Static Worksharing Strategies for Heterogeneous Computers with Unrecoverable Failures

Anne Benoit[1,4], Yves Robert[1,4], Arnold Rosenberg[2], and Frédéric Vivien[3,4]

[1] Ecole Normale Supérieure de Lyon, France
{Anne.Benoit,Yves.Robert,Frederic.Vivien}@ens-lyon.fr
[2] Colorado State University, Fort Collins, USA
rsnbrg@colostate.edu
[3] INRIA, France
[4] LIP, UMR 5668 ENS-CNRS-INRIA-UCBL, Lyon, France

Abstract. One has a large workload that is "divisible" (its constituent work's 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 toward this end? 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, given the risk of interruptions. We consider three versions of the preceding problem. Two versions envision *heterogeneous* computing resources: the remote computers may differ from one another in speed; one version envisions *homogeneous* computing resources: the remote computers are identical. One of the heterogeneous versions ignores communication costs (i.e., assumes that they are negligible); the other two versions account explicitly for communication costs. We provide exact expressions for the optimal work expectation for all three versions of the problem. For the most general version (heterogeneous resources, with communication costs), we provide a recurrence for computing this expectation; for the other two versions, we provide closed-form expressions.

1 Introduction

This paper extends well-known results from divisible load theory [11] concerning master-worker computing platforms. Our goal is to optimally distribute a large workload to p remote computers (the "workers") that may differ in speeds; the workers are connected to the "master" via a bus or network. The master wants to send a fraction of the load to each worker, seriatim. The problem is to determine, for each worker, the fraction of the load that it should be sent and the order in which it should be served. This problem has received considerable attention in recent years, and closed-form expressions have been derived for these load fractions [7, 4]. We revisit this problem in the context of remote computers that are subject to *unrecoverable interruptions* [5], and we aim to maximize the expected amount of total work that will be completed. For intuition: An "unrecoverable interruption" may correspond to a hardware crash, an event that is increasingly likely with the advent of massively parallel grid platforms [1, 2]; it may also correspond to the unexpected return of a remote computer's user/owner during an episode of cycle-stealing [3, 10, 12]. Consider the following scenario: on Friday evening, a PhD student has a large set of simulations to run. S/he 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 to work on Monday 8am. What is the student's best strategy? How much simulation data should s/he send to, and execute on, each accessible computer?

We cleave to the preceding scenario and assume that remote computers are vulnerable to interruption, with a risk that grows *linearly with the time the computer has been available.* Other probability distributions can be envisioned, but the linear distribution is very natural in the absence of further information. Also, the linear risk function turns out to be tractable: we have derived optimality results for this distribution. The major achievement of this paper is to provide a distribution strategy that maximizes the expected total amount of work completed.

The paper is organized as follows. We describe the formal framework in detail, in Section 2. We then address three optimization problems. The simplest problem ignores communication costs, but considers a heterogeneous set of resources that may differ in speed (Section 3). The other two problems account for communication costs, first with identical remote computers (Section 4) and then with computers that may differ in speed (Section 5). We provide exact expressions for the optimal work expectation for all three problems. For the first two problems we provide explicit, closed-form expressions; for the last (and most general) problem, we have to resort to a complicated recurrence formula that yields the optimal solution for specific instances in linear time. We provide a brief overview of related work in Section 6; in particular, we compare our approach and results with those of our previous work [5]. We close with some conclusions and perspectives in Section 7.

2 Framework

We have W units of divisible work to execute on p remote computers. Each computer is susceptible to unrecoverable interruptions that "kill" all work in progress (on that computer). All remote computers share the same perfectly known instantaneous probability of being interrupted, and this probability increases with the amount of time the computer has been operating (whether working or not). Within our model, all computers obey the same *risk function* $Pr(T)$ of having been interrupted by the end of the first T time units.

The risk function that is the focus of our study is the linear function $Pr(w) = \kappa w$. It is the most natural model in the absence of further information: the risk of interruption grows linearly with the time that the computer has been available, or equivalently with the amount of work that it could have done. The density function is then $dPr = \kappa dt$ for $t \in [0, 1/\kappa]$ and 0 otherwise, so that

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

We assume that all p computing remote computers obey the same probability failure distribution. For instance, in the earlier-mentioned cycle-stealing scenario, the remote computers are computers from the CS department that can be loaned during the week-end, so they have the same probability of having their owner returning. With more information about the owners, we could refine the scenario and assume different laws for, say, students and staff.

The speed of computer P_i is speed_i. The computers are interconnected by a bus or homogeneous network of bandwidth bw. Each computer will receive a single message from the master that contains its *work chunk*, i.e., all the data necessary to execute its assigned work: in the terminology of [7], this is a single-round distribution strategy. Work is transmitted sequentially to each remote computer, as in the standard divisible load model of [7]. This corresponds to a (somewhat pessimistic) *one-port* model [9], with single-threaded execution and blocking send/receive MPI primitives [13]. We introduce two important notations:

- $\mathsf{z} = \frac{\kappa}{\mathsf{bw}}$, the failure-rate per unit-load communication from the master to any computer;
- $\mathsf{x}_i = \frac{\kappa}{\mathsf{speed}_i}$, the failure-rate per unit-load computation by computer P_i.

If we send a load w_1 to computer P_1 and then a load w_2 to computer P_2, then the expected amount of work executed by P_1 is

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

To understand (Eq. 1), simply observe that P_1 is communicating during the first w_1/bw time-units and is computing during the next w_1/speed_1 time-units. P_1's risk of being interrupted increases linearly with elapsed time, regardless of whether it is communicating or computing. Similarly, we derive that the expected amount of work executed by P_2 is

$$E_2 = w_2 \left(1 - \mathsf{z}(w_1 + w_2) - \mathsf{x}_2 w_2\right).$$

Indeed, P_2 starts computing only after both communications from the master to P_1 and P_2 have completed, which takes $(w_1 + w_2)/\mathsf{bw}$ time-steps; then it computes during w_2/speed_2 additional time-steps. Again, P_2's risk of being interrupted increases linearly with elapsed time, regardless of whether it is waiting (while the master communicates to P_1), or communicating, or computing. If we had only these two remote computers ($p = 2$), then our goal would be to maximize $E_1 + E_2$, the expected total amount of work done.

Note that the formula for expectation E_1 (Eq. 1) assumes that $(\mathsf{z}+\mathsf{x}_1)w_1 \leq 1$. If this condition is not satisfied, then $E_1 = 0$. To avoid such cases, we make a technical assumption and assume that the total load is small enough so that we distribute it entirely to the p computers. Indeed, if the total load is too large, then all computers will be interrupted with certainty (probability 1) before they complete their chunk. In the following, we assume that the p chunks received by the computers *partition* the original load, and that there is a nonzero probability that the last computer is not interrupted before or during its computation. A sufficient condition for this latter condition to hold is $W \leq \frac{1}{\mathsf{z}+\mathsf{x}_{\max}}$, where $\mathsf{x}_{\max} = \frac{\kappa}{\min_{1\leq i\leq p} \mathsf{speed}_i}$ is the failure-rate per unit-load computation of the slowest computer. To see this, simply note that the last computer, say P_i, can always start computing at time-step Y/bw, where $Y \leq W$ is the total load sent to all preceding computers: introducing idle times in the communication cannot improve the solution, as the risk of interruption grows with time. Then P_i needs V/speed_i time-steps to execute its own chunk of size V, where $Y + V \leq W$, whence the claim. We can now formally state the optimization problem:

Definition 1. *We let* DISTRIB(p) *denote the problem of computing* $\mathcal{E}^{opt}(W,p)$*, the optimal value of the expected total amount of work completed when partitioning and distributing the entire workload* $W \leq \frac{1}{\mathsf{z}+\mathsf{x}_{\max}}$ *to the* p *remote computers.*

We turn now to the case $\mathsf{z} = 0$, wherein communication costs can be neglected. Then we shall tackle the case with communication costs and identical remote computers ($\mathsf{x}_i = \mathsf{x}$ for $1 \leq i \leq p$) before moving on to the general case with communication costs and different-speed remote computers.

3 Heterogeneous Workers, No Communication Costs

In this section we deal with the DISTRIB problem when we have p remote computers that differ in speed, but we ignore communication cost (the case $\mathsf{z} = 0$). This scenario models situations wherein computations dominate in the application. We need to introduce symmetric functions to state our result.

Definition 2. *For integers* $n \geq 1$ *and* $i \in \{0, \ldots, n\}$*, we denote by* $\sigma_i^{(n)}$ *the* i*-th symmetric function of* $\mathsf{x}_1, \ldots, \mathsf{x}_n$*:* $\sigma_i^{(n)} = \sum_{1\leq j_1<\cdots<j_i\leq n} \prod_{k=1}^{i} \mathsf{x}_{j_k}$*. 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) *sends a chunk of size* $\frac{\prod_{k\neq i} \mathsf{x}_k}{\sigma_{p-1}^{(p)}} W = \frac{\sigma^{(p)}}{\mathsf{x}_i \sigma_{p-1}^{(p)}}$ *to computer* P_i*. In this case,*

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

Proof. Let $\alpha_{i,p} = \frac{\prod_{k\neq i} \mathsf{x}_k}{\sigma_{p-1}^{(p)}}$ and $\mathsf{f}_p = \frac{\sigma_p^{(p)}}{\sigma_{p-1}^{(p)}}$. We show the result by induction. Note that it 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 done is

$$E(Y) = Y\left(1 - \mathsf{x}_1 Y\right) + \left(W - Y\right)\left(1 - \mathsf{x}_2(W - Y)\right).$$

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

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

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 $\leq n$ workers. 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 optimal expected amount of work done by the first n computers is $\mathcal{E}^{\mathrm{opt}}(Y, n) = Y\left(1 - \mathsf{f}_n Y\right)$, and this is achieved by sending a chunk of size $\alpha_{i,n}Y$ to P_i for $1 \leq i \leq n$. The expected amount of work done by the $n + 1$ workers is then

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

We proceed as in the preceding case. The optimal value is $Y^{(\mathrm{opt})} = \frac{\mathsf{x}_{n+1}}{\mathsf{f}_n+\mathsf{x}_{n+1}}W$, and we derive that $\mathcal{E}^{\mathrm{opt}}(W, n+1) = E(Y^{(\mathrm{opt})}) = W - \mathsf{f}_{n+1}W^2$ 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 $\mathsf{f}_n + \mathsf{x}_{n+1} = \frac{\sigma_n^{(n+1)}}{\sigma_{n-1}^{(n)}}$, 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, $Y^{(\mathrm{opt})} = \frac{\mathsf{x}_{n+1}}{\mathsf{f}_n+\mathsf{x}_{n+1}}W = \frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}}W$. By induction, for $1 \leq i \leq n$, we get $\alpha_{i,n+1} = \alpha_{i,n}\frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} = \frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)}\prod_{1\leq k\leq n, k\neq i}\mathsf{x}_k}{\sigma_{n-1}^{(n)}\sigma_n^{(n+1)}} = \frac{\mathsf{x}_{n+1}\prod_{1\leq k\leq n, k\neq i}\mathsf{x}_k}{\sigma_n^{(n+1)}} = \frac{\prod_{1\leq k\leq n+1, k\neq i}\mathsf{x}_k}{\sigma_n^{(n+1)}}$ as desired. It remains to check the value of $\alpha_{n+1,n+1} = 1 - \frac{\mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} = \frac{\sigma_n^{(n+1)} - \mathsf{x}_{n+1}\sigma_{n-1}^{(n)}}{\sigma_n^{(n+1)}} = \frac{\prod_{1\leq k\leq n}\mathsf{x}_k}{\sigma_n^{(n+1)}}$ which concludes the proof. $\square$

We see that the optimal solution is *symmetric*: each worker's "contribution" to the expectation is a (somewhat complicated, but) symmetric function of all computer speeds.

4 Homogeneous Workers, with Communication Costs

We move now to the case with communication costs. Before dealing with the general case of heterogeneous remote computers, which turns out to be difficult, we address the homogeneous problem, with identical remote computers:

Theorem 2. *When* $\mathsf{x}_i \equiv \mathsf{x}$ *(i.e., speeds are identical), the optimal solution to* DISTRIB(p) *distributes equal-size chunks to the workers, i.e., chunks of size* W/p. *In this case,*

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

Proof. The proof is similar to that of Theorem 1. Let $\mathsf{f}_p = \frac{(p+1)\mathsf{z}+2\mathsf{x}}{2p}$. We show the result by induction. Note that it holds for $p=1$, because $\mathsf{f}_1 = \mathsf{z}+\mathsf{x}$.

Assume that the result holds for $\leq n$ workers. 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 optimal expected amount of work done by the first n workers is $\mathcal{E}^{\text{opt}}(Y,n) = Y\left(1-\mathsf{f}_nY\right)$, and this is achieved by sending a chunk of size Y/n to each P_i, for $1 \leq i \leq n$. The expected amount of work done is then

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

To understand the value of the contribution of P_{n+1} to $E(Y)$, simply note that it 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 is $Y^{(\text{opt})} = \frac{\mathsf{z}+2\mathsf{x}}{2(\mathsf{f}_n+\mathsf{x})}W$ and we derive that $\mathcal{E}^{\text{opt}}(W,n+1) = E(Y^{(\text{opt})}) = W - \mathsf{f}_{n+1}W^2$, where $\mathsf{f}_{n+1} = \mathsf{z}+\mathsf{x} - \frac{(\mathsf{z}+2\mathsf{x})^2}{4(\mathsf{f}_n+\mathsf{x})}$.

Using the induction hypothesis, we get $\mathsf{f}_n + \mathsf{x} = \frac{(n+1)\mathsf{z}+2\mathsf{x}}{2n} + \mathsf{x} = \frac{(n+1)(\mathsf{z}+2\mathsf{x})}{2n}$, so that

$$\mathsf{f}_{n+1} = \mathsf{z}+\mathsf{x} - \frac{n(\mathsf{z}+2\mathsf{x})}{2(n+1)} = \frac{(n+2)\mathsf{z}+2\mathsf{x}}{2(n+1)}$$

as expected. We also obtain $Y^{(\text{opt})} = \frac{n}{n+1}W$, so that each P_i (with $i \leq n$) receives a chunk of size $\frac{Y^{(\text{opt})}}{n} = \frac{W}{n+1}$. We deduce that P_{n+1} receives a chunk of that same size (or we can directly check that $W - Y^{(\text{opt})} = \frac{W}{n+1}$). □

Interestingly, the optimal solution mandates sending *equal-size* chunks to all workers. This contrasts with the standard divisible load setting. In that setting, when one aims to minimize the total time needed to execute a certain amount of work, one has all workers terminate their computations simultaneously [7], so that the earlier workers served by the master receive larger chunks than do the later workers.

5 Heterogeneous Workers, with Communication Costs

We are now ready for the general case, which accounts for communication costs while serving different-speed computers. We need a few notations before stating the main result of this paper:

Definition 3. *Define the sequence $\lambda = \lambda_0, \lambda_1, \ldots$, as follows:*
$\lambda_0 = \lambda_1 = 4$; *for* $n \geq 2$, $\lambda_n = \lambda_{n-1} - \frac{1}{4}\lambda_{n-2}$
For convenience, let $\lambda_{-1} = 0$.

Note that $\lambda_n = 4(1+n)/2^n$ for all $n \geq 0$. We use the sequence λ is used to characterize the optimal solution to the general problem.

Theorem 3. *In the general case, the optimal solution to* Distrib(p) *does not depend upon the ordering of the communications from the master. When using the ordering $P_1, P_2, \ldots, P_p$, the optimal solution sends a chunk of size $\alpha_{i,p}W$ to P_i, where*

1. $\alpha_{p,p} = \dfrac{2\mathsf{f}_{p-1} - \mathsf{z}}{2(\mathsf{f}_{p-1} + \mathsf{x}_p)}$ *for* $p \geq 2$ *and* $\alpha_{1,1} = 1$*;*
2. $\alpha_{i,p} = \alpha_{i,i}\dfrac{\mathsf{z} + 2\mathsf{x}_{i+1}}{2(\mathsf{f}_i + \mathsf{x}_{i+1})}$ *for* $p - 1 \geq i \geq 2$*;*
3. $\alpha_{1,p} = 1 - \alpha_{2,p}$ *for* $p \geq 2$.

In this case, $\mathcal{E}^{opt}(W,p) = W - \mathsf{f}_p W^2$, *where* $\mathsf{f}_p = \dfrac{\sum_{i=0}^{p} \lambda_i \sigma_{p-i}^{(p)} \mathsf{z}^i}{\sum_{i=0}^{p-1} \lambda_i \sigma_{p-i-1}^{(p)} \mathsf{z}^i}$ *for* $p \geq 1$

Proof. The proof is similar to those of Theorems 1 and 2 but it is more involved. Due to lack of space, we refer to the companion research report [6] for a complete proof. Note that the theorem holds for $p = 1$, because $\mathsf{f}_1 = \frac{\lambda_0 \mathsf{x}_1 + \lambda_1 \mathsf{z}}{\lambda_0} = \mathsf{z} + \mathsf{x}_1$.

To give intuition for the result, in particular why the ordering of the communications is not important, consider the case with 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, we derive that the expected amount of work completed is

$$E(W) = Y(1 - \mathsf{f}_1 Y) + (W - Y)(1 - (\mathsf{z}W + \mathsf{x}_2(W - Y))).$$

As before, to understand this reckoning, we note that P_2 waits for the first chunk to be sent to P_1; then it receives its own chunk; both steps account for the term $\mathsf{z}W$ on the righthand side. Finally, P_2 computes its chunk, whence the term $\mathsf{x}_2(W - Y)$. We rewrite

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

The optimal value is $Y^{(\mathrm{opt})} = \frac{\mathsf{z}+2\mathsf{x}_2}{2(\mathsf{f}_1+\mathsf{x}_2)}W = \alpha_{1,2}W$, and we derive that

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

where $\mathsf{f}_2 = \mathsf{z} + \mathsf{x}_2 - \frac{(\mathsf{z}+2\mathsf{x}_2)^2}{4(\mathsf{f}_1+\mathsf{x}_2)}$. After some easy manipulation, we see that

$$\mathsf{f}_2 = \frac{4\mathsf{x}_1\mathsf{x}_2 + 4(\mathsf{x}_1 + \mathsf{x}_2)\mathsf{z} + 3\mathsf{z}^2}{4(\mathsf{x}_1 + \mathsf{x}_2 + \mathsf{z})},$$

as desired. We see that the formula is symmetric in x_1 and x_2, thereby showing that the ordering of the communications has no significance. Please refer to [6] for the general case of the induction. □

The interested reader can check that we end up with the values of f_p and $\alpha_{i,p}$ given in Theorem 1 when $\mathsf{z} = 0$, and those given in Theorem 2 when $\mathsf{x}_i = \mathsf{x}$.

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 [11] and the journal special issue [8] 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 [5], and its predecessors [3, 10, 12], also consider computers with unrecoverable failures/interruptions, but with major differences in the models. The current paper allows *heterogeneous* computers and communication costs, while [5] 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 remote computers, a strategy that is often suboptimal, even when scheduling a single remote computer [5]. Furthermore, we have not considered here the possible benefits of replicating the execution of some work units on several remote computers, a key tool for enhancing expected work production in [5]. Obviously, it would be highly desirable to combine the sophisticated platforms of the current study with the sophisticated algorithmics of [5]. 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 remote computers.

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, thereby providing a nice counterpart to existing results in the classical context of makespan minimization. In particular, our demonstration that the ordering of communications has no impact on the optimal solution is a very interesting (and somewhat unexpected) result, as it shows that the scheduling problem has polynomial complexity: there is no need to explore the combinatorial space of all possible orderings.

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 remote 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 remote 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. Y. Robert is 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 Grants CNS-0842578 and -0905399.

References

[1] Abawajy, J.: Fault-tolerant scheduling policy for grid computing systems. In: International Parallel and Distributed Processing Symposium IPDPS 2004. IEEE Computer Society Press, Los Alamitos (2004)
[2] Albers, S., Schmidt, G.: Scheduling with unexpected machine breakdowns. Discrete Applied Mathematics 110(2-3), 85–99 (2001)
[3] Awerbuch, B., Azar, Y., Fiat, A., Leighton, F.T.: Making commitments in the face of uncertainty: How to pick a winner almost every time. In: 28th ACM SToC, pp. 519–530 (1996)
[4] Beaumont, O., Casanova, H., Legrand, A., Robert, Y., Yang, Y.: Scheduling divisible loads on star and tree networks: results and open problems. IEEE Trans. Parallel Distributed Systems 16(3), 207–218 (2005)
[5] Benoit, A., Robert, Y., Rosenberg, A., Vivien, F.: Static strategies for worksharing with unrecoverable interruptions. In: IPDPS 2009, the 23rd IEEE International Parallel and Distributed Processing Symposium. IEEE Computer Society Press, Los Alamitos (2009)
[6] Benoit, A., Robert, Y., Rosenberg, A., Vivien, F.: Static worksharing strategies for heterogeneous computers with unrecoverable failures. Research Report 2009-23, LIP, ENS Lyon, France (July 2009), `graal.ens-lyon.fr/~yrobert/`

[7] Bharadwaj, V., Ghose, D., Mani, V., Robertazzi, T.: Scheduling Divisible Loads in Parallel and Distributed Systems. IEEE Computer Society Press, Los Alamitos (1996)
[8] Bharadwaj, V., Ghose, D., Robertazzi, T.: Divisible load theory: a new paradigm for load scheduling in distributed systems. Cluster Computing 6(1), 7–17 (2003)
[9] Bhat, P., Raghavendra, C., Prasanna, V.: Efficient collective communication in distributed heterogeneous systems. Journal of Parallel and Distributed Computing 63, 251–263 (2003)
[10] Bhatt, S., Chung, F., Leighton, F., Rosenberg, A.: On optimal strategies for cycle-stealing in networks of workstations. IEEE Trans. Computers 46(5), 545–557 (1997)
[11] Robertazzi, T.: Ten reasons to use divisible load theory. IEEE Computer 36(5), 63–68 (2003)
[12] Rosenberg, A.L.: Optimal schedules for cycle-stealing in a network of workstations with a bag-of-tasks workload. IEEE Trans. Parallel Distrib. Syst. 13(2), 179–191 (2002)
[13] Snir, M., Otto, S.W., Huss-Lederman, S., Walker, D.W., Dongarra, J.: MPI the complete reference. The MIT Press, Cambridge (1996)