hal-00788641, version 1 - 14 Feb 2013

Energy-aware checkpointing of divisible tasks with soft or hard deadlines

Guillaume Aupy, Anne Benoit, Rami Melhem, Paul Renaud-Goud, Yves Robert

ISSN 0249-6399 ISRN INRIA/RR--8238--FR+ENG

RESEARCH REPORT
N° 8238
February 2013
Project-Team ROMA

hal-00788641, version 1 - 14 Feb 2013

hal-00788641, version 1 - 14 Feb 2013

Energy-aware checkpointing of divisible tasks with soft or hard deadlines

Guillaume Aupy[*†], Anne Benoit[*†‡], Rami Melhem[§], Paul Renaud-Goud[*†¶], Yves Robert[*†‡‖]

Project-Team ROMA

Research Report n° 8238 — February 2013 — 30 pages

Abstract: In this paper, we aim at minimizing the energy consumption when executing a divisible workload under a bound on the total execution time, while resilience is provided through checkpointing. We discuss several variants of this multi-criteria problem. Given the workload, we need to decide how many chunks to use, what are the sizes of these chunks, and at which speed each chunk is executed. Furthermore, since a failure may occur during the execution of a chunk, we also need to decide at which speed a chunk should be re-executed in the event of a failure. The goal is to minimize the expectation of the total energy consumption, while enforcing a deadline on the execution time, that should be met either in expectation (soft deadline), or in the worst case (hard deadline). For each problem instance, we propose either an exact solution, or a function that can be optimized numerically. The different models are then compared through an extensive set of experiments.

Key-words: Fault-tolerance, checkpointing, energy, model

* LIP, École Normale Supérieure de Lyon, France
† INRIA
‡ Institut Universitaire de France
§ University of Pittsburgh, USA
¶ LABRI - University of Bordeaux, France
‖ University of Tennessee Knoxville, USA

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

hal-00788641, version 1 - 14 Feb 2013

Protocoles de checkpoint à faible consommation énergétique pour des tâches divisibles, sous contrainte faible ou forte du temps d'exécution

Résumé : Ce travail vise à minimiser la consommation d'énergie lors de l'exécution d'une quantité de travail divisible, sous contrainte de temps d'exécution, sur une plateforme soumise à des fautes passagères. La résilience est fournie grâce à un protocole de sauvegarde de points de reprise (*checkpoints*). Nous étudions différentes variantes de ce problème multi-critère. Etant donnée une quantité de travail, nous devons décider du découpage optimal de celle-ci en morceaux qui seront sauvegardés ; plus précisément, nous devons déterminer le nombre, la taille et la vitesse d'exécution de chacun de ces morceaux. De plus, sachant qu'une faute peut arriver pendant l'exécution de l'un d'entre eux, nous devons décider de la vitesse de ré-exécution de ces morceaux en cas de faute. Le but de ce travail est de minimiser l'espérance de la consommation d'énergie, sous une contrainte temporelle, qui peut être atteinte en moyenne (contrainte faible), ou dans le pire cas (contrainte forte). Pour chaque instance du problème, nous fournissons soit une solution exacte, soit une fonction qui peut être optimisée numériquement. Les différents modèles sont ensuite comparés via un ensemble étendu d'expériences.

Mots-clés : Tolérance aux pannes, checkpoint, énergie, modèle

hal-00788641, version 1 - 14 Feb 2013

1 Introduction

Divisible load scheduling has been extensively studied in the past years [5, 10]. For divisible applications, the computational workload can be divided into an arbitrary number of chunks, whose sizes can be freely chosen by the user. Such applications occur for instance in the processing of very large data files, e.g., signal processing, linear algebra computation, or DNA sequencing. Traditionally, the goal is to minimize the makespan of the application, i.e., the total execution time.

Nowadays, high performance computing is facing a major challenge with the increasing frequency of failures [9]. There is a need to use fault tolerance or resilience mechanisms to ensure the efficient progress and correct termination of the applications in the presence of failures. A well-established method to deal with failures is *checkpointing*: a checkpoint is taken at the end of the execution of each chunk. During the checkpoint, we check for the accuracy of the result; if the result is not correct, due to a transient failure (such as a memory error or software error), the chunk is re-executed. This model with transient failures is one of the most used in the literature, see for instance [18, 8].

Furthermore, energy-awareness is now recognized as a first-class constraint in the design of new scheduling algorithms. To help reduce energy dissipation, current processors from AMD, Intel and Transmetta allow the speed to be set dynamically, using a dynamic voltage and frequency scaling technique (DVFS). Indeed, a processor running at speed s dissipates s^3 watts per unit of time [4]. We therefore focus on two objective functions: execution time and energy consumption, while resilience is ensured through checkpointing. More precisely, we aim at minimizing energy consumption, including that of checkpointing and re-execution in case of failure, while enforcing a bound on execution time.

Given a workload W, we need to decide how many chunks to use, and of which sizes. Using more chunks leads to a higher checkpoint cost, but smaller chunks imply less computation loss (and less re-execution) when a failure occurs. We assume that a chunk can fail only once, i.e., we re-execute each chunk at most once. Indeed, the probability that a fault would strike during both the first execution and the re-execution is negligible. We discuss the accuracy of this assumption in Section 4.

Due to the probabilistic nature of failure hits, it is natural to study the expectation $\mathbb{E}(E)$ of the energy consumption, because it represents the average cost over many executions. As for the bound D on execution time (the deadline), there are two relevant scenarios: either we enforce that this bound is a *soft deadline* to be met in expectation, or we enforce that this bound is a *hard deadline* to be met in the worst case. The former scenario corresponds to flexible environment where task deadlines can be viewed as average response times [6], while the latter scenario corresponds to real-time environments where task deadlines are always strictly enforced [14]. In both scenarios, we have to determine the number of chunks, their sizes, and the speed at which to execute (and possibly re-execute) every chunk.

Our first contribution is to formalize this important multi-objective problem. The general problem consists of finding n, the number of chunks, as well as the speeds for the execution and the re-execution of each chunk, both for soft and hard deadlines. We identify and discuss two important sub-cases that help tackling the most general problem instance: (i) a single chunk (the task is

hal-00788641, version 1 - 14 Feb 2013

atomic); and (ii) re-execution speed is always identical to first execution speed. The second contribution is a comprehensive study of all problem instances; for each instance, we propose either an exact solution, or a function that can be optimized numerically. We also analytically prove the accuracy of our model that enforces a single re-execution per chunk. We then compare the different models through an extensive set of experiments. We compare the optimal energy consumption under various models with a set of different parameters. It turns out that when λ is small, it is sufficient to restrict the study to a single chunk, while when λ increases, it is better to use multiple chunks and different re-execution speeds.

The rest of the paper is organized as follows. First we discuss related work in Section 2. The model and the optimization problems are formalized in Section 3. We discuss the accuracy of the model in Section 4. We first focus in Section 5 on the simpler case of an atomic task, i.e., with a single chunk. The general problem with multiple chunks, where we need to decide for the number of chunks and their sizes, is discussed in Section 6. In Section 7, we report several experiments to assess the differences between the models, and the relative gain due to chunking or to using different speeds for execution and re-execution. Finally, we provide some concluding remarks and future research directions in Section 8.

2 Related work

Dynamic power management through voltage/frequency scaling [15] utilizes the slack in a given computation to reduce energy consumption while checkpointing.The authors of [7, 11] utilize that slack to improve the reliability of the computation. Hence, it is natural to explore the interplay of power management and fault tolerance [12], when both techniques result in delaying the completion time of tasks, thus resulting in a tradeoff between power consumption, reliability and performance. This tri-criteria optimization problem has been explored by many researchers, especially in real-time and embedded systems where the completion time of a task is as important as the reliability of its result.

The power/reliability/performance tradeoff has been explored from many different angles. In [16], an adaptive scheme is presented to place checkpoints based on the expected frequency of faults and is combined with dynamic speed scaling depending on the actual occurrence of faults. Similarly, in [12], the placement of checkpoints is chosen in a way that minimizes the total energy consumption assuming that the slack reserved for rollback recovery is used for speed scaling if faults do not occur. In [18], the effect of frequency scaling on the fault rate was considered and incorporated into the optimization problem. In [17], the study of the tri-criteria optimization was extended to the case of multiple tasks executing on the same processor. In [13], a constraint logic programming-based approach is presented to decide for the voltage levels, the start times of processes and the transmission times of messages, in such a way that transient faults are tolerated, timing constraints are satisfied and energy is minimized.

Recently, off-line scheduling heuristics that consider the three criteria were presented for systems where active replication, rather than fault recovery, is used to enhance reliability [1]. Selective re-execution of some tasks were considered

hal-00788641, version 1 - 14 Feb 2013

in [3] to achieve a given level of reliability while minimizing energy, when tasks graphs are scheduled on multiprocessors with hard deadlines. Approximation algorithms for particular types of task graphs were presented to efficiently solve the same problem in [2].

In this work, we consider two types of deadlines that are commonly used for real-time tasks; hard and soft deadlines. In hard real-time systems [14], deadlines should be strictly met and any computation that does not meet its deadline is not useful to the system. These systems are built to cope with worst-case scenarios, especially in critical applications where catastrophic consequences may result from missing deadlines. Soft real-time systems [6] are more flexible and are designed to adapt to system changes that may prevent the meeting of the deadline. They are suited to novel applications such as multimedia and interactive systems. In these systems, it is desired to reduce the expected completion time rather than to meet hard deadlines.

3 Framework

Given a workload W, the problem is to divide W into a number of chunks and to decide at which speed each chunk is executed. In case of a transient failure during the execution of one chunk, this chunk is re-executed, possibly at a different speed. We formalize the model in Section 3.1, and then different variants of the optimization problem are defined in Section 3.2. Table 1 summarizes the main notations.

W	total amount of work
s	processor speed for first execution
σ	processor speed for re-execution
T_C	checkpointing time
E_C	energy spent for checkpointing

Table 1: List of main notations.

3.1 Model

Consider first the case of a single chunk (or atomic task) of size W, denoted as SingleChunk. We execute this chunk on a processor that can run at several speeds. We assume continuous speeds, i.e., the speed of execution can take an arbitrary positive real value. The execution is subject to failure, and resilience is provided through the use of checkpointing. The overhead induced by checkpointing is twofold: execution time T_C, and energy consumption E_C.

We assume that failures strike with uniform distribution, hence the probability that a failure occurs during an execution is linearly proportional to the length of this execution. Consider the first execution of a task of size W executed at speed s: the execution time is $T_{\text{exec}} = W/s + T_C$, hence the failure probability is $P_{\text{fail}} = \lambda T_{\text{exec}} = \lambda(W/s+T_C)$, where λ is the instantaneous failure rate. If there is indeed a failure, we re-execute the task at speed σ (which may or may not differ from s); the re-execution time is then $T_{\text{reexec}} = W/\sigma + T_C$ so

hal-00788641, version 1 - 14 Feb 2013

that the expected execution time is

$$\begin{aligned}\mathbb{E}(T) &= T_{\text{exec}} + P_{\text{fail}} T_{\text{reexec}} \\ &= (W/s + T_C) + \lambda(W/s + T_C)(W/\sigma + T_C)\ . \end{aligned} \quad (1)$$

Similarly, the worst-case execution time is

$$\begin{aligned}T_{wc} &= T_{\text{exec}} + T_{\text{reexec}} \\ &= (W/s + T_C) + (W/\sigma + T_C)\ . \end{aligned} \quad (2)$$

Remember that we assume success after re-execution, so we do not account for second and more re-executions. Along the same line, we could spare the checkpoint after re-executing the last task in a series of tasks, but this unduly complicates the analysis. In Section 4, we show that this model with only a single re-execution is accurate up to second order terms when compared to the model with an arbitrary number of failures that follows an Exponential distribution of parameter λ.

What is the expected energy consumed during execution? The energy consumed during the first execution at speed s is $Ws^2 + E_C$, where E_C is the energy consumed during a checkpoint. The energy consumed during the second execution at speed σ is $W\sigma^2 + E_C$, and this execution takes place with probability $P_{\text{fail}} = \lambda T_{\text{exec}} = \lambda(W/s + T_C)$, as before. Hence the expectation of the energy consumed is

$$\mathbb{E}(E) = (Ws^2 + E_C) + \lambda\,(W/s + T_C)\left(W\sigma^2 + E_C\right). \quad (3)$$

With multiple chunks (MultipleChunks model), the execution times (worst case or expected) are the sum of the execution times for each chunk, and the expected energy is the sum of the expected energy for each chunk (by linearity of expectations).

We point out that the failure model is coherent with respect to chunking. Indeed, assume that a divisible task of weight W is split into two chunks of weights w_1 and w_2 (where $w_1 + w_2 = W$). Then the probability of failure for the first chunk is $P^1_{\text{fail}} = \lambda(w_1/s + T_C)$ and that for the second chunk is $P^2_{\text{fail}} = \lambda(w_2/s + T_C)$. The probability of failure $P_{\text{fail}} = \lambda(W/s + T_C)$ with a single chunk differs from the probability of failure with two chunks only because of the extra checkpoint that is taken; if $T_C = 0$, they coincide exactly. If $T_C > 0$, there is an additional risk to use two chunks, because the execution lasts longer by a duration T_C. Of course this is the price to pay for a shorter re-execution time in case of failure: Equation (1) shows that the expected re-execution time is $P_{\text{fail}}T_{\text{reexec}}$, which is quadratic in W. There is a trade-off between having many small chunks (many T_C to pay, but small re-execution cost) and a few larger chunks (fewer T_C, but increased re-execution cost).

3.2 Optimization problems

The optimization problem is stated as follows: given a deadline D and a divisible task whose total computational load is W, the problem is to partition the task into n chunks of size w_i, where $\sum_{i=1}^{n} w_i = W$, and choose for each chunk an execution speed s_i and a re-execution speed σ_i in order to minimize the expected energy consumption:

$$\mathbb{E}(E) = \sum_{i=1}^{n} (w_i s_i^2 + E_C) + \lambda \left(\frac{w_i}{s_i} + T_C \right) \left(w_i \sigma_i^2 + E_C \right),$$

subject to the constraint that the deadline is met either in expectation or in the worst case:

EXPECTED-DEADLINE	$\mathbb{E}(T) = \sum_{i=1}^{n} \left(\frac{w_i}{s_i} + T_C + \lambda \left(\frac{w_i}{s_i} + T_C \right) \left(\frac{w_i}{\sigma_i} + T_C \right) \right) \leq D$
HARD-DEADLINE	$T_{wc} = \sum_{i=1}^{n} \left(\frac{w_i}{s_i} + T_C + \frac{w_i}{\sigma_i} + T_C \right) \leq D$

The unknowns are the number of chunks n, the sizes of these chunks w_i, the speeds for the first execution s_i and the speeds for the second execution σ_i. We consider two variants of the problem, depending upon re-execution speeds:

- SINGLESPEED : in this simpler variant, the re-execution speed is always the same as the speed chosen for the first execution. We then have to determine a single speed for each chunk: $\sigma_i = s_i$ for all i.
- MULTIPLESPEEDS : in this more general variant, the re-execution speed is freely chosen, and there are two different speeds to determine for each chunk.

We also consider the variant with a single chunk (SINGLECHUNK), i.e., the task is atomic and we only need to decide for its execution speed (in the SINGLESPEED model), or for its execution and re-execution speeds (in the MULTIPLESPEEDS model). We start the study in Section 5 with this simpler problem.

4 Accuracy of the model

In this section, we discuss the accuracy of this model, which accounts for a single re-execution. We compare the expressions of the expected deadline and energy (in Equations (1) and (3)) to those obtained when adopting the more advanced model where an arbitrary number of Exponentially distributed failures can strike during execution and re-execution. We only deal with soft deadlines here, because no hard deadline can be enforced for the model with Exponentially distributed failures (the execution time of a chunk can be arbitrarily large, although such an event has low probability to occur).

Assume that failures are distributed using an Exponential distribution of parameter λ: the probability of failure during a time interval of length t is $P_{\text{fail}} = 1 - e^{-\lambda t}$. Consider a single task of size W that we first execute at speed s. If we detect a transient failure at the end of the execution, we re-execute the task until success, using speed σ at each of these new attempts. To the best of our knowledge, the expressions for $\mathbb{E}(T)$ and $\mathbb{E}(E)$ are unknown for this model, and we establish them below:

Proposition 1. *With an arbitrary number of Exponentially distributed failures and one single task of size W,*

$$\mathbb{E}(T) = W/s + T_C \quad + e^{\lambda(W/\sigma + T_C)} \left(1 - e^{-\lambda(W/s + T_C)}\right) (W/\sigma + T_C) \tag{4}$$

$$\mathbb{E}(E) = W s^2 + E_C \quad + e^{\lambda(W/\sigma + T_C)} \left(1 - e^{-\lambda(W/s + T_C)}\right) \left(W \sigma^2 + E_C\right) \tag{5}$$

hal-00788641, version 1 - 14 Feb 2013

hal-00788641, version 1 - 14 Feb 2013

Proof. With an Exponential distribution, Equation (1) can be rewritten as $\mathbb{E}(T) = T_{\text{exec}} + P_{\text{fail}}\mathbb{E}(T_{\text{reexec}})$, where $T_{\text{exec}} = W/s + T_C$ and $P_{\text{fail}} = 1 - e^{-\lambda(W/s+T_C)}$. Since all re-executions are done at speed σ, the expectation of the re-execution time obeys the following equation:

$$\mathbb{E}(T_{\text{reexec}}) = (W/\sigma + T_C) + \left(1 - e^{-\lambda(W/\sigma+T_C)}\right)\mathbb{E}(T_{\text{reexec}})$$

We use the memoryless property of the Exponential distribution here: after a failure, the expectation of the time to re-execute the task is exactly the same as before the failure This leads to $\mathbb{E}(T_{\text{reexec}}) = e^{\lambda(W/\sigma+T_C)}(W/\sigma + T_C)$. Reporting in the first equation, we end up with Equation (4). The expression of the expected energy consumption (Equation (5)) is derived using the same line of reasoning. □

Proposition 2. *With an arbitrary number of Exponentially distributed failures and one single task of size W, when $\lambda \to 0$,*

$$\mathbb{E}(T) = (W/s + T_C) + \lambda(W/s + T_C)(W/\sigma + T_C) + O\left(\lambda^2\right) \quad (6)$$

$$\mathbb{E}(E) = (Ws^2 + E_C) + \lambda(W/s + T_C)(W\sigma^2 + E_C) + O\left(\lambda^2\right) \quad (7)$$

Proof. The first-order Taylor expansion of $x \mapsto e^x$ around 0 gives:

$$\mathbb{E}(T) \underset{\lambda\to 0}{=} (W/s + T_C) + \left(1 + \lambda(W/s + T_C) + O\left(\lambda^2(W/s + T_C)^2\right)\right) \times \left(\lambda(W/\sigma + T_C) + O\left(\lambda^2(W/\sigma + T_C)^2\right)\right)(W/\sigma + T_C)$$

Hence,

$$\mathbb{E}(T) \underset{\lambda\to 0}{=} (W/s + T_C) + \left(\lambda(W/s + T_C) + O\left(\lambda^2\right)\right)(W/\sigma + T_C)$$
$$\mathbb{E}(T) \underset{\lambda\to 0}{=} (W/s + T_C) + \lambda(W/s + T_C)(W/\sigma + T_C) + O\left(\lambda^2\right)$$

Again, the energy formula is built using the same rationale. □

As a consequence of Proposition 2, the formulas that we consider with one single re-execution (Equations (1) and (3)) are accurate up to second order terms when compared to the model with an arbitrary number of Exponential failures. Note that this result is not obvious, because we drop a potentially arbitrarily large number of re-executions in the linear model with at most one re-execution. Furthermore, the result extends naturally when considering a divisible task and MultipleChunks, since the result holds for each chunk, and by summation, one single re-execution of each chunk is accurate up to second order terms.

5 With a single chunk

In this section, we consider the case of a single chunk, or equivalently of an atomic task: given a non-divisible workload W and a deadline D, find the values of s and σ that minimize

$$\mathbb{E}(E) = (Ws^2 + E_C) + \lambda\left(\frac{W}{s} + T_C\right)(W\sigma^2 + E_C)$$

hal-00788641, version 1 - 14 Feb 2013

subject to

$$\mathbb{E}(T) = \left(\frac{W}{s} + T_C\right) + \lambda\left(\frac{W}{s} + T_C\right)\left(\frac{W}{\sigma} + T_C\right) \leq D$$

in the Expected-Deadline model, and subject to

$$\frac{W}{s} + T_C + \frac{W}{\sigma} + T_C \leq D$$

in the Hard-Deadline model. We first deal with the SingleSpeed model, where we enforce $\sigma = s$, before moving on to the MultipleSpeeds model.

5.1 Single speed model

In this section, we express $\mathbb{E}(E)$ as functions of the speed s. That is, $\mathbb{E}(E)(s) = (Ws^2+E_C)(1+\lambda(W/s+T_C))$. The following result is valid for both Expected-Deadline and Hard-Deadline models.

Lemma 1. *$\mathbb{E}(E)$ is convex on $\mathbb{R}_+^\star$. It admits a unique minimum*

$$s^\star = \frac{\lambda W}{6(1+\lambda T_C)}\left(\frac{-(3\sqrt{3}\sqrt{27a^2-4a}-27a+2)^{1/3}}{2^{1/3}} - \frac{2^{1/3}}{(3\sqrt{3}\sqrt{27a^2-4a}-27a+2)^{1/3}} - 1\right) \quad (8)$$

where $a = \lambda E_C\left(\frac{2(1+\lambda T_C)}{\lambda W}\right)^2$.

Proof. Let us prove that $g(s) = \mathbb{E}(E)(s)$ is convex and admits a unique minimum: we have $g'(s) = s(2W(1+\lambda T_C)) + \lambda W^2 - \frac{\lambda W E_C}{s^2}$, $g''(s) = (2W(1+\lambda T_C)) + \frac{2\lambda W E_C}{s^3} > 0$. This function is strictly convex in $\mathbb{R}_+^\star$, and $g' \underset{0^+}{\rightarrow} -\infty$, $g' \underset{\infty}{\rightarrow} \infty$ thus there exist a unique minimum.

Let us find the minimum. For $s > 0$, we have:

$$g'(s) = 0 \Leftrightarrow \left(\frac{2(1+\lambda T_C)}{\lambda W}\right)^3 s^3 + \left(\frac{2(1+\lambda T_C)}{\lambda W}\right)^2 s^2 - \lambda E_C\left(\frac{2(1+\lambda T_C)}{\lambda W}\right)^2 = 0$$

$$\Leftrightarrow X^3 + X^2 - \lambda E_C\left(\frac{2(1+\lambda T_C)}{\lambda W}\right)^2 = 0 \quad \text{where } X = \frac{2(1+\lambda T_C)}{\lambda W}s$$

Using a computer algebra software, it is easy to show that the minimum is obtained at the value $s = s^\star$ given by Equation 8. □

5.1.1 Expected deadline

In the SingleSpeed Expected-Deadline model, we denote $\mathbb{E}(T)(s) = (W/s + T_C)(1+\lambda(W/s+T_C))$ the constraint on the execution time.

Lemma 2. *For any D, if $T_C + \lambda T_C^2 \geq D$, then there is no solution. Otherwise, the constraint on the execution time can be rewritten as* $s \in \left[W\frac{1+2\lambda T_C+\sqrt{4\lambda D+1}}{2(D-T_C(1+\lambda T_C))}, +\infty\right($.

hal-00788641, version 1 - 14 Feb 2013

Proof. The function $s \mapsto \mathbb{E}(T)(s)$ is strictly decreasing and converges to $T_C + \lambda T_C^2$. Hence, if $T_C + \lambda T_C^2 \geq D$, then there is no solution. Else there exist a minimum speed s_0 such that, $\mathbb{E}(T)(s_0) = D$, and for all $s \geq s_0$, $\mathbb{E}(T)(s) \leq D$.

More precisely, $s_0 = W \frac{1 + 2\lambda T_C + \sqrt{4\lambda D + 1}}{2(D - T_C(1 + \lambda T_C))}$: since there is a unique solution to $\mathbb{E}(T)(s) = D$, we can solve this equation in order to find s_0. □

To simplify the following results, we define

$$s_0 = W \frac{1 + 2\lambda T_C + \sqrt{4\lambda D + 1}}{2(D - T_C(1 + \lambda T_C))}. \tag{9}$$

Proposition 3. *In the* SingleSpeed *model, it is possible to numerically compute the optimal solution for* SingleChunk *as follows:*

1. *If $T_C + \lambda T_C^2 \geq D$, then there is no solution;*
2. *Else, the optimal speed is* $\max(s_0, s^\star)$.

Proof. This is a corollary of Lemma 1: because $s \mapsto \mathbb{E}(T)(s)$ is convex on $\mathbb{R}_+^\star$, then its restriction to the interval $[s_0, +\infty($ is also convex and admits a unique minimum:

- if $s^\star < s_0$, then $\mathbb{E}(T)$ (s) is increasing on $[s_0, +\infty($, then the optimal solution is s_0
- else, clearly the minimum is reached when $s = s^\star$.

The optimal solution is then $\max(s_0, s^\star)$. □

5.1.2 Hard deadline

In the Hard-Deadline model, the bound on the execution time can be written as $2\left(\frac{W}{s} + T_C\right) \leq D$

Lemma 3. *In the* SingleSpeed Hard-Deadline *model, for any D, if $2T_C \geq D$, then there is no solution. Otherwise, the constraint on the execution time can be rewritten as* $s \in \left[\frac{W}{\frac{D}{2} - T_C}; +\infty\right($

Proof. The constraint on the execution time is now $2\left(\frac{W}{s} + T_C\right) \leq D$. □

Proposition 4. *Let $s^\star$ the solution indicated in Equation 8. In the* Single-Speed Hard-Deadline *model if $2T_C \geq D$, then there is no solution. Otherwise, the minimum is reached when* $s = \max\left(s^\star, \frac{W}{\frac{D}{2} - T_C}\right)$.

Proof. The fact that there is no solution when $2T_C \geq D$ comes from Lemma 3. Otherwise, the result is obvious by convexity of the expected energy function. □

hal-00788641, version 1 - 14 Feb 2013

5.2 Multiple speeds model

In this section, we consider the general MultipleSpeeds model. We use the following notations:

$$\mathbb{E}(E)(s,\sigma) = (Ws^2 + E_C) + \lambda(W/s + T_C)(W\sigma^2 + E_C)$$

Let us first introduce a preliminary Lemma:

Lemma 4 (Convexity SingleChunk)**.** *The problem of minimizing $A_0 + \alpha_0 x^2$ under the constraint $A_1 + \frac{\alpha_1}{x} \leq A_2$ where A_0, A_1, A_2 are constants and α_0, α_1 are positive constants is solved when x is minimum, that is when $A_1 + \frac{\alpha_1}{x} = A_2$.*

Proof. The function $A_0 + \alpha_0 x^2$ is strictly increasing, so it is is minimized when x is minimum. The function $A_1 + \frac{\alpha_1}{x}$ is strictly decreasing with $\lim_{x\to 0} = +\infty$, hence an upper bound is reached when x is minimum. With those two results, we can say that the constraint should be tight in order to solve our problem. □

5.2.1 Expected deadline

The execution time in the MultipleSpeeds Expected-Deadline model can be written as

$$\mathbb{E}(T)(s,\sigma) = (W/s + T_C) + \lambda(W/s + T_C)(W/\sigma + T_C)$$

We start by giving a useful property, namely that the deadline is always tight in the MultipleSpeeds Expected-Deadline model:

Lemma 5. *In the* MultipleSpeeds Expected-Deadline *model, in order to minimize the energy consumption, the deadline should be tight.*

Proof. Considering s and W fixed, then $\mathbb{E}(T)(s,\sigma) = T_0 + \frac{\alpha}{\sigma} \leq D$, and $\mathbb{E}(E)(s,\sigma) = E_0 + \alpha\sigma^2$, where $T_0 = (W/s + E_C) + \lambda T_C(W/s + T_C)$, $E_0 = (Ws^2 + E_C) + \lambda E_C(W/s + T_C)$ and $\alpha = W(W/s + T_C)$ are constant. With Lemma 4 we conclude that the deadline should be tight. □

This lemma allows us to express σ as a function of s:

$$\sigma = \frac{\lambda W}{\frac{D}{\frac{W}{s}+T_C} - (1 + \lambda T_C)}.$$

Also we reduce the bi-criteria problem to the minimization problem of the single-variable function:

$$s \mapsto Ws^2 + E_C + \lambda\left(\frac{W}{s} + T_C\right)\left(W\left(\frac{\lambda W}{\frac{D}{\frac{W}{s}+T_C} - (1 + \lambda T_C)}\right)^2 + E_C\right) \quad (10)$$

which can be solved numerically.

5.2.2 Hard deadline

In this model we have similar results as with EXPECTED-DEADLINE. The constraint on the execution time writes: $\frac{W}{s} + T_C + \frac{W}{\sigma} + T_C \leq D$. Another corollary of Lemma 4 is:

Lemma 6. *In the* MULTIPLESPEEDS EXPECTED-DEADLINE *model, in order to minimize the energy consumption, the deadline should be tight.*

This lemma allows us to express σ as a function of s:

$$\sigma = \frac{W}{(D - 2T_C)s - W} s$$

Finally, we reduce the bi-criteria problem to the minimization problem of the single-variable function:

$$s \mapsto Ws^2 + E_C + \lambda \left(\frac{W}{s} + T_C \right) \left(W \left(\frac{W}{(D - 2T_C)s - W} s \right)^2 + E_C \right) \qquad (11)$$

which can be solved numerically.

6 Several chunks

In this section, we deal with the general problem of a divisible task of size W that can be split into an arbitrary number of chunks. We divide the task into n chunks of size w_i such that $\sum_{i=1}^{n} w_i = W$. Each chunk is executed once at speed s_i, and re-executed (if necessary) at speed σ_i. The problem is to find the values of n, w_i, s_i and σ_i that minimize

$$\mathbb{E}(E) = \sum_i \left(w_i s_i^2 + E_C \right) + \lambda \sum_i \left(\frac{w_i}{s_i} + T_C \right) \left(w_i \sigma_i^2 + E_C \right)$$

subject to

$$\sum_i \left(\frac{w_i}{s_i} + T_C \right) + \lambda \sum_i \left(\frac{w_i}{s_i} + T_C \right) \left(\frac{w_i}{\sigma_i} + T_C \right) \leq D$$

in the EXPECTED-DEADLINE model, and subject to

$$\sum_i \left(\frac{w_i}{s_i} + T_C \right) + \sum_i \left(\frac{w_i}{\sigma_i} + T_C \right) \leq D$$

in the HARD-DEADLINE model. We first deal with the SINGLESPEED model, where we enforce $\sigma_i = s_i$, before dealing with the MULTIPLESPEEDS model.

hal-00788641, version 1 - 14 Feb 2013

hal-00788641, version 1 - 14 Feb 2013

6.1 Single speed model

6.1.1 Expected deadline

In this section, we deal with the SingleSpeed Expected-Deadline model and consider that for all i, $\sigma_i = s_i$. Then:

$$\mathbb{E}(T)(\cup_i(w_i, s_i, s_i)) = \sum_i \left(\frac{w_i}{s_i} + T_C\right) + \lambda \sum_i \left(\frac{w_i}{s_i} + T_C\right)^2$$
$$\mathbb{E}(E)(\cup_i(w_i, s_i, s_i)) = \sum_i \left(w_i s_i^2 + E_C\right)\left(1 + \lambda\left(\frac{w_i}{s_i} + T_C\right)\right)$$

Theorem 1. *In the optimal solution to the problem with the* SingleSpeed Expected-Deadline *model, all n chunks are of equal size W/n and executed at the same speed s.*

Proof. Consider the optimal solution, and assume by contradiction that it includes two chunks w_1 and w_2, executed at speeds s_1 and s_2, where either $s_1 \neq s_2$, or $s_1 = s_2$ and $w_1 \neq w_2$. Let us assume without loss of generality that $\frac{w_1}{s_1} \geq \frac{w_2}{s_2}$.

We show that we can find a strictly better solution where both chunks have size $w = \frac{1}{2}(w_1 + w_2)$, and are executed at same speed s (to be defined later). The size and speed of the other chunks are kept the same. We will show that the execution time of the new solution is not larger than in the optimal solution, while its energy consumption is strictly smaller, hence leading to the contradiction.

We have seen that

$$\mathbb{E}(T)((w_1, s_1), (w_2, s_2)) = \frac{w_1}{s_1} + T_C + \frac{w_2}{s_2} + T_C + \lambda\left(\frac{w_1}{s_1} + T_C\right)^2 + \lambda\left(\frac{w_2}{s_2} + T_C\right)^2$$
$$\mathbb{E}(T)((w, s), (w, s)) = 2\left(\frac{w}{s} + T_C\right) + 2\lambda\left(\frac{w}{s} + T_C\right)^2$$

Hence,

$$\mathbb{E}(T)((w_1, s_1), (w_2, s_2)) - \mathbb{E}(T)((w, s), (w, s)) = \left(\frac{w_1}{s_1} + \frac{w_2}{s_2} - \frac{2w}{s}\right) + \lambda\left(\left(\frac{w_1}{s_1}\right)^2 + \left(\frac{w_2}{s_2}\right)^2 - 2\left(\frac{w}{s}\right)^2\right)$$

Similarly, we know that:

$$\mathbb{E}(E)((w_1, s_1), (w_2, s_2)) = w_1 s_1^2 + E_C + w_2 s_2^2 + E_C + \lambda\left(\frac{w_1}{s_1} + T_C\right)(w_1 s_1^2 + E_C) + \lambda\left(\frac{w_2}{s_2} + T_C\right)(w_2 s_2^2 + E_C)$$
$$\mathbb{E}(E)((w, s), (w, s)) = 2\left(ws^2 + E_C\right) + 2\lambda\left(\frac{w}{s} + T_C\right)\left(ws^2 + E_C\right)$$

and deduce

$$\mathbb{E}(E)((w_1, s_1), (w_2, s_2)) - \mathbb{E}(E)((w, s), (w, s)) = \left(w_1 s_1^2 + w_2 s_2^2 - 2ws^2\right)(1 + \lambda T_C) + \lambda E_C\left(\frac{w_1}{s_1} + \frac{w_2}{s_2} - \frac{2w}{s}\right) + \lambda\left(w_1^2 s_1 + w_2^2 s_2 - 2w^2 s\right) \quad (12)$$

hal-00788641, version 1 - 14 Feb 2013

Let us now define

$$s_A = \frac{2w}{\frac{w_1}{s_1}+\frac{w_2}{s_2}} = \frac{w_1+w_2}{\frac{w_1}{s_1}+\frac{w_2}{s_2}}$$

$$s_B = \frac{\sqrt{2}w}{\left(\left(\frac{w_1}{s_1}\right)^2+\left(\frac{w_2}{s_2}\right)^2\right)^{1/2}} = \frac{w_1+w_2}{\left(2\left(\frac{w_1}{s_1}\right)^2+2\left(\frac{w_2}{s_2}\right)^2\right)^{1/2}}$$

We then fix $s = \max(s_A, s_B)$. Then, since $s \geq s_A$, we have $\frac{w_1}{s_1}+\frac{w_2}{s_2}-\frac{2w}{s} \geq 0$, and since $s \geq s_B$, we have $\left(\frac{w_1}{s_1}\right)^2+\left(\frac{w_2}{s_2}\right)^2-2\left(\frac{w}{s}\right)^2 \geq 0$. This ensures that $\mathbb{E}(T)((w_1,s_1),(w_2,s_2)) - \mathbb{E}(T)((w,s),(w,s)) \geq 0$.

Note that

$$\frac{(w_1+w_2)^2}{s_B^2} - \frac{(w_1+w_2)^2}{s_A^2} = 2\left(\frac{w_1}{s_1}\right)^2+2\left(\frac{w_2}{s_2}\right)^2-\left(\frac{w_1}{s_1}+\frac{w_2}{s_2}\right)^2 = \left(\frac{w_1}{s_1}-\frac{w_2}{s_2}\right)^2 \geq 0$$

This means that $s_A \geq s_B$, hence $s = s_A$. To prove that $\mathbb{E}(E)((w_1,s_1),(w_2,s_2)) - \mathbb{E}(E)((w,s),(w,s)) > 0$, we want to show that:

1. $w_1 s_1^2 + w_2 s_2^2 - 2ws^2 \geq 0$
2. $\frac{w_1}{s_1}+\frac{w_2}{s_2}-\frac{2w}{s} \geq 0$
3. $w_1^2 s_1 + w_2^2 s_2 - 2w^2 s \geq 0$
4. and that one of the previous inequalities is strict.

Note that by definition of $s = s_A$, the second inequality is true.

Let us first show that $w_1 s_1^2 + w_2 s_2^2 - 2ws_A^2 \geq 0$

$$\begin{aligned}
&\left(\frac{w_1}{s_1}+\frac{w_2}{s_2}\right)^2 \left(w_1 s_1^2 + w_2 s_2^2 - (w_1+w_2)\left(\frac{w_1+w_2}{\frac{w_1}{s_1}+\frac{w_2}{s_2}}\right)^2\right) \\
&= w_1^3 + w_1 w_2^2\left(\frac{s_1}{s_2}\right)^2 + 2\frac{w_1 w_2}{s_1 s_2} w_1 s_1^2 + w_2^3 + w_2 w_1^2 \left(\frac{s_2}{s_1}\right)^2 + 2\frac{w_2 w_1}{s_1 s_2} w_2 s_2^2 - (w_1+w_2)^3 \\
&= w_1 w_2^2\left(\left(\frac{s_1}{s_2}\right)^2 + 2\frac{s_2}{s_1} - 3\right) + w_1^2 w_2\left(\left(\frac{s_2}{s_1}\right)^2 + 2\frac{s_1}{s_2} - 3\right) \\
&= w_1 w_2^2 g\left(\frac{s_1}{s_2}\right) + w_1^2 w_2 g\left(\frac{s_2}{s_1}\right)
\end{aligned}$$

where $g : u \mapsto u^2 + \frac{2}{u} - 3$. It is easy to show that g is nonnegative on $\mathbb{R}_+^\star$: indeed, $g'(u) = \frac{2}{u^2}(u^3-1)$ is negative in $[0,1[$ and positive in $]1,\infty[$, and the unique minimum is $g(1) = 0$. We derive that $w_1 s_1^2 + w_2 s_2^2 - 2ws_A^2 \geq 0$.

hal-00788641, version 1 - 14 Feb 2013

Let us now show that $w_1^2 s_1 + w_2^2 s_2 - 2w^2 s \geq 0$ Remember that $2w = w_1 + w_2$.

$$\begin{aligned} 2\left(\frac{w_1}{s_1}+\frac{w_2}{s_2}\right)&\left(w_1^2 s_1 + w_2^2 s_2 - \frac{(w_1+w_2)^3}{2\left(\frac{w_1}{s_1}+\frac{w_2}{s_2}\right)}\right) \\ &= 2w_1^3 + 2w_1^2 w_2 \frac{s_1}{s_2} + 2w_2^3 + 2w_1 w_2^2 \frac{s_2}{s_1} - (w_1+w_2)^3 \\ &= w_1^3 + w_2^3 + w_1^2 w_2 \left(2\frac{s_1}{s_2} - 3\right) + w_1 w_2^2 \left(2\frac{s_2}{s_1} - 3\right) \end{aligned}$$

Remember that we assumed without loss of generality that $\frac{w_1}{s_1} \geq \frac{w_2}{s_2}$.

$$\begin{aligned} 2\left(\frac{w_1}{s_1}+\frac{w_2}{s_2}\right)&\left(w_1^2 s_1 + w_2^2 s_2 - \frac{(w_1+w_2)^3}{2\left(\frac{w_1}{s_1}+\frac{w_2}{s_2}\right)}\right) \\ &\geq w_2^3\left(\left(\frac{s_1}{s_2}\right)^3 + 1 + \left(\frac{s_1}{s_2}\right)^2\left(2\frac{s_1}{s_2}-3\right) + \frac{s_1}{s_2}\left(2\frac{s_2}{s_1}-3\right)\right) \\ &\geq 3w_2^3\left(\left(\frac{s_1}{s_2}\right)^3 - \left(\frac{s_1}{s_2}\right)^2 - \frac{s_1}{s_2} + 1\right) \\ &\geq 3w_2^3\left(\left(\frac{s_1}{s_2}-1\right)^2\left(\frac{s_1}{s_2}+1\right)\right) \geq 0 \end{aligned}$$

Let us now conclude our study: if $\frac{s_1}{s_2} \neq 1$, then the energy consumption of the optimal solution is strictly greater than the one from our solution which is a contradiction. Hence we must have $s_1 = s_2$, and $w_1 \neq w_2$ (in fact, since we assumed that $\frac{w_1}{s_1} \geq \frac{w_2}{s_2}$, we must have $w_1 > w_2$). Then we can refine the previous analysis, and obtain that $w_1^2 s_1 + w_2^2 s_2 - 2w^2 s > 0$: again, the optimal energy consumption is strictly greater than in our solution; this is the final contradiction and concludes the proof. □

Thanks to this result, we know that the problem with n chunks can be rewritten as follows: find s such that

$$n\left(\frac{W}{ns}+T_C\right) + n\lambda\left(\frac{W}{ns}+T_C\right)^2 = \frac{W}{s} + nT_C + \frac{\lambda}{n}\left(\frac{W}{s}+nT_C\right)^2 \leq D$$

in order to minimize

$$n\left(\frac{W}{n}s^2 + E_C\right) + n\lambda\left(\frac{W}{ns}+T_C\right)\left(\frac{W}{n}s^2+E_C\right) = \left(Ws^2 + nE_C\right)\left(1 + \frac{\lambda}{n}\left(\frac{W}{s}+nT_C\right)\right)$$

One can see that this reduces to the SingleChunk problem with the SingleSpeed model (Section 5.1) up to the following parameter changes:

- $\lambda \leftarrow \frac{\lambda}{n}$
- $T_C \leftarrow nT_C$
- $E_C \leftarrow nE_C$

hal-00788641, version 1 - 14 Feb 2013

If the number of chunks n is given, we can express the minimum speed such that there is a solution with n chunks:

$$s_0(n) = W \frac{1 + 2\lambda T_C + \sqrt{4\frac{\lambda D}{n} + 1}}{2(D - nT_C(1 + \lambda T_C))}. \tag{13}$$

We can verify that when $D \leq nT_C(1 + \lambda n)$, there is no solution, hence obtaining an upper bound on n. Therefore, the two variables problem (with unknowns n and s) can be solved numerically.

6.1.2 Hard deadline

In the Hard-Deadline model, all results still hold, they are even easier to prove since we do not need to introduce a second speed.

Theorem 2. *In the optimal solution to the problem with the* SingleSpeed Hard-Deadline *model, all n chunks are of equal size W/n and executed at the same speed s.*

Proof. The proof is similar to the one of Theorem 1, except we do not need to study the case where $s_B > s_A$. □

6.2 Multiple speeds model

6.2.1 Expected deadline

In this section, we still deal with the problem of a divisible task of size W that we can split into an arbitrary number of chunks, but using the more general MultipleSpeeds model. We start by proving that all re-execution speeds are equal:

Let us first introduce a preliminary Lemma:

Lemma 7 (Convexity MultipleChunks)**.** *The problem of minimizing $A_0 + \alpha_0 x_0^2 + \alpha_1 x_1^2$ under the constraint $\frac{\alpha_0}{x_0} + \frac{\alpha_1}{x_1} \leq A_1$ where A_0 is a constant, and A_1, α_0, α_1 are positive constants, is solved when $x_0 = x_1$, and when the constraint is tight: $\frac{\alpha_0}{x_0} + \frac{\alpha_1}{x_1} = A_1$.*

Proof. First remark that when x_1 is fixed, then according to Lemma 4, the constraint should be tight. Hence this is true for the optimal solution (any optimal solution when the constraint is not tight can be improved by reducing one of the variables).

To prove the result now that we know that the constraint is tight, it suffices to replace in the function we wish to minimize, $x_0 = \frac{\alpha_0}{A_1 - \frac{\alpha_1}{x_1}}$. Differentiating $A_0 + \alpha_0 \times \left(\frac{\alpha_0}{A_1 - \frac{\alpha_1}{x_1}}\right)^2 + \alpha_1 x_1^2$ with respect to x_1 gives $-\frac{2\alpha_1\alpha_0^3}{x_1^2\left(A_1 - \frac{\alpha_1}{x_1}\right)^3} + 2\alpha_1 x_1$. Then we obtain that the equation is minimized (by differentiating again, we can see that the function is convex) when $-\frac{2\alpha_1\alpha_0^3}{x_1^2\left(A_1 - \frac{\alpha_1}{x_1}\right)^3} + 2\alpha_1 x_1 = 0$, that is $-x_0 + x_1 = 0$, hence the result. □

Note that if A_1 is nonpositive, then there is no solution.

hal-00788641, version 1 - 14 Feb 2013

Lemma 8. *In the* MULTIPLESPEEDS *model, all re-execution speeds are equal in the optimal solution:* $\exists\sigma, \forall i, \sigma_i = \sigma$, *and the deadline is tight.*

Proof. This is a direct corollary of Lemma 7. If we consider the w_i and s_i to be fixed, then we can write $\mathbb{E}(T)(\cup_i(w_i, s_i, \sigma_i)) = T_0 + \sum_i \frac{\alpha_i}{\sigma_i}$, and $\mathbb{E}(E)(\cup_i(w_i, s_i, \sigma_i)) = E_0 + \sum_i \alpha_i \sigma_i^2$, where T_0, E_0 and α_i are constant. Assuming $D - T_0 > 0$ (otherwise there is no solution), we can apply Lemma 7, then the problem is minimized when the deadline is tight, and when for all i, $\sigma_i = \frac{\sum_i \alpha_i}{D - T_0}$. □

We can now redefine

$$\mathbb{E}(T)(\cup_i(w_i, s_i, \sigma_i)) = T(\cup_i(w_i, s_i), \sigma)$$
$$\mathbb{E}(E)(\cup_i(w_i, s_i, \sigma_i)) = E(\cup_i(w_i, s_i), \sigma)$$

Theorem 3. *In the* MULTIPLESPEEDS *model, all chunks have the same size* $w_i = \frac{W}{n}$, *and are executed at the same speed* s, *in the optimal solution.*

Proof. We first prove that chunks are of equal size. Assume first, by contradiction, that the optimal solution has two chunks of different sizes, for instance $w_1 < w_2$. These chunks are executed at speeds s_1 and s_2. Thanks to Lemma 8, both chunks are re-executed at a same speed σ. We consider the solution with two chunks of size $w = \frac{1}{2}(w_1 + w_2)$, executed at a same speed s (to be defined later), and re-executed at speed σ (the value of the re-execution speed in the optimal solution). The size and speed of the other chunks are kept the same. We show that the execution time is not greater than in the optimal solution, while the energy consumption is strictly smaller, hence leading to the contradiction.

We have seen that

$$\mathbb{E}(T)((w_1, s_1), (w_2, s_2), \sigma) = \frac{w_1}{s_1} + T_C + \frac{w_2}{s_2} + T_C + \lambda\left(\frac{w_1}{s_1} + T_C\right)\left(\frac{w_1}{\sigma} + T_C\right) + \lambda\left(\frac{w_2}{s_2} + T_C\right)\left(\frac{w_2}{\sigma} + T_C\right)$$

$$\mathbb{E}(T)((w, s), (w, s), \sigma) = 2\left(\frac{w}{s} + T_C\right) + 2\lambda\left(\frac{w}{s} + T_C\right)\left(\frac{w}{\sigma} + T_C\right)$$

Hence,

$$\mathbb{E}(T)((w_1, s_1), (w_2, s_2), \sigma) - \mathbb{E}(T)((w, s), (w, s), \sigma) = (1 + \lambda T_C)\left(\frac{w_1}{s_1} + \frac{w_2}{s_2} - \frac{2w}{s}\right) + \frac{\lambda}{\sigma}\left(\frac{w_1^2}{s_1} + \frac{w_2^2}{s_2} - \frac{2w^2}{s}\right)$$

Similarly, we know that:

$$\mathbb{E}(E)((w_1, s_1), (w_2, s_2), \sigma) = w_1 s_1^2 + E_C + w_2 s_2^2 + E_C + \lambda\left(\frac{w_1}{s_1} + T_C\right)\left(w_1\sigma^2 + E_C\right) + \lambda\left(\frac{w_2}{s_2} + T_C\right)\left(w_2\sigma^2 + E_C\right)$$

$$\mathbb{E}(E)((w, s), (w, s), \sigma) = 2\left(ws^2 + E_C\right) + 2\lambda\left(\frac{w}{s} + T_C\right)\left(w\sigma^2 + E_C\right)$$

hal-00788641, version 1 - 14 Feb 2013

and deduce

$$\mathbb{E}(E)((w_1, s_1), (w_2, s_2), \sigma) - \mathbb{E}(E)((w, s), (w, s), \sigma) = \left(w_1 s_1^2 + w_2 s_2^2 - 2ws^2\right) + \lambda E_C \left(\frac{w_1}{s_1} + \frac{w_2}{s_2} - \frac{2w}{s}\right) + \lambda\sigma^2 \left(\frac{w_1^2}{s_1} + \frac{w_2^2}{s_2} - \frac{2w^2}{s}\right) \quad (14)$$

Let us now define

$$s_A = \frac{2w}{\frac{w_1}{s_1} + \frac{w_2}{s_2}} = \frac{w_1 + w_2}{\frac{w_1}{s_1} + \frac{w_2}{s_2}}$$
$$s_B = \frac{2w^2}{\frac{w_1^2}{s_1} + \frac{w_2^2}{s_2}} = \frac{1}{2} \frac{(w_1 + w_2)^2}{\frac{w_1^2}{s_1} + \frac{w_2^2}{s_2}}$$

We then fix $s = \max(s_A, s_B)$. Then, since $s \geq s_A$, we have $\frac{w_1}{s_1} + \frac{w_2}{s_2} - \frac{2w}{s} \geq 0$, and since $s \geq s_B$, we have $\frac{w_1^2}{s_1} + \frac{w_2^2}{s_2} - \frac{2w^2}{s} \geq 0$. This ensures that $\mathbb{E}(T)((w_1, s_1), (w_2, s_2), \sigma) - \mathbb{E}(T)((w, s), (w, s), \sigma) \geq 0$. To prove that $\mathbb{E}(E)((w_1, s_1), (w_2, s_2), \sigma) - \mathbb{E}(E)((w, s), (w, s), \sigma) \geq 0$, there remains to show that $w_1 s_1^2 + w_2 s_2^2 - 2ws^2 \geq 0$.

Let us first suppose that $s_A > s_B$ Then we have $s = s_A$, and let us show that $w_1 s_1^2 + w_2 s_2^2 - 2ws_A^2 \geq 0$:

$$\begin{aligned}
&\left(\frac{w_1}{s_1} + \frac{w_2}{s_2}\right)^2 \left(w_1 s_1^2 + w_2 s_2^2 - (w_1 + w_2)\left(\frac{w_1 + w_2}{\frac{w_1}{s_1} + \frac{w_2}{s_2}}\right)^2\right) \\
&= w_1^3 + w_1 w_2^2 \left(\frac{s_1}{s_2}\right)^2 + 2\frac{w_1 w_2}{s_1 s_2} w_1 s_1^2 + w_2^3 + w_2 w_1^2 \left(\frac{s_2}{s_1}\right)^2 + 2\frac{w_2 w_1}{s_1 s_2} w_2 s_2^2 - (w_1 + w_2)^3 \\
&= w_1 w_2^2 \left(\left(\frac{s_1}{s_2}\right)^2 + 2\frac{s_2}{s_1} - 3\right) + w_1^2 w_2 \left(\left(\frac{s_2}{s_1}\right)^2 + 2\frac{s_1}{s_2} - 3\right) \\
&= w_1 w_2^2 g\left(\frac{s_1}{s_2}\right) + w_1^2 w_2 g\left(\frac{s_2}{s_1}\right)
\end{aligned}$$

where $g : u \mapsto u^2 + \frac{2}{u} - 3$. We know from the proof of Theorem 1 that g is positive on $\mathbb{R}_+^\star$, hence $w_1 s_1^2 + w_2 s_2^2 - 2ws_A^2 \geq 0$.

Finally, since $s > s_B$, we have $\frac{w_1^2}{s_1} + \frac{w_2^2}{s_2} - \frac{2w^2}{s} > 0$, and all other terms of $\mathbb{E}(E)((w_1, s_1), (w_2, s_2), \sigma) - \mathbb{E}(E)((w, s_A), (w, s_A), \sigma)$ are non-negative, hence proving that the new solution is strictly better than the optimal one, and leading to a contradiction.

Let us now suppose that $s_A \leq s_B$ Then we have $s = s_B$. Moreover, we have $(w_2 - w_1)(\frac{w_2}{s_2} - \frac{w_1}{s_1}) \leq 0$ (this comes directly from $s_A \leq s_B$), and since we assume that $w_2 > w_1$, $\frac{w_2}{s_2} - \frac{w_1}{s_1} \leq 0$. Let us show that $w_1 s_1^2 + w_2 s_2^2 - 2ws_B^2 > 0$:

$$
\begin{aligned}
4\left(\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}\right)^2 \times \left(w_1 s_1^2 + w_2 s_2^2 - (w_1+w_2)\left(\frac{1}{2}\frac{(w_1+w_2)^2}{\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}}\right)^2\right) \\
=&4w_1^5 + 8w_1^3w_2^2\frac{s_1}{s_2} + 4w_1w_2^4\left(\frac{s_1}{s_2}\right)^2 + 4w_2^5 + 8w_1^2w_2^3\frac{s_2}{s_1} + 4w_1^4w_2\left(\frac{s_2}{s_1}\right)^2 - (w_1+w_2)^5 \\
=&3\left(w_1^5+w_2^5\right) + w_1^3w_2^2\left(8\frac{s_1}{s_2}-10\right) + w_2^3w_1^2\left(8\frac{s_2}{s_1}-10\right) \\
&+ w_1w_2^4\left(4\left(\frac{s_1}{s_2}\right)^2-5\right) + w_1^4w_2\left(4\left(\frac{s_2}{s_1}\right)^2-5\right)
\end{aligned}
$$

$$
\begin{aligned}
&4\left(\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}\right)^2\left(w_1 s_1^2 + w_2 s_2^2 - (w_1+w_2)\left(\frac{1}{2}\frac{(w_1+w_2)^2}{\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}}\right)^2\right) \\
&= 4w_1^5 + 8w_1^3w_2^2\frac{s_1}{s_2} + 4w_1w_2^4\left(\frac{s_1}{s_2}\right)^2 + 4w_2^5 + 8w_1^2w_2^3\frac{s_2}{s_1} \\
&\quad + 4w_1^4w_2\left(\frac{s_2}{s_1}\right)^2 - (w_1+w_2)^5 \\
&= 3\left(w_1^5+w_2^5\right) + w_1^3w_2^2\left(8\frac{s_1}{s_2}-10\right) + w_2^3w_1^2\left(8\frac{s_2}{s_1}-10\right) \\
&\quad + w_1w_2^4\left(4\left(\frac{s_1}{s_2}\right)^2-5\right) + w_1^4w_2\left(4\left(\frac{s_2}{s_1}\right)^2-5\right)
\end{aligned}
$$

Now because $w_1 \geq \frac{w_2 s_1}{s_2}$, we can bound the last equation. Let $u = \frac{s_1}{s_2}$ (and hence $w_1 \geq u \times w_2$):

$$
\begin{aligned}
4\left(\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}\right)^2 \times \left(w_1 s_1^2 + w_2 s_2^2 - (w_1+w_2)\left(\frac{1}{2}\frac{(w_1+w_2)^2}{\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}}\right)^2\right) \\
\geq& w_2^5\left(3\left(u^5+1\right) + u^3\left(8u-10\right) + u^2\left(8\frac{1}{u}-10\right) + u\left(4u^2-5\right) + u^4\left(4\frac{1}{u^2}-5\right)\right) \\
=& w_2^5\left(3u^5+3u^4-6u^3-6u^2+3u+3\right) \\
=& 3w_2^5\left(u-1\right)^2\left(u+1\right)^3
\end{aligned}
$$

hal-00788641, version 1 - 14 Feb 2013

$$
\begin{aligned}
&4\left(\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}\right)^2\left(w_1s_1^2+w_2s_2^2-(w_1+w_2)\left(\frac{1}{2}\frac{(w_1+w_2)^2}{\frac{w_1^2}{s_1}+\frac{w_2^2}{s_2}}\right)^2\right)\\
&\geq w_2^5\left(3\left(u^5+1\right)+u^3\left(8u-10\right)+u^2\left(8\frac{1}{u}-10\right)\right.\\
&\quad\left.+u\left(4u^2-5\right)+u^4\left(4\frac{1}{u^2}-5\right)\right)\\
&= w_2^5\left(3u^5+3u^4-6u^3-6u^2+3u+3\right)\\
&= 3w_2^5\left(u-1\right)^2\left(u+1\right)^3
\end{aligned}
$$

Since $w_2 > w_1$, $0 < u < 1$, and this polynomial is strictly positive, hence we have $w_1s_1^2 + w_2s_2^2 - 2ws_B^2 > 0$.

Finally, we can conclude that in both cases, $\mathbb{E}(E)((w_1, s_1), (w_2, s_2), \sigma) - \mathbb{E}(E)((w, s_B), (w, s_B), \sigma) > 0$, so there exist a better solution with two chunks of same sizes, hence leading to a contradiction.

We had proven that all chunks have the same size. We use the same line of reasoning to prove that all chunks are executed at a same speed s. If there are two chunks executed at speeds $s_1 < s_2$ (with $w_1 = w_2 = w$), then we have $s_A = s_B$. Considering that $s = s_A$, it is easy to see that $w_1s_1^2+w_2s_2^2-2ws_A^2 > 0$ since $w_1w_2^2g\left(\frac{s_1}{s_2}\right) + w_1^2w_2g\left(\frac{s_2}{s_1}\right) > 0$. Indeed, g is null only in 1, and $s_1 \neq s_2$. We exhibit a solution strictly better, hence showing a contradiction. This concludes the proof. □

Thanks to this result, we know that the n chunks problem can be rewritten as follows: find s such that

- $\frac{W}{s} + nT_C + \frac{\lambda}{n}\left(\frac{W}{s} + nT_C\right)\left(\frac{W}{\sigma} + nT_C\right) = D$
- in order to minimize $Ws^2 + nE_C + \frac{\lambda}{n}\left(\frac{W}{s} + nT_C\right)\left(W\sigma^2 + nE_C\right)$

One can see that this reduces to the SingleChunk MultipleSpeeds Expected-Deadline task problem where

- $\lambda \leftarrow \frac{\lambda}{n}$
- $T_C \leftarrow nT_C$
- $E_C \leftarrow nE_C$

and allows us to write the problem to solve as a two parameters function:

$$
(n,s) \mapsto Ws^2 + nE_C + \frac{\lambda}{n}\left(\frac{W}{s} + nT_C\right)\left(W\left(\frac{\frac{\lambda}{n}W}{\frac{D}{\frac{W}{s}+nT_C} - (1+\lambda T_C)}\right)^2 + nE_C\right) \tag{15}
$$

which can be minimized numerically.

hal-00788641, version 1 - 14 Feb 2013

hal-00788641, version 1 - 14 Feb 2013

6.2.2 Hard deadline

In this section, the constraint on the execution time can be written as:

$$\sum_i \left(\frac{w_i}{s_i} + T_C + \frac{w_i}{\sigma_i} + T_C \right) \leq D.$$

Lemma 9. *In the* MultipleSpeeds Hard-Deadline *model with divisible chunk, the deadline should be tight.*

Proof. This result is obvious with Lemma 4: if we have a solution such that the deadline is not tight, if we fix every variable but σ_1 (the re-execution speed of the first task), we can improve the solution with a tight deadline. □

Lemma 10. *In the optimal solution, for all* i, j, $\lambda \left(\frac{w_i}{s_i} + T_C \right) \sigma_i^3 = \lambda \left(\frac{w_j}{s_j} + T_C \right) \sigma_j^3$.

Proof. Consider any solution to our problem. Thanks to Lemma 9, we know that the deadline should be tight. Let T_i and T_j two tasks of er-execution speed σ_i, σ_j. We show that those speed can be optimally defined such that $\lambda \left(\frac{w_i}{s_i} + T_C \right) \sigma_i^3 = \lambda \left(\frac{w_j}{s_j} + T_C \right) \sigma_j^3$. Let us call $u_i = \lambda \left(\frac{w_i}{s_i} + T_C \right)$ and $u_j = \lambda \left(\frac{w_j}{s_j} + T_C \right)$.

The minimization problem for those speeds can be written as $A_0 + u_i w_i \sigma_i^2 + u_j w_j \sigma_j^2$ under the constraint that $A_1 + \frac{w_i}{\sigma_i} + \frac{w_j}{\sigma_j} = D$ where neither A_0 nor A_1 depends on σ_i, σ_j.

Replacing $\sigma_i = \frac{w_i}{D - A_1 - \frac{w_j}{\sigma_j}}$ in the function we need to minimize, we obtain $A_0 + u_i w_i \left(\frac{w_i}{D - A_1 - \frac{w_j}{\sigma_j}} \right)^2 + u_j w_j \sigma_j^2$. A simple differentiation gives $-2 w_j u_i \frac{w_i^3}{\left(D - A_1 - \frac{w_j}{\sigma_j} \right)^3 \sigma_j^2} + 2 u_j w_j \sigma_j$. Another differentiation shows the convexity of the function we want to minimize. Hence one can see that the function is minimized when $u_j \sigma_j^3 = u_i \left(\frac{w_i}{D - A_1 - \frac{w_j}{\sigma_j}} \right)^3 = u_i \sigma_i^3$. □

Lemma 11. *If we enforce the condition that the execution speeds of the chunks are all equal, and that the re-execution speeds of the chunks are all equal, then all chunks should have same size in the optimal solution.*

Proof. This result is obvious since the problem can be reformulated as the minimization of $\alpha \sum w_i + \beta \sum w_i^2$ where neither α nor β depends on any w_i, under the constraints $\gamma \sum w_i + \zeta \leq D$, and $\sum w_i = W$. It is easy to see the result when there are only two chunks since there is only one variable, and the problem generalizes well in the case of n chunks. □

We have not been able to prove a stronger result than Lemma 11. However we conjecture the following result:

Conjecture 1. In the MultipleSpeeds Hard-Deadline, in the optimal solution, the re-execution speeds are identical, the deadline is tight. The re-execution speed is equal to $\sigma = \frac{W}{(D - 2nT_C)s - W} s$. Furthermore the chunks should have the same size $\frac{W}{n}$ and should be executed at the same speed s.

hal-00788641, version 1 - 14 Feb 2013

This conjecture reduces the problem to the SingleChunk MultipleSpeeds problem where

- $\lambda \leftarrow \frac{\lambda}{n}$
- $T_C \leftarrow nT_C$
- $E_C \leftarrow nE_C$

and allows us to write the problem to solve as a two-parameter function:

$$(n, s) \mapsto Ws^2 + nE_C + \frac{\lambda}{n}\left(\frac{W}{s} + nT_C\right)\left(W\left(\frac{W}{(D-2nT_C)s - W}s\right)^2 + nE_C\right) \tag{16}$$

which can be solved numerically.

7 Simulations

7.1 Simulation settings

We performed a large set of simulations in order to illustrate the differences between all the models studied in this paper, and to show upon to which extent each additional degree of freedom improves the results, i.e., allowing for multiple speeds instead of a single speed, or for multiple smaller chunks instead of a single large chunk. All these experiments are conducted under both constraint types, expected and hard deadlines.

We envision reasonable settings by varying parameters within the following ranges:

- $\frac{W}{D} \in [0.2, 10]$
- $\frac{T_C}{D} \in [10^{-4}, 10^{-2}]$
- $E_C \in [10^{-3}, 10^3]$
- $\lambda \in [10^{-8}, 1]$.

In addition, we set the deadline to 1. Note that since we study $\frac{W}{D}$ and $\frac{T_C}{D}$ instead of W and T_C, we do not need to study how the variation of the deadline impacts the simulation, this is already taken into account.

We use the Maple software to solve numerically the different minimization problems. Results are showed from two perspectives: on the one hand (Figures 1 and 2), for a given constraint (Hard-Deadline or Expected-Deadline), we normalize all variants according to SingleSpeed SingleChunk, under the considered constraint. For instance, on the plots, the energy consumed by MultipleChunks MultipleSpeeds (denoted as MCMS) for Hard-Deadline is divided by the energy consumed by SingleChunk SingleSpeed (denoted as SCSS) for Hard-Deadline, while the energy of MultipleChunks SingleSpeed (denoted as MCSS) for Expected-Deadline is normalized by the energy of SingleChunk SingleSpeed for Expected-Deadline.

On the other hand (Figures 3 and 4), we study the impact of the constraint hardness on the energy consumption. For each solution form (SingleSpeed or

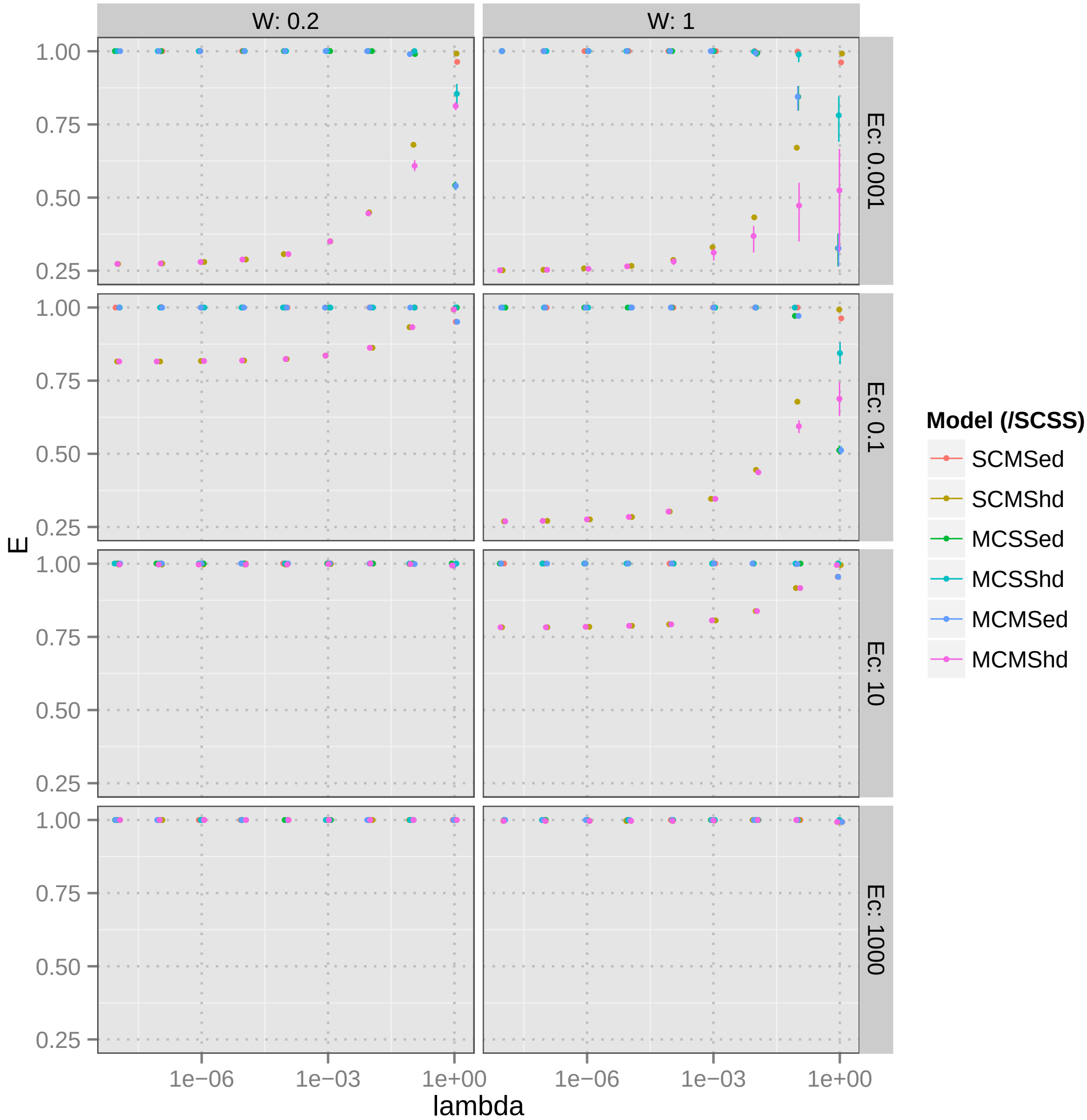

Figure 1: Comparison with SingleChunk SingleSpeed.

hal-00788641, version 1 - 14 Feb 2013

hal-00788641, version 1 - 14 Feb 2013

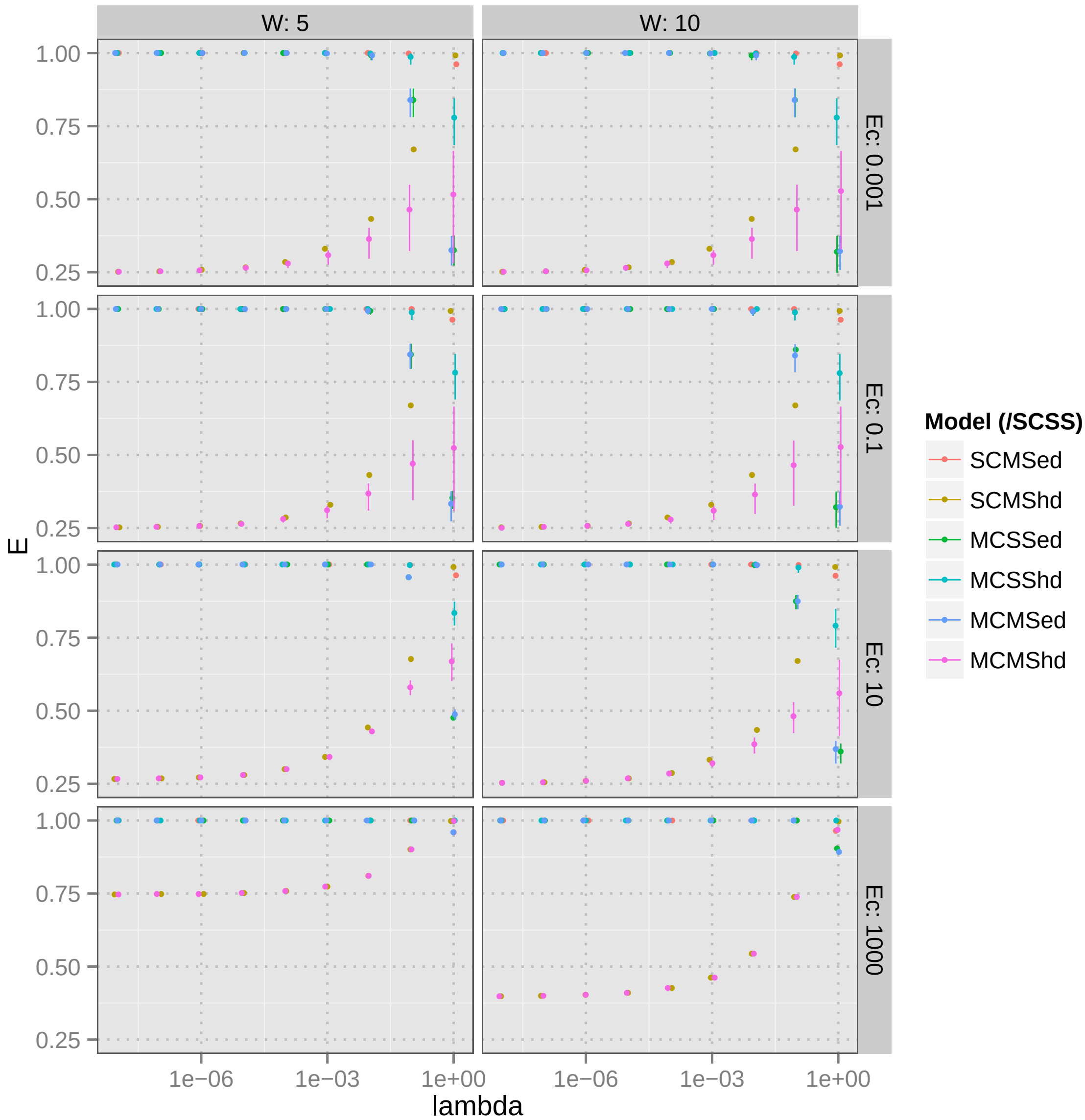

Figure 2: Comparison with SingleChunk SingleSpeed.

hal-00788641, version 1 - 14 Feb 2013

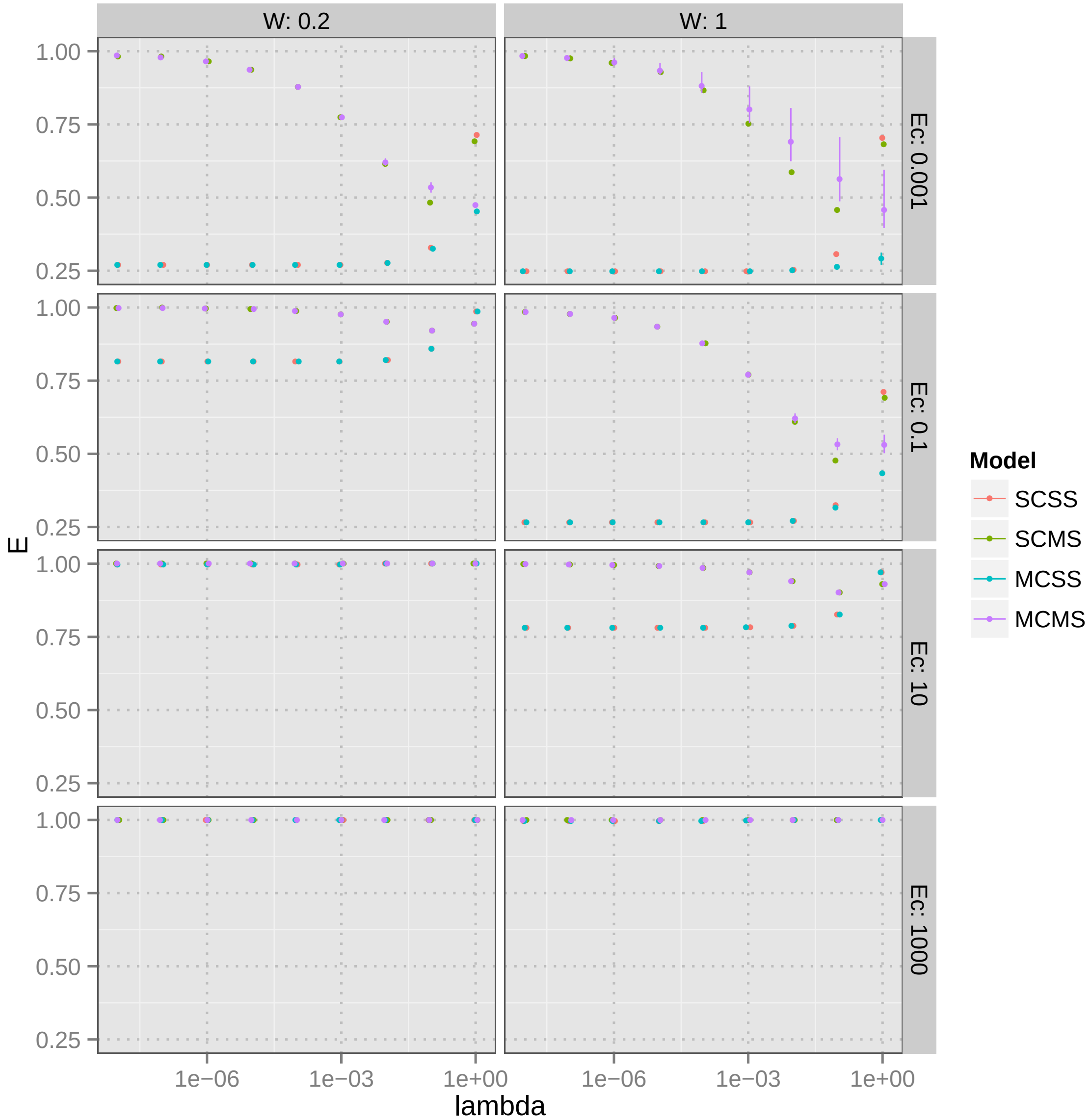

Figure 3: Comparison Hard-Deadline versus Expected-Deadline.

hal-00788641, version 1 - 14 Feb 2013

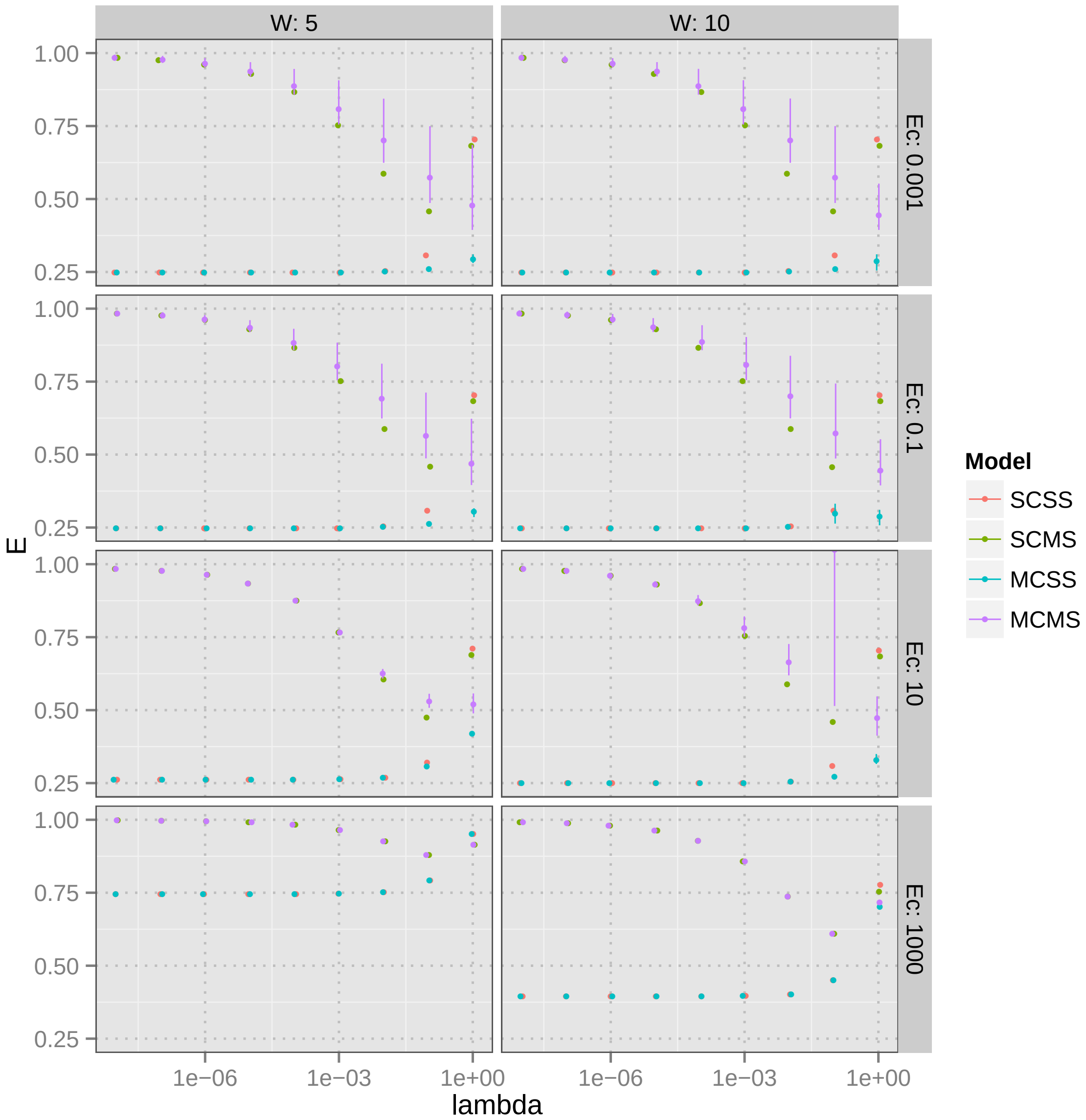

Figure 4: Comparison HARD-DEADLINE versus EXPECTED-DEADLINE.

MultipleSpeeds, and SingleChunk or MultipleChunks), we plot the ratio energy consumed for Expected-Deadline over energy consumed for Hard-Deadline.

Note that for each figure, we plot for each function different values that depend on the different values of T_C/D (hence the vertical intervals for points where T_C/D has an impact). In addition, the lower the value of T_C/D, the lower the energy consumption.

7.2 Comparison with single speed

At first, we observe that the results are identical for any value of W/D, up to a translation of E_C (see ($W/D = 0.2, E_C = 10^{-3}$) vs. ($W/D = 5, E_C = 1000$) on Figures 1 and 2, or see ($W/D = 1, E_C = 10^{-3}$) vs. ($W/D = 5, E_C = 0.1$) on Figures 1 and 2, for instance).

Then the next observation is that for Expected-Deadline, with a small λ ($< 10^{-2}$), MultipleChunks or MultipleSpeeds models do not improve the energy ratio. This is due to the fact that, in both expressions for energy and for execution time, the re-execution term is negligible relative to the execution one, since it has a weighting factor λ. However, when λ increases, if the energy of a checkpoint is small in front of the total work (which is the general case), we can see a huge improvement (between 25% and 75% energy saving) with MultipleChunks.

On the contrary, as expected, for small λ's, re-executing at a different speed has a huge impact for Hard-Deadline, where we can gain up to 75% energy when the failure rate is low. We can indeed run at around half speed during the first execution (leading to the $1/2^2 = 25\%$ saving), and at a high speed for the second one, because the very low failure probability avoids the explosion of expected energy consumption. For both MultipleChunks and SingleChunk, this saving ratio increases with λ (the energy consumed by the second execution cannot be neglected any more, and both executions need to be more balanced), the latter being more sensitive to λ. But the former is the only configuration where T_C has a significant impact: its performance decreases with T_C; still it remains strictly better than SingleChunk MultipleSpeeds.

7.3 Comparison between Expected-Deadline and Hard-Deadline

As before, the value of W/D does not change the energy ratios up to translations of E_C. As expected, the difference between the Expected-Deadline and Hard-Deadline models is very important for the SingleSpeed variant: when the energy of the re-execution is negligible (because of the failure rate parameter), it would be better to spend as little time as possible doing the re-execution in order to have a speed as slow as possible for the first execution, however we are limited in the SingleSpeed Hard-Deadline model by the fact that the re-execution time is fully taken into account (its speed is the same as the first execution, and there is no parameter λ to render it negligible).

Furthermore, when λ is minimum, MultipleSpeeds consumes the same energy for Expected-Deadline and for Hard-Deadline. Indeed, as expected, the λ in the energy function makes it possible for the re-execution speed to be maximal: it has little impact on the energy, and it is optimal for the execution

hal-00788641, version 1 - 14 Feb 2013

hal-00788641, version 1 - 14 Feb 2013

time; this way we can focus on slowing down the first execution of each chunk. For HARD-DEADLINE, we already run the first execution at half speed, thus we cannot save more energy, even considering EXPECTED-DEADLINE instead. When λ increases, speeds of HARD-DEADLINE cannot be lowered but the expected execution time decreases, making room for a downgrade of the speeds in the EXPECTED-DEADLINE problems.

8 Conclusion

In this work, we have studied the energy consumption of a divisible computational workload on volatile platforms. In particular, we have studied the expected energy consumption under different deadline constraints: a soft deadline (a deadline for the expected execution time), and a hard deadline (a deadline for the worst case execution time).

We have been able to show mathematically, for all cases but one, that when using the MULTIPLECHUNKS model, then (i) every chunk should be equally sized; (ii) every execution speed should be equal; and (iii) every re-execution speed should also be equal. This problem remains open in the MULTIPLESPEEDS HARD-DEADLINE variant.

Through a set of extensive simulations, we were able to show the following: (i) when the fault parameter λ is small, for EXPECTED-DEADLINE constraints, the SINGLECHUNK SINGLESPEED model leads to almost optimal energy consumption. This is not true for the HARD-DEADLINE model, which accounts equally for execution and re-execution, thereby leading to higher energy consumption. Therefore, for the HARD-DEADLINE model and for small λ, the model of choice should be the SINGLECHUNK MULTIPLESPEEDS model. When the fault parameter rate λ increases, using a single chunk is no longer energy-efficient, and one should focus on the MULTIPLECHUNKS MULTIPLESPEEDS model for both deadline types.

An interesting direction for future work is to extend this study to the case of an application workflow: instead of dealing with a single divisible task, we would deal with a DAG of tasks, that could be either divisible (checkpoints can take place anytime) or atomic (checkpoints can only take place at the end of the execution of some tasks). Again, we can envision both soft or hard constraints on the execution time, and we can keep the same model with a single re-execution per chunk/task, at the same speed or possibly at a different speed. Deriving complexity results and heuristics to solve this difficult problem is likely to be very challenging, but could have a dramatic impact to reduce the energy consumption of many scientific applications.

hal-00788641, version 1 - 14 Feb 2013

References

[1] I. Assayad, A. Girault, and H. Kalla. Tradeoff exploration between reliability, power consumption, and execution time. In *Proc. of SAFECOMP, the 30th int. conf. on computer safety, reliability, and security*, pages 437–451, 2011.

[2] G. Aupy and A. Benoit. Approximation algorithms for energy, reliability and makespan optimization problems. Research report RR-8107, INRIA, France, Oct. 2012.

[3] G. Aupy, A. Benoit, and Y. Robert. Energy-aware scheduling under reliability and makespan constraints. In *Proc. of HiPC, the 19th int. conf. on High Performance Computing*, 2012. Also available as INRIA Research Report RR-7757.

[4] H. Aydin and Q. Yang. Energy-aware partitioning for multiprocessor real-time systems. In *Proc. of Int. Parallel and Distributed Processing Symposium (IPDPS)*, pages 113–121, 2003.

[5] V. Bharadwaj, T. G. Robertazzi, and D. Ghose. *Scheduling Divisible Loads in Parallel and Distributed Systems.* IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.

[6] G. Buttazzo, G. Lipari, L. Abeni, and M. Caccamo. *Soft Real-Time Systems: Predictability vs. Efficiency.* Springer series in Computer Science, 2005.

[7] K. M. Chandy and L. Lamport. Distributed snapshots: determining global states of distributed systems. *ACM Trans. Comput. Syst.*, 3(1):63–75, Feb. 1985.

[8] V. Degalahal, L. Li, V. Narayanan, M. Kandemir, and M. J. Irwin. Soft errors issues in low-power caches. *IEEE Trans. Very Large Scale Integr. Syst.*, 13:1157–1166, October 2005.

[9] J. Dongarra, P. Beckman, P. Aerts, F. Cappello, T. Lippert, S. Matsuoka, P. Messina, T. Moore, R. Stevens, A. Trefethen, and M. Valero. The international exascale software project: a call to cooperative action by the global high-performance community. *Int. J. High Perform. Comput. Appl.*, 23(4):309–322, 2009.

[10] M. Drozdowski. Divisible load. In *Scheduling for Parallel Processing*, Computer Communications and Networks, pages 301–365. Springer, 2009.

[11] R. Geist, R. Reynolds, and J. Westall. Selection of a checkpoint interval in a critical-task environment. *IEEE Transactions on Reliability*, 37(4):395–400, Oct. 1988.

[12] R. Melhem, D. Mossé, and E. Elnozahy. The interplay of power management and fault recovery in real-time systems. *IEEE Trans. on Computers*, 53(2):217–231, 2004.

[13] P. Pop, K. H. Poulsen, V. Izosimov, and P. Eles. Scheduling and voltage scaling for energy/reliability trade-offs in fault-tolerant time-triggered embedded systems. In *Proc. of IEEE/ACM Int. Conf. on Hardware/software codesign and system synthesis (CODES+ISSS)*, pages 233–238, 2007.

[14] J. A. Stankovic, K. Ramamritham, and M. Spuri. *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[15] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for reduced CPU energy. *Operating Systems Design and Implementation*, pages 13–23, 1994.

[16] Y. Zhang and K. Chakrabarty. Energy-aware adaptive checkpointing in embedded real-time systems. In *Proc. of the Design, Automation and Test in Europe Conf. (DATE)*, pages 918–923. IEEE CS Press, 2003.

[17] D. Zhu and H. Aydin. Energy management for real-time embedded systems with reliability requirements. In *Proc. of IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, pages 528–534, 2006.

[18] D. Zhu, R. Melhem, and D. Mossé. The effects of energy management on reliability in real-time embedded systems. In *Proc. of IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, pages 35–40, 2004.

hal-00788641, version 1 - 14 Feb 2013

hal-00788641, version 1 - 14 Feb 2013

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399