Comments on "Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpoint"

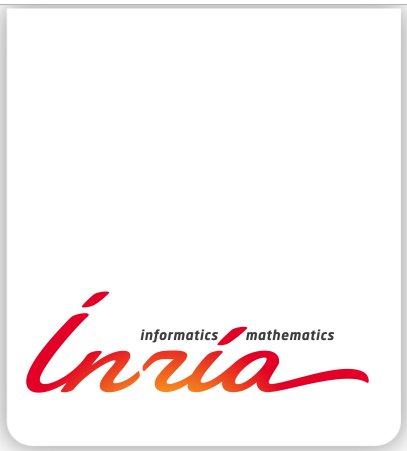
Guillaume Aupy, Yves Robert, Frédéric Vivien, Dounia Zaidouni

Comments on "Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpoint"

Guillaume Aupy, Yves Robert, Frédéric Vivien, Dounia Zaidouni

RESEARCH REPORT
N° 8318
June 2013
Project-Team Roma

ISSN 0249-6399 ISRN INRIA/RR--8318--FR+ENG



Comments on "Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpoint"

Guillaume Aupy*, Yves Robert*†‡, Frédéric Vivien§*, Dounia Zaidouni§*

Project-Team Roma

Research Report n° 8318 — June 2013 — 4 pages

Abstract: In this short note, we provide some comments on the recent paper "Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpointing" by Bouguerra et al., published in [3]. We start by identifying some errors in their equations. Then we explain that they do not actually use the distribution of lead times, contrary to statements by the authors. Finally, we show that their algorithm does not change policy at the best possible moment, and we point to our own work [2] for the (correct version of the) optimal algorithm.

Key-words: fault tolerance, checkpointing, prediction, algorithms, model, exascale

* LIP, École Normale Supérieure de Lyon, France
† University of Tennessee Knoxville, USA
‡ Institut Universitaire de France
§ INRIA

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

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

Commentaires sur l'article "Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpoint"

Résumé : Dans cette courte note nous commentons l'article "Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpointing" de Bouguerra et al. [3]. Nous commençons par identifier des erreurs dans la mise en équation du problème. Nous expliquons ensuite que, contrairement à ce qu'ils prétendent, les auteurs n'utilisent pas la distribution du délai de prédiction (*lead time*). Finalement, nous montrons que leur algorithme ne change pas de politique au moment optimum, et nous indiquons que nous avons présenté l'algorithme optimal dans un rapport de recherche [2].

Mots-clés : Tolérance aux pannes, checkpoint, prédiction, algorithmes, modèle, exascale

1 Introduction

In this short note, we provide some comments on the recent paper "Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpointing" by Bouguerra et al., published in [3]. The authors of [3] claim that they use the distribution of prediction lead times, thereby improving upon our previous work [1]. We explain why this claim is not correct in Section 3. Beforehand, we start in Section 2 by identifying some errors in the equations of [3]. Finally in Section 4, we show that their algorithm does not change policy at the right moment, and we point to our own previous work for the correct version of the optimal algorithm. The main objective of this note is to take timely credit for the respective contributions of [3] and [1, 2].

2 List of corrections

In this section, we list some errors done in the equations of Section IV, *Analytical modeling and optimization*, in [3].

1. Equation 3: the authors state that $\sigma(t) = c_1 \frac{t}{\tau}$, where σ is the overhead due to periodic checkpointing of non-faulty periods. However, time spent in non-faulty periods at time t is not equal to t, but to $t - \delta(t) - \gamma(t)$ (the total time spent minus the time not actually spent working because of failures or of proactive actions). Furthermore, τ "represents the units of useful work between two consecutive preventive checkpoints". The checkpointing period is, therefore, $\tau + c_1$ and not τ. Overall, we obtain: $\sigma(t) = c_1 \frac{t-\delta(t)-\gamma(t)}{\tau+c_1}$.

2. Equation 5: the authors state that $\gamma(t) = \frac{c_2 \bar{p} q s r t}{p\mu} + \frac{q s r t}{\mu}(\mathbb{E}[\Delta_l] + R)$. The error in this equation comes from conditional probabilities. The error lies in the expression for $\mathbb{E}[\Delta_l]$: the case considered here is when there is enough lead time to take proactive action (see the factor s). Thus $\mathbb{E}[\Delta_l]$ should be replaced by $\mathbb{E}[\Delta_l | \Delta_l \geq c_2]$. Note that $\mathbb{E}[\Delta_l | \Delta_l \geq c_2] \geq \mathbb{E}[\Delta_l]$.

3. In paragraph IV.B.3.1, the expected lost work due to non predicted failures is, according to the authors, on average $\frac{t\bar{r}}{\mu}(\frac{\tau}{2} + R)$, because the expected lost time due to one failure is $\tau/2$. However one should note that the expected lost time is half a period, and that is $(\tau + c_1)/2$. The time between two checkpoints is not τ, which is the time between the end of the first checkpoint and the beginning of the second checkpoint. However, the time that we are interested in is the time between the end of two checkpoints. In other words one should not forget to take the preventive checkpoint into account since a fault can occur during a checkpoint. Then the expected lost work should be $\frac{t\bar{r}}{\mu}(\frac{\tau+c_1}{2} + R)$.

4. Similarly, in paragraph IV.B.3.2, the expected lost time due to short lead time intervals should be $\frac{t\bar{s}r}{\mu}(\frac{\tau+c_1}{2} + R)$ instead of $\frac{t\bar{s}r}{\mu}(\frac{\tau}{2} + R)$.

5. Similarly to the error in Equation 5, in Equation 6, the term $\mathbb{E}[\Delta_l]$ should be replaced by $\mathbb{E}[\Delta_l | \Delta_l \geq c_2]$: again, it is considered for this waste that the lead time is large enough to take proactive actions. Furthermore note that in this case, $\mathbb{E}[t_a]$ should be equal to $\min(\frac{h}{2}, \frac{\tau}{2})$ instead of $\frac{h}{2}$ (for the case where $h \geq \tau$).

6. Finally, in order to obtain Equation 7, the authors summed Equations 3, 4, 5, and 6, but the summation is incomplete: in the first part of Equation 7, when $h < \tau$, a term is missing, and $\frac{sr}{\mu}\mathbb{E}[\Delta_l]$ should be subtracted from the result.

3 Distribution of prediction lead times

In [3], the authors claim to use the probability distribution law of lead times (the lead time is the time between the alert and the actual fault). However, when computing the waste, they simply consider two kinds of lead time intervals: those which are bigger than C_p (in which case one has enough time to take proactive actions) and those that are smaller (in which case one does not). They consider a proportion s of the former and $1-s$ of the latter out of all lead time intervals. Then they update the recall r using a simple transformation: the only true predictions where one can actually take some proactive action have a recall of sr. They consider the $(1-s)r$ other predictions as unpredicted faults (which added together with the $1-r$ originally unpredicted faults, makes $1-sr$ unpredicted faults).

Altogether, the probability distribution law of lead times is, therefore, irrelevant. This directly contradicts the authors of [3] who wrote "the lead time distribution should be modeled carefully and the exponential distribution can not be chosen arbitrarily to represent it". The only parameter that indeed matters is the fraction of predictions that are made enough time in advance so that a proactive action can be taken. This observation is already made in [1, 2].

4 Optimal algorithm

In [3], W_p is defined as the expected wasted time if the decision is to perform proactive action. According to the Equation that defines W_p, the strategy in [3], to the best of our understanding, seems to decide whether to checkpoint, or not, at the beginning of the lead time interval. This strategy is not optimal: once the alert of a future fault occurs, one knows the exact moment of the fault (otherwise one could never know whether a false positive is a false positive or a very long lead time). Then the obvious strategy is to checkpoint right before the fault.

With this in mind, the Equation defining W_p, $W_p = p(R + c_2 + \Delta_l - c_2) + \bar{p}c_2$, should be replaced by $W_p = p(R+c_2)+\bar{p}c_2$, and the correct condition whether or not one should checkpoint becomes: one should checkpoint if and only if the sum of the time lost before the alert and of the lead time (assuming that this lead time is greater than C_p) is greater $\frac{C_p}{p}$. In [3], the condition is that the time lost (without the lead time) is greater than $\frac{(1-p)C_p}{p}$.

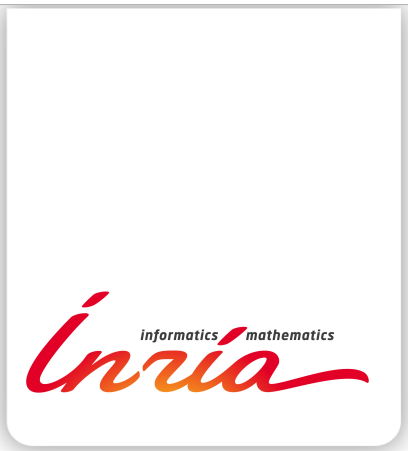
The correct version of the optimal algorithm can be found in [2].

References

[1] G. Aupy, Y. Robert, F. Vivien, and D. Zaidouni. Impact of fault prediction on checkpointing strategies. Technical Report RR-8023,v1, INRIA, July 2012.

[2] G. Aupy, Y. Robert, F. Vivien, and D. Zaidouni. Checkpointing algorithms and fault prediction. Research report RR-8237, INRIA, February 2013.

[3] M. Bouguerra, A. Gainaru, L. Gomez, F. Cappello, S. Matsuoka, and N. Maruyama. Improving the computing efficiency of HPC systems using a combination of proactive and preventive checkpointing. In *Proceedings of the 27th International Parallel & Distributed Processing Symposium (IPDPS'13)*, pages 501–512. IEEE, May 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