we select the profile with the largest work production to be profile P_i's successor, P_{i+i}. In case of ties—wherein speeding up computers C_j and C_k yield the same work-production—we choose to speed up the computer with the larger index. (The independence of the FIFO protocol from computer ordering guarantees that our choice has no impact on subsequent speedups.)

The first phase of the experiment, during which we observe condition (1) of Theorem 4 "in action," is depicted in Fig. 3. (We increased τ from 1 to 200 μsec/work unit to make the figure legible.) We observe $\mathcal{C}$'s profile "improving" (because of the speedups) in 16 steps from its initial value, $\mathsf{P} = \langle 1, 1, 1, 1 \rangle$, to the value $\mathsf{P}' = \left\langle \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16} \right\rangle$, by repeated speedup of $\mathcal{C}$'s *fastest* computer. In detail: this phase of the experiment begins with an invocation of our tie-breaking mechanism because $\mathcal{C}$ is homogeneous before any speedups. We subsequently observe the repeated selection of the then-current fastest computer as the best one to speed up in rounds 2–16. Observe that we speed up computer C_4 in round 1 because of our tie-breaking mechanism, but we speed it up in rounds 2–4 because of condition (1) of Theorem 4. At round 5, condition (2) of Theorem 4 tells us not to speed up computer C_4 again. At that point, we again invoke the tie-breaking mechanism to speed up computer C_3, and the just-described cycle repeats, until $\mathcal{C}$ ends up in round 17 with the profile $\left\langle \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16} \right\rangle$. At this point, we enter phase 2 of the experiment. (The procedure stays fixed, but the results change.)

Once all of $\mathcal{C}$'s computers achieve the speed $\rho_i \equiv 1/16$, all subsequent speedups follow condition (2) in the theorem. Although we continue to speed up one of cluster $\mathcal{C}$'s computers by the factor $\psi = 1/2$, we observe the very different result predicted by condition (2). This phase of the experiment is depicted in Fig. 4. (We changed the scale from that of Fig. 3 to make the new snapshots legible.) In this second phase, we observe condition (2) of Theorem 4 invoked at every step, which means that, at every step, $\mathcal{C}$'s *slowest* computer is the best one to speed up (with the tie-breaking mechanism used as necessary).

4 Predicting Clusters' Powers via Profiles

Proposition 2 tells us that if cluster $\mathcal{C}_1$'s profile $\langle \rho_{11}, \ldots, \rho_{1n} \rangle$ "minorizes" cluster $\mathcal{C}_2$'s profile $\langle \rho_{21}, \ldots, \rho_{2n} \rangle$, in the sense that (a) for every i, $\rho_{1i} \le \rho_{2i}$, (b) for at least one i, $\rho_{1i} < \rho_{2i}$, then

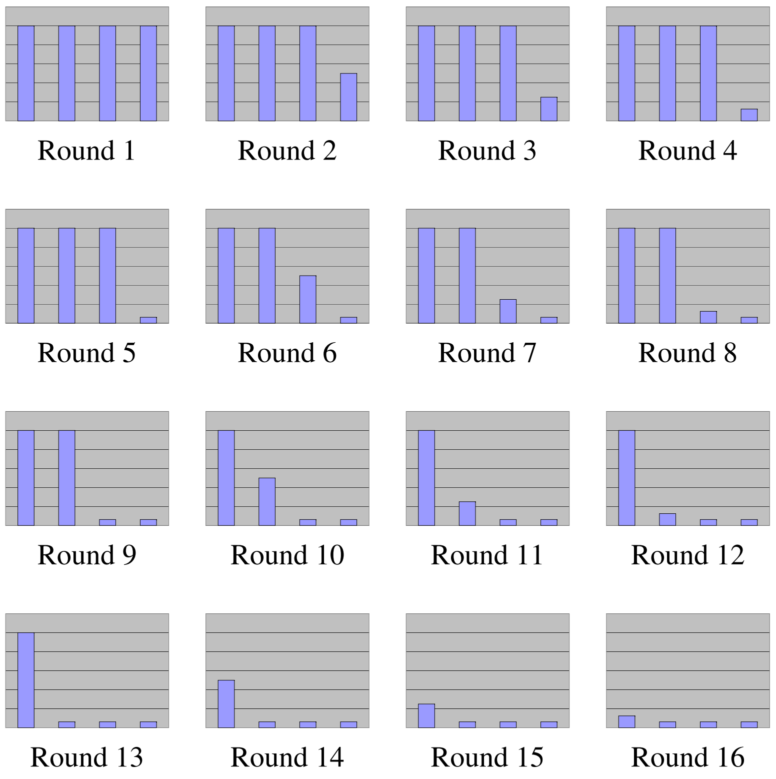

Figure 3: Speeding up a cluster when not all computers are "very fast." Bar heights correspond to ρ-values: The highest bars have height (i.e., ρ-value) 1 and thereafter go down by factors of 2, so the lowest bars have height (i.e., ρ-value) $1/16$.

$\mathcal{C}_1$ outperforms $\mathcal{C}_2$. It is easy to show that the "minorization" condition is sufficient but not necessary: *$\mathcal{C}_1$ can outperform $\mathcal{C}_2$ even though some of $\mathcal{C}_1$'s computers are slower than any of $\mathcal{C}_2$'s.* For instance, a simple calculation shows that the cluster $\mathcal{C}_1$ with profile $\langle 0.99,\ 0.02\rangle$ has a larger X-value than—hence, outperforms—the cluster $\mathcal{C}_2$ with profile $\langle 0.5,\ 0.5\rangle$. Note that in this example, $\mathcal{C}_1$'s *mean ρ-value* exceeds $\mathcal{C}_2$'s—which shows that *mean speeds are* not *valid predictors of clusters' relative computing powers*. Are there valid predictors of clusters' relative performance that are based solely on clusters' profiles—other than "minorization," and, of course, other than computing the clusters' X-values or HECRs?

This section is devoted to studying the use of the *symmetric functions* and the *statistical moments* of two clusters' profiles[6] to predict the clusters' relative computing powers.

[6]Of course, we use the profile only to extract the set of ρ-values.

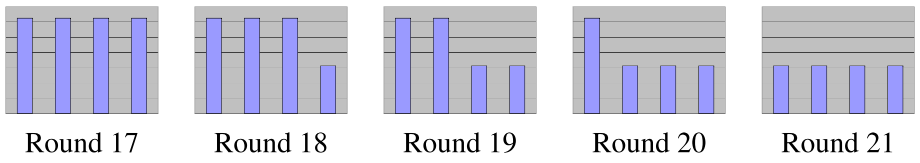

Figure 4: Speeding up a cluster when all computers are "very fast." Bar heights correspond to ρ-values: The highest bars have height/ρ-value $1/16$ and thereafter go down by factors of 2, so the lowest bars have height/ρ-value $1/32$.

4.1 Symmetric Functions of Profiles as Predictors of Power

A function $F(x_1, \ldots, x_n)$ is symmetric if its value is unchanged by every reordering of values for its variables. When $n = 3$, for instance, we must have

$$F(a,b,c) = F(a,c,b) = F(b,a,c) = F(b,c,a) = F(c,a,b) = F(c,b,a)$$

for all values a, b, c for the variables x_1, x_2, x_3. For integers $n > 1$ and $k \in \{1, \ldots n\}$, $F_k^{(n)}$ denotes the symmetric function that has n variables grouped as products of k variables. It simplifies analyses clerically if we allow the index k to assume the value 0 also, with the convention that, for all n, $F_0^{(n)} \equiv 1$. The first three families of $F_k^{(n)}$ functions of ρ-values (excluding the degenerate $F_0^{(n)}$) are exhibited in Table 5.

$F_1^{(2)}(\rho_1, \rho_2)$	$=$	$\rho_1 + \rho_2$
$F_2^{(2)}(\rho_1, \rho_2)$	$=$	$\rho_1\rho_2$
$F_1^{(3)}(\rho_1, \rho_2, \rho_3)$	$=$	$\rho_1 + \rho_2 + \rho_3$
$F_2^{(3)}(\rho_1, \rho_2, \rho_3)$	$=$	$\rho_1\rho_2 + \rho_1\rho_3 + \rho_2\rho_3$
$F_3^{(3)}(\rho_1, \rho_2, \rho_3)$	$=$	$\rho_1\rho_2\rho_3$
$F_1^{(4)}(\rho_1, \rho_2, \rho_3, \rho_4)$	$=$	$\rho_1 + \rho_2 + \rho_3 + \rho_4$
$F_2^{(4)}(\rho_1, \rho_2, \rho_3, \rho_4)$	$=$	$\rho_1\rho_2 + \rho_1\rho_3 + \rho_1\rho_4 + \rho_2\rho_3 + \rho_2\rho_4 + \rho_3\rho_4$
$F_3^{(4)}(\rho_1, \rho_2, \rho_3, \rho_4)$	$=$	$\rho_1\rho_2\rho_3 + \rho_1\rho_2\rho_4 + \rho_1\rho_3\rho_4 + \rho_2\rho_3\rho_4$
$F_4^{(4)}(\rho_1, \rho_2, \rho_3, \rho_4)$	$=$	$\rho_1\rho_2\rho_3\rho_4$

Table 5: The first three families of symmetric functions of ρ-values.

One can use the symmetric functions of clusters' profiles to compare the clusters' powers. We assume henceforth that $\tau\delta \leq A \leq B$.[7] (Consider the semantics of our architectural parameters to see why this inequality is reasonable.)

[7]Recall: $A = \pi + \tau$; $B = 1 + (1 + \delta)\pi$.

Lemma 1. *There exist positive constants, α_0, α_1, …, α_{n-1} and β_0, β_1, …, β_n, such that*

$$X(\mathsf{P}) \;=\; \frac{\alpha_0 + \alpha_1 F_1^{(n)}(\mathsf{P}) + \cdots + \alpha_{n-1} F_{n-1}^{(n)}(\mathsf{P})}{\beta_0 + \beta_1 F_1^{(n)}(\mathsf{P}) + \cdots + \beta_{n-1} F_{n-1}^{(n)}(\mathsf{P}) + \beta_n F_n^{(n)}(\mathsf{P})}. \tag{4}$$

where for $i \in \{0, \ldots, n-1\}$, $\alpha_i = (B/A)^i \displaystyle\sum_{k=0}^{n-i-1} A^{n-k-1} (\tau\delta)^k$,

and for $i \in \{0, \ldots, n\}$, $\beta_i = (B/A)^i A^n$.

Proof. Focus on a fixed, but arbitrary profile $\mathsf{P} = \langle \rho_1, \ldots, \rho_n \rangle$, and expand expression (1) to express $X(\mathsf{P})$ as a single fraction, $X(\mathsf{P}) = X_{\text{num}}/X_{\text{denom}}$.

Analyzing X_{denom}. Consider first the *denominator*, X_{denom}, of the fraction, which is simpler to analyze than the numerator. Easily, X_{denom} is the n-factor product $X_{\text{denom}} = \prod_{i=1}^{n} (B\rho_i + A)$. Using reasoning analogous to the proof of the Binomial Theorem, it is clear that, for each $i \in \{0, \ldots, n\}$, the coefficient, β_i, of $F_i(\mathsf{P})$ in X_{denom} is $\beta_i = B^i \cdot A^{n-i}$.

Analyzing X_{num}. We begin to analyze the *numerator*, X_{num}, of the fraction by expressing it as an n-term sum of products, where each product can be factored into an "I-J product," as follows.

$$X_{\text{num}} = \sum_{j=1}^{n} I_j \cdot J_j \quad \text{where} \quad I_j = \prod_{k=j+1}^{n} (B\rho_k + A) \quad \text{and} \quad J_j = \prod_{k=1}^{j-1} (B\rho_k + \tau\delta)\,.$$

Note that, for each $j \in \{0, \ldots, n\}$, the jth I-J product, $I_j \cdot J_j$, is the unique one that does not "mention" ρ_j.

Focus now on an arbitrary $i \in \{0, \ldots, n\}$ and an arbitrary i-*monomial* $\mu = \rho_{k_1} \cdots \rho_{k_i}$. Consider the coefficient of μ in $F_i(\mathsf{P})$. As just noted, μ appears as a subproduct of every I-J product $I_\ell \cdot J_\ell$ where $\ell \in \{0, \ldots, n\} \setminus \{k_1, \ldots, k_i\}$; focus on an arbitrary such index ℓ. Say that μ is "split" between I_ℓ and J_ℓ, in the sense that $0 \le h \le i$ of the ρ-values that appear in μ are "mentioned" in I_ℓ, and the other $i - h$ ρ-values are "mentioned" in J_ℓ. (The extreme cases, $h = 0$ and $h = i$, correspond, respectively, to μ's being a subproduct of J_ℓ or I_ℓ.) Reasoning analogous to that used in analyzing X_{denom} shows that μ's coefficient in the product $I_\ell \cdot J_\ell$ is

$$c_\mu \;\stackrel{\text{def}}{=}\; B^i \cdot \left(A^{n-h-\ell} \cdot (\tau\delta)^{\ell-(i-h)-1}\right).$$

Next, note that, given μ, the coefficient c_μ identifies index ℓ uniquely. Note also that, for each of the $i+1$ possible values for h, there is an I-J product containing μ as a subproduct, within which μ provides h ρ-values to the I-portion of the product and $i - h$ ρ-values to the J-portion. The just-exposed correspondences between I-J products and monomials and conversely allow us to conclude that the coefficient of $F_i(\mathsf{P})$ in X_{num} is a sum over I-J products, whose summands represent allocations of monomials the the I and J portions of the products. In detail: for each i, $\alpha_i = B^i \cdot \sum_{k=0}^{n-i-1} A^k \cdot (\tau\delta)^{n-k-i-1}$. $\square$

Expression (4) for $X(\mathsf{P})$ suggests a method for comparing clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ by comparing the symmetric functions of their respective profiles; see footnote 6.

Proposition 3. *Let clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ have, respectively, profiles P_1 and P_2. Cluster $\mathcal{C}_1$ outperforms cluster $\mathcal{C}_2$ whenever the following system of inequalities holds.*

For all pairs of indices $i, j \in \{0, \ldots, n\}$, with $i < j$

$$F_i^{(n)}(\mathsf{P}_1) \cdot F_j^{(n)}(\mathsf{P}_2) \ \geq \ F_i^{(n)}(\mathsf{P}_2) \cdot F_j^{(n)}(\mathsf{P}_1) \tag{5}$$

and for at least one i-j pair, the inequality is strict.

Proof. After "cross-multiplying" the fractions that express $X(\mathsf{P}_1)$ and $X(\mathsf{P}_2)$ in the form (4), we see that $X(\mathsf{P}_1) > X(\mathsf{P}_2)$ if and only if the following "α-β difference" is positive:

$$\Big(\alpha_0 F_0^{(n)}(\mathsf{P}_1) \ + \cdots + \ \alpha_{n-1} F_{n-1}^{(n)}(\mathsf{P}_1)\Big) \cdot \Big(\beta_0 F_0^{(n)}(\mathsf{P}_2) \ + \cdots + \ \beta_n F_n^{(n)}(\mathsf{P}_2)\Big)$$
$$- \Big(\alpha_0 F_0^{(n)}(\mathsf{P}_2) \ + \cdots + \ \alpha_{n-1} F_{n-1}^{(n)}(\mathsf{P}_2)\Big) \cdot \Big(\beta_0 F_0^{(n)}(\mathsf{P}_1) \ + \cdots + \ \beta_n F_n^{(n)}(\mathsf{P}_1)\Big)$$

Consider now arbitrary indices $i, j \in \{0, \ldots, n\}$, with $i < j$, and focus on the portion of the "α-β difference" that involves exactly the four quantities $F_i^{(n)}(\mathsf{P}_1)$, $F_i^{(n)}(\mathsf{P}_1)$, $F_j^{(n)}(\mathsf{P}_2)$, and $F_j^{(n)}(\mathsf{P}_2)$. One sees easily that this portion of the difference is precisely the product

$$(\alpha_i \beta_j - \alpha_j \beta_i) \cdot \Big(F_i^{(n)}(\mathsf{P}_1) \cdot F_j^{(n)}(\mathsf{P}_2) \ - \ F_i^{(n)}(\mathsf{P}_2) \cdot F_j^{(n)}(\mathsf{P}_1)\Big) \tag{6}$$

The following result will allow us to complete the proof.

Claim 1 *For all indices* i and $j > i \quad \alpha_i \beta_j > \alpha_j \beta_i$.

We verify Claim 1 by direct calculation. From Lemma 1, we know that

$$\Big[\alpha_i = B^i \cdot \sum_{k=0}^{n-1-i} A^{n-1-k-i} \cdot (\tau\delta)^k\Big] \quad \text{and} \quad \Big[\beta_i = B^i \cdot A^{n-i}\Big]$$

It follows that $\alpha_i \beta_j > \alpha_j \beta_i$; to wit:

$$\begin{aligned}
\alpha_i \beta_j - \alpha_j \beta_i \ &= \ \Big[B^i \cdot \sum_{k=0}^{n-1-i} A^{n-1-k-i} \cdot (\tau\delta)^k\Big] \cdot \Big[B^j \cdot A^{n-j}\Big] \\
&\qquad - \Big[B^j \cdot \sum_{k=0}^{n-1-j} A^{n-1-k-j} \cdot (\tau\delta)^k\Big] \cdot \Big[B^i \cdot A^{n-i}\Big] \\
&= \ B^{i+j} \cdot \Big(\sum_{k=0}^{n-1-i} A^{2n-1-k-i-j} \cdot (\tau\delta)^k \ - \ \sum_{k=0}^{n-1-j} A^{2n-1-k-j-i} \cdot (\tau\delta)^k\Big) \\
&= \ B^{i+j} \cdot \sum_{k=n-j}^{n-1-i} A^{2n-1-k-i-j} \cdot (\tau\delta)^k
\end{aligned}$$

The claimed inequality holds because every term in the last summation is positive. This verifies Claim 1.

To complete the argument, note that whenever Claim 1 holds for a pair of indices i and j, the product (6) is positive whenever (in fact, precisely when) the difference

$$F_i^{(n)}(\mathsf{P}_1) \cdot F_j^{(n)}(\mathsf{P}_2) \; - \; F_i^{(n)}(\mathsf{P}_2) \cdot F_j^{(n)}(\mathsf{P}_1)$$

is positive. Because Claim 1 in fact holds for all i and $j > i$, we see that the "α-β difference" is positive whenever (5) holds. This means, however, that $X(\mathsf{P}_1) > X(\mathsf{P}_2)$ whenever (5) holds, whence the proposition. $\square$

4.2 Statistical Moments of Profiles as Predictors of Power

The results in this section are enabled by a close relationship between some of the symmetric functions and standard statistical measures. For any profile $\mathsf{P} = \langle \rho_1, \ldots, \rho_n \rangle$:

• The *arithmetic* and *geometric* means of the ρ_i are:

$$\left[\bar{\rho} \stackrel{\text{def}}{=} \mathit{ARITH\text{-}MEAN}(\mathsf{P}) \; = \; \frac{1}{n} F_1^{(n)}\right]$$

and

$$\left[\mathit{GEO\text{-}MEAN}(\mathsf{P}) \; = \; \left(F_n^{(n)}\right)^{1/n}\right].$$

• The *variance* of the ρ_i is

$$\mathit{VAR}(\mathsf{P}) \; = \; \frac{1}{n}\left(\rho_1^2 + \cdots + \rho_n^2\right) \; - \; \left(\frac{1}{n} F_1^{(n)}(\mathsf{P})\right)^2 \tag{7}$$

while

$$F_2^{(n)}(\mathsf{P}) \; = \; \frac{1}{2}\left(F_1^{(n)}(\mathsf{P})\right)^2 \; - \; \frac{1}{2}\left(\rho_1^2 + \cdots + \rho_n^2\right). \tag{8}$$

Using these connections, we note that Proposition 3 has the following immediate consequence.

Theorem 5. *Say that cluster $\mathcal{C}_1$, with profile P_1, and cluster $\mathcal{C}_2$, with profile P_2, share the same mean speed.*

1. *If $\mathcal{C}_1$ outperforms $\mathcal{C}_2$ because of the system of inequalities (5), then $\mathit{VAR}(\mathsf{P}_1) > \mathit{VAR}(\mathsf{P}_2)$.*
2. *When $\mathcal{C}_1$ and $\mathcal{C}_2$ each has 2 computers, then the preceding sentence becomes a biconditional: $\mathcal{C}_1$ outperforms $\mathcal{C}_2$ if and only if $\mathit{VAR}(\mathsf{P}_1) > \mathit{VAR}(\mathsf{P}_2)$.*

Proof. Let $\mathsf{P}_1 = \langle \rho_{11}, \ldots, \rho_{1n} \rangle$ and $\mathsf{P}_2 = \langle \rho_{21}, \ldots, \rho_{2n} \rangle$, and say that $F_1^{(n)}(\mathsf{P}_1) = F_1^{(n)}(\mathsf{P}_2)$.

1. By (7), in this case:

$$[VAR(\mathsf{P}_1) \ > \ VAR(\mathsf{P}_2)] \ \text{ if and only if } \ [\rho_{11}^2 + \cdots + \rho_{1n}^2 \ > \ \rho_{21}^2 + \cdots + \rho_{2n}^2].$$

Because $(\rho_{11} + \cdots + \rho_{1n})^2 = \left(F_1^{(n)}(\mathsf{P}_1)\right)^2 = \left(F_1^{(n)}(\mathsf{P}_2)\right)^2 = (\rho_{21} + \cdots + \rho_{2n})^2$, equation (8) thus implies that $[F_2^{(n)}(\mathsf{P}_1) \ < \ F_2^{(n)}(\mathsf{P}_2)]$.

2. When $n = 2$, there are only two symmetric functions, $F_1^{(2)}$ and $F_2^{(2)}$. Hence, in this case, the sufficient condition

$$F_1^{(n)}(\mathsf{P}_1) \cdot F_2^{(n)}(\mathsf{P}_2) \ > \ F_1^{(n)}(\mathsf{P}_2) \cdot F_2^{(n)}(\mathsf{P}_1)$$

of Proposition 3 becomes "$F_2^{(n)}(\mathsf{P}_1) \ < \ F_2^{(n)}(\mathsf{P}_2)$." □

Theorem 5(2) exposes an unexpected fact: *Heterogeneity can be a source of computational power*. This discovery contrasts dramatically with the view of heterogeneity as a computational encumbrance that must be coped with—but that we would be better off without. The following result is immediate from Theorem 5(2).

Corollary 1. *Heterogeneity can actually lend power to a cluster. To wit, if one has two* 2*-computer clusters that share the same mean speed—*$\mathcal{C}_2$*, which is homogeneous, and* $\mathcal{C}_1$*, which is not—then* $\mathcal{C}_1$ *outperforms* $\mathcal{C}_2$*.*

4.3 Going beyond Theorem 5 and Corollary 1

It would be exciting if Theorem 5(2) held for clusters of arbitrary sizes, not just $n = 2$, for this would allow us to strengthen Corollary 1 to larger cluster sizes. This is an intuitively plausible hope because when $VAR(\mathsf{P}_1) > VAR(\mathsf{P}_2)$, one would expect P_1 to contain some ρ-values that are smaller than any of P_2's, and one might hope that these small values would pull $\mathcal{C}_1$'s HECR down below $\mathcal{C}_2$'s. (Because each $\rho_i \leq 1$, the small ρ-values should have greater impact on HECRs than do the large values.) But, alas, such is not the case. We performed the following simple simulation-based experiment for n-computer clusters, for various integers n; each trial consisted of the following steps. (We only sketch these steps roughly, because they are described in detail in our companion paper [13].)

1. Generate n-computer clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ with respective profiles P_1 and P_2, such that: (a) $F_1^{(n)}(\mathsf{P}_1) = F_1^{(n)}(\mathsf{P}_2)$ (so that $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same mean speed); (b) $VAR(\mathsf{P}_1) \neq VAR(\mathsf{P}_2)$.
2. Compare the HECRs of $\mathcal{C}_1$ and $\mathcal{C}_2$. Label $(\mathcal{C}_1, \mathcal{C}_2)$ "good" if the cluster with *larger* variance has the *smaller* HECR (i.e, is more powerful); otherwise, label the pair "bad."

We found "bad" cluster-pairs for clusters of every size $n = 2^k$ for $k \in \{2, 3, \ldots, 16\}$. Our disappointment was moderated by two facts.

1. Although the percentage of "bad" pairs grew to roughly 23% (reached when $n = 128$), it stayed steady thereafter. Thus, variance is a rather good predictor of the relative power of clusters that have equal mean speeds, being "correct" roughly 76% of the time.
2. The clusters in the "bad" pairs had rather small differences in HECR.

These results led us to seek a *variance threshold* θ, such that having variances differ by at least θ was (empirically) a "perfect" predictor of relative power. Specifically, we repeated a modified version of our simulation-based experiment, which replaced the condition "cluster with larger variance" by the condition

"cluster whose variance is larger by at least θ."

Our goal was to find the smallest value of θ for which this condition correctly identified the more powerful cluster in 100% of our trials! Thus, assuming, with no loss of generality, that $VAR(\mathsf{P}_1) > VAR(\mathsf{P}_2)$, we wanted to find in every trial that $\mathrm{HECR}(\mathcal{C}_1) < \mathrm{HECR}(\mathcal{C}_2)$. We determined experimentally that the value $\theta = 0.167$ achieves our goal:

Fact. *Using the described experimental procedures, we observe* $\mathrm{HECR}(\mathcal{C}_1) < \mathrm{HECR}(\mathcal{C}_2')$ 100% *of our trials when* $VAR(\mathsf{P}_1) > VAR(\mathsf{P}_2) + 0.167$.

We thus have an empirical version of Theorem 5(2) for clusters as large as 2^{16}. Ongoing simulation-based experiments in our companion paper [13] are extending this work with the goal of deepening our understanding of the role of statistical moments as predictors of computational power.

5 Conclusions and Projections

Heterogeneity is almost ubiquitous in modern computing platforms, yet sources such as [1] show that we have yet to unlock some very basic secrets about this phenomenon. One finds in [1] a simple computational problem (the CEP) all of whose optimal solutions for a given cluster $\mathcal{C}$ can be characterized and shown to be functions of $\mathcal{C}$'s *(heterogeneity) profile* (Theorems 1 and 2). We build on these results by using the quality of cluster $\mathcal{C}$'s solution to the CEP as a measure of $\mathcal{C}$'s computational power. We thereby expose properties of $\mathcal{C}$'s profile that determine its computational power. Perhaps our most interesting results—certainly our favorites—show the following: (1) If one can replace just one of $\mathcal{C}$'s computers by a faster one, then: (a) If the new computer is *additively* faster than the old one, then the most advantageous computer to replace is $\mathcal{C}$'s fastest one (Theorem 3). (b) The same is true for *multiplicative* speedups, *unless either* $\mathcal{C}$'s fastest computer is already "very fast" *or* the speedup factor is "very aggressive" (Theorem 4). (2) The symmetric functions of $\mathcal{C}$'s computers' speeds play a major role in determining $\mathcal{C}$'s power (Lemma 1 and Proposition 3); this fact suggests a similarly large role for the statistical moments of $\mathcal{C}$'s computers' speeds (Theorem 5). (3) Heterogeneity can enhance the power of a cluster (Corollary 1). Ongoing research, whose initial results are reported in [13], strives to better understand topics $(2, 3)$, via both (simulation-based) experimentation and analysis.

Acknowledgments. This research was supported in part by NSF Grants CNS-0615170 and CNS-0905399. The authors benefited from conversations with A.M. Maciejewski, H.J. Siegel, and F. Vivien.

References

[1] M. Adler, Y. Gong, A.L. Rosenberg (2008): On "exploiting" node-heterogeneous clusters optimally. *Theory of Computing Systems 42*, 465–487.

[2] T.E. Anderson, D.E. Culler, D.A. Patterson, and the NOW Team (1995): A case for NOW (networks of workstations). *IEEE Micro 15*, 54–64.

[3] M. Banikazemi, V. Moorthy, D.K. Panda (1998): Efficient collective communication on heterogeneous networks of workstations. *Intl. Conf. on Parallel Processing (ICPP)*, 460–467.

[4] C. Banino, O. Beaumont, L. Carter, J. Ferrante, A. Legrand, Y. Robert (2004): Scheduling strategies for master-slave tasking on heterogeneous processor grids. *IEEE Trans. Parallel and Distr. Systs. 15*, 319–330.

[5] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, Y. Robert (2002): Bandwidth-centric allocation of independent tasks on heterogeneous platforms. *Intl. Parallel and Distr. Processing Symp. (IPDPS).*

[6] O. Beaumont, A. Legrand, Y. Robert (2003): The master-slave paradigm with heterogeneous processors. *IEEE Trans. Parallel and Distr. Systs. 14*, 897–908.

[7] O. Beaumont, L. Marchal, Y. Robert (2005): Scheduling divisible loads with return messages on heterogeneous master-worker platforms. *12th Intl. High-Performance Computing Conf. Lecture Notes in Computer Science 3769*, Springer, Berlin, 498–507.

[8] P.B. Bhat, V.K. Prasanna, C.S. Raghavendra (1999): Efficient collective communication in distributed heterogeneous systems. *19th IEEE Intl. Conf. on Distributed Computing Systems (ICDCS).*

[9] R. Buyya, D. Abramson, J. Giddy (2001): A case for economy Grid architecture for service oriented Grid computing. *10th Heterogeneous Computing Wkshp. (HCW).*

[10] 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 Systs. 25,* 599–616.

[11] W. Cirne and K. Marzullo (1999): The Computational Co-Op: Gathering clusters into a metacomputer. *13th Intl. Parallel Processing Symp. (ICPP)*, 160–166.

[12] F. Cappello, P. Fraigniaud, B. Mans, A.L. Rosenberg (2005): An algorithmic model for heterogeneous clusters: rationale and experience. *Intl. J. Foundations of Computer Science 16*, 195–216.

[13] R.C. Chiang, A.A. Maciejewski, A.L. Rosenberg, H.J. Siegel (2010): Statistical predictors of computing power in heterogeneous clusters. Submitted for publication; available at (http://www.engr.colostate.edu/∼chilung/hetero2.pdf).

[14] P.-F. Dutot (2003): Master-slave tasking on heterogeneous processors. *17th Intl. Parallel and Distributed Processing Symp. (IPDPS).*

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

[16] P. Fraigniaud, B. Mans, A.L. Rosenberg (2005): Efficient trigger-broadcasting in heterogeneous clusters. *J. Parallel and Distributed Computing 65,* 628–642.

[17] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, M. Lebofsky (2001): SETI@home—Massively distributed computing for SETI. In *Computing in Science and Engineering 3*, 78–83.

[18] P. Liu and T.-H. Sheng (2000): Broadcast scheduling optimization for heterogeneous clusters systems. *12th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, 129–136.

[19] P. Liu and D.-W. Wang (2000): Reduction optimization in heterogeneous cluster environments. *Intl. Parallel and Distr. Processing Symp. (IPDPS).*

[20] J. Mache, R. Broadhurst, J. Ely (2000): Ray tracing on cluster computers. *Intl. Conf. on Parallel and Distr. Processing Techniques and Applications (PDPTA)*, 509–515.

[21] G. Malewicz, A.L. Rosenberg, M. Yurkewych (2006): Toward a theory for scheduling DAGs in Internet-based computing. *IEEE Trans. Comput. 55*, 757–768.

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

[23] R. Prakash and D.K. Panda (1998): Designing communication strategies for heterogeneous parallel systems. *Parallel Computing 24*, 2035–2052.

[24] A.L. Rosenberg (1994): Needed: a theoretical basis for heterogeneous parallel computing. In *Developing a Computer Science Agenda for High-Performance Computing* (U. Vishkin, ed.), ACM Press, N.Y. (1994) 137–142.

[25] A.L. Rosenberg and R.C. Chiang (2009): Toward understanding heterogeneity in computing. Available at (http://www.cs.umass.edu/∼rsnbrg/hetero.pdf).

[26] A.S. Tosun and A. Agarwal (2000): Efficient broadcast algorithms for heterogeneous networks of workstations. *13th Intl. Conf. on Parallel and Distr. Comput. Systs. (PDCS).*

[27] S.W. White and D.C. Torney (1993): Use of a workstation cluster for the physical mapping of chromosomes. *SIAM NEWS*, March, 1993, 14–17.