



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A new flexible Checkpoint/Restart model

Bouguerra Mohamed Slim — Thierry Gautier — Denis Trystram — Jean-Marc Vincent

N° 6751

December 2008

Thème NUM



*R*apport
de recherche

A new flexible Checkpoint/Restart model

Bouguerra Mohamed Slim , Thierry Gautier , Denis Trystram , Jean-Marc
Vincent

Thème NUM — Systèmes numériques
Équipes-Projets MOAIS

Rapport de recherche n° 6751 — December 2008 — 25 pages

Abstract: The utilization of new generation computing platforms like computational grids or desktop grids introduces new challenging problems. In particular, due to the huge number of the involved processors, security and fault-tolerance aspects are key issues that must be taken into account. Coordinated checkpointing is one of the most popular technique to deal with failures in such platforms. The approach of application-directed checkpointing in fault-tolerance puts an incredible strain on the storage system and the communications. This results in large overheads on the execution times of applications that severely impact the performance and the scalability. This work presents a new model of coordinated checkpoint/restart mechanism for several types of computing platforms. Its main feature is that it is independent from the failure law which makes it very flexible. We will show that such a model may be used to determine the optimal periodic checkpoint interval and to reduce the checkpoint overhead through mathematical analysis of reliability. Moreover, unlike most of the existing checkpointing models, the proposed model is able to take into account a variable checkpoint cost. Finally, we report some experiments based on simulations for random failure distributions corresponding to the two most popular laws, namely, the Poisson's process and Weibull's law.

Key-words: Checkpointing , Fault tolerance, Parallel processing

Une nouvelle modélisation flexible du mécanisme de sauvegarde/reprise

Résumé : L'utilisation des nouvelles plates-formes de calcul parallèle comme les grilles de calcul ou les *desktop grids* introduisent de nouvelles problématiques. En particulier, à cause du nombre élevé des noeuds de calcul, la sécurité et la tolérance aux pannes sont des aspects très importants et ils doivent être pris en considération. Le mécanisme de sauvegarde et reprise coordonnée est un des protocoles le plus populaire dans ces nouvelles plates-formes. Néanmoins, ce genre de mécanisme crée une congestion importante aux cours de l'exécution. Cela touche directement les performances et la scalabilité des applications avec ce type de mécanisme. Ce travail présente une nouvelle modélisation flexible du mécanisme de sauvegarde et reprise dans divers environnements de calcul. En effet, ce modèle permet de déterminer les intervalles optimaux entre chaque sauvegarde dont le but est de réduire le surcoût engendré par le mécanisme de sauvegarde. Nous notons que ce modèle est indépendant de la loi de panne ce qui le rend flexible. En plus ce modèle prend en compte un coût variable de sauvegarde, ce qui représente une originalité de ce modèle par rapport aux modèles existants. Finalement nous proposons des expérimentations qui valident le modèle avec deux types de loi de panne (Processus de Poisson et loi de Weibull).

Mots-clés : Sauvegarde, Tolérance aux pannes, Calcul parallèle

1 Introduction

Today the most powerful computing systems involve more and more processors. Reliability is a crucial issue to address while running applications on such large systems because the failure rate grows with the system size. The users expect several processors to fail during the execution of their applications. Thus, it is necessary to develop strategies for providing a reliable completion of large applications. From the IBM source, the BlueGene/L system with 65,536 nodes is expected to have a mean time between failure less than 24 hours [1]. Fault-tolerant systems have been extensively studied on various computing or real-time systems. Many approaches have been proposed for dealing with fault-tolerance. We focus in this work on the coordinated checkpoint/restart which is one of the most popular mechanism used in practice [5][6][9][11][14]. In checkpointing-based approaches, the state of computations is saved periodically in a reliable storage [2]. Informally, checkpointing is a technique that enables to reduce the completion time of the application by saving intermediate states in a stable storage, and then, to restore from the last stored state when a failure occurs. In the case of distributed computations, checkpointing methods differ from each other in the way the processes are coordinated in order to capture the global state in a consistent manner or not. Coordinated checkpointing requires that all processes coordinate the construction of a consistent global state before they write the individual checkpoints in the stable storage. After one or many failures, the restart mechanism sets up each processor from the last checkpoint and then, it schedules again the tasks of the crashed processors on new ones.

In this paper, we focus on the minimization of both the completion time of an application and the checkpoint/restart overhead by determining the optimal interval between two consecutive checkpoints. The main contribution of this work is to derive a new probabilistic model of the execution time of parallel applications with stochastic processes. This explicit analytical cost model enables to address and solve several important problems. First, like most of other models, it can be used to determine the optimal interval length between two successive checkpoints that minimizes the congestion on the network in the case of periodic checkpoints. Moreover, it is possible to take into account a variable checkpoint cost that depends on residual workload. Finally, the proposed model can be adapted to several types of computations since it is independent from the failure law. We show how to use it for the two most popular laws (Exponential and Weibull distribution). This makes it very flexible.

The rest of this paper is organized as follows: Section 2 briefly recalls and discusses the other existing models for coordinated checkpointing on parallel platforms. Section 3 describes the proposed model which leads to a formula that expresses the expected completion time of the execution. In Section 4, we detail how to apply this model to two case-studies for the distribution of failure occurrences with Poisson's process and Weibull's law. Before concluding, we report in Section 5 some experiments based on simulations for assessing the model in several scenarii.

2 Review of Related Works

Let us first recall the principle of the coordinated checkpoint/restart protocol proposed by Chandy et al. [2]. It served as a basis of many implementations of fault tolerant systems for high performance computing. Coordinated checkpointing requires that all processes periodically coordinate the construction of a consistent global state before writing the state of individual checkpoints in a stable storage. Based on simulation results, Elnozahy et al. [6] showed that this approach is the most effective fault tolerance mechanism in large-scale parallel platforms. The aims of fault tolerance models is to find strategies that minimize the completion time of the execution. Young [14] proposed a checkpoint/restart model where the failures follow an exponential law. Moreover, checkpointing is assumed to be fault-free and to have a constant execution time. The basic periodic version of this model has been recently extended by Oliner et al. [11]. Daly proposed also in [5] an extension of Young's model where failures can occur during checkpointing and he derived a higher order of approximation. Both Young's and Daly's models are able to compute an optimal checkpoint interval that minimizes the completion time considering constant checkpoint times. Other works from Geist et al. [8], Plank et al. [12] studied stochastic models and determined an optimal checkpoint date that minimizes a cost function which corresponds to maximize the availability of the system. Yudan et al. [9] proposed a stochastic model and an optimal checkpoint interval which does not depend on the specific failure law, such as the Poisson's process used in [8, 12]. Moreover, in this model, it is possible to compute the optimal checkpoint interval that minimizes the expected wasting time in the protocol itself (checkpoint overhead, restart time and rollback time). Another variant of this model is proposed in [10], that modelize the incremental checkpoint under Poisson's process. We present in this paper a new model that expresses the expected time to completion with respect to the distribution law of failures taking into account variable checkpoint overheads. The two most important differences between this new model and the previous works is that it does not depend on a particular failure law and it takes into account the variable size of checkpoint as an input of the problem.

3 Probabilistic Model

In this section a detailed description of the proposed probabilistic execution model is presented. Firstly, we present a short description of the application model with no failures, then we add a failure process which allows to compute the expected completion time of the execution with a formal method. Finally, using this abstraction that models a classical execution without checkpointing, we derive a global model that contains the checkpoint/restart mechanism. Moreover, we establish a global formula that expresses the expected completion time of the execution.

3.1 Application Model

The first input parameter in the considered problem is the application model. In this work a parallel application is modeled by a residual computation work function denoted by ω_t , where t is time index and ω_t represents the amount of work remaining at t . The typical shape of this cost function is represented in Figure 1. The slope of the curve indicates the parallel degree of the application. In the general case, the curve can be divided into two parts. The first part before the moment τ is characterized by a linear speed-up with a slope which depends on the number of processors m and overhead factor α (due to the parallelism). In this case, all the processors are busy. The second part presents a more curved shape. This means that there is not enough work for all the processors, and thus, some idle time appear. These two parts will be determined by the amount of parallelism in the target application.

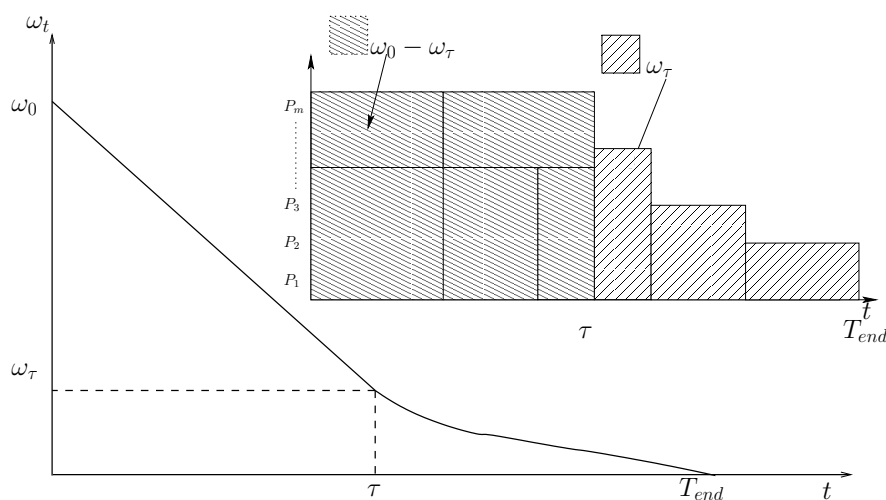


Figure 1: General execution model

In this work we assume, that having a linear speed-up until the end of the application, where $\frac{\omega_0}{\alpha m} = \tau = T_{end}$. This hypothesis will be relaxed in future works, but is not really a limitation for large applications. Under this hypothesis the ω_t function can be expressed as follows:

Definition 1. Let ω_0 be the total amount of work to be executed, α the overhead factor $0 < \alpha \leq 1$ and m the number of available processors. The residual workload at time t is:

$$\omega_t = \omega_0 - \alpha m t$$

3.2 Probabilistic Execution Model

After having described the application model, the next step is to add to this abstraction the failures process and the checkpoint/restart mechanism. The global execution model

is drawn in Figure 2. From this figure we deduce the following renewal process. At the first step, there is a start-up or a restart phase in where the time is elapsed but the residual workload does not decrease. It is denoted by R^j which is the restart cost in the j^{th} checkpoint interval. Thus, the cost of this restart depend on the cost of the preview checkpoint. Then, the computation is started and the residual workload is decreased with a speed-up depending on the number of computing nodes. After an interval of work denoted by I_j a checkpoint is triggered. Finally, when a failure comes into one of these preview phases, all the computed work after the last checkpoint is lost. Thus, the application is restarted from the last checkpoint and the process is restarted from the beginning. Using this abstraction will allow to establish some important theorems under the following hypothesis.

Statistical hypothesis on the failures process Mainly we consider permanent failures that affect the hardware or software system. In this model, if a processor crashes, all of its tasks will be reassigned to another new processor. We also assume that we have available processors to replace the lost ones so that the number of processors remains m . Also, we assume reliability of the failures detection tool and the time of failures detection is negligible and that the failures do not propagate. Thus, the time between failures on different processors in the system are independent and identically distributed. Finally, the phases of checkpoint and recovery may also suffer from failures. From the above assumptions, we deduce that $\{X_i^j\}$ process (which denote the i^{th} inter-arrival of failures in the j^{th} interval of checkpoint) is a renewal process, thus $\{X_i^j\}$ are positive independent and identically distributed random variables with distribution function $F(t)$ and probability density function $f(t)$.

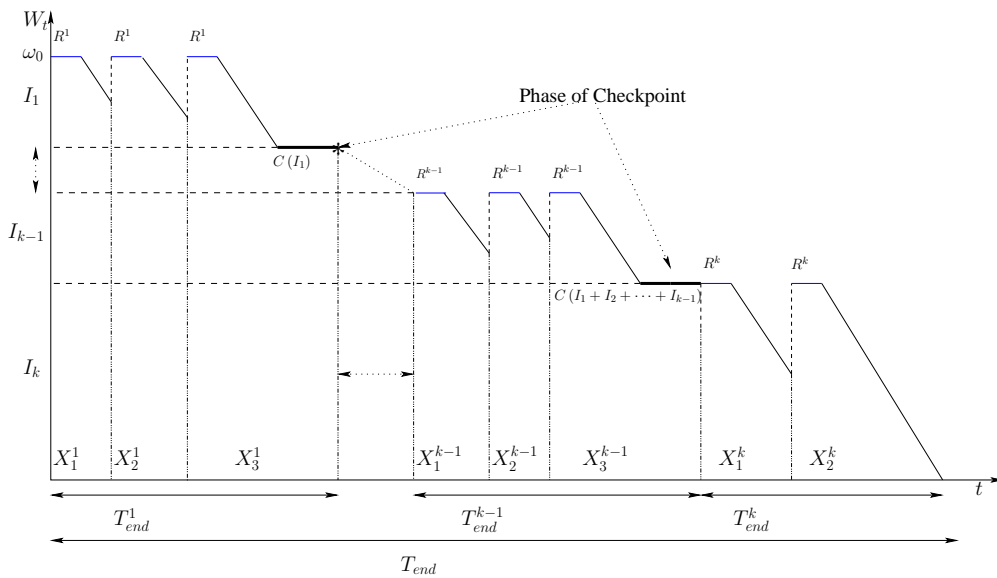


Figure 2: Model of execution with the checkpoint/restart mechanism

Hypothesis on the Checkpoint/Restart process The checkpoint process is considered as a deterministic process and the cost of a checkpoint depends only on the amount of the work already done. It is denoted by $C(S_j)$ where $S_j = \sum_{i=1}^j I_i$ the amount of work already done. Thus, the restart process is considered as a deterministic process and its cost depends only on the cost of the last checkpoint. It is denoted by $R(S_{j-1})$ where $S_{j-1} = \sum_{i=1}^{j-1} I_i$ is the work already done before the last checkpoint.

State variables

- $\{Z_i\}_{n \in \mathbb{N}}$: The failure arrival process.
- $\{X_i\}_{n \in \mathbb{N}}$: The inter-arrival time of failure process (which does not depend on the j index).
- $\{I_j\}_{j \in \mathbb{N}}$: Each I_j represent the amount of work to be done before each checkpoint.
- T_{end} : The global completion time in presence of failures.
- $\{T_{end}^j\}_{j \in \mathbb{N}}$: Each T_{end}^j represents the sub-completion time of a sub-quantity of workload.
- $k \in \mathbb{N}$: The number of checkpoint barriers.

3.2.1 Model of Execution without Checkpoint

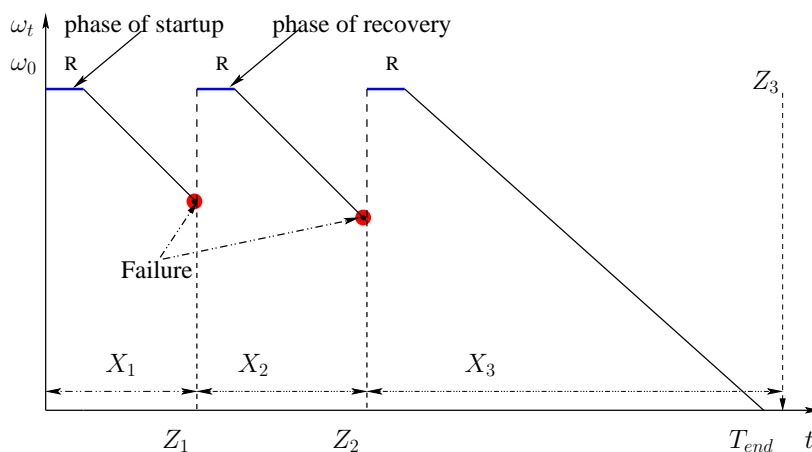


Figure 3: Execution Model without checkpoint mechanism.

To establish the first fundamental theorem, we start by studying the execution without checkpoint mechanism. In such an execution model, the execution is restarted from the beginning after a failure (see Figure 3). We recall that in this case the recovery process is a deterministic process with duration $R^j = R$. From this figure, we deduce a formula that expresses the expected completion time T_{end} . Consider a number of failures N during the execution such as:

$$\{N = n\} = \{X_1 < \frac{\omega_0}{\alpha m} + R, X_2 < \frac{\omega_0}{\alpha m} + R \dots X_n \geq \frac{\omega_0}{\alpha m} + R\}.$$

Consequently N is a stopping time process associated to the process $\{X_i\}$. Based on the Wald's equation [13] we can establish the following theorem.

Theorem 1. Let T_{end} be the completion time of the execution, ω_0 the quantity of work at the beginning, m the number of processors, α the overhead factor ($0 < \alpha \leq 1$) and p the probability of failure occurrence during the execution with $p = \mathbb{P}[X \leq \frac{\omega_0}{\alpha m} + R] = F(\frac{\omega_0}{\alpha m} + R)$ then:

$$\mathbb{E}(T_{end}) = \frac{1}{1-p} \mathbb{E}(X_1) + \frac{\omega_0}{\alpha m} + R - \mathbb{E}(X_N).$$

Proof. From Figure 3, We have:

$T_{end} = \sum_{i=1}^N X_i + \frac{\omega_0}{\alpha m} + R - X_N$, where X_N is the first interval between failures which is greater than $(\frac{\omega_0}{\alpha m} + R)$.

Taking the expectation of both sides gives:

$$\mathbb{E}(T_{end}) = \mathbb{E}\left(\sum_{i=1}^N X_i\right) + \frac{\omega_0}{\alpha m} + R - \mathbb{E}(X_N).$$

Using Wald's equation [13] we obtain:

$$\mathbb{E}(T_{end}) = \mathbb{E}(N)\mathbb{E}(X_1) + \frac{\omega_0}{\alpha m} + R - \mathbb{E}(X_N).$$

Since the $\{X_i\}$ process is independent and identically distributed, hence the stopping time N is geometrically distributed with parameter $p = \mathbb{P}[X \leq \frac{\omega_0}{\alpha m} + R]$, thus $\mathbb{E}(N) = \frac{1}{1-p}$.

Consequently:

$$\mathbb{E}(T_{end}) = \frac{1}{1-p} \mathbb{E}(X_1) + \frac{\omega_0}{\alpha m} + R - \mathbb{E}(X_N).$$

Moreover given the density $f(x)$ of inter-arrival time of failures, the quantity $\mathbb{E}(X_N)$ is computed by the following expression:

$$\mathbb{E}(X_N) = \mathbb{E}(X | X > \frac{\omega_0}{\alpha m} + R) = \frac{1}{\mathbb{P}[X > \frac{\omega_0}{\alpha m} + R]} \int_{\frac{\omega_0}{\alpha m} + R}^{+\infty} x f(x) dx. \quad (1)$$

□

3.2.2 Model of Execution with a Checkpoint Mechanism

To add the checkpoint mechanism in the previous model, we divide the initial workload $\frac{\omega_0}{\alpha m}$ in k intervals denoted by $I_1, I_2 \dots I_k$ (see Figure 2). Then, we use the above model for each I_j , as I_j is the amount of workload that should be done between the checkpoint number $j-1$ and j . Because the computation is restarted again after a checkpoint, we suppose that the failures process is also restarted after each checkpoint barrier. From this hypothesis we establish the following theorem:

Theorem 2. Let k be the number of checkpoints, $I_1, I_2, I_3 \dots I_k$ be the amount of work between each checkpoint, such as $\sum_{j=1}^k I_j = \frac{\omega_0}{\alpha m}$, $C(S_j)$ the cost of the checkpoint after the

amount of work S_j and $R(S_{j-1})$ the restart cost before the j^{th} checkpoint then:

$$\mathbb{E}(T_{end}) = \mathbb{E}(X_1) \sum_{j=1}^k \frac{1}{1-p_j} + \sum_{j=1}^k \eta_j - \mathbb{E}(X_N^j),$$

$$\text{where } \eta_j = I_j + C(S_j) + R(S_{j-1}), p_j = F(\eta_j).$$

Proof. As it can be seen in Figure 2, T_{end} can be written as $T_{end} = T_{end}^1 + T_{end}^2 + \dots + T_{end}^k$. Taking the expectation of both sides gives:

$$\mathbb{E}(T_{end}) = \mathbb{E}(T_{end}^1) + \mathbb{E}(T_{end}^2) + \dots + \mathbb{E}(T_{end}^k).$$

Thus, using Theorem 1 with η_j as initial amount of work, $\mathbb{E}(T_{end}^j)$ is given as follows:

$$\mathbb{E}(T_{end}^j) = \frac{1}{1-p_j} \mathbb{E}(X_1^j) + \eta_j - \mathbb{E}(X_N^j).$$

Then,

$$\mathbb{E}(T_{end}) = \mathbb{E}(X_1) \sum_{j=1}^k \frac{1}{1-p_j} + \sum_{j=1}^k \eta_j - \mathbb{E}(X_N^j) \quad (2)$$

□

Thus, Theorem 2 expresses the expected value of the completion time. We notice that this formula does not depend on the type of failures law, which is very important. Then, in order to find the optimal amount of work between each checkpoint we have to minimize the equation in Theorem 2.

4 Case Studies

In this section, we propose two exhaustive case studies. In the first one, we express the optimal checkpoint interval when the failures process is modeled by a Poisson's process. Then, in the second one we also express the optimal solution when the failures are modeled by a Weibull's law.

4.1 Poisson's Process Failures

One of the most common method to modelize the failures in electronic device, is the Poisson's process with a constant rate (denoted by λ).

Proposition 1. *If the distribution of failures time follows a Poisson's process with rate λ then:*

$$\mathbb{E}(T_{end}) = \frac{1}{\lambda} \sum_{j=1}^k [e^{\lambda \eta_j} - 1] \text{ where } \eta_j = I_j + C(S_j) + R(S_{j-1}).$$

Proof. First due to the Poisson's process distribution, we have:

- $\mathbb{E}(X_1) = \frac{1}{\lambda}$,
- $p_j = 1 - e^{-\lambda\eta_j}$,
- From Equation (1), we have $\mathbb{E}(X_N^j) = \eta_j + \frac{1}{\lambda}$.

Then, from Theorem 2 we have:

$$\begin{aligned}\mathbb{E}(T_{end}) &= \mathbb{E}(X_1) \sum_{j=1}^k \frac{1}{1-p_j} + \sum_{j=1}^k \eta_j - \mathbb{E}(X_N^j), \\ &= \frac{1}{\lambda} \sum_{j=1}^k e^{\lambda\eta_j} + \sum_{j=1}^k \eta_j - \sum_{j=1}^k \eta_j - \frac{k}{\lambda}, \\ &= \frac{1}{\lambda} \sum_{j=1}^k [e^{\lambda\eta_j} - 1].\end{aligned}$$

□

4.1.1 Case with a Constant Checkpoint Cost

When the cost of the checkpoint barrier is constant such that $C(I) = C$ and $R(S) = R$, the general form of the equation in Proposition 1 becomes:

$$\mathbb{E}(T_{end}) = \frac{1}{\lambda} \sum_{j=1}^k [e^{\lambda(I_j+C+R)} - 1]. \quad (3)$$

Lemma 1. *When the cost of Checkpoint and restart is constant, then the optimal interval between each checkpoint is $\frac{\omega_0}{\alpha m k}$ where k is the checkpoint number.*

Proof. Let L the Lagrange function and l_1 a Lagrange multiplier, such as:

$$L(I_1, I_2, \dots, I_k) = \frac{1}{\lambda} \sum_{j=1}^k e^{I_j+C+R} + l_1 \left(\frac{\omega_0}{\alpha m} - \sum_{j=1}^k I_j \right).$$

Then, the minimal value is the value where the gradient of L is equal to zero.

$$\begin{aligned}\frac{\partial L}{\partial I_1} &= e^{\lambda(I_1+C+R)} + l_1 = 0 \\ \frac{\partial L}{\partial I_2} &= e^{\lambda(I_2+C+R)} + l_1 = 0 \\ \frac{\partial L}{\partial I_k} &= e^{\lambda(I_k+C+R)} + l_1 = 0 \\ \frac{\partial L}{\partial l_1} &= \frac{\omega_0}{\alpha m} - \sum_{j=1}^k I_j = 0\end{aligned}$$

hence,

$$\begin{aligned} I_1 &= \frac{\log(-l1)}{\lambda} - C - R \\ I_2 &= \frac{\log(-l1)}{\lambda} - C - R \\ I_k &= \frac{\log(-l1)}{\lambda} - C - R \\ \frac{\omega_0}{\alpha m} &= \sum_{j=1}^k I_j \end{aligned}$$

Thus,

$$I_1 = I_2 = \dots = I_k = \frac{\omega_0}{\alpha m k}$$

□

After finding the optimal interval between each checkpoint using Lemma 1 the Equation (3) becomes:

$$\mathbb{E}(T_{end}) = \frac{k}{\lambda} (e^{\lambda(\frac{\omega_0}{\alpha m k} + C + R)} - 1) \quad (4)$$

Theorem 3. *If the failures follows a Poisson's process and the checkpoint/restart cost is constant, the optimal checkpoint number is:¹*

$$\hat{k} = \frac{\omega_0 \lambda}{\alpha m (1 + \mathcal{W}(-e^{-1-\lambda(C+R)}))}$$

Proof. To minimize Equation (4), let define ϕ by:

$$\phi(k) = \frac{k}{\lambda} (e^{\lambda(\frac{\omega_0}{\alpha m k} + C + R)} - 1)$$

The optimal value is the root of the derivation $\frac{d(\phi)}{dk} = 0$

$$\begin{aligned} \frac{\left(e^{\lambda(\frac{\omega_0}{\alpha m k} + C + R)} - 1 \right)}{\lambda} - \frac{\omega_0}{\alpha m k} e^{\lambda(\frac{\omega_0}{\alpha m k} + C + R)} &= 0 \\ e^{\frac{\lambda \omega_0}{\alpha m k}} e^{\lambda(C+R)} \left(\frac{\lambda \omega_0}{\alpha m k} - 1 \right) &= -1, \\ e^{\frac{\lambda \omega_0}{\alpha m k} - 1} \left(\frac{\lambda \omega_0}{\alpha m k} - 1 \right) &= -e^{-\lambda(C+R)-1}, \end{aligned}$$

Then, using the Lambert equation denoted by \mathcal{W} such as:

$$X = Y e^Y \iff Y = \mathcal{W}(X)$$

We obtain:

$$\hat{k} = \frac{\omega_0 \lambda}{\alpha m (1 + \mathcal{W}(-e^{-1-\lambda(C+R)}))}$$

□

Notice the Lambert's function is multivalued except when we restrict to real arguments, then the function is defined only for $X \geq -1/e$ which is true for $(-e^{-1-\lambda(C+R)})$. Using Taylor's series one can compute a numeric value of this function.

¹ \mathcal{W} denote the Lambert function [4].

4.2 Using Weibull's Law

In this second case study, a Weibull's law with shape parameter β and scale parameter λ is used to modelize the failure distribution occurrences.

Proposition 2. *If the distribution of inter-arrival failures time follows a Weibull's law with a shape parameter β and a scale λ parameter then:*

$$\mathbb{E}(T_{end}) = \sum_{j=1}^k e^{(\lambda\eta_j)^\beta} \int_0^{\eta_j} e^{-(\lambda x)^\beta} dx.$$

Proof. Let $1 - p_j = \bar{p}_j$. Using Theorem 2, we have:

$$\mathbb{E}(T_{end}) = \mathbb{E}(X_1) \sum_{j=1}^k \frac{1}{\bar{p}_j} + \sum_{j=1}^k \eta_j - \sum_{j=1}^k \mathbb{E}(X_{N_j}).$$

Then, from Weibull's law we have:

- $\mathbb{E}(X_1) = \Gamma(1 + \frac{1}{\beta}) \frac{1}{\lambda}$ such as Γ is the complete Gamma function,
- $p_j = 1 - e^{-(\lambda\eta_j)^\beta}$,
- $\mathbb{E}(X_{N_j}^j) = e^{(\lambda\eta_j)^\beta} \Gamma(1 + \frac{1}{\beta}) \frac{1}{\lambda} + \eta_j - e^{(\lambda\eta_j)^\beta} \int_0^{\eta_j} e^{-(\lambda x)^\beta} dx$.

Hence,

$$\begin{aligned} \mathbb{E}(T_{end}) &= \frac{\Gamma(1 + \frac{1}{\beta})}{\lambda} \sum_{j=1}^k e^{(\lambda\eta_j)^\beta} + \sum_{j=1}^k \eta_j - \sum_{j=1}^k \eta_j - \\ &\quad \sum_{j=1}^k e^{(\lambda\eta_j)^\beta} \left(\frac{\Gamma(1 + \frac{1}{\beta})}{\lambda} - \int_0^{\eta_j} e^{-(\lambda x)^\beta} dx \right), \\ \mathbb{E}(T_{end}) &= \sum_{j=1}^k e^{(\lambda\eta_j)^\beta} \int_0^{\eta_j} e^{-(\lambda x)^\beta} dx. \end{aligned}$$

□

4.2.1 Case with Constant Checkpoint Cost

Let us now assess the proposed model when the checkpoint cost is constant. Thus, the general form of the equation in Proposition 2 becomes:

$$\mathbb{E}(T_{end}) = \sum_{j=1}^k e^{(\lambda(I_j+C+R))^\beta} \int_0^{I_j+C+R} e^{-(\lambda x)^\beta} dx \quad (5)$$

Lemma 2. *If the failure distribution is a Weibull's law and the checkpoint/restart cost is constant, the optimal interval between each checkpoint is $\frac{\omega_0}{\alpha mk}$, where k is the optimal number of checkpoints.*

Proof. Let: $\tau(I_1, I_2 \cdots I_k)$.

$$\begin{aligned} \tau(I_1, I_2 \cdots I_k) = & e^{(\lambda(I_1+C+R))^\beta} \int_0^{I_1+C+R} e^{-(\lambda x)^\beta} dx + \\ & e^{(\lambda(I_2+C+R))^\beta} \int_0^{I_2+C+R} e^{-(\lambda x)^\beta} dx + \dots \\ & \dots + e^{(\lambda(I_k+C+R))^\beta} \int_0^{I_k+C+R} e^{-(\lambda x)^\beta} dx \end{aligned}$$

τ is symmetric then:

$$\tau(I_1, I_2 \cdots I_k) = \frac{1}{k} (\tau(I_1, I_2 \cdots I_k) + \tau(I_k, I_1 \cdots I_{k-1}) + \dots + \tau(I_2, I_3 \cdots I_k, I_1)) \quad (6)$$

τ is convex then:

$$\begin{aligned} \frac{1}{k} (\tau(I_1, I_2 \cdots I_k) + \tau(I_k, I_1 \cdots I_{k-1}) + \dots + \tau(I_2, I_3 \cdots I_k, I_1)) \geq \\ \tau \left(\frac{(I_1 + I_2 \cdots + I_k)}{k}, \frac{(I_k + I_1 + \dots + I_{k-1})}{k}, \dots, \frac{(I_2 + I_3 + \dots + I_k + I_1)}{k} \right) \end{aligned} \quad (7)$$

$$7 + 6 \implies \forall (I_1, I_2 \cdots I_k) \quad \tau(I_1, I_2 \cdots I_k) \geq \tau \left(\frac{\omega_0}{mk}, \frac{\omega_0}{mk}, \dots, \frac{\omega_0}{\alpha mk} \right)$$

□

Then, using Lemma 2 the new expression of the expected completion time becomes:

$$k e^{(\lambda(\frac{\omega_0}{\alpha mk} + C + R))^\beta} \int_0^{\frac{\omega_0}{\alpha mk} + C + R} e^{-(\lambda x)^\beta} dx \quad (8)$$

Thus to minimize Equation (8) the Newton's numerical method is used to find the root of the derivative function. To achieve this goal we use the *fsolve* function in MAPEL softwar.

5 Simulations

In the first set of simulations, a Poisson's process is used to validate the proposed model and to compare it with Daly's model [5]. Then, a second set of simulations is presented that confirm that the proposed model can predict the optimal checkpoint location even when the failure process is modeled by a Weibull's law. We also compare this model with Yudan's model [9].

5.1 Using Poisson's Process

In this scenario, we study the behavior of the proposed model with a Poisson's process when the cost of the checkpoint barrier C varies in a given interval such that the amount of initial work and the failures rate is known. For each value of C we compute the average completion times over 10^4 executions and the associated 95% confidence interval. In the second scenario, we set the checkpoint cost and the initial amount of work at a given value and we compute the average completion times when the failure rate increases and becomes very large.

5.1.1 Variation of Checkpoint Cost

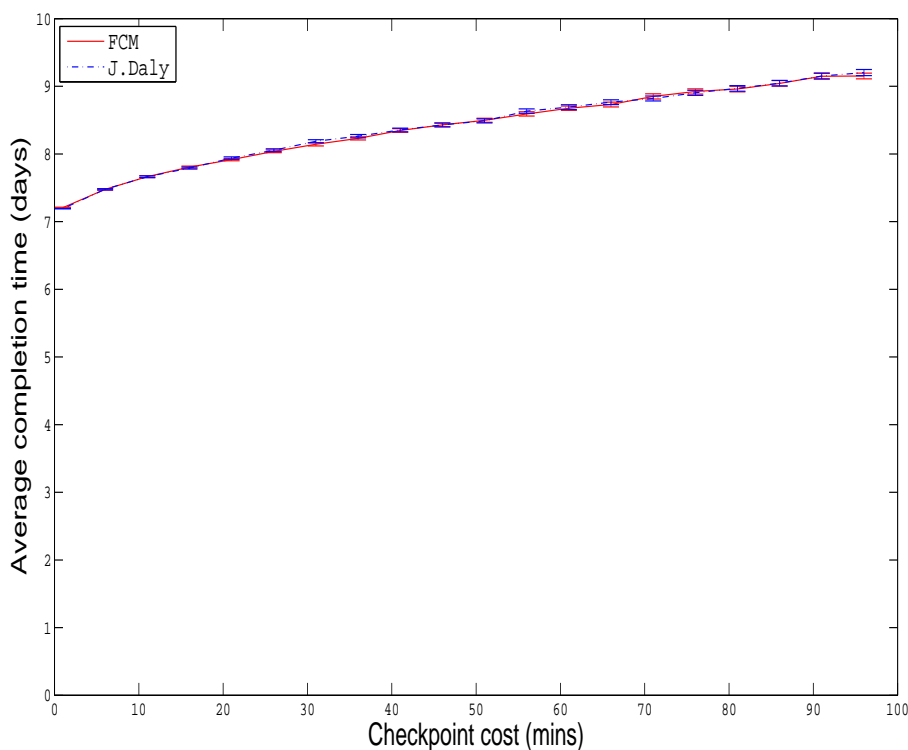


Figure 4: Variation of the average completion times with $C \in [1, 95]$ mins, $\frac{\omega_0}{\alpha m} = 7$ days, $\lambda = \frac{1}{2}$ per day

In this simulation the initial amount of work per computing node is 7 days and the failures rate per day is equal to $\frac{1}{2}$, then the checkpoint cost parameter C will increase in the interval $[1, 95]$ minutes. The results are displayed in Figure 4. It is clear that the two curves are not distinguishable. Thus, we conclude that our flexible checkpoint model (denoted by FCM) achieves about the same performance as in Daly's model.

5.1.2 Variation of Failure Rate

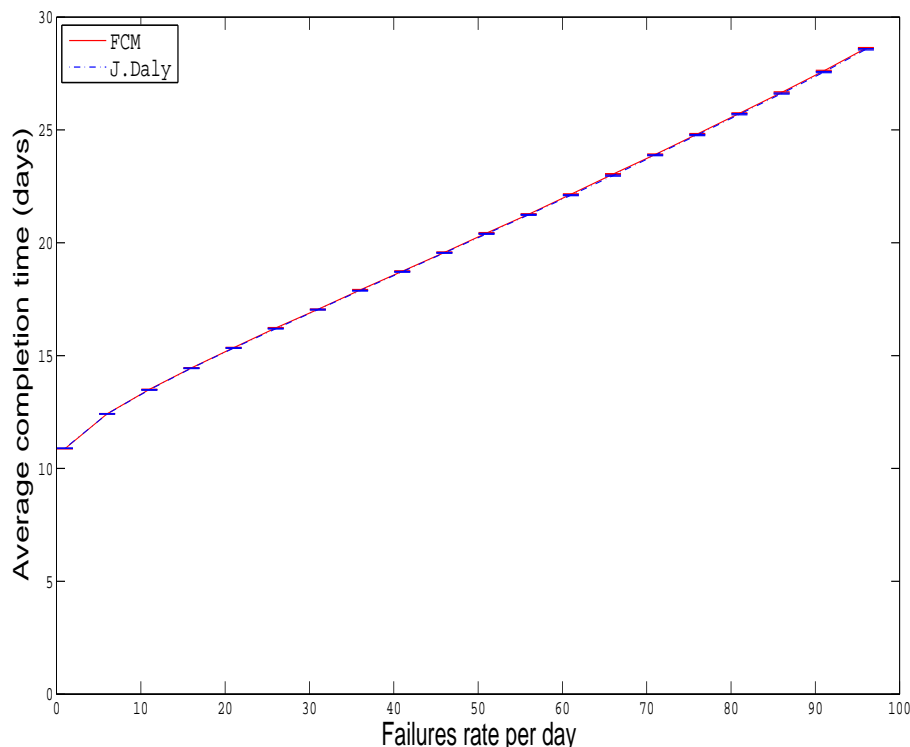


Figure 5: Variation of the average completion times with $\lambda \in [\frac{1}{2}, 96]$ per day, $\frac{\omega_0}{\alpha m} = 10$ days, $C = 10$ mins

In this section, we study the behaviors of our model when the failure rate λ becomes very large. To develop this second series of simulations, the scenario is the following: We set the duration of the parallel application to 10 days and the cost of checkpoint is constant at 10 minutes. Then, the failure rate per day varies in the interval $[\frac{1}{2}, 96]$, for each λ we report the average completion times for 10^3 executions. In Figure 5 it is clear that again both curves are not distinguishable. Therefore it confirms that our model keeps the same performance even when we increase highly the failures rate. The most important new result in these simulations is presented in Figure 6 that reports the number of checkpoints of our model together with Daly's one. This figure shows that our model reduces up to 20% the number of checkpoints compared to the number of checkpoints made by Daly's model. Indeed, we notice that this point is very important even if we have the same average completion time of the execution in simulation, i.e, since the checkpoint mechanism may generate high of network congestion.

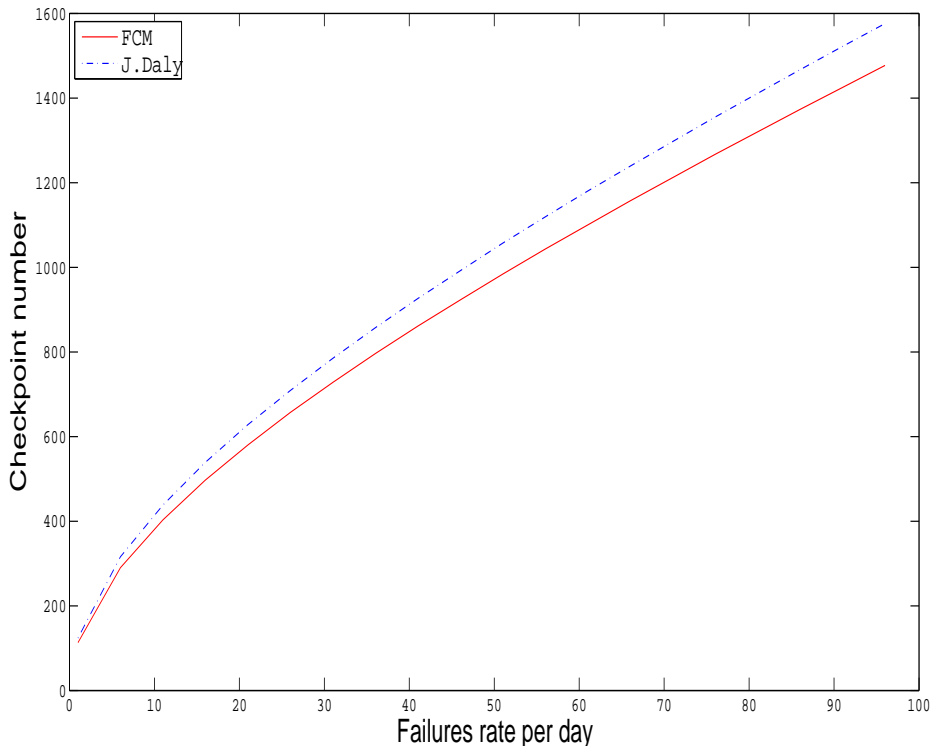


Figure 6: Variation of the optimal checkpoint number with $\lambda \in [\frac{1}{2}, 96]$ per day, $\frac{\omega_0}{\alpha m} = 10$ days, $C = 10$ mins

5.2 Variable cost of checkpoint

In this section, we validate the proposed model in cases where the cost of the checkpoint depends on the residual workload. We use traces of execution collected from real execution on the middleware KAAPI [7]. The trace contains sum of the data size that have been written at each checkpoint barrier (see Figure 7). Notice that the sampling time between each checkpoint is 10 minutes and the application execution time is 220 minutes, it corresponds to N-Queens benchmark on 572 processors and 10 stable storage system in Grid5000 platform. From these experiments it is clear that the cost of the checkpoint is not constant but it depends on the residual work load. Thus, from this set of values a cost function is established by polynomial interpolation with degree 7 (see Figure 8). Thus, the cost of a checkpoint at S is equal to the evaluation of the polynomial at this point multiplied by the data writing cost on a stable storage. Determining the optimal solutions to this optimization problem with a formal method is hard. However, a first approach is to solve the problem with numerical method. We use an implementation of the trust-region-reflective algorithm [3] from the Optimization Toolbox in MATLAB. We

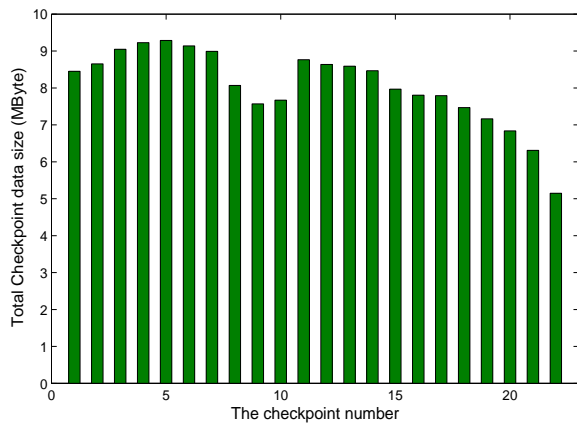


Figure 7: Variation of the Checkpoint size to write in the stable storage

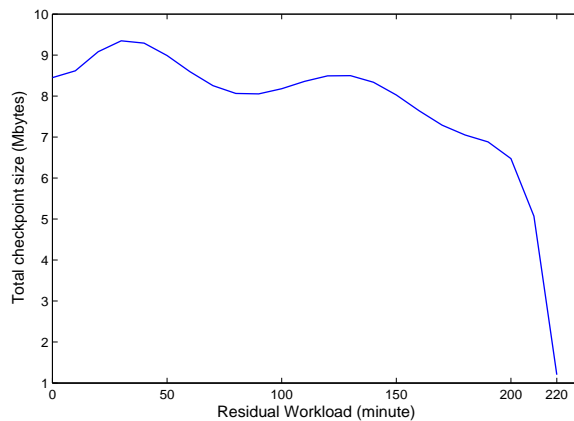


Figure 8: Polynomial interpolation of the checkpoint variation

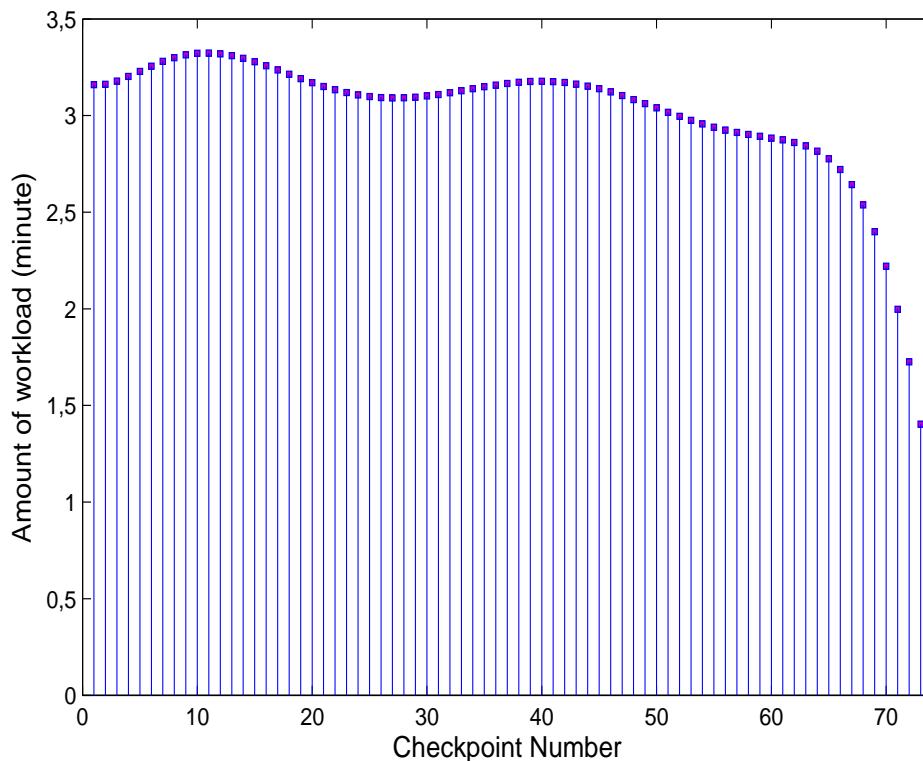


Figure 9: Optimal workload intervals between each checkpoint

use this algorithm to minimize $f(I, k)$ such as:

$$f(I, k) = \frac{1}{\lambda} (e^{\lambda(I(1)+C(S_1)+R(S_0))} - 1) + \frac{1}{\lambda} (e^{\lambda(I(2)+C(S_2)+R(S_1))} - 1) \dots$$

$$+ \frac{1}{\lambda} (e^{\lambda(I(k)+C(S_k)+R(S_{k-1}))} - 1).$$

Under the following constraints:

1. Equality constraints $\sum_{j=1}^k I_j = \frac{\omega_0}{m}$
2. In-equality constraints $\forall I_j \mid 1 \leq j \leq k$ and each $I_j \geq 0$.

In the last figure 9 each bar represents the optimal amount of work between the checkpoint j and $j + 1$. This figure shows that the proposed solution fit very well with the variation of the checkpoint cost. Since, it is clear that the bar length increases when the cost of checkpoint increases which means that the model reduces the number of checkpoint when the cost of checkpoint increases. Then when the cost of the checkpoint decreases the bar length decreases also that means it is more effective to increase the number of checkpoints when it has a lower cost.

5.3 Using Weibull's Law

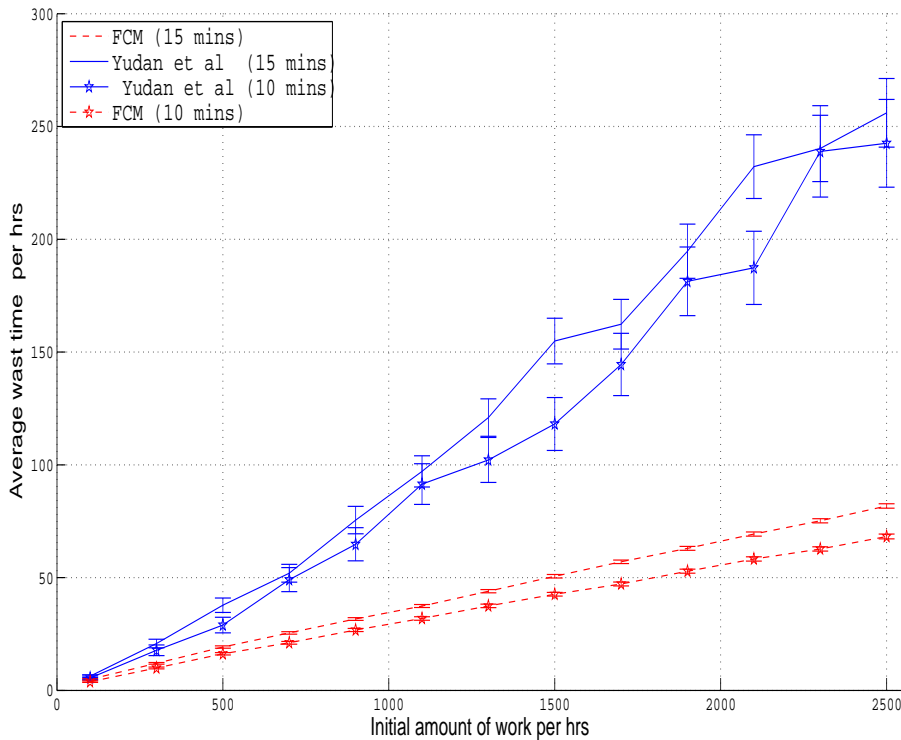


Figure 10: Variation of the average wasted time with $\frac{\omega_0}{\alpha m} \in [100, 2500]$ hrs, $\lambda = \frac{1}{20.584}$, $\beta = 0.509$

In this new simulation a Weibull's law is used to generate the successive interval between failures. We use the same value for $(\beta, \lambda, \frac{\omega_0}{\alpha m})$, as those used in [9] to compare the results. Thus, the shape parameter ($\beta = 0.509$) and $\lambda = \frac{1}{20.584}$. In this scenario, the total amount of work is increased in the interval [100, 2500] hours. Then, we compare the average wasted times on 10^4 simulations produced by the two models. The wasted time is the same as defined in [9], it is equal to the overhead due to the checkpoint plus the amount of work that should be re-executed due to failures plus restart costs.

Figure 10 represent the average wasted time for each model when the checkpoint overhead is 10 minutes and 15 minutes. This figure shows that our model reduces up to factor 4 the wasted time, especially when the initial amount of work is relatively large. Also, we notice that our model reduces very well the confidence interval.

6 Concluding Remarks

We have presented in this work a new flexible model for checkpointing applications in large-scale computing platforms. The main result was to establish a simple analytical model which expresses the expected completion time of the application. We showed that it is possible to use this model with various probability laws that modelize the failures process, such as the Poisson's process or the Weibull's law. Moreover, it is also this model is designed, for determining the optimal interval between checkpoints considering a variable checkpoint cost and the wasted time. A comparison with other existing mechanisms revealed that our model reduces the congestion of the network and the waste time.

We are currently working at implementing this mechanism in an actual system. For this purpose, it would be interesting to take into account a variable number of processors (instead of unbounded number of processors) since it is not obvious to always have extra available processors to replace those which failed.

References

- [1] NR et al Adiga. An Overview of the BlueGene/L Supercomputer. In *Supercomputing, ACM/IEEE 2002 Conference*, pages 60–60, 2002.
- [2] K. M. Chandy and L. Lamport. Distributed snapshots: determining global states of distributed systems. *ACM Trans. Comput. Syst.*, 3(1):63–75, 1985.
- [3] T.F. Coleman and Y. Li. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds . *SIAM Journal on Optimization*, 6:418–445, 1996.
- [4] R.M. Corless, D.J. Jeffrey, and D.E. Knuth. A sequence of series for the Lambert W function. In *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pages 197–204. ACM New York, NY, USA, 1997.

-
- [5] J. T. Daly. A higher order estimate of the optimum checkpoint interval for restart dumps. *Future Generation Computer Systems*, 22(3):303–312, 2006.
- [6] E. N. Elnozahy and J. S. Plank. Checkpointing for peta-scale systems: A look into the future of practical rollback-recovery. *IEEE Trans. Dependable Secur. Comput.*, 1(2):97–108, 2004.
- [7] T. Gautier, X. Besseron, and L. Pigeon. Kaapi: A thread scheduling runtime system for data flow computations on cluster of multi-processors. In *PASCO '07: Proceedings of the 2007 international workshop on Parallel symbolic computation*, pages 15–23, New York, NY, USA, 2007. ACM.
- [8] R. Geist, R. Reynolds, and J. Westall. Selection of a checkpoint interval in a critical-task environment. *IEEE Transactions on Reliability*, 37:395–400, October 1988.
- [9] Y. Liu, R. Nassar, C. Leangsuksun, N. Naksinehaboon, M. Paun, and S.L. Scott. An optimal checkpoint/restart model for a large scale high performance computing system. *IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008.*, pages 1–9, April 2008.
- [10] N. Naksinehaboon, Y. Liu, C.B. Leangsuksun, R. Nassar, M. Paun, and S.L. Scott. Reliability-Aware Approach: An Incremental Checkpoint/Restart Model in HPC Environments. In *Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, pages 783–788, 2008.
- [11] A. J. Oliner, L. Rudolph, and R. K. Sahoo. Cooperative checkpointing: a robust approach to large-scale systems reliability. In *ICS '06: Proceedings of the 20th annual international conference on Supercomputing*, pages 14–23, New York, NY, USA, 2006. ACM.
- [12] J. S. Plank and M. G. Thomason. The average availability of parallel checkpointing systems and its importance in selecting runtime parameters. In *29th International Symposium on Fault-Tolerant Computing*, pages 250–259, 1999.
- [13] H. C. Tijms. *A First Course in Stochastic Models*. John Wiley, 2003.
- [14] J. W. Young. A first order approximation to the optimum checkpoint interval. *Commun. ACM*, 17(9):530–531, 1974.



Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399