

Toward Understanding Heterogeneity in Computing*

Arnold L. Rosenberg and Ron C. Chiang

Electrical & Computer Engineering, Colorado State Univ., 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 the most 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 just “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 formal, 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, 21]; modalities of Internet-based computing [20] such as grid computing [9, 14], global computing [11], volunteer computing [16], and cloud computing [10]. The difficulty of scheduling complex computations 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 [23]. There has since been an impressive amount of first-rate work on this topic—focusing largely on collective communication [3, 4, 8, 15, 17, 22, 24], but also studying important scheduling issues [1, 5, 6, 7, 13, 18]. That said, sources such as [1] show that there is still much to learn about this important topic—including the questions in the abstract.

This research was supported in part by NSF Grants CNS-0615170 and CNS-0905399.

1.1 “Understanding” Heterogeneity

We have access to $n + 1$ computers: the *server* C_0 and a *cluster* \mathcal{C} comprising n computers, C_1, \dots, C_n , which may differ dramatically in computing powers. (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 $\langle \rho_1, \dots, \rho_n \rangle$ is \mathcal{C} ’s (**heterogeneity**) profile. For convenience:

- we index the C_i in nonincreasing order of power, so that $\rho_1 \geq \dots \geq \rho_n$;
- 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} “outperforms” (or, is more “powerful” than) cluster \mathcal{C}' ? We invoke the framework of a remarkable result from [1] that *characterizes all optimal solutions* to a simple scheduling problem for heterogeneous clusters. We thereby *isolate the heterogeneity* of \mathcal{C} and \mathcal{C}' as the only respect in which they differ: both are performing the same computation optimally, given their respective resources.

Highlight 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 3.2). This is always true for 2-computer clusters; for other sizes, the advantage takes hold when the difference in variances is sufficiently large. (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, e.g., data smoothing, pattern matching, ray tracing, Monte-Carlo simulations, chromosome mapping [16, 19, 25].) *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 message, to C_0 . At most one intercomputer message can be in transit at a time. Consider the following problem.

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

¹Note that faster computers have smaller ρ -values.

²“Size” quantifies specification; “complexity” quantifies computation.

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 our study is on experimental illustration and elucidation of the analytical results we derive; therefore, we relegate all proofs of new results to an appendix.

2 Worksharing Protocols and Work Production

2.1 The Architectural Model [12]

We assume that C ’s computers are (*architecturally*) *balanced*: 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.³ We ignore the fixed costs associated with transmitting \mathcal{M} —the end-to-end latency of the first packet and the set-up cost—because their impacts fade over long lifespans L . A final important feature: *At most one intercomputer message can be in transit at any moment*. The following table provides intuition about the sizes of the model’s parameters.

<i>Parameter</i>	<i>Wall-Clock Time/Rate</i>
Transit rate (pipelined network): τ	1 μ sec per work unit
Packaging rate: π	10 μ sec per work unit
Result-size rate: δ	1 work unit per work unit

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, 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.*

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 following action/time diagram (*not to scale*):

³We equate packaging and unpackaging times; this is consistent with most actual architectures.

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	$\pi_i w$	$\rho_i w$	$\pi_i \delta w$	$\tau \delta w$	$\pi_0 \delta w$

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

The described protocol is summarized in diagram (2.1) (*not to scale*). Note that in this diagram, 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
				$(\pi_3 + \rho_3)w_3$		$(\pi_3 + \tau)\delta w_3$

(2.1)

2.3 Protocols that Solve the CEP Optimally

The **FIFO protocol** is defined by coincident startup and finishing indexings ($\Sigma = \Phi$), as in (2.1). Provided only that 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 for *every* heterogeneity profile, we use these solutions as our vehicle for studying clusters’ heterogeneity.

2.4 Two Ways to Measure a Cluster’s Computing Power

2.4.1 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 1.

Sample Values for Perspective	
Quantity	Wall-Clock Time/Rate
$A = \pi + \tau$:	11 μ sec 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 1: Sample parameter values.

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

$$X(\mathbf{P}) = \sum_{i=1}^n \frac{1}{B\rho_i + A} \cdot \prod_{j=1}^{i-1} \frac{B\rho_j + \tau\delta}{B\rho_j + A}, \quad (2.2)$$

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

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

In Section A, we verify that FIFO protocols allocate work to \mathcal{C} ’s computers in proportion to their speeds. This is a sort of “reality check” on our model, because intuition strongly suggests that optimal work allocations must be proportional.

2.4.2 The Homogeneous-Equivalent Computing Rate (HECR). $X(\mathbf{P})$ is a viable and tractable measure but not very perspicuous. We propose, therefore, the following alternative measure for a *heterogeneous* cluster \mathcal{C} with profile $\mathbf{P} = \langle \rho_1, \dots, \rho_n \rangle$. Consider a *homogeneous* cluster $\mathcal{C}^{(\rho)}$, with profile $\mathbf{P}^{(\rho)} = \langle \rho, \dots, \rho \rangle$ for some $\rho \leq 1$. \mathcal{C} 's homogeneous-equivalent computation rate (HECR), $\rho_{\mathcal{C}}$, is the largest ρ such that $X(\mathbf{P}^{(\rho_{\mathcal{C}})}) \geq X(\mathbf{P})$.⁴

Proposition 1.⁵
$$\rho_{\mathcal{C}} = \frac{A - \tau\delta}{B - \left(1 - (A - \tau\delta)X(\mathbf{P})\right)^{1/n} B} - \frac{A}{B}.$$

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 integer function f , abbreviate the sequence $\langle f(1), \dots, f(n) \rangle$ via the notation $\langle f(i)|_{i=1}^n \rangle$.

Cluster \mathcal{C}_1 has profile $\mathbf{P}_1^{(n)} = \langle (1 - (i - 1)/n)|_{i=1}^n \rangle$, meaning that each $\rho_i = 1 - (i - 1)/n$; cluster \mathcal{C}_2 has profile $\mathbf{P}_2^{(n)} = \langle (1/i)|_{i=1}^n \rangle$, meaning that each $\rho_i = 1/i$. Note that 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 this range, namely, $[1/n, 1/2]$. When $n = 8$, for example, $\mathbf{P}_1^{(8)} = \langle 1, \frac{7}{8}, \dots, \frac{1}{8} \rangle$, and $\mathbf{P}_2^{(8)} = \langle 1, \frac{1}{2}, \dots, \frac{1}{8} \rangle$. Note that most of \mathcal{C}_2 's computers are faster than their counterparts in \mathcal{C}_1 , a fact that should be reflected in the HECR-values of the two clusters: \mathcal{C}_1 , being slower than \mathcal{C}_2 , should have a larger HECR-value (Proposition 1). Table 2 presents the HECR-values of three instantiations of clusters \mathcal{C}_1 and \mathcal{C}_2 : with 8, 16, and 32 computers. As expected, \mathcal{C}_1 's HECR-value 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 $\leq 1/2$, while half of \mathcal{C}_1 's computers have ρ -values $> 1/2$, we know intuitively that \mathcal{C}_2 's speed advantage over \mathcal{C}_1 should increase with larger instantiations of the two clusters. Indeed, the entries in Table 2 demonstrate this trend, as the ratio of \mathcal{C}_2 's HECR-value to \mathcal{C}_1 's improves from roughly 1.7 for 8 computers 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	$\langle (1 - (i - 1)/n) _{i=1}^n \rangle$	0.366	0.298	0.251
\mathcal{C}_2	$\langle (1/i) _{i=1}^n \rangle$	0.216	0.116	0.060

Table 2: HECR values for sample competing heterogeneous clusters

⁴Because we use the value of ρ to calibrate a *heterogeneous* cluster's power, we must violate our normalization convention and allow ρ to assume any value ≤ 1 .

⁵Proof appears in Section B.1.

3 What Determines a Cluster’s Power?

3.1 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 $0 < \psi < 1$.

3.1.1 Faster clusters complete more work. Speedups always matter for FIFO protocols.

Proposition 2. ⁶ *FIFO protocols complete more work on faster clusters; i.e., given profiles $\mathbf{P} = \langle \rho_1, \dots, \rho_{i-1}, \rho_i, \rho_{i+1}, \dots, \rho_n \rangle$ and $\mathbf{P}' = \langle \rho_1, \dots, \rho_{i-1}, \rho'_i, \rho_{i+1}, \dots, \rho_n \rangle$: if $\rho'_i < \rho_i$, then for all L , $W(L; \mathbf{P}') > W(L; \mathbf{P})$.*

3.1.2 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? We focus on a cluster \mathcal{C} whose heterogeneity profile is $\mathbf{P} = \langle \rho_1, \dots, \rho_n \rangle$, where each $\rho_k \geq \rho_{k+1}$. Let i and $j > i$ be two of \mathcal{C} ’s power indices. We compare the benefits of speeding up C_i vs. speeding up C_j . Of course, in order for this question to make sense, we must have $\rho_i > \rho_j$, i.e., a *strict* inequality between ρ_i and ρ_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 Theorem 1.2 assures us that Σ has no impact on $W(L; \mathbf{P})$. Specifically, we have \mathcal{C} employ a startup ordering $\Sigma = \langle s_1, \dots, s_{n-1}, s_n \rangle$ for which $s_n = i$ and $s_{n-1} = j$. Under such an ordering, we can rewrite expression (2.2) for $X(\mathbf{P})$ in the following convenient way, using two quantities that are independent of ρ_i and ρ_j and that, importantly, are both *positive*.

$$X(\mathbf{P}) = \frac{A + B(\rho_{s_{n-1}} + \rho_{s_n}) + \tau\delta}{A^2 + AB(\rho_{s_{n-1}} + \rho_{s_n}) + B^2\rho_{s_{n-1}}\rho_{s_n}} \cdot Y(\mathbf{P}) + Z(\mathbf{P}) \quad (3.1)$$

where

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

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

A. The additive-speedup scenario. We compare two profiles: $\mathbf{P}^{(i)}$ is obtained by speeding up the slower computer (of the two we are focusing on), C_i ; $\mathbf{P}^{(j)}$ is obtained by speeding up

⁶Proof appears in Section B.2.

the faster computer, 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 .)

$$\begin{aligned} \mathbf{P}^{(i)} &= \langle \rho_1, \dots, \rho_{i-1}, \rho_i - \varphi, \rho_{i+1}, \dots, \rho_{j-1}, \rho_j, \rho_{j+1}, \dots, \rho_n \rangle \\ \mathbf{P}^{(j)} &= \langle \rho_1, \dots, \rho_{i-1}, \rho_i, \rho_{i+1}, \dots, \rho_{j-1}, \rho_j - \varphi, \rho_{j+1}, \dots, \rho_n \rangle \end{aligned}$$

Theorem 3.⁷ *Under the additive-speedup scenario, the most advantageous single computer to speed up is \mathcal{C} 's fastest computer.*

Additive speedup “in action.” We compare $\mathbf{P}^{(i)}$ and $\mathbf{P}^{(j)}$ via the work ratios $W(L; \mathbf{P}^{(i)})/W(L; \mathbf{P})$ and $W(L; \mathbf{P}^{(j)})/W(L; \mathbf{P})$. Proposition 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 $\mathbf{P} = \langle 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \rangle$ and the (additive) speedup term $\varphi = \frac{1}{16}$. (Note that C_1 is \mathcal{C} 's slowest computer, and C_4 is its fastest.) Table 3 presents the work ratios obtained by speeding up each of \mathcal{C} 's computers in turn by the additive term φ . Fig. 1 presents the same results graphically.

i	Profile $\mathbf{P}^{(i)}$	Work ratio $W(L; \mathbf{P}^{(i)}) \div W(L; \mathbf{P})$
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 3: The work ratios as each of \mathcal{C} 's 4 computers is sped up additively.

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

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

$$\begin{aligned} \mathbf{P}^{[i]} &= \langle \rho_1, \dots, \rho_{i-1}, \psi\rho_i, \rho_{i+1}, \dots, \rho_{j-1}, \rho_j, \rho_{j+1}, \dots, \rho_n \rangle \\ \mathbf{P}^{[j]} &= \langle \rho_1, \dots, \rho_{i-1}, \rho_i, \rho_{i+1}, \dots, \rho_{j-1}, \psi\rho_j, \rho_{j+1}, \dots, \rho_n \rangle \end{aligned}$$

The driving question of which computer to speed up has a more complicated answer in the multiplicative-speedup scenario than in the additive-speedup scenario.

Theorem 4.⁸ *Let \mathcal{C} contain computers C_i and C_j , with respective computation rates ρ_i and*

⁷Proof appears in Section B.3.

⁸Proof appears in Section B.4.

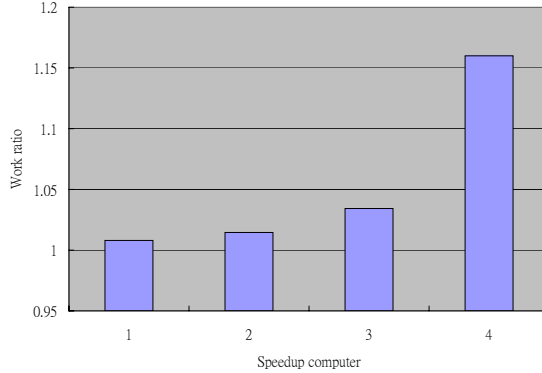


Figure 1: The work ratios as each of \mathcal{C} 's 4 computers is sped up additively.

$\rho_j < \rho_i$. Under the multiplicative-speedup scenario with speedup factor ψ :

- 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 .
- 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 .

Informal translation. *It is more advantageous to speed up the faster computer, unless either both computers are already “very fast” or the speedup factor ψ is “very small.”*

The values of “very fast” and “very small” depend on the relation between the problem-specific quantity $\psi\rho_i\rho_j$ and the environment-specific quantity $A\tau\delta/B^2$. For perspective, with our earlier values (see Table 1), we have $A\tau\delta/B^2 \approx 1.1 \times 10^{-5}$; hence, *we expect that speeding up the faster computer will usually be the better option.*

Multiplicative speedup “in action.” The experiment that illustrates multiplicative speedup “in action” is quite different from the one we used to illustrate additive speedup. We observe the two conditions of Theorem 4 “in action” via a sequence of snapshots of a cluster that experiences a sequence of multiplicative speedups. The snapshots depict a two-phase experiment that begins with a 4-computer homogeneous cluster \mathcal{C} whose profile is $\mathbf{P} = \langle 1, 1, 1, 1 \rangle$ and that iteratively optimally speeds \mathcal{C} up via the speedup factor $\psi = 1/2$. The first phase illustrates the first condition in the proposition, as \mathcal{C} 's profile “improves” (because of the speedups) from its initial value of $\mathbf{P} = \langle 1, 1, 1, 1 \rangle$ to the value $\mathbf{P}' = \langle 5/80, 5/80, 5/80, 5/80 \rangle$. Once all of \mathcal{C} 's computers achieve this speed, subsequent speedups follow the second condition in the proposition. Although we continue to speed up cluster \mathcal{C} via the factor $\psi = 1/2$, we observe the very different result predicted by the second condition.

We have increased the value of the τ parameter for this experiment, from its earlier $1 \mu\text{sec}/\text{work unit}$ to $200 \mu\text{sec}/\text{work unit}$. With the original value of τ , the ρ -value of \mathcal{C} 's fastest computer becomes too small to be seen when displayed with the ρ -values of the slower computers. Increasing the value of τ was an easy expedient for enhancing visibility while still illustrating the proposition.

Each bar-graph in Figs. 2 and 3 represents the then-current profile of \mathcal{C} after one round of the experiment: when the four bars in each graph have respective heights $\rho_1, \rho_2, \rho_3,$ and 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 successor profiles to profile P_i , call them $P_i^{[1]}, P_i^{[2]}, P_i^{[3]},$ and $P_i^{[4]}$. Each profile $P_i^{[j]}$ 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, and we select the profile with the largest work-production to be profile P_i 's successor, P_{i+1} . In case of ties—wherein speeding up computers C_j and C_k yield the same work-production—then we choose to speed up the computer with the larger index. We discuss each phase in turn.

Phase 1: not all computers are “very fast.” As we observe in Fig. 2, this phase of the ex-

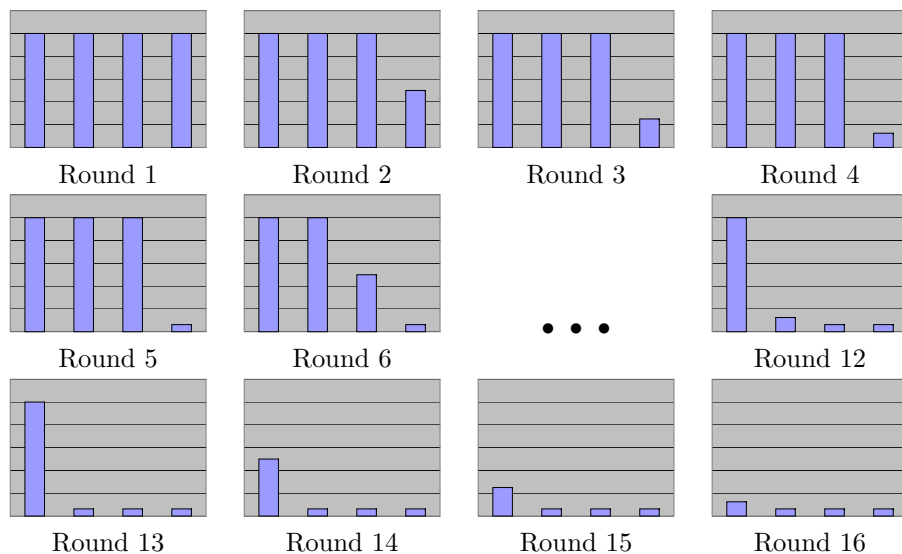


Figure 2: Speeding up a cluster when not all computers are “very fast.”

periment 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. Note that we choose to speed up computer C_4 in round 1 because of our tie-breaking mechanism, but we then select it in rounds 2–4 because of the first condition in Theorem 4. At round 5, the second condition in

Theorem 4 tells us not to speed up computer C_4 again. At that point, we again invoke the tie-breaking mechanism to select computer C_3 , and the just-described cycle repeats, until \mathcal{C} ends up in round 17 with the profile $\langle 0.0625, 0.0625, 0.0625, 0.0625 \rangle$. At this point, phase 2 of the experiment begins.

Phase 2: all computers are “very fast.” At this point, the second condition in the proposition is triggered, and we observe the repeated selection of the slowest computer as the best one to speed up (with the tie-breaking mechanism used as necessary). Fig. 3 illustrates the pattern of speeding up a cluster under the second condition of Theorem 4.

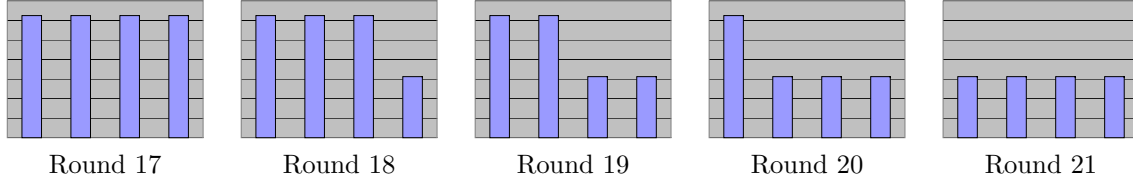


Figure 3: Speeding up a cluster when all computers are “very fast.”

3.2 Predicting Power via Moments of Heterogeneity Profiles

Proposition 2 tells us that if cluster \mathcal{C}_1 's profile $\langle \rho_{1,1}, \dots, \rho_{1,n} \rangle$ *minorizes* cluster \mathcal{C}_2 's profile $\langle \rho_{2,1}, \dots, \rho_{2,n} \rangle$, in the sense that (a) for every index i , $\rho_{1,i} \leq \rho_{2,i}$, (b) for at least one index i , $\rho_{1,i} < \rho_{2,i}$, then \mathcal{C}_1 outperforms \mathcal{C}_2 . It is easy to identify situations in which \mathcal{C}_1 outperforms \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 $\langle 0.5, 0.5 \rangle$. This section is devoted to identifying situations in which the *symmetric functions* and (statistical) moments of two clusters' sets of ρ -values can be used to predict their relative performance. Regarding such moments: Note that the preceding 2-computer clusters show that having a better *mean speed* does not guarantee better performance.

A function $F(x_1, \dots, x_n)$ 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 $k \in \{1, \dots, n\}$, we denote by $F_k^{(n)}$ the symmetric function that has n variables grouped as products of k . It simplifies our analysis clerically if we allow the index k to assume the value $k = 0$ also, with the convention that, for all n , $F_0^{(n)} \equiv 1$. All of our values are computation rates ρ_i , so the first two families of $F_k^{(n)}$ functions (excluding the degenerate value $k = 0$) are exhibited in Table 4.

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

Table 4: The first two families of symmetric functions.

Note. *There is a close relationship between some of the symmetric functions and standard statistical “moments.” For any profile $\mathbf{P} = \langle \rho_1, \dots, \rho_n \rangle$:*

- $F_1^{(n)}$ (resp., $F_n^{(n)}$) is the unnormalized arithmetic (resp., geometric) mean of the ρ_i .
- On the one hand, the variance of the ρ_i is given by

$$\text{VAR}(\mathbf{P}) = \frac{1}{n}(\rho_1^2 + \dots + \rho_n^2) - \left(\frac{1}{n}(\rho_1 + \dots + \rho_n) \right)^2 \quad (3.2)$$

$$\text{while } F_2^{(n)}(\mathbf{P}) = \frac{1}{2}((\rho_1 + \dots + \rho_n)^2 - (\rho_1^2 + \dots + \rho_n^2)). \quad (3.3)$$

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

Lemma 1.⁹ *There exist positive constants, $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$ and $\beta_0, \beta_1, \dots, \beta_n$, such that*

$$X(\mathbf{P}) = \frac{\alpha_0 + \alpha_1 F_1^{(n)}(\mathbf{P}) + \dots + \alpha_{n-1} F_{n-1}^{(n)}(\mathbf{P})}{\beta_0 + \beta_1 F_1^{(n)}(\mathbf{P}) + \dots + \beta_{n-1} F_{n-1}^{(n)}(\mathbf{P}) + \beta_n F_n^{(n)}(\mathbf{P})} \quad (3.4)$$

$$\text{Specifically: } \begin{cases} \text{for each } i \in \{0, \dots, n-1\}, & \alpha_i = B^i \cdot \sum_{k=0}^{n-i-1} A^k \cdot (\tau\delta)^{n-k-i-1} \\ \text{for each } i \in \{0, \dots, n\}, & \beta_i = B^i \cdot A^{n-i}. \end{cases}$$

Expression (3.4) suggests a method for comparing profiles \mathbf{P}_1 and \mathbf{P}_2 by comparing their respective sets of symmetric functions.

Proposition 3.¹⁰ *Let clusters \mathcal{C}_1 and \mathcal{C}_2 have, respectively, profiles \mathbf{P}_1 and \mathbf{P}_2 . Cluster \mathcal{C}_1 outperforms cluster \mathcal{C}_2 whenever the following system of inequalities holds.*

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

$$F_i^{(n)}(\mathbf{P}_1) \cdot F_j^{(n)}(\mathbf{P}_2) \geq F_i^{(n)}(\mathbf{P}_2) \cdot F_j^{(n)}(\mathbf{P}_1) \quad (3.5)$$

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

⁹Proof appears in Section B.5.

¹⁰Proof appears in Section B.6.

Theorem 5.¹¹ Say that cluster \mathcal{C}_1 , with profile \mathbf{P}_1 , and cluster \mathcal{C}_2 , with profile \mathbf{P}_2 , share the same mean speed. If cluster \mathcal{C}_1 outperforms cluster \mathcal{C}_2 because of the system of inequalities (3.5), then $\text{VAR}(\mathbf{P}_1) > \text{VAR}(\mathbf{P}_2)$. 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 $\text{VAR}(\mathbf{P}_1) > \text{VAR}(\mathbf{P}_2)$.

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 .

It would be exciting if the final sentence of Theorem 5 held for clusters of arbitrary sizes, not just $n = 2$. This is an intuitively plausible hope because when $\text{VAR}(\mathbf{P}_1) > \text{VAR}(\mathbf{P}_2)$, one would expect \mathbf{P}_1 to contain some ρ -values that are smaller than any of \mathbf{P}_2 's, and one would 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 experiment for n -computer clusters, for various integers n ; each trial consisted of the following steps.

1. Randomly generate n -computer clusters \mathcal{C}_1 and \mathcal{C}_2 , with respective profiles \mathbf{P}_1 and \mathbf{P}_2 and mean speeds $\bar{\rho}_1$ and $\bar{\rho}_2$.
 - (a) Alter the speeds of \mathcal{C}_2 's computers by the factor $\bar{\rho}_1/\bar{\rho}_2$ (giving us cluster \mathcal{C}'_2 with profile \mathbf{P}'_2) so that \mathcal{C}_1 and \mathcal{C}'_2 have the same mean speed (namely, $\bar{\rho}_1$).
 - (b) Reject the current pair if $\text{VAR}(\mathbf{P}_1) = \text{VAR}(\mathbf{P}'_2)$ (which should be quite unlikely).
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 every size $n > 2$. Moderating our disappointment is the fact that the clusters in the “bad” pairs had rather small differences in HECR. We therefore selected a *variance threshold* θ , and we repeated a modified version of our experiment. Say, to be definite, with no loss of generality, that $\text{VAR}(\mathbf{P}_1) > \text{VAR}(\mathbf{P}'_2)$. We replaced the condition “cluster with larger variance”—in this case, $\text{VAR}(\mathbf{P}_1) > \text{VAR}(\mathbf{P}'_2)$ —by the condition

“cluster whose variance is larger by at least θ ”—in this case, $\text{VAR}(\mathbf{P}_1) \geq \text{VAR}(\mathbf{P}'_2) + \theta$.

Our goal was to find the smallest values of θ for which $\text{HECR}(\mathcal{C}_1) < \text{HECR}(\mathcal{C}'_2)$ 100% of the time! We experimentally determined for pairs $(\mathcal{C}_1, \mathcal{C}'_2)$ of 8-computer clusters:

Fact. Using the described experimental procedures, we observe $\text{HECR}(\mathcal{C}_1) < \text{HECR}(\mathcal{C}'_2)$ 100% of the time when $\theta = 0.15$, i.e., when $\text{VAR}(\mathbf{P}_1) > \text{VAR}(\mathbf{P}'_2) + 0.15$.

We thus have a version of Theorem 5's final sentence that, empirically, holds for 8-computer clusters. Ongoing experiments are extending this work to larger clusters, with the hope that θ_n , the n -computers/cluster analogue of threshold $\theta(= \theta_8)$, grows slowly as a function of n .

¹¹Proof appears in Section B.7.

4 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 (Theorem 1) and shown to be functions of \mathcal{C} 's (*heterogeneity profile*) (Theorem 2). We build on these results to expose properties of \mathcal{C} 's profile that determine the quality of solutions to the CEP for \mathcal{C} . 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* all of \mathcal{C} 's computers are already “very fast” *or* the speedup factor is “very aggressive” (Theorem 4). (2) The symmetric functions of \mathcal{C} 's computers' speeds play a large role in determining \mathcal{C} 's power (Lemma 1, Proposition 3), which 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 strives to better understand topics (2,3), via experimentation and analysis.

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. *ICPP'00*, 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. Sys.* 15, 319–330.
- [5] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, Y. Robert (2002): Bandwidth-centric allocation of independent tasks on heterogeneous platforms. *IPDPS'02*.
- [6] O. Beaumont, A. Legrand, Y. Robert (2003): The master-slave paradigm with heterogeneous processors. *IEEE Trans. Parallel and Distr. Sys.* 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. LNCS 3769*, Springer, Berlin, 498–507.
- [8] P.B. Bhat, V.K. Prasanna, C.S. Raghavendra (1999): Efficient collective communication in distributed heterogeneous systems. *ICDCS'99*.

- [9] R. Buyya, D. Abramson, J. Giddy (2001): A case for economy Grid architecture for service oriented Grid computing. *HCW'01*.
- [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 Sys.*, to appear.
- [11] W. Cirne and K. Marzullo (1999): The Computational Co-Op: gathering clusters into a metacomputer. *ICPP'99*, 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] P.-F. Dutot (2003): Master-slave tasking on heterogeneous processors. *IPDPS'03*.
- [14] I. Foster and C. Kesselman [eds.] (2004): *The Grid: Blueprint for a New Computing Infrastructure (2nd Ed.)*. Morgan-Kaufmann, San Francisco.
- [15] P. Fraigniaud, B. Mans, A.L. Rosenberg (2005): Efficient trigger-broadcasting in heterogeneous clusters. *J. Parallel and Distributed Computing 65* (2005) 628–642.
- [16] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, M. Lebofsky (2000): SETI@home: massively distributed computing for SETI. In *Computing in Science and Engineering* (P.F. Dubois, Ed.) IEEE Computer Soc. Press, Los Alamitos, CA.
- [17] P. Liu and T.-H. Sheng (2000): Broadcast scheduling optimization for heterogeneous clusters systems. *SPAA'00*, 129–136.
- [18] P. Liu and D.-W. Wang (2000): Reduction optimization in heterogeneous cluster environments. *IPDPS'00*.
- [19] J. Mache, R. Broadhurst, J. Ely (2000): Ray tracing on cluster computers. *PDPTA'00*, 509–515.
- [20] G. Malewicz, A.L. Rosenberg, M. Yurkewych (2006): Toward a theory for scheduling DAGs in Internet-based computing. *IEEE Trans. Comput. 55*, 757–768.
- [21] G.F. Pfister (1995): *In Search of Clusters*. Prentice-Hall.
- [22] R. Prakash and D.K. Panda (1998): Designing communication strategies for heterogeneous parallel systems. *Parallel Computing 24*, 2035–2052.
- [23] 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.
- [24] A.S. Tosun and A. Agarwal (2000): Efficient broadcast algorithms for heterogeneous networks of workstations. *PDCS'00*.
- [25] 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.

A FIFO Protocols Allocate Work Proportionally

How should one define *work allocation that is proportional to computer speeds* within the context of our model? Certainly, the parameters in our model will not permit the ideal notion of proportionality that is embodied in the equation $w_i/w_{i+1} = \rho_{i+1}/\rho_i$. The following result shows that FIFO-based work allocations do exhibit a strong level of proportionality in their work allocations. Inequality (A.1) is validated in Section A. Focus on a cluster \mathcal{C} that has the heterogeneity profile $\mathbf{P} = \langle \rho_1, \dots, \rho_n \rangle$, where $\rho_1 \geq \dots \geq \rho_n$.

Proposition 4. *FIFO Protocols allocate work in proportion to computer speeds, in the following sense. If the FIFO protocol employs the startup indexing $s_i = i$ for all $i \in \{1, \dots, n-1\}$, then the work allocations satisfy*

$$\frac{\rho_{i+1}}{\rho_i} + A/B < \frac{w_i}{w_{i+1}} < (1 + A/B + \tau/B) \cdot \frac{\rho_{i+1}}{\rho_i}. \quad (\text{A.1})$$

For perspective, using our sample parameter values, inequalities (A.1) become

$$\begin{aligned} \text{fine-grain tasks:} \quad & \frac{\rho_{i+1}}{\rho_i} + 0.0001 < \frac{w_i}{w_{i+1}} < 1.00012 \cdot \frac{\rho_{i+1}}{\rho_i} \\ \text{coarse-grain tasks:} \quad & \frac{\rho_{i+1}}{\rho_i} + 0.00001 < \frac{w_i}{w_{i+1}} < 1.000012 \cdot \frac{\rho_{i+1}}{\rho_i}. \end{aligned}$$

Proof. The proof of Theorem 2 in [1] actually gives more information than we have thus far indicated. Let cluster \mathcal{C} that has the heterogeneity profile $\mathbf{P} = \langle \rho_1, \dots, \rho_n \rangle$, where $\rho_1 \geq \dots \geq \rho_n$. Let the FIFO protocol employ the startup indexing $s_i = i$ for all $i \in \{1, \dots, n-1\}$. Then each work allocation w_i (for computer C_i) is given *exactly* (i.e., not asymptotically) by:

$$w_i = \left[\frac{1}{A + B\rho_i} \cdot \prod_{j=1}^{i-1} \frac{B\rho_j + \tau\delta}{A + B\rho_j} \right] \cdot (L - \tau\delta W(L; \mathbf{P}) - (n+1)\sigma).$$

(The parameter σ , which measures the cost of setting up an intercomputer communication, appears in the full model of [12], but not its asymptotic simplification.)

It follows that

$$\frac{w_i}{w_{i+1}} = \frac{B\rho_{i+1} + A}{B\rho_i + A} \cdot \frac{B\rho_i + A}{B\rho_i + \tau\delta} = \frac{B\rho_{i+1} + A}{B\rho_i + \tau\delta}$$

Elementary estimates then yield (A.1), because $A < B$ and both ρ_i and ρ_{i+1} are ≥ 1 . \square

Sample Values for Perspective	
Quantity	Value
A/B (coarse tasks):	0.0000101
A/B (finer tasks):	0.000101
$1 + A/B + \tau/B$ (coarse tasks):	1.0000111
$1 + A/B + \tau/B$ (finer tasks):	1.00011

B Proofs

B.1 Proof of Proposition 1

By (2.2),

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

By (B.2), then,

$$\frac{B\rho + \tau\delta}{B\rho + A} = \left(1 - (A - \tau\delta)X(\mathbf{P}^{(\rho)}) \right)^{1/n}$$

Therefore,

$$B\rho + \tau\delta = (B\rho + A) \left(1 - (A - \tau\delta)X(\mathbf{P}^{(\rho)}) \right)^{1/n}$$

so that

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

and

$$\rho = \frac{1}{B} \cdot \frac{A \left(1 - (A - \tau\delta)X(\mathbf{P}^{(\rho)}) \right)^{1/n} - \tau\delta}{1 - \left(1 - (A - \tau\delta)X(\mathbf{P}^{(\rho)}) \right)^{1/n}}$$

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

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

B.2 Proof of Proposition 2

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

We begin by refining the expression (2.2) for $X(\mathbf{P})$ to make explicit the startup order $\Sigma = \langle s_1, \dots, s_n \rangle$ used by \mathcal{C} . (By Theorem 1.2, this has no impact on \mathcal{C} 's work production.) As we write $X(\mathbf{P}; \Sigma)$ to announce the use of Σ , the only impact on (2.2) 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 $\Sigma = \langle s_1, \dots, s_{n-1}, i \rangle$. We then form the appropriate versions of (2.2) that use startup order Σ . For the sake of perspicuity, we write these versions in the following way, which emphasize that $X(\mathbf{P}; \Sigma)$ and $X(\mathbf{P}'; \Sigma)$ differ only in their first terms.

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

Direct calculation now shows that

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

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

B.3 Proof of Theorem 3

As we compare $X(\mathbf{P}^{(i)})$ and $X(\mathbf{P}^{(j)})$, we lose no generality by using a startup ordering $\Sigma = \langle s_1, \dots, s_{n-1}, s_n \rangle$ for \mathcal{C} 's computers for which $s_n = i$ and $s_{n-1} = j$. We then obtain the following expressions via (3.1).

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

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 $X(\mathbf{P}^{(i)})$ and $X(\mathbf{P}^{(j)})$, respectively. (The ‘‘lead fraction’’ in both expressions is the fraction that multiplies $Y(\mathbf{P})$.) Because $\rho_i > \rho_j$, it follows that $X(\mathbf{P}^{(j)}) > X(\mathbf{P}^{(i)})$, whence the result. \square

B.4 Proof of Theorem 4

We have \mathcal{C} employ the same startup order Σ as we compare $X(\mathbf{P}^{[i]})$ and $X(\mathbf{P}^{[j]})$ as we did when we compared $X(\mathbf{P}^{(i)})$ and $X(\mathbf{P}^{(j)})$ (in Section B.3); hence, $s_n = i$ and $s_{n-1} = j$. Specializing (3.1) therefore yields

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

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

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

(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. □

B.5 Proof of Lemma 1

Focus on a fixed, but arbitrary profile $\mathbf{P} = \langle \rho_1, \dots, \rho_n \rangle$, and expand (2.2) to express $X(\mathbf{P})$ as a single fraction, $X(\mathbf{P}) = X_{\text{num}}/X_{\text{denom}}$.

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 $i \in \{0, \dots, n\}$, the coefficient, β_i , of $F_i(\mathbf{P})$ 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 $j \in \{0, \dots, n\}$, the j th I - J product, $I_j \cdot J_j$, is the unique one that does not “mention” ρ_j .

Focus now on an arbitrary $i \in \{0, \dots, n\}$ and an arbitrary i -*monomial* $\mu = \rho_{k_1} \cdots \rho_{k_i}$. Consider the coefficient of μ in $F_i(\mathbf{P})$. As just noted, μ appears as a subproduct of every I - J product $I_\ell \cdot J_\ell$ where $\ell \in \{0, \dots, n\} \setminus \{k_1, \dots, k_i\}$; focus on an arbitrary such index ℓ . Say that μ is “split” between I_ℓ and J_ℓ , in the sense that $0 \leq h \leq i$ 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

$$B^i \cdot \left(A^{n-h-\ell} \cdot (\tau\delta)^{\ell-(i-h)-1} \right). \quad (\text{B.3})$$

Next, note that, given μ , the coefficient (B.3) 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 $F_i(\mathbf{P})$ 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

B.6 Proof of Proposition 3

After ‘‘cross-multiplying’’ the fractions in expression (3.4), we see that $X(\mathbf{P}_1) > X(\mathbf{P}_2)$ if, and only if, the following ‘‘ α - β difference’’ is positive:

$$\begin{aligned} & \left(\alpha_0 F_0^{(n)}(\mathbf{P}_1) + \cdots + \alpha_{n-1} F_{n-1}^{(n)}(\mathbf{P}_1) \right) \cdot \left(\beta_0 F_0^{(n)}(\mathbf{P}_2) + \cdots + \beta_n F_n^{(n)}(\mathbf{P}_2) \right) \\ & - \left(\alpha_0 F_0^{(n)}(\mathbf{P}_2) + \cdots + \alpha_{n-1} F_{n-1}^{(n)}(\mathbf{P}_2) \right) \cdot \left(\beta_0 F_0^{(n)}(\mathbf{P}_1) + \cdots + \beta_n F_n^{(n)}(\mathbf{P}_1) \right) \end{aligned}$$

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

$$(\alpha_i \beta_j - \alpha_j \beta_i) \cdot \left(F_i^{(n)}(\mathbf{P}_1) \cdot F_j^{(n)}(\mathbf{P}_2) - F_i^{(n)}(\mathbf{P}_2) \cdot F_j^{(n)}(\mathbf{P}_1) \right) \quad (\text{B.4})$$

The following result will allow us to complete the proof.

$$\text{Claim. For all indices } i \text{ and } j > i \quad \alpha_i \beta_j > \alpha_j \beta_i \quad (\text{B.5})$$

We verify claim (B.5) by direct calculation. From Lemma 1, we know that

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

It follows that

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

The last inequality holds because every term in the last summation is positive. This verifies claim (B.5).

To complete the argument, note that whenever (B.5) holds for a pair of indices i and j , the product (B.4) is positive whenever (in fact, precisely when) the difference

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

is positive. Because (B.5) in fact holds for all i and $j > i$, we see that the “ α - β difference” is positive whenever (3.5) holds. This means, however, that $X(\mathbf{P}_1) > X(\mathbf{P}_2)$ whenever (3.5) holds, whence the proposition. \square

B.7 Proof of Theorem 5

Let $\mathbf{P}_1 = \langle \rho_{11}, \dots, \rho_{1n} \rangle$ and $\mathbf{P}_2 = \langle \rho_{21}, \dots, \rho_{2n} \rangle$. By (3.2), if $F_1^{(n)}(\mathbf{P}_1) = F_1^{(n)}(\mathbf{P}_2)$, then:

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

But we know that $(\rho_{11} + \dots + \rho_{1n})^2 = (\rho_{21} + \dots + \rho_{2n})^2$ (because of the equal mean speeds); hence we have, by (3.3), $[F_2^{(n)}(\mathbf{P}_1) < F_2^{(n)}(\mathbf{P}_2)]$.

When $n = 2$, there are only two symmetric functions, $F_1^{(2)}$ and $F_2^{(2)}$, so the relations between the clusters’ mean speeds and variances determine the relations between their profiles’ symmetric functions. \square