ARTICLE IN PRESS

Parallel Computing xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Static worksharing strategies for heterogeneous computers with unrecoverable interruptions

Anne Benoit [a,*], Yves Robert [a], Arnold Rosenberg [b], Frédéric Vivien [a]

[a] Ecole Normale Supérieure de Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France
[b] Colorado State University, Fort Collins, USA

ARTICLE INFO

Article history:
Received 11 December 2009
Accepted 17 March 2010
Available online xxxx

Keywords:
Divisible workload
Unrecoverable interruptions
Heterogeneous computers
Master–Worker platforms

ABSTRACT

One has a large computational workload that is “divisible” (its constituent tasks' granularity can be adjusted arbitrarily) and one has access to p remote computers that can assist in computing the workload. How can one best utilize the computers? Two features complicate this question. First, the remote computers may differ from one another in speed. Second, each remote computer is subject to interruptions of known likelihood that kill all work in progress on it. One wishes to orchestrate sharing the workload with the remote computers in a way that maximizes the expected amount of work completed. We deal with three versions of this problem. The simplest version ignores communication costs but allows computers to differ in speed (a *heterogeneous* set of computers). The other two versions account for communication costs, first with identical remote computers (a *homogeneous* set of computers), and then with computers that may differ in speed. We provide exact expressions for the optimal work expectation for all three versions of the problem – via explicit closed-form expressions for the first two versions, and via a recurrence that computes this optimal value for the last, most general version.

1. Introduction

This paper extends work on divisible-load theory [1] that focuses on computing platforms that employ a master–worker scheduling paradigm. Our goal is to optimally distribute a given *divisible* computational workload (whose constituent tasks' granularity can be adjusted arbitrarily) to p remote *worker* computers that may differ in speeds. The workers are connected to the *master* computer via a bus or network. The master serially sends a fraction of the workload to each worker. The problem is to determine what fraction of the load should be sent to each remote computer, and in which order. This problem has been considered many times in the recent past, and closed-form expressions have been derived to compute these load fractions [2,3]. We revisit this problem in the context of workers that are subject to *unrecoverable failures* [4], and we strive to maximize the *expected* amount of total work that the workers will complete. For intuition: An “unrecoverable failure” may result from a hardware crash, an increasingly likely event with the advent of massively parallel grid platforms [5,6]; the “failure” may also result from the (unexpected) return of a remote computer's user/owner in a cycle-stealing episode [7–9]. Consider the following scenario: On Friday evening, a Ph.D. student has a large set of simulations to run. She has access to a set of computers from the lab, but each computer can be reclaimed at any instant by its owner. In any case, everybody will be back

* Corresponding author.
E-mail addresses: Anne.Benoit@ens-lyon.fr (A. Benoit), Yves.Robert@ens-lyon.fr (Y. Robert), rsnbrg@colostate.edu (A. Rosenberg), Frederic.Vivien@ens-lyon.fr (F. Vivien).

0167-8191/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.parco.2010.03.001

to work by 8am on Monday. What is the student's best strategy: How much simulation data should she attempt to execute on each accessible computer?.

We study the preceding scenario under the assumption that the risk of each worker computer's being reclaimed is known and that it grows with time. In detail: *The probability that a worker computer will be interrupted increases linearly with the time the computer has been available.* Other failure probability distributions could be envisioned, but the linear distribution is natural in the absence of further information. Also, the linear risk function turns out to be tractable, in the sense that we have succeeded in deriving optimality results for this distribution. Indeed, the major achievement of this paper is to expose a strategy for distributing work optimally, i.e., in a way that maximizes the expected total amount of work completed by the workers.

A roadmap. After describing the formal framework of our study in detail, in Section 2, we address three versions of our optimization problem. Section 3 treats the first, and simplest, version of the problem, which does not assess a charge for intercomputer communication; this models compute-intensive workloads wherein compute costs render communication costs negligible. Within this version, the set of workers is heterogeneous, in that they may differ in speed. The other two versions of our problems do assess costs for intercomputer communication: the version studied in Section 4 assumes that all workers are identical; the version in Section 5 considers workers that may differ in speed, bandwidth, and/or failure rate. For the versions of Sections 3 and 4, we provide explicit closed-form expressions for the optimal expected amount of work completed by the optimal strategy; for the version of Section 5, which is the most general, we provide a recurrence that computes the optimal work expectation in linear time as long as computers only differ by a single of their three characteristics. We follow the analytical sections with a brief overview of related work in Section 6, particularly comparing the current approach and results with those of our previous work [4]. We end in Section 7 with conclusions and perspectives.

2. The technical framework

We (the master computer) have W units of divisible work to execute on p worker computers. We wish to determine how much work to allocate to each worker and when to transmit this work, with the goal of having the workers complete as much work as possible. Having made these determinations, we send each worker a single message containing the data that it needs in order to execute its fraction of the workload. In the terminology of [2], this is the single-round distribution strategy. Note that the load fractions received by the workers are *rational* quantities instead of *integer* quantities: this is the key relaxation of divisible-load theory. Communications are done sequentially to each worker, which corresponds to a (somewhat pessimistic) *one-port* model [10], with single-threaded execution and blocking send/receive MPI primitives [11]. In order to simplify analyses, we index workers in the order in which they receive work: $P_1, \ldots, P_p$. Our study is devoted to determining the sizes of work allocations and the order of transmitting them to workers that maximizes the aggregate amount of work that the workers complete.

The preceding paragraph omits the crucial aspect of our problem that makes our study difficult and significant: Each worker is vulnerable to unrecoverable interruptions that "kill" all work in progress (on that computer). Without somehow constraining the problem, we could not prevent a "malicious adversary" from preventing our workers from completing any work. The constraint that we posit is a rather mild one, which only slightly idealizes what one could achieve in practice. We assume that we have *exact knowledge* of: each worker's computing rate on the workload, its communication rate (with the master), and the instantaneous probability of its being interrupted. We measure time from the beginning of the "episode" during which we have access to the workers, and we assume that the probability of each worker's being interrupted increases with the amount of time that it has been available, whether working on our workload or not. (From another perspective, a worker's probability of being interrupted increases with the amount of work it *could have done*.) Formally, each computer is subject to a *risk function* $Pr(T)$ for worker P_i, which denotes the probability that P_i has been interrupted by the end of the first T time-units.

The interruption model that we study is embodied in *linear risk* functions: Worker P_i's risk of being interrupted by the end of w time-units has the form $Pr(w) = \kappa_i w$ for some constant κ_i. Linear risk is the most natural model in the absence of further information. The probability density function for P_i is then $dPr = \kappa_i\, dt$ for $t \in [0, 1/\kappa_i]$ and 0 otherwise, so that

$$Pr(T) = \min\left\{1, \int_0^T \kappa_i\, dt\right\} = \min\{1, \kappa_i T\}.$$

Each worker P_i computes at the rate s_i work-units/time-unit; it is connected to the master by a link of bandwidth bw_i. We introduce two abbreviation that we use selectively to simplify quantification of the impact of interruption risks on work completion.

- $\mathsf{z}_{i,j} = \kappa_i/\mathsf{bw}_j$. (Under our indexing convention, we use this notation only for $j \leqslant i$.) This is the interruption rate for P_i per unit-load communication from the master to P_j. It reflects the risk that P_i incurs as it waits for earlier workers to get their work ($j < i$) and as it gets its work ($j = i$).
- $\mathsf{x}_i = \kappa_i/\mathsf{s}_i$. This is the interruption rate for P_i for each unit-load of computation that it does.

We use these quantities as follows. Say that we send w_1 units of work to P_1 and then w_2 units to P_2. The expected amount of work completed by P_1 is, then,

$$E_1 = w_1(1 - (\mathsf{z}_{1,1} + \mathsf{x}_1)w_1). \tag{1}$$

As explanation: Observe that P_1 receives work during the first w_1/bw_1 time-units and computes during the next w_1/s_1 time-units. Its risk of being interrupted increases linearly with elapsed time, whether it is communicating or computing. Next, note that the expected amount of work completed by P_2 is

$$E_2 = w_2(1 - (\mathsf{z}_{2,1}w_1 + \mathsf{z}_{2,2}w_2) - \mathsf{x}_2 w_2).$$

To wit, before P_2 starts computing, it waits for P_1 to receive its work (which takes the first w_1/bw_1 time-units) and to receive its own work (which takes the next w_2/bw_2 time-units). Only after these two communications does P_2 start computing, for the next w_2/s_2 time-units. P_2's risk of being interrupted increases linearly *with elapsed time* – whether it is waiting (as P_1 receives work), communicating (as it receives work), or computing. If we had only these two workers (the case $p = 2$), then our goal would be to maximize $E_1 + E_2$, the expected total amount of completed work.

Note that the formula (1) for E_1 assumes that $(\mathsf{z}_{1,1} + \mathsf{x}_1)w_1 \leqslant 1$. If this condition is not satisfied, then $E_1 = 0$. To avoid such situations, we make a technical assumption that the total workload is small enough so that we distribute it entirely to the p workers. Indeed, if the total load is too large, then, with probability 1, all workers will be interrupted before completing their work. Henceforth, we assume that the p chunks allocated to the workers *partition* the original workload and that there is a nonzero probability that the last worker can complete its allocated work. A sufficient condition for this situation is that $W \leqslant 1/(\mathsf{z}_{\max} + \mathsf{x}_{\max})$, where $\mathsf{z}_{\max} = \kappa_{\max}/\min_{1 \leqslant i \leqslant p}\{\mathsf{bw}_i\}$ and $\mathsf{x}_{\max} = \kappa_{\max}/\min_{1 \leqslant i \leqslant p}\{\mathsf{s}_i\}$ are calibrated to the the slowest link and the slowest worker, respectively. To see this, note that the last computer, P_n, can always start computing after it and all preceding computers have their work. Allowing idle periods in the communication cannot improve the solution, because interruption risk grows with elapsed time. Thus, P_n needs V_n/s_n time-steps to execute its size-V_n allocation, which it receives not later than times-step $(W - V_n)/\min_{1 \leqslant i \leqslant p}\{\mathsf{bw}_i\}$. We can now formally state our optimization problem.

Definition 1. *Distrib*(p) denotes the problem of computing $\mathscr{E}^{\mathrm{opt}}(W, p)$, the optimal value of the expected total amount of work done when partitioning and distributing the entire workload $W \leqslant 1/(\mathsf{z}_{\max} + \mathsf{x}_{\max})$ to the p worker computers.

We have defined "personalized" versions of the parameters that we use to characterize a collection of workers: s, κ, bw, and z. In fact, in only one section (Section 5.4) do we "personalize" all four parameters; generally, at least one parameter is constant across the collection. We employ the just-indicated unsubscripted notation when a parameter's value is shared by all workers.

3. Heterogeneous computers with free communication

We first study the DISTRIB problem when our p workers: (a) may differ in speed; (b) do not incur any cost for communication (the case $\mathsf{z}_i \equiv \mathsf{z} = 0$); ($c$) share the same risk of interruption (so all $\kappa_i \equiv \kappa$). This case models compute-intensive applications wherein computation costs render communication costs negligible. Our result for this case is expressed most perspicuously using symmetric functions.

Definition 2. Given $n \geqslant 1$, for $0 \leqslant i \leqslant n$, $\sigma_i^{(n)}$ denotes the ith symmetric function of $\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n$: $\sigma_i^{(n)} = \sum_{1 \leqslant j_1 < j_2 < \cdots < j_i \leqslant n} \prod_{k=1}^{i} \mathsf{x}_{j_k}$. By convention $\sigma_0^{(n)} = 1$.

For instance with $n = 3$, $\sigma_1^{(3)} = \mathsf{x}_1 + \mathsf{x}_2 + \mathsf{x}_3$, $\sigma_2^{(3)} = \mathsf{x}_1\mathsf{x}_2 + \mathsf{x}_1\mathsf{x}_3 + \mathsf{x}_2\mathsf{x}_3$ and $\sigma_3^{(3)} = \mathsf{x}_1\mathsf{x}_2\mathsf{x}_3$.

Theorem 1. *When* $\mathsf{z} = 0$ *the optimal solution to Distrib*(p) *is obtained by sending each worker* P_i *a chunk of size* $\prod_{k \neq i} \mathsf{x}_k \cdot W/\sigma_{p-1}^{(p)} = \sigma^{(p)}/(\mathsf{x}_i \sigma_{p-1}^{(p)})$. *This leads to expected work production*

$$\mathscr{E}^{\mathrm{opt}}(W, p) = W - \frac{\sigma_p^{(p)}}{\sigma_{p-1}^{(p)}} W^2 = W - \frac{1}{\sum_{i=1}^{p}(1/\mathsf{x}_i)} W^2.$$

Proof. Let $\alpha_{i,p} = \left(\prod_{k \neq i} \mathsf{x}_k\right)/\sigma_{p-1}^{(p)}$ and $\mathsf{f}_p = \sigma_p^{(p)}/\sigma_{p-1}^{(p)}$. We proceed by induction on p, noting that the theorem holds for $p = 1$, because $\alpha_{1,1} = 1$ and $\mathsf{f}_1 = \mathsf{x}_1$.

To help the reader follow the derivation, we prove the result for $p = 2$ before dealing with the general case. Assume that the size of the chunk sent to P_1 is Y. The size of the chunk sent to P_2 is thus W – Y. Both chunks are sent in parallel, as no cost is assessed for communications. The expected amount of work completed is

$$E(Y) = Y(1 - \mathsf{x}_1 Y) + (W - Y)(1 - \mathsf{x}_2(W - Y)) = W - \mathsf{x}_2 W^2 - (\mathsf{x}_1 + \mathsf{x}_2)Y^2 + 2\mathsf{x}_2 WY.$$

The optimal value is $Y^{(\mathrm{opt})} = \frac{\mathsf{x}_2}{\mathsf{x}_1 + \mathsf{x}_2} W = \alpha_{1,2} W$ as desired (and $W - Y^{(\mathrm{opt})} = \frac{\mathsf{x}_1}{\mathsf{x}_1 + \mathsf{x}_2} W = \alpha_{2,2} W$). Importing the value of $Y^{(\mathrm{opt})}$ into the expression of $E(Y)$, we derive that

Please cite this article in press as: A. Benoit et al., Static worksharing strategies for heterogeneous computers with unrecoverable interruptions, Parallel Comput. (2010), doi:10.1016/j.parco.2010.03.001

$$\mathcal{E}^{\text{opt}}(W,2) = E(Y^{(\text{opt})}) = W - \mathsf{f}_2 W^2,$$

where

$$\mathsf{f}_2 = \mathsf{x}_2 - \frac{\mathsf{x}_2^2}{\mathsf{x}_1 + \mathsf{x}_2} = \frac{\mathsf{x}_1\mathsf{x}_2}{\mathsf{x}_1 + \mathsf{x}_2} = \frac{\sigma_2^{(2)}}{\sigma_1^{(2)}}.$$

This proves the claim for $p = 2$.

Assume now that the result holds for any collection of $m \leqslant n$ computers. Consider the case of $n+1$ computers, and assume that the size of the chunk sent to P_{n+1} is $W - Y$. By induction, the optimal expected amount of work done by the first n computers is $\mathcal{E}^{\text{opt}}(Y,n) = Y(1 - \mathsf{f}_n Y)$, and this is achieved by sending a chunk of size $\alpha_{i,n}Y$ to P_i for $1 \leqslant i \leqslant n$. The expected amount of work done by the $n+1$ computers is then

$$E(Y) = Y(1 - \mathsf{f}_n Y) + (W - Y)(1 - \mathsf{x}_{n+1}(W - Y)).$$

Proceeding as above, the optimal value is $Y^{(\text{opt})} = \frac{\mathsf{x}_{n+1}}{\mathsf{f}_n + \mathsf{x}_{n+1}} W$, whence $\mathcal{E}^{\text{opt}}(W,n+1) = E(Y^{(\text{opt})}) = W - \mathsf{f}_{n+1}W^2$, where $\mathsf{f}_{n+1} = \mathsf{x}_{n+1} - \frac{\mathsf{x}_{n+1}^2}{\mathsf{f}_n + \mathsf{x}_{n+1}}$.

We recognize that $\sigma_n^{(n)} + \mathsf{x}_{n+1}\sigma_{n-1}^{(n)} = \sigma_n^{(n+1)}$ so that $\mathsf{f}_n + \mathsf{x}_{n+1} = \sigma_n^{(n+1)}/\sigma_{n-1}^{(n)}$ and

$$\mathsf{f}_{n+1} = \mathsf{x}_{n+1} - \frac{\mathsf{x}_{n+1}^2\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} = \frac{\mathsf{x}_{n+1}\left(\sigma_n^{(n+1)} - \mathsf{x}_{n+1}\sigma_{n-1}^{(n)}\right)}{\sigma_n^{(n+1)}} = \frac{\mathsf{x}_{n+1}\sigma_n^{(n)}}{\sigma_n^{(n+1)}} = \frac{\sigma_{n+1}^{(n+1)}}{\sigma_n^{(n+1)}},$$

as desired. Also, $Y^{(\text{opt})} = \frac{\mathsf{x}_{n+1}}{\mathsf{f}_n + \mathsf{x}_{n+1}} W = \frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} W$.

By induction, for $1 \leqslant i \leqslant n$, we get

$$\alpha_{i,n+1} = \alpha_{i,n}\frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} = \frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)} \prod\limits_{1\leqslant k\leqslant n, k\neq i} \mathsf{x}_k}{\sigma_{n-1}^{(n)}\sigma_n^{(n+1)}} = \frac{\mathsf{x}_{n+1} \prod\limits_{1\leqslant k\leqslant n, k\neq i} \mathsf{x}_k}{\sigma_n^{(n+1)}} = \frac{\prod\limits_{1\leqslant k\leqslant n+1, k\neq i} \mathsf{x}_k}{\sigma_n^{(n+1)}},$$

as desired. It remains to check the value of

$$\alpha_{n+1,n+1} = 1 - \frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} = \frac{\sigma_n^{(n+1)} - \mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} = \frac{\prod\limits_{1\leqslant k\leqslant n} \mathsf{x}_k}{\sigma_n^{(n+1)}},$$

which concludes the proof. □

Thus, the optimal solution is *symmetric*: the contribution of each computer is a (somewhat complicated) symmetric function of all computer speeds.

4. Homogeneous computers with communication costs

For the remainder of the paper, we account for every communication, via the (possibly "personalized") parameter $\mathsf{z} \neq 0$. We first study the case of homogeneous, identical workers (so $\mathsf{s}_i \equiv \mathsf{s}$, $\mathsf{bw}_i \equiv \mathsf{bw}$, and $\mathsf{x}_i \equiv \mathsf{x}$), in preparation for the technically more challenging case of heterogeneous workers.

Theorem 2. *If workers have identical speeds, then the optimal solution to Distrib*(p) *allocates equal-size chunks (of size* W/p*) to all workers. In expectation, this completes the following amount of work.*

$$\mathcal{E}^{\text{opt}}(W,p) = W - \frac{(p+1)\mathsf{z} + 2\mathsf{x}}{2p} W^2.$$

Proof. The proof is similar to that of Theorem 1. Let $\mathsf{f}_p = \frac{(p+1)\mathsf{z}+2\mathsf{x}}{2p}$. We proceed by induction on p, noting that the theorem holds when $p = 1$, because $\mathsf{f}_1 = \mathsf{z} + \mathsf{x}$.

Assume that the result holds for $m \leqslant n$ computers. Consider the case of $n+1$ workers and assume that the size of the chunk sent to P_{n+1} is W – Y. By induction, the first n computers operating optimally produce $\mathcal{E}^{\text{opt}}(Y,n) = Y(1 - \mathsf{f}_n Y)$ units of work, by sending a chunk of size Y/n to each P_i $(1 \leqslant i \leqslant n)$. Thus, the expected amount of work completed by our $n+1$ workers is

$$E(Y) = Y(1 - \mathsf{f}_n Y) + (W - Y)(1 - \mathsf{z}W - \mathsf{x}(W - Y)).$$

To understand this reckoning, note that P_{n+1} has to wait for the whole workload to be distributed (accounted for by the term $\mathsf{z}W$) before it can start computing its own chunk (accounted for by the term $\mathsf{x}(W - Y)$). We rewrite $E(Y)$ as

$$E(Y) = W - (\mathsf{z} + \mathsf{x})W^2 - (\mathsf{f}_n + \mathsf{x})Y^2 + (\mathsf{z} + 2\mathsf{x})WY.$$

The optimal value of Y is $Y^{(\mathrm{opt})} = \frac{z+2x}{2(f_n+x)}W$, whence $\mathcal{E}^{\mathrm{opt}}(W, n+1) = E(Y^{(\mathrm{opt})}) = W - f_{n+1}W^2$, where $f_{n+1} = z + x - \frac{(z+2x)^2}{4(f_n+x)}$.

By the induction hypothesis, we get $f_n + x = \frac{n+1}{2n}(z+2x) + x = \frac{n+1}{2n}(z+2x)$, so that

$$f_{n+1} = z + x - \frac{n(z+2x)}{2(n+1)} = \frac{(n+2)z + 2x}{2(n+1)},$$

as expected. We find also that $Y^{(\mathrm{opt})} = \frac{n}{n+1}W$, so that, for each $i \leqslant n$, P_i receives a chunk of size $\frac{1}{n}Y^{(\mathrm{opt})} = \frac{1}{n+1}W$. We deduce that P_{n+1} receives a chunk of that same size (or we can directly check that $W - Y^{(\mathrm{opt})} = \frac{1}{n+1}W$). This concludes the proof. □

Interestingly, the optimal solution mandates sending equal-size chunks to all computers, which contrasts with the classical divisible-load setting. In that setting, one minimizes the total time needed to execute a fixed workload by having all computers finish computing simultaneously [2], so that the first computers served by the master receive larger chunks.

5. Heterogeneous computers with communication costs

5.1. Computers that differ only in computation speed

Workers have "personalized" speeds in this scenario (each P_i has speed s_i), but they share link bandwidth ($bw_i \equiv bw$) and interruption risk ($\kappa_i \equiv \kappa$).

Definition 3. Define the sequence $\vec{\lambda}$ as follows: $\lambda_0 = \lambda_1 = 4$, and for $n \geqslant 2$, $\lambda_n = \lambda_{n-1} - \frac{1}{4}\lambda_{n-2}$. For convenience, let $\lambda_{-1} = 0$. Note that $\lambda_n = 4(1+n)/2^n$ for all $n \geqslant 0$.

The sequence $\vec{\lambda}$ is used to characterize the optimal solution to this version of *Distrib*(p).

Theorem 3. *Say that the master serves workers in the order $P_1, P_2, \ldots, P_p$. In the current scenario, the optimal schedule for Distrib(p) allocates $\alpha_{i,p}W$ units of work to P_i, for $i \in [1, p]$, where:*

- *for* $p \geqslant 1$: $f_p = \frac{\sum_{i=0}^{p} \lambda_i \sigma^{(p)}_{p-i} z^i}{\sum_{i=0}^{p-1} \lambda_i \sigma^{(p)}_{p-i-1} z^i}$,
- $\alpha_{1,p} = \prod_{j=1}^{p} Y_j^{(\mathrm{opt})}$; *and, for* $i \in [2, p]$, $\alpha_{i,p} = (1 - Y_i^{(\mathrm{opt})})\prod_{j=i+1}^{p} Y_j^{(\mathrm{opt})}$,
- $Y_1^{(\mathrm{opt})} = 1$; *and, for* $i \in [2, p]$, $Y_i^{(\mathrm{opt})} = \frac{z+2x_i}{2(f_{i-1}+x_i)}$.

In expectation, this strategy completes

$$\mathcal{E}^{\mathrm{opt}}(W, p) = W - f_p W^2.$$

The optimal solution does not depend on the order in which the master serves workers.

Proof. The proof is a more involved analogue of those of Theorems 1 and 2. Note that the theorem holds for $p = 1$, because $f_1 = \frac{\lambda_0 x_1 + \lambda_1 z}{\lambda_0} = z + x_1$.

To supply intuition, particularly for why the order of serving workers is not important, consider the case of two workers, P_1 and P_2, that are served in this order (first P_1, then P_2). If we send a chunk of size Y to P_1 and one of size $W - Y$ to P_2, the expected amount of work completed is

$$E(Y) = Y(1 - f_1 Y) + (W - Y)(1 - (zW + x_2(W - Y))).$$

Note that the term zW accounts for P_2's two waiting periods: for the first chunk to be sent to P_1 and for the second chunk to be sent to it. Finally, P_2 computes its chunk, whence the term $x_2(W - Y)$. We rewrite

$$E(Y) = W - (z + x_2)W^2 - (f_1 + x_2)Y^2 + (z + 2x_2)WY.$$

The optimal value for Y is $Y^{(\mathrm{opt})} = \frac{z+2x_2}{2(f_1+x_2)}W = \alpha_{1,2}W$, and we derive that

$$\mathcal{E}^{\mathrm{opt}}(W, 2) = W - f_2 W^2,$$

where

$$f_2 = z + x_2 - \frac{(z+2x_2)^2}{4(f_1+x_2)} = \frac{4x_1x_2 + 4(x_1+x_2)z + 3z^2}{4(x_1+x_2+z)},$$

as desired. We note that the expression is symmetric in x_1 and x_2, meaning that the order of serving the workers has no significance.

Assume now that the theorem holds for up to n workers, and consider the case of $n+1$ workers that are served in the order $P_1, \ldots, P_{n+1}$. Say that we send a chunk of size $W - Y$ to P_{n+1}. We know by induction that the best way to distribute the remaining Y units of work to the first n workers is independent of their ordering, and that the optimal expectation $\mathcal{E}^{\mathrm{opt}}(W, n)$ is given by $\mathcal{E}^{\mathrm{opt}}(W, n) = W - f_n W^2$.

Please cite this article in press as: A. Benoit et al., Static worksharing strategies for heterogeneous computers with unrecoverable interruptions, Parallel Comput. (2010), doi:10.1016/j.parco.2010.03.001

The total expectation $E(Y)$ for the $n+1$ computers is obtained as previously:

$$E(Y) = W - (z + x_{n+1})W^2 - (f_n + x_{n+1})Y^2 + (z + 2x_{n+1})WY.$$

The optimal value for Y is $Y^{(\mathrm{opt})} = \frac{z+2x_{n+1}}{2(f_n+x_{n+1})}W$, and we derive that

$$f_{n+1} = z + x_{n+1} - \frac{(z+2x_{n+1})^2}{4(f_n + x_{n+1})}.$$

We know by induction that $f_n = a_n/b_n$, where $a_n = \sum_{i=0}^{n} \lambda_i \sigma_{n-i}^{(n)} z^i$ and $b_n = \sum_{i=0}^{n-1} \lambda_i \sigma_{n-i-1}^{(n)} z^i$. We have $f_n + x_{n+1} = \frac{a_n + x_{n+1} b_n}{b_n}$ and

$$a_n + x_{n+1} b_n = \sum_{i=0}^{n-1} \lambda_i \left(\sigma_{n-i}^{(n)} + x_{n+1}\sigma_{n-i-1}^{(n)}\right) z^i + \lambda_n z^n.$$

But we recognize that for $0 \leqslant i \leqslant n-1$, we have

$$\sigma_{n-i}^{(n)} + x_{n+1}\sigma_{n-i-1}^{(n)} = \sigma_{n-i}^{(n+1)}. \tag{2}$$

We also have $\sigma_0^{(n)} = \sigma_0^{(n+1)} = 1$, so that

$$b_{n+1} = a_n + x_{n+1} b_n = \sum_{i=0}^{n} \lambda_i \sigma_{n-i}^{(n+1)} z^i.$$

Now we import this value into the expression for f_{n+1}, and we obtain $f_{n+1} = a_{n+1}/b_{n+1}$, where

$$\begin{aligned} a_{n+1} &= b_{n+1}(z + x_{n+1}) - \frac{1}{4}(z + 2x_{n+1})^2 b_n = \left(\sum_{i=0}^{n} \lambda_i \sigma_{n-i}^{(n+1)} z^i\right)(z + x_{n+1}) - \left(\frac{z^2}{4} + x_{n+1} z + x_{n+1}^2\right)\left(\sum_{i=0}^{n-1} \lambda_i \sigma_{n-i-1}^{(n)} z^i\right) \\ &= z^{n+1}\left(\lambda_n - \frac{\lambda_{n-1}}{4}\right) + \sum_{i=1}^{n} z^i (A_i + B_i - C_i - D_i - E_i) + \lambda_0(\sigma_n^{(n+1)} x_{n+1} - \sigma_{n-1}^{(n)} x_{n+1}^2). \end{aligned}$$

In the last expression, we have

$$\begin{aligned} A_i &= \lambda_i \sigma_{n-i}^{(n+1)} x_{n+1} \\ B_i &= \lambda_{i-1} \sigma_{n-i+1}^{(n+1)} \\ C_i &= \frac{1}{4}\lambda_{i-2}\sigma_{n-i+1}^{(n)} \\ D_i &= \lambda_{i-1}\sigma_{n-i}^{(n)} x_{n+1} \\ E_i &= \lambda_i \sigma_{n-i-1}^{(n)} x_{n+1}^2 \end{aligned}$$

Next we use Eq. (2) to derive $A_i - E_i = \lambda_i \sigma_{n-i}^{(n)} x_{n+1}$ and $B_i - D_i = \lambda_{i-1}\sigma_{n-i+1}^{(n)}$, so that $B_i - D_i - C_i = (\lambda_{i-1} - \frac{\lambda_{i-2}}{4})\sigma_{n-i+1}^{(n)} = \lambda_i \sigma_{n-i+1}^{(n)}$, and finally $A_i + B_i - C_i - D_i - E_i = \lambda_i \sigma_{n-i+1}^{(n+1)}$. As for the first and last terms, we get $\lambda_n - \frac{1}{4}\lambda_{n-1} = \lambda_{n+1} = \lambda_{n+1}\sigma_0^{(n+1)}$, and $\lambda_0(\sigma_n^{(n+1)} x_{n+1} - \sigma_{n-1}^{(n)} x_{n+1}^2) = \lambda_0 \sigma_n^{(n)} x_{n+1} = \lambda_0 \sigma_{n+1}^{(n+1)}$. In summation, we find that

$$a_{n+1} = \sum_{i=0}^{n+1} \lambda_i \sigma_{n+1-i}^{(n+1)} z^i,$$

which establishes the inductive expression. Because the expression is symmetric, we verify that the order serving workers has no impact. We thereby have the value of $\mathscr{E}^{\mathrm{opt}}(W,p)$.

As for the sizes of the allocated chunks, we find that $Y^{(\mathrm{opt})} = \frac{z+2x_{n+1}}{2(f_n+x_{n+1})}W$; hence with p computers,

$$\alpha_{p,p} = 1 - \frac{z + 2x_p}{2(f_{p-1} + x_p)} = \frac{2f_{p-1} - z}{2(f_{p-1} + x_p)},$$

as desired. We proceed by induction to determine the value of $\alpha_{i,p}$ for $i = p-1$ down to $i = 2$, and then $i = 1$. With $p = 2$, we check that $\alpha_{2,2} = \frac{z+2x_1}{2(z+x_1+x_2)}$ (remember that $f_1 = z + x_1$) and then $\alpha_{1,2} = \frac{z+2x_2}{2(z+x_1+x_2)}$. □

As a "reality check" on the values of f_p and $\alpha_{i,p}$, we see that: when $z_i \equiv z = 0$, we retrieve the values given in Theorem 1; and, when $x_i \equiv x$, we retrieve the values given in Theorem 2.

Corollary 1. *When communication costs are not assessed* $(z = 0)$, *the expression for* f_p *reduces to* $f_p = \sigma_p^{(p)}/\sigma_{p-1}^{(p)}$, *and the chunk sent to each* P_j *is of size* $\frac{\prod_{k \neq j} x_k}{\sigma_{p-1}^{(p)}} W$.

Proof. When $z = 0$, we have $a_p = \sigma_p^{(p)}$ and $b_p = \sigma_{p-1}^{(p)}$. Also, the chunk sent to P_p is of size $W - Y^{(\mathrm{opt})} = (a_{p-1}/b_p)W$. □

Corollary 2. *When workers are identical* $(x_j \equiv x)$, *the expression for* f_p *reduces to* $f_p = \frac{1}{2p}(2x + (p+1)z)$, *and all allocated chunks have size* $\frac{1}{p}W$.

Proof. When $\mathsf{x}_j \equiv \mathsf{x}$, one computes f_p via recurrence, starting with $\mathsf{f}_1 = \mathsf{x} + \mathsf{z}$ and using the relation $\mathsf{f}_{n+1} = \mathsf{z} + \mathsf{x}_{n+1} - \frac{\mathsf{z}+2\mathsf{x}_{n+1}}{4(\mathsf{f}_n+\mathsf{x}_{n+1})}$. Also, the chunk sent to P_p has size $W - Y^{(\mathrm{opt})} = \frac{2\mathsf{f}_{p-1}-\mathsf{z}}{2(\mathsf{f}_p+\mathsf{x})}W = \frac{1}{p}W$. □

5.2. Computers that differ only in communication bandwidth

We focus now on workers that are identical except for having different link bandwidths to the master. Such configurations are encountered, for instance, when one borrows resources from clusters that are similar/identical in computing power ($\mathsf{s}_i \equiv \mathsf{s}$) and risk of interruption ($\mathsf{z}_i \equiv \mathsf{z}$) but are geographically dispersed.

In contrast to the case where computers differ only in computing speed, we now find that the order of serving workers impacts the expected work production.

Lemma 1. *In an optimal schedule for Distrib(p), the master sends work to workers in non-increasing order of their bandwidths.*

Proof. Consider any schedule for *Distrib*(p), and focus on two computers, P_i and P_j, that receive their work consecutively in this solution: First, P_i receives a chunk of size Y, then P_j receives a chunk of size Z. Denote by X the cumulative amount of work distributed to computers that receive their work before P_i, and let T_X denote the time that was needed to distribute that work. We can isolate within the expected amount of work completed by this schedule the portion of P_i's and P_j's contributions that depends on the relative service orders of P_i and P_j:

$$E_{i,j}(W) = Y\left(1 - \left(T_X + \frac{Y}{\mathsf{bw}_i} + \frac{Y}{\mathsf{s}}\right)\kappa\right) + Z\left(1 - \left(T_X + \frac{Y}{\mathsf{bw}_i} + \frac{Z}{\mathsf{bw}_j} + \frac{Z}{\mathsf{s}}\right)\kappa\right).$$

Consider now the schedule that differs from the preceding one *only* in its reversing the order in which P_i and P_j receive their work; i.e., all workers still receive the same amount of work and, except for P_i and P_j, they still are served in the same order. Because this modification impacts only the contributions of P_i and P_j to the overall expectation, the new analogue of $E_{i,j}(W)$ is:

$$E_{j,i}(W) = Z\left(1 - \left(T_X + \frac{Z}{\mathsf{bw}_j} + \frac{Z}{\mathsf{s}}\right)\kappa\right) + Y\left(1 - \left(T_X + \frac{Z}{\mathsf{bw}_j} + \frac{Y}{\mathsf{bw}_i} + \frac{Y}{\mathsf{s}}\right)\kappa\right).$$

The difference in expected work production is then

$$E_{i,j}(W) - E_{j,i}(W) = \frac{(\mathsf{bw}_1 - \mathsf{bw}_2)}{\mathsf{bw}_1\mathsf{bw}_2}YZ\kappa. \qquad \square$$

Let the workers be indexed in non-increasing order of bandwidth: $\mathsf{bw}_1 \geqslant \mathsf{bw}_2 \geqslant \cdots \geqslant \mathsf{bw}_n$.

Theorem 4. *In the current scenario, the optimal schedule for Distrib(p) distributes work to computers in non-increasing order of their bandwidth; for each $i \in [1,n]$, it allocates $\alpha_{i,p}W$ units of work to P_i, where:*

- $\alpha_{1,p} = \prod_{j=1}^{p} Y_j^{(opt)}$; *and, for* $i \in [2,p]: \ \alpha_{i,p} = (1 - Y_i^{(opt)})\prod_{j=i+1}^{p} Y_j^{(opt)}$.
- $Y_1^{(opt)} = 1$; *and, for* $i \in [2,p]$ $Y_i^{(opt)} = \frac{(2\mathsf{bw\_eq}_{i-1}(\mathsf{bw}_i+\mathsf{s}) - \mathsf{bw}_i\cdot\mathsf{s})b_{i-1}}{2(a_{i-1}\cdot\mathsf{bw\_eq}_{i-1}\cdot\mathsf{bw}_i\cdot\mathsf{s} + b_{i-1}\cdot(\mathsf{bw\_eq}_{i-1}(\mathsf{bw}_i+\mathsf{s}) - \mathsf{bw}_i\cdot\mathsf{s}))}$.
- $a_1 = \mathsf{s} + \mathsf{bw}_1$; *and, for* $i \in [2,p]: \ a_i = a_{i-1}\cdot\mathsf{bw\_eq}_{i-1}^2(\mathsf{bw}_i + \mathsf{s}) - \frac{1}{4}b_{i-1}\cdot\mathsf{bw}_i\cdot\mathsf{s}$.
- $b_1 = \mathsf{s}.\mathsf{bw}_1$; *and, for* $i \in [2,p]: \ b_i = \mathsf{bw\_eq}_{i-1}(a_{i-1}\cdot\mathsf{bw\_eq}_{i-1}\cdot\mathsf{bw}_i\cdot\mathsf{s} + b_{i-1}((\mathsf{bw\_eq}_{i-1} - \mathsf{bw}_i)\mathsf{s} + \mathsf{bw\_eq}_{i-1}\cdot\mathsf{bw}_i))$.
- $\frac{1}{\mathsf{bw\_eq}_i} = \sum_{j=1}^{i}\frac{\alpha_{j,i}}{\mathsf{bw}_j}$.

The resulting expected work production is $\mathscr{E}^{\mathrm{opt}}(W,p) = W - \frac{a_p}{b_p}W^2\kappa$.

Proof. Noting that the optimal ordering of serving workers is given by Lemma 1, we begin with a technical remark.

By definition and Lemma 1, $\mathsf{bw\_eq}_i \geqslant \mathsf{bw}_i \geqslant \mathsf{bw}_{i+1}$ for all $i \in [1,n]$. Therefore, if $a_i > 0$ and $b_i > 0$, then we have $b_{i+1} > 0$. We must then have $a_{i+1} > 0$, because the expected work production cannot be greater than the amount of work distributed, which is W. Therefore, because $a_1 > 0$ and $b_1 > 0$, we must have $a_n > 0$ and $b_n > 0$.

On to the proof, which proceeds by induction on p. When $p = 1$, if one sends W units of work to P_1, then the expected work production is

$$E(W,1) = W\left(1 - \left(\frac{W}{\mathsf{bw}_1} + \frac{W}{\mathsf{s}}\right)\kappa\right) = W - \frac{\mathsf{s} + \mathsf{bw}_1}{\mathsf{s}\cdot\mathsf{bw}_1}W^2\kappa,$$

which satisfies the theorem.

Assume that the result holds for up to n computers, and consider the case of $n+1$ computers. Denote by W-Y the size of the chunk sent to P_{n+1}. By induction, the optimal expected amount of work done by the first n computers is $\mathscr{E}_n^{\mathrm{opt}}(Y,n) =$

Please cite this article in press as: A. Benoit et al., Static worksharing strategies for heterogeneous computers with unrecoverable interruptions, Parallel Comput. (2010), doi:10.1016/j.parco.2010.03.001

$Y\left(1-\frac{a_n}{b_n}Y\kappa\right)$, and this is achieved by giving a chunk of size $\alpha_{i,n}Y$ to each P_i, for $i \in [1,n]$. The expected amount of work completed is then:

$$E_{n+1}(Y) = Y - \frac{a_n}{b_n}Y^2\kappa + (W-Y)\left(1 - \left(\sum_{i=1}^{n}\frac{\alpha_{i,n}Y}{\mathsf{bw}_i} + \frac{W-Y}{\mathsf{bw}_{n+1}} + \frac{W-Y}{\mathsf{s}}\right)\kappa\right).$$

If we let $\frac{1}{\mathsf{bw\_eq}_n} = \sum_{i=1}^{n}\frac{\alpha_{i,n}}{\mathsf{bw}_i}$, then the preceding expectation can be rewritten as:

$$E_{n+1}(Y) = W - \frac{a_n \cdot \mathsf{bw\_eq}_n \cdot \mathsf{bw}_{n+1} \cdot \mathsf{s} + b_n(\mathsf{bw\_eq}_n(\mathsf{bw}_{n+1} + \mathsf{s}) - \mathsf{bw}_{n+1} \cdot \mathsf{s})}{b_n \cdot \mathsf{bw\_eq}_n \cdot \mathsf{bw}_{n+1} \cdot \mathsf{s}}Y^2\kappa + \frac{2\mathsf{bw\_eq}_n(\mathsf{s} + \mathsf{bw}_{n+1}) - \mathsf{bw}_{n+1} \cdot \mathsf{s}}{\mathsf{bw\_eq}_n \cdot \mathsf{bw}_{n+1} \cdot \mathsf{s}}WY\kappa$$
$$- \frac{\mathsf{bw}_{n+1} + \mathsf{s}}{\mathsf{bw}_{n+1} \cdot \mathsf{s}}W^2\kappa.$$

Therefore, $E_{n+1}(Y)$ is maximized when $Y = Y_{n+1}^{(\mathrm{opt})}W$, where

$$Y_{n+1}^{(\mathrm{opt})} = \frac{(2\mathsf{bw\_eq}_n(\mathsf{bw}_{n+1} + \mathsf{s}) - \mathsf{bw}_{n+1} \cdot \mathsf{s})b_n}{2(a_n \cdot \mathsf{bw\_eq}_n \cdot \mathsf{bw}_{n+1} \cdot \mathsf{s} + b_n(\mathsf{bw\_eq}_n(\mathsf{bw}_{n+1} + \mathsf{s}) - \mathsf{bw}_{n+1} \cdot \mathsf{s}))}.$$

Recalling our earlier inequalities involving $\mathsf{bw\_eq}_n$, bw_n, and bw_{n+1}, the numerator and denominator of the expression for $Y_{n+1}^{(\mathrm{opt})}$ are both positive (whenever a_n and b_n are both positive). Furthermore, the numerator of $Y_{n+1}^{(\mathrm{opt})}$ is always strictly smaller than its denominator. Consequently, $Y_{n+1}^{(\mathrm{opt})}$ is always strictly smaller than W. Therefore, the optimal schedule employs the following work fractions: $\alpha_{n+1,n+1} = 1 - Y_{n+1}^{(\mathrm{opt})}$ and, for $i \in [1,n]$, $\alpha_{i,n+1} = \alpha_{i,n}Y_{n+1}^{(\mathrm{opt})}$. Therefore, we find, as claimed, that

$$\begin{cases} \alpha_{1,n+1} = \prod_{j=1}^{n+1} Y_j^{(\mathrm{opt})}; \\ \text{and, for } i \in [2, n+1], \quad \alpha_{i,n+1} = (1 - Y_i^{(\mathrm{opt})}) \prod_{j=i+1}^{n+1} Y_j^{(\mathrm{opt})}. \end{cases}$$

Finally we compute the optimal expected work production, $E(Y_{n+1}^{(\mathrm{opt})})$. We find that

$$\mathscr{E}_{n+1}^{\mathrm{opt}} = W - \frac{a_{n+1}}{b_{n+1}}W^2\kappa,$$

where $a_{n+1} = a_n \cdot \mathsf{bw\_eq}_n^2(\mathsf{bw}_{n+1} + \mathsf{s}) - \frac{1}{4} \cdot b_n \cdot \mathsf{bw}_{n+1} \cdot \mathsf{s}$, and $b_{n+1} = \mathsf{bw\_eq}_n(a_n.\mathsf{bw\_eq}_n \cdot \mathsf{bw}_{n+1} \cdot \mathsf{s} + b_n((\mathsf{bw\_eq}_n - \mathsf{bw}_{n+1}) \cdot \mathsf{s} + \mathsf{bw\_eq}_n \cdot \mathsf{bw}_{n+1}))$. □

5.3. Computers that differ only in risk of interruption

We now study the *Distrib*(p) problem in the case of workers that are identical in computing speed ($\mathsf{s}_i \equiv \mathsf{s}$) and link bandwidth ($\mathsf{bw}_i \equiv \mathsf{bw}$), but that are subject to different linear risk functions (personalized risk parameters: κ_i for P_i). One might encounter such a situation, for instance, when borrowing computers that are part of the same cluster but that have different owners.

When workers differ only in computing speed, the order in which they receive their work allocations does not impact the overall expected work production; when workers differ only in bandwidth, they *must* be served in the order of non-increasing bandwidth. We see in this section that when workers differ only in interruption risk, *there exists* an optimal solution in which workers are served in non-increasing order of interruption risk. In fact, the proof of the following lemma shows that this ordering can be ignored only for workers that are not employed in the schedule – but Theorem 5 will state that *all* workers participate in an optimal solution.

Lemma 2. *There exists an optimal solution to Distrib*(p) *in which the master sends work to the workers in non-increasing order of their interruption risk.*

Proof. Consider any schedule for *Distrib*(p), and focus on two computers, P_i and P_j, that are served consecutively – first P_i, then P_j. Say that P_i receives Y units of work and that P_i and P_j jointly receive W units. Let V denote the cumulative share of work distributed to computers that receive their work before P_i. We can isolate within the expected amount of work completed by this schedule the portion of P_i's and P_j's contributions that depends on the relative service orders of P_i and P_j:

$$\begin{aligned} E_{i,j}(W) &= Y\left(1 - \left(\frac{V}{\mathsf{bw}} + \frac{Y}{\mathsf{bw}} + \frac{Y}{\mathsf{s}}\right)\kappa_i\right) + (W-Y)\left(1 - \left(\frac{V}{\mathsf{bw}} + \frac{Y}{\mathsf{bw}} + \frac{W-Y}{\mathsf{bw}} + \frac{W-Y}{\mathsf{s}}\right)\kappa_j\right) \\ &= \left(1 - \frac{V}{\mathsf{bw}}\kappa_j\right)W - \frac{\mathsf{bw}(\kappa_i + \kappa_j) + \mathsf{s} \cdot \kappa_i}{\mathsf{s} \cdot \mathsf{bw}}Y^2 + \frac{\mathsf{s} + 2\mathsf{bw}}{\mathsf{s} \cdot \mathsf{bw}}WY\kappa_j + \frac{\kappa_j - \kappa_i}{\mathsf{bw}}VY - \frac{\mathsf{s} + \mathsf{bw}}{\mathsf{s} + \mathsf{bw}}W^2\kappa_j. \end{aligned}$$

Consider now the same schedule modified *only* by exchanging the order in which P_i and P_j are served. Note that this reordering impacts only the contributions of P_i and P_j to the overall expectation. The analogue of $E_{i,j}(W)$ for this modified schedule is:

$$\begin{aligned}E_{j,i}(W) &= Y\left(1-\left(\frac{V}{\mathsf{bw}}+\frac{Y}{\mathsf{bw}}+\frac{Y}{\mathsf{s}}\right)\kappa_j\right)+(W-Y)\left(1-\left(\frac{V}{\mathsf{bw}}+\frac{Y}{\mathsf{bw}}+\frac{W-Y}{\mathsf{bw}}+\frac{W-Y}{\mathsf{s}}\right)\kappa_i\right)\\ &= \left(1-\frac{V}{\mathsf{bw}}\kappa_i\right)W-\frac{\mathsf{bw}(\kappa_i+\kappa_j)+\mathsf{s}\cdot\kappa_j}{\mathsf{s}\cdot\mathsf{bw}}Y^2+\frac{\mathsf{s}+2\mathsf{bw}}{\mathsf{s}\cdot\mathsf{bw}}WY\kappa_i+\frac{\kappa_i-\kappa_j}{\mathsf{bw}}VY-\frac{\mathsf{s}+\mathsf{bw}}{\mathsf{s}+\mathsf{bw}}W^2\kappa_i.\end{aligned}$$

Fixing all aspects of the two schedules other than the order of serving the W units of work to P_i and P_j, we determine the values of Y that maximize each expectation, $E_{i,j}(W)$ and $E_{j,i}(W)$. For $E_{i,j}(W)$, this value of Y is:

$$Y^{(\mathrm{opt})}_{i,j} = \frac{\mathsf{s}\cdot V(\kappa_j-\kappa_i)+(\mathsf{s}+2\mathsf{bw})W\kappa_j}{2(\mathsf{bw}(\kappa_i+\kappa_j)+\mathsf{s}\cdot\kappa_i)};$$

for $E_{j,i}(W)$, this value of Y is:

$$Y^{(\mathrm{opt})}_{j,i} = \frac{\mathsf{s}\cdot V(\kappa_i-\kappa_j)+(\mathsf{s}+2\mathsf{bw})W\kappa_i}{2(\mathsf{bw}(\kappa_i+\kappa_j)+\mathsf{s}\cdot\kappa_j)}.$$

Obviously, under each service order, the optimal value of Y is feasible only if it lies in the range $[0,W]$; we insist, therefore, that $0\leqslant Y^{(\mathrm{opt})}_{i,j}, Y^{(\mathrm{opt})}_{j,i}\leqslant W$, which leaves us with three cases to consider.

Case 1: $[Y^{(\mathrm{opt})}_{i,j}=0]$ **or** $[Y^{(\mathrm{opt})}_{j,i}=0]$.

If $Y^{(\mathrm{opt})}_{i,j}=0$, then $\kappa_i\geqslant\kappa_j$ and $\mathsf{s}\cdot V(\kappa_i-\kappa_j)\geqslant(\mathsf{s}+2\mathsf{bw})W\kappa_j$. In this case,

$$Y^{(\mathrm{opt})}_{j,i} \;\geqslant\; \frac{(\mathsf{s}+2\mathsf{bw})W(\kappa_j+\kappa_i)}{2(\mathsf{bw}(\kappa_i+\kappa_j)+\mathsf{s}\cdot\kappa_j)}=\frac{2\mathsf{bw}(\kappa_j+\kappa_i)+\mathsf{s}(\kappa_j+\kappa_i)}{2(\mathsf{bw}(\kappa_i+\kappa_j)+\mathsf{s}\cdot\kappa_j)}W \;\geqslant\; W.$$

Therefore, if P_i is allocated no work under the P_i-then-P_j schedule, then the same is true under the P_j-then-P_i schedule. Symmetrically, if P_j is allocated no work under the P_j-then-P_i schedule, then the same is true under the P_i-then-P_j schedule. The service order of P_i and P_j thus does not impact the expected work production in this case – thereby satisfying the lemma.

Case 2: $[Y^{(\mathrm{opt})}_{i,j}=W]$ **or** $[Y^{(\mathrm{opt})}_{j,i}=W]$).

If $[Y^{(\mathrm{opt})}_{i,j}=W]$, then $\kappa_j\geqslant\kappa_i$, and P_j receives no work under the P_i-then-P_j schedule. The expected work production due to P_i and P_j is then

$$E_{i,j}(W) = W\left(1-\left(\frac{V}{\mathsf{bw}}+\frac{W}{\mathsf{bw}}+\frac{W}{\mathsf{s}}\right)\kappa_i\right).$$

Comparing the expected work production of the P_i-then-P_j schedule, with $Y=W$, to that of the P_j-then-P_i schedule, with $Y=Y^{(\mathrm{opt})}_{j,i}$, we find that

$$E_{i,j}(W)-E_{j,i}(Y^{(\mathrm{opt})}_{j,i}) = -\frac{((\mathsf{s}+2\mathsf{bw})W\kappa_i+((\kappa_i-\kappa_j)\cdot V\cdot\mathsf{s}))^2}{4\mathsf{s}\cdot\mathsf{bw}(\mathsf{bw}(\kappa_i+\kappa_j)+\mathsf{s}\cdot\kappa_j)}.$$

This difference is always non-positive, so that, in expectation, the P_j-then-P_i schedule always completes as much work as the P_i-then-P_j schedule. This means that the P_j-then-P_i service order is the preferable one, in accordance with the lemma (because $\kappa_j\geqslant\kappa_i$).

Case 3: $[0<Y^{(\mathrm{opt})}_{i,j}, Y^{(\mathrm{opt})}_{j,i}<W]$.

To show that one service order is always better than the other, we compare $\mathcal{E}^{\mathrm{opt}}_{i,j}=E_{i,j}(Y^{(\mathrm{opt})}_{i,j})$ and $\mathcal{E}^{\mathrm{opt}}_{j,i}=E_{j,i}(Y^{(\mathrm{opt})}_{j,i})$. One shows easily that

$$\mathcal{E}^{\mathrm{opt}}_{i,j}=E_{i,j}(Y^{(\mathrm{opt})}_{i,j}) = W-\alpha_{i,j}W^2-\beta_{i,j}WV+\gamma_{i,j}V^2,\quad \text{where } \alpha_{i,j}=\frac{(\mathsf{s}+\mathsf{bw})^2\kappa_i-\frac{1}{4}\mathsf{s}^2\kappa_j}{\mathsf{s}\cdot\mathsf{bw}((\mathsf{s}+\mathsf{bw})\kappa_i+\mathsf{bw}\cdot\kappa_j)}\kappa_j,$$

$$\beta_{i,j}=\frac{2((3\kappa_i-\kappa_j)\mathsf{s}+4\mathsf{bw}\cdot\kappa_i)}{4\mathsf{bw}(\kappa_i(\mathsf{s}+\mathsf{bw})+(\mathsf{bw}\cdot\kappa_j))}\kappa_j,\quad \gamma_{i,j}=\frac{\mathsf{s}(\kappa_i-\kappa_j)^2}{4\mathsf{bw}(\mathsf{s}\cdot\kappa_i+\mathsf{bw}(\kappa_i+\kappa_j))}.$$

The corresponding expression for $\mathcal{E}^{\mathrm{opt}}_{j,i}$ is symmetrical. We compare $\mathcal{E}^{\mathrm{opt}}_{i,j}$ and $\mathcal{E}^{\mathrm{opt}}_{j,i}$ by studying their difference:

$$\mathscr{E}^{\mathrm{opt}}_{i,j} - \mathscr{E}^{\mathrm{opt}}_{j,i} = (\kappa_i - \kappa_j)\frac{\kappa_i\kappa_j\mathsf{bw}^2 - \frac{1}{4}(\kappa_j^2 - 3\kappa_i\kappa_j + \kappa_i^2)\mathsf{s}^2 - \frac{1}{4}\mathsf{s}\cdot\mathsf{bw}(\kappa_i^2 - 6\kappa_i\cdot\kappa_j + \kappa_j^2)}{\mathsf{bw}(\mathsf{bw}(\kappa_i+\kappa_j) + \mathsf{s}\cdot\kappa_i)(\mathsf{bw}(\kappa_i+\kappa_j) + \mathsf{s}\cdot\kappa_j)}W^2$$
$$- \frac{(\kappa_i - \kappa_j)^3\mathsf{s}(\mathsf{s}+\mathsf{bw})}{2\mathsf{bw}(\mathsf{bw}(\kappa_i+\kappa_j) + \mathsf{s}\cdot\kappa_i)(\mathsf{bw}(\kappa_i+\kappa_j) + \mathsf{s}\cdot\kappa_j)}VW - \frac{(\kappa_i - \kappa_j)^3\mathsf{s}^2}{4\mathsf{bw}(\mathsf{bw}(\kappa_i+\kappa_j) + \mathsf{s}\cdot\kappa_i)(\mathsf{bw}(\kappa_i+\kappa_j) + \mathsf{s}\cdot\kappa_j)}V^2.$$

Assuming, with no loss of generality, that $\kappa_j > \kappa_i$, we want to determine the sign of $E = (\mathscr{E}^{\mathrm{opt}}_{i,j} - \mathscr{E}^{\mathrm{opt}}_{j,i})/(\kappa_j - \kappa_i)$. We let $\kappa_j = (1+\theta)\kappa_i$ and note that E's numerator becomes a quadratic in θ:

$$\frac{-1}{4}\Big((W(W+2V)\mathsf{bw} + (W+V)^2\mathsf{s})\mathsf{s}\cdot\theta^2 + (\mathsf{s}+2\mathsf{bw})^2W^2\cdot\theta + (\mathsf{s}+2\mathsf{bw})^2W^2\Big)\kappa_i^2$$

Letting $a = -(W(W+2V)\mathsf{bw} + (W+V)^2\mathsf{s})\mathsf{s},\ b = (\mathsf{s}+2\mathsf{bw})^2W^2$ and $c = (\mathsf{s}+2\mathsf{bw})^2W^2$, the two solutions for θ are:

$$\theta_- = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad \theta_+ = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

The first solution, θ_-, is negative (because $a < 0$ and $c > 0$); the second solution, θ_+, is positive; and the polynomial is positive when $(\theta_- < \theta < \theta_+)$. We want to show that, under the hypotheses defining this case, θ can never be as large as θ_+. This means that E's numerator, hence, E itself, is always positive.

Because $Y^{(\mathrm{opt})}_{i,j} \leqslant W$, we know that

$$\mathsf{s}\cdot V(\kappa_j - \kappa_i) + \mathsf{s}\cdot W\cdot\kappa_j \leqslant 2(\mathsf{bw}+\mathsf{s})W\kappa_i.$$

Replacing κ_j by $(1+\theta)\kappa_i$, this leads to:

$$2(\mathsf{bw}+\mathsf{s})W - \theta(V+W)\mathsf{s} \geqslant 0.$$

We therefore study the sign of this expression when $\theta = \theta_+$. From solving the quadratic, we know that this latter expression has the same sign as

$$(2\mathsf{bw}+\mathsf{s})(-2a)W - (b + \sqrt{b^2 - 4ac})(V+W)\mathsf{s},$$

which is equal to the positive term $\mathsf{s}(2\mathsf{bw}+\mathsf{s})W$ times the expression:

$$(\mathsf{s}\cdot W^2 + (2\mathsf{bw}+3\mathsf{s})WV + 2\mathsf{s}\cdot V^2) - (W+V)\sqrt{((2(\mathsf{s}+\mathsf{bw})W) + (2\mathsf{s}\cdot V))^2 + \mathsf{s}^2W^2}.$$

It follows that both expressions have the same sign. Because an expression $r - s\sqrt{t}$ with r, s, t all positive has the same sign as $r^2 - s^2t$, the preceding expression has the same sign as

$$-4\Big((\mathsf{s}+\mathsf{bw})\Big((\mathsf{s}+\mathsf{bw})W^4 + (2\mathsf{bw}+3\mathsf{s})VW^3 + 3\mathsf{s}\cdot V^2W^2\Big) + \mathsf{s}^2WV^3\Big),$$

which is obviously negative. Therefore, we cannot have $Y^{(\mathrm{opt})}_{i,j} \leqslant W$ when $\theta = \theta_+$, so that, in this case, we always have $\theta < \theta_+$. It follows that $\mathscr{E}^{\mathrm{opt}}_{i,j} - \mathscr{E}^{\mathrm{opt}}_{j,i}$ has the same sign as $\kappa_i - \kappa_j$, and is nonzero whenever $\kappa_i \neq \kappa_j$. We conclude finally that P_j must be served before P_i whenever $\kappa_j > \kappa_i$. □

Let the workers be indexed by non-increasing risk of interruption: $\kappa_1 \geqslant \kappa_2 \geqslant \cdots \geqslant \kappa_n$,

Theorem 5. *In the current scenario, the optimal schedule for Distrib(p) distributes work to computers in non-increasing order of their interruption risks; it allocates $\alpha_{i,p}W$ units of work to each P_i, for $i \in [1,p]$, where*

- $\alpha_{1,p} = \prod_{j=1}^{p} Y_j^{(\mathrm{opt})}$; *for* $i \in [2,p]:\ \alpha_{i,p} = (1 - Y_i^{(\mathrm{opt})})\prod_{j=i+1}^{p} Y_j^{(\mathrm{opt})}$,
- $Y_1^{(\mathrm{opt})} = 1$; *for* $i \in [2,p]:\ Y_i^{(\mathrm{opt})} = \frac{(2\mathsf{bw}+\mathsf{s})b_{i-1}}{2\mathsf{bw}(a_{i-1}\cdot\mathsf{s}\cdot\kappa_{i-1} + b_{i-1}\cdot\kappa_i)}\kappa_i$,
- $a_1 = \mathsf{s}+\mathsf{bw}$; *for* $i \in [2,p]:\ a_i = a_{i-1}\cdot\mathsf{bw}(\mathsf{s}+\mathsf{bw})\kappa_{i-1} - \frac{1}{4}b_{i-1}\cdot\mathsf{s}\cdot\kappa_i$,
- $b_1 = \mathsf{s}\cdot\mathsf{bw}$; *for* $i \in [2,p]:\ b_i = (a_{i-1}\cdot\mathsf{s}\cdot\kappa_{i-1} + b_{i-1}\cdot\kappa_i)\mathsf{bw}^2$.

The resulting expected work production is $\mathscr{E}^{\mathrm{opt}}(W,p) = W - \frac{a_p}{b_p}W^2\kappa_p$.

Proof. The order of serving workers is given in Lemma 2. We proceed by induction on p. If W units of work are allocated to P_1, then the expected work production is:

$$E(W,1) = W\left(1 - \left(\frac{W}{\mathsf{bw}} + \frac{W}{\mathsf{s}}\right)\kappa_1\right) = W - \frac{\mathsf{s}+\mathsf{bw}}{\mathsf{s}\cdot\mathsf{bw}}W^2\kappa_1.$$

Thus the theorem holds for the base case $p = 1$.

Please cite this article in press as: A. Benoit et al., Static worksharing strategies for heterogeneous computers with unrecoverable interruptions, Parallel Comput. (2010), doi:10.1016/j.parco.2010.03.001

Assume, for induction, that the result holds for up to n workers, and consider the case of $n+1$ workers. Let us study this case in the presence of three hypotheses that do not appear in the theorem: (1) $a_n > 0$; (2) $b_n > 0$; (3) $2\mathsf{bw}\cdot a_n > b_n$. These properties obviously hold when $p = 1$; we shall verify that they hold for arbitrary p.

Let us allocate W-Y units of work to worker P_{n+1}. By induction, the optimal expected work production by the first n workers is:

$$\mathscr{E}_n^{\text{opt}}(Y,n) = Y\left(1 - \frac{a_n}{b_n} Y \kappa_n\right),$$

and this is achieved by allocating $\alpha_{i,n}Y$ units of work to worker P_i, for $i \in [1,n]$. The expected work production with our additional worker, P_{n+1}, is:

$$\begin{aligned} E_{n+1}(Y) &= Y - \frac{a_n}{b_n} Y^2 \kappa_n + (W-Y)\left(1 - \left(\frac{Y}{\mathsf{bw}} + \frac{W-Y}{\mathsf{bw}} + \frac{W-Y}{\mathsf{s}}\right)\kappa_{n+1}\right) \\ &= W - \frac{a_n\cdot\mathsf{s}\cdot\kappa_n + b_n\cdot\kappa_{n+1}}{b_n\cdot\mathsf{s}} Y^{\,2} + \frac{2\mathsf{bw}+\mathsf{s}}{\mathsf{bw}\cdot\mathsf{s}} WY\kappa_{n+1} - \frac{\mathsf{bw}+\mathsf{s}}{\mathsf{bw}\cdot\mathsf{s}} W^2\kappa_{n+1}. \end{aligned}$$

This quantity is maximized when

$$Y = Y_{n+1}^{(\text{opt})} = \frac{(2\mathsf{bw}+\mathsf{s})b_n}{2\mathsf{bw}(a_n\cdot\mathsf{s}\cdot\kappa_n + b_n\cdot\kappa_{n+1})}\kappa_{n+1}.$$

Hypotheses (1) and (2) assert that both a_n and b_n are positive, which implies that $Y_{n+1}^{(\text{opt})}$ also is positive; hypothesis (3) asserts that $b_n < 2\mathsf{bw}\cdot a_n$. Therefore, because $\kappa_{n+1} \leqslant \kappa_n$, we have:

$$\begin{array}{ll} \text{if} & (2\mathsf{bw}+\mathsf{s})b_n < 2\mathsf{bw}(a_n\cdot\mathsf{s} + b_n), \\ \text{then} & (2\mathsf{bw}+\mathsf{s})b_n\kappa_{n+1} < 2\mathsf{bw}(a_n\cdot\mathsf{s}+b_n)\kappa_{n+1} \ \leqslant\ 2\mathsf{bw}(a_n\cdot\mathsf{s}\cdot\kappa_n + b_n\kappa_{n+1}). \end{array}$$

Because a_n and b_n are positive, the last inequality implies that $Y_{n+1}^{(\text{opt})} < 1$; consequently, we see that $0 < Y_{n+1}^{(\text{opt})} < 1$. Therefore, the optimal schedule allocates the following work fractions:

$$\alpha_{n+1,n+1} = 1 - Y_{n+1}^{(\text{opt})}; \quad \text{for } i \in [1,n], \quad \alpha_{i,n+1} = \alpha_{i,n} Y_{n+1}^{(\text{opt})}.$$

We thus find, as claimed:

$$\begin{cases} \alpha_{1,n+1} = \prod_{j=1}^{n+1} Y_j^{(\text{opt})}; \\ \text{and, for } i \in [2,n+1], \quad \alpha_{i,n+1} = \left(1 - Y_i^{(\text{opt})}\right)\prod_{j=i+1}^{n+1} Y_j^{(\text{opt})}. \end{cases}$$

Finally we compute the optimal expected work production, $E(Y_{n+1}^{(\text{opt})})$. We find that

$$\mathscr{E}_{n+1}^{\text{opt}} = W - \frac{a_{n+1}}{b_{n+1}} W^2 \kappa_{n+1},$$

where

$$\begin{aligned} a_{n+1} &= a_n\cdot\mathsf{bw}(\mathsf{s}+\mathsf{bw})\kappa_n - \frac{1}{4} b_n\cdot\mathsf{s}\cdot\kappa_{n+1}, \\ b_{n+1} &= (a_n\cdot\mathsf{s}\cdot\kappa_n + b_n\cdot\kappa_{n+1})\mathsf{bw}^2. \end{aligned}$$

We can now finally verify our three "additional" hypotheses for $n+1$ workers. By hypotheses (1) and (2) for n workers, we have $b_{n+1} > 0$; by hypothesis (3) for n workers, we have $a_{n+1} > 0$, because $\kappa_{n+1} \leqslant \kappa_n$; finally, we note that

$$2\mathsf{bw}\cdot a_{n+1} - b_{n+1} = a_n\cdot\mathsf{bw}^2(\mathsf{s}+2\mathsf{bw})\kappa_n - b_n\cdot\mathsf{bw}\left(\frac{1}{2}\mathsf{s}+\mathsf{bw}\right)\cdot\kappa_{n+1} = (\mathsf{s}+2\mathsf{bw})\cdot\mathsf{bw}\left(a_n\cdot\mathsf{bw}\cdot\kappa_n - \frac{1}{2}b_n\cdot\kappa_{n+1}\right) > 0,$$

because $\kappa_{n+1} \leqslant \kappa_n$, and because of hypothesis (3) for n workers. The three hypothesis are verified for $n+1$ workers. □

5.4. The general case: total heterogeneity

We now consider the general case, where computers have different computation speeds, different communication bandwidths, and different failure rates. To show the intrinsic difficulty of the general case, we focus on the system with only two computers, P_1 and P_2. If we decide for the ordering P_1 then P_2, sending first a chunk of size Y to P_1 and then one of size W – Y to P_2, we derive that the expectation of the amount of work done is

$$E(Y) = Y\left(1 - \left(\frac{Y}{\mathrm{bw}_1} + \frac{Y}{\mathrm{s}_1}\right)\kappa_1\right) + (W - Y)\left(1 - \left(\frac{Y}{\mathrm{bw}_1} + \frac{W - Y}{\mathrm{bw}_2} + \frac{W - Y}{\mathrm{s}_2}\right)\kappa_2\right).$$

We rewrite $E(Y)$ as follows:

$E(Y) = W - \alpha Y^2 + \beta WY - \gamma W^2$ where

$$\alpha = \frac{(\mathrm{s}_1 + \mathrm{bw}_1)\mathrm{s}_2 \cdot \mathrm{bw}_2 \cdot \kappa_1 + (\mathrm{s}_2 + \mathrm{bw}_2) \cdot \mathrm{s}_1 \cdot \mathrm{bw}_1 \cdot \kappa_2 - \mathrm{bw}_2 \cdot \mathrm{s}_1 \cdot \mathrm{s}_2 \cdot \kappa_2}{\mathrm{bw}_1 \cdot \mathrm{bw}_2 \cdot \mathrm{s}_1 \cdot \mathrm{s}_2},$$

$$\beta = \frac{2\mathrm{bw}_1(\mathrm{s}_2 + \mathrm{bw}_2) - \mathrm{bw}_2 \cdot \mathrm{s}_2}{\mathrm{bw}_1 \cdot \mathrm{bw}_2 \cdot \mathrm{s}_2}\kappa_2 \quad \text{and} \quad \gamma = \frac{\mathrm{s}_2 + \mathrm{bw}_2}{\mathrm{s}_2 \cdot \mathrm{bw}_2}\kappa_2.$$

The maximum of $E(Y)$ depends on whether α and/or β are null:

Case $\alpha = \beta = 0$. In this case we have:

$$\mathrm{bw}_1 = \frac{\mathrm{bw}_2 \cdot \mathrm{s}_2}{2(\mathrm{s}_2 + \mathrm{bw}_2)} \quad \text{and} \quad \kappa_1 = \frac{\mathrm{s}_1(\mathrm{s}_2 + \mathrm{bw}_2)}{\mathrm{s}_2 \cdot \mathrm{bw}_2 + 2\mathrm{s}_1(\mathrm{s}_2 + \mathrm{bw}_2)}\kappa_2,$$

and the expectation does not depend on the way the load is distributed among P_1 and P_2. Indeed, we then have $E(Y) = W - \frac{\mathrm{s}_2 + \mathrm{bw}_2}{\mathrm{s}_2 \cdot \mathrm{bw}_2}W^2\kappa_2$, which is the expectation when all the work is given to P_2.

Case $\alpha = 0$ and $\beta \neq 0$. In this case we have:

$$\kappa_1 = \frac{\mathrm{s}_1(\mathrm{s}_2 \cdot \mathrm{bw}_2 - \mathrm{bw}_1(\mathrm{s}_2 + \mathrm{bw}_2))}{\mathrm{s}_2 \cdot \mathrm{bw}_2(\mathrm{s}_1 + \mathrm{bw}_1)}\kappa_2.$$

This can be achieved whatever the parameters of P_2 as soon as bw_1 is small enough (for the numerator to be positive). We then have two sub-cases to consider, depending on the sign of β:

Case $\beta > 0$. This is equivalent to: $\mathrm{bw}_1 > \frac{\mathrm{bw}_2 \cdot \mathrm{s}_2}{2(\mathrm{s}_2 + \mathrm{bw}_2)}$. Then the expectation is increasing with Y, the optimal solution is $Y = W$, and is achieved by giving all the work to P_1.

Case $\beta < 0$. This is equivalent to: $\mathrm{bw}_1 < \frac{\mathrm{bw}_2 \cdot \mathrm{s}_2}{2(\mathrm{s}_2 + \mathrm{bw}_2)}$. Then the expectation is decreasing with Y, the optimal solution is $Y = 0$, and is achieved by giving all the work to P_2.

Case $\alpha > 0$ and $\beta \leqslant 0$. The expectation is then a decreasing function of Y, the optimal solution is $Y = 0$, and is achieved by giving all the work to P_2.

Case $\alpha > 0$ and $\beta \geqslant 0$. The expectation is then an increasing function of Y and then a decreasing one, reaching its maximum over the real line for $Y = \frac{\beta}{2\alpha}W$. Once again, we have two sub-cases to consider:

Case $\beta \geqslant 2\alpha$. This case is equivalent to $\mathrm{s}_1 \cdot \kappa_2 \geqslant 2(\mathrm{bw}_1 + \mathrm{s}_1)\kappa_1$. Then, the expectation reaches its maximum for $Y = W$, that is, by giving all the work to P_1. (For instance, the three conditions defining this case are met when $\mathrm{bw}_1 = \frac{3\mathrm{bw}_2 \cdot \mathrm{s}_2}{4(\mathrm{s}_2 + \mathrm{bw}_2)}$ and $\kappa_2 = 2\frac{(\mathrm{bw}_1 + \mathrm{s}_1)\kappa_1}{\mathrm{s}_1}$.)

Case $\beta < 2\alpha$. In that case, the expectation achieves its maximum for $Y = \frac{\beta}{2\alpha}$ and both computers receive a non-empty share of the work. The expectation is then:

$$\mathscr{E}^{\mathrm{opt}}(W, 2) = W - \left(\gamma - \frac{\beta^2}{4\alpha}\right)W^2.$$

Case $\alpha < 0$ and $\beta \geqslant 0$. The expectation is then an increasing function of Y, the optimal solution is $Y = W$, and is achieved by giving all the work to P_1.

Case $\alpha < 0$ and $\beta \leqslant 0$. The expectation is then a decreasing function of Y and then an increasing one, reaching its minimum over the real line for $Y = \frac{\beta}{2\alpha}W$. Once again, we have two sub-cases to consider:

Case $\beta \geqslant 2\alpha$. Then the expectation is decreasing on the interval of valid values for Y. The optimal solution is $Y = 0$, and is achieved by giving all the work to P_2.

Case $\beta < 2\alpha$. Then the global minimum of the expectation is reached in the interval and the maximum is reached for one of the two bounds $Y = 0$ and $Y = W$. Then the optimal solution is achieved by giving all the work to P_1 if $\frac{\mathrm{s}_1 + \mathrm{bw}_1}{\mathrm{s}_1 \cdot \mathrm{bw}_1}\kappa_1 \geqslant \frac{\mathrm{s}_2 + \mathrm{bw}_2}{\mathrm{s}_2 \cdot \mathrm{bw}_2}\kappa_2$, and by giving it to P_2 otherwise.

In summary, finding the optimal work distribution among two computers whose ordering is given, is quite an involved case study. The next question is: what would be the optimal computer ordering? To attempt to answer this question, we consider that, under both orderings, we are in the non-trivial case that assigns a non-empty share of work to both computers. (To convince one-self that this case indeed exists, remember that we encountered it in Sections 5.1–5.3.) We re-use the above notations while adding to them the subscripts 12 and 21 to denote the computer ordering. Then, under the hypothesis $0 \leqslant \beta_{12} < 2\alpha_{12}$ and $0 \leqslant \beta_{21} < 2\alpha_{21}$, we can form the difference between the optimum expectations of both cases:

$$\mathscr{E}_{12}^{\mathrm{opt}}(W) - \mathscr{E}_{21}^{\mathrm{opt}}(W) = \mathrm{s}_1 \cdot \mathrm{s}_2(\mathrm{bw}_1 \cdot \kappa_1 - \mathrm{bw}_2 \cdot \kappa_2)\frac{core}{4\mathrm{bw}_1 \cdot \mathrm{bw}_2 \cdot denominator}W^2,$$

where

$$\begin{aligned}core = &-(\mathsf{bw}_2\mathcal{K}_2 + \mathsf{bw}_1\mathcal{K}_1)(\mathsf{s}_1\mathsf{s}_2(\mathsf{bw}_1\mathcal{K}_2 + \mathsf{bw}_2\mathcal{K}_1) + \mathsf{bw}_1\mathsf{bw}_2(\mathcal{K}_1\mathsf{s}_2 + \mathcal{K}_2\mathsf{s}_1)) + 5\mathsf{bw}_1\mathsf{bw}_2\mathcal{K}_1\mathcal{K}_2\mathsf{s}_1\mathsf{s}_2 + 4\mathsf{bw}_1\mathsf{bw}_2\mathcal{K}_1\mathcal{K}_2(\mathsf{bw}_1\mathsf{s}_2\\ &+ \mathsf{bw}_2\mathsf{s}_1 + \mathsf{bw}_1\mathsf{bw}_2),\end{aligned}$$

and where *denominator* is the product of the numerators of α_{12}, and α_{21}, both being positive by hypothesis. The difference suggests the importance of the product $\mathsf{bw}_i.\mathcal{K}_i$ for the computer ordering. This is obviously consistent with the results of Section 5.2 ($\mathcal{K}_1 = \mathcal{K}_2$) and Section 5.3 ($\mathsf{bw}_1 = \mathsf{bw}_2$). We thus propose the following conjecture:

Conjecture 1. *In an optimal solution to Distrib(p), the master sends work to the computers in non-increasing order of their products* $\mathsf{bw}_i.\mathcal{K}_i$.

On one side, a numerical search for a counter-example for the two computer case was fruitless. On the other side, we were unable to prove this conjecture even in the simple two computer case. This can be partially explained by the complexity of the proof of Lemma 2, lemma that this conjecture subsumes.

6. Related work

The divisible-load model is a reasonable abstraction of an application made up of a large number of identical, fine-grained parallel computations. Such applications are found in many scientific areas, and we refer the reader to the survey paper [1] and the journal special issue [12] for detailed examples. Also, the divisible-load approach has been applied successfully to a variety of computing platforms, such as bus-shaped, star-shaped, and even tree-shaped platforms. Despite the extensive literature on the divisible-load model, to the best of our knowledge, the current study is the first to consider the divisible-load problem on master–worker platforms whose computers are subject to unrecoverable failures/interruptions.

Our earlier work [4], and its predecessors [7–9], also consider computers with unrecoverable failures/interruptions, but with major differences in the models. In this paper, we allow for *heterogeneous* computers, and we take communication costs into account, while [4] focuses only on identical computers without communication costs. To "compensate" for the additional complexity in the model we study here, we have restricted ourselves in this paper to scenarios where the entire workload is distributed to the worker computers, a strategy that is often suboptimal, even when scheduling a single worker computer [4]. Furthermore, we have not considered here the possible benefits of replicating the execution of some work-units on several worker computers, a key tool for enhancing expected work production in [4]. Obviously, it would be highly desirable to combine the sophisticated platforms of the current study with the sophisticated algorithmics of [4].

We hope to do so in future work, in order to deal with the most general master–worker problem instances – instances that allow heterogeneous computing resources and communication costs, that do not insist that all work be distributed, and that give the scheduler the option of replicating work on multiple worker computers. However, the complexity of the proofs derived in Sections 5.1–5.3, and the fact that we were unable to tackle the general case (Section 5.4), all suggest that we should content ourselves with efficient heuristics rather than searching for optimal solutions.

7. Conclusion

In this paper we have revisited the well-known master–worker paradigm for divisible-load applications, adding the hypothesis that the computers are subject to unrecoverable failures/interruptions. In this novel context, the natural objective of a schedule is to maximize the expected amount of work that gets completed. We have succeeded in providing either closed-form formulas or linear recurrences to characterize optimal solutions for all platforms subject to a single source of heterogeneity: either heterogeneous communications, heterogeneous computing speeds, or heterogeneous failure rates. This provides a nice counterpart to existing results in the classical context of makespan minimization.

In particular, we establish the optimal processor orderings for any platform subject to a single source of heterogeneity:

- any processor ordering for platforms where the only heterogeneity comes from computing speeds,
- non-increasing bandwidths for platforms where the only heterogeneity comes from communication bandwidths,
- non-increasing failure rates for platforms where the only heterogeneity comes from failure rates.

These are very interesting (and somewhat unexpected for computing speeds) results, as they show that the scheduling problem has polynomial complexity: there is no need to explore the combinatorial space of all possible orderings. We conjecture that in the general case, where all three sources of heterogeneity are simultaneously present, processors should be order by non-increasing product of the bandwidths and failure rates.

As discussed in Section 6, we have adopted certain simplifications to the general problem we ultimately aspire to. We have insisted on distributing the entire workload to the worker computers, without replication of work. Our not allowing work replication is particularly unfortunate when contemplating environments that have access to abundant computing resources. This, then, is the first projected avenue for extending the current work. Several other extensions of this work would be desirable also, for instance: (*i*) including a start-up overhead-cost each time a computer executes a piece of work (e.g., to account for the cost of initiating a communication or a checkpointing); (*ii*) studying computers that obey not only linear, but also different risk functions (e.g., when several user categories have different probabilities of returning to reclaim their computers); (*iii*) studying risk functions that are no longer linear (e.g., standard exponential or, importantly, heavy-tailed distributions); and (*iv*) analyzing multi-round strategies, wherein each worker computer receives its share of work in several rounds. Altogether, there are many challenging algorithmic problems to address!.

Acknowledgments

A. Benoit, Y. Robert, and F. Vivien are with Université de Lyon, France. A. Benoit and Y. Robert are with the Institut Universitaire de France. F. Vivien is with INRIA, France. The work of A. Benoit and Y. Robert was supported in part by the ANR StochaGrid project. The work of A. Rosenberg was supported in part by US NSF Grant CNS-0905399.

References

[1] T. Robertazzi, Ten reasons to use divisible load theory, IEEE Computer 36 (5) (2003) 63–68.
[2] V. Bharadwaj, D. Ghose, V. Mani, T. Robertazzi, Scheduling Divisible Loads in Parallel and Distributed Systems, IEEE Computer Society Press, 1996.
[3] O. Beaumont, H. Casanova, A. Legrand, Y. Robert, Y. Yang, Scheduling divisible loads on star and tree networks: results and open problems, IEEE Transactions on Parallel and Distributed Systems 16 (3) (2005) 207–218.
[4] A. Benoit, Y. Robert, A. Rosenberg, F. Vivien, Static strategies for worksharing with unrecoverable interruptions, in: IPDPS'2009, the 23rd IEEE International Parallel and Distributed Processing Symposium, IEEE Computer Society Press, 2009.
[5] J. Abawajy, Fault-tolerant scheduling policy for grid computing systems, in: International Parallel and Distributed Processing Symposium IPDPS'2004, IEEE Computer Society Press, 2004.
[6] S. Albers, G. Schmidt, Scheduling with unexpected machine breakdowns, Discrete Applied Mathematics 110 (2–3) (2001) 85–99.
[7] B. Awerbuch, Y. Azar, A. Fiat, F.T. Leighton, Making commitments in the face of uncertainty: how to pick a winner almost every time, in: 28th ACM SToC, 1996, pp. 519–530.
[8] S. Bhatt, F. Chung, F. Leighton, A. Rosenberg, On optimal strategies for cycle-stealing in networks of workstations, IEEE Transactions on Computers 46 (5) (1997) 545–557.
[9] A.L. Rosenberg, Optimal schedules for cycle-stealing in a network of workstations with a bag-of-tasks workload, IEEE Transactions on Parallel and Distributed Systems 13 (2) (2002) 179–191.
[10] P. Bhat, C. Raghavendra, V. Prasanna, Efficient collective communication in distributed heterogeneous systems, Journal of Parallel and Distributed Computing 63 (2003) 251–263.
[11] M. Snir, S.W. Otto, S. Huss-Lederman, D.W. Walker, J. Dongarra, MPI the Complete Reference, The MIT Press, 1996.
[12] V. Bharadwaj, D. Ghose, T. Robertazzi, Divisible load theory: a new paradigm for load scheduling in distributed systems, Cluster Computing 6 (1) (2003) 7–17.

Please cite this article in press as: A. Benoit et al., Static worksharing strategies for heterogeneous computers with unrecoverable interruptions, Parallel Comput. (2010), doi:10.1016/j.parco.2010.03.001