University of Texas at El Paso
DigitalCommons@UTEP

Departmental Technical Reports (CS) Department of Computer Science

4-1-2008

Impact of Checkpoint Latency on the Optimal Checkpoint Interval and Execution Time

Sarala Arunagiri
University of Texas at El Paso, sarunagiri@utep.edu

Seetharami Seelam

Ron A. Oldfield

Maria Ruiz

Patricia J. Teller
University of Texas at El Paso, pteller@utep.edu

See next page for additional authors

Follow this and additional works at: http://digitalcommons.utep.edu/cs_techrep

Part of the Computer Engineering Commons

Comments:
Technical Report: UTEP-CS-07-55a

Recommended Citation
Arunagiri, Sarala; Seelam, Seetharami; Oldfield, Ron A.; Ruiz, Maria; Teller, Patricia J.; and Riesen, Rolf, "Impact of Checkpoint Latency on the Optimal Checkpoint Interval and Execution Time" (2008). *Departmental Technical Reports (CS).* Paper 224.
http://digitalcommons.utep.edu/cs_techrep/224

This Article is brought to you for free and open access by the Department of Computer Science at DigitalCommons@UTEP. It has been accepted for inclusion in Departmental Technical Reports (CS) by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.

Authors
Sarala Arunagiri, Seetharami Seelam, Ron A. Oldfield, Maria Ruiz, Patricia J. Teller, and Rolf Riesen

This article is available at DigitalCommons@UTEP: http://digitalcommons.utep.edu/cs_techrep/224

Impact of Checkpoint Latency on the Optimal Checkpoint Interval and Execution Time

Sarala Arunagiri* Seetharami Seelam† Ron A. Oldfield‡ Maria Ruiz Varela*
Patricia J. Teller* Rolf Riesen‡

Abstract

The massive scale of current and next-generation massively parallel processing (MPP) systems presents significant challenges related to fault tolerance. In particular, the standard approach to fault tolerance, application-directed checkpointing, puts an incredible strain on the storage system and the interconnection network. This results in overheads on the application that severely impact performance and scalability. The checkpoint overhead can be reduced by decreasing the checkpoint latency, which is the time to write a checkpoint file, or by increasing the checkpoint interval, which is the compute time between writing checkpoint files. However, increasing the checkpoint interval may increase execution time in the presence of failures. The relationship among the mean time to interruption (MTTI), the checkpoint parameters, and the expected application execution time can be explored using a model, e.g., the model developed by researchers at Los Alamos National Laboratory (LANL). Such models may be used to calculate the optimal periodic checkpoint interval. In this paper, we use the LANL model of checkpointing and thorough mathematical analysis we show the impact of a change in the checkpoint latency on the optimal checkpoint interval and the overall execution time of the application.

For checkpoint latencies, δ_1 and δ_2, and the corresponding optimal checkpoint intervals, τ_1 and τ_2, our analysis shows the following results: (1) For a given MTTI, if δ_1 is greater than δ_2, τ_1 is greater than or equal to τ_2. (2) When the checkpoint interval is fixed, a decrease in checkpoint latency results in a decrease in application execution time. (3) A reduction in checkpoint latency, from δ_1 to δ_2, and a corresponding change of the checkpoint interval from the optimal checkpoint interval associated with δ_1, τ_1, to that associated with δ_2, τ_2, translates to reduced application execution time when the difference between τ_1 and τ_2 exceeds a certain threshold value, which can be as large as 12% of τ_{opt}.

In terms of application execution times, the approximation error of the optimal checkpoint interval is not significant. However, when we consider other performance metrics of the application, such as network bandwidth consumption and I/O bandwidth consumption, we conjecture that the information obtained by the analysis presented in this report could be of value in reducing resource consumption.

*The University of Texas at El Paso
†IBM TJ Watson Research Center
‡Sandia National Laboratories

1 Introduction

Modern high-end massively parallel processing (MPP) systems have tens of thousands of processors [3, 22], and next-generation systems are expected to have in excess of one-hundred thousand processors. These systems are designed specifically to support scientific applications that simulate fusion [**?**], combustion [19], climate [2], and other phenomena that require tremendous computing resources to provide scientists with useful insights in a reasonable amount of time. The massive scale of these systems translates into an increase in the expected number of failures per time period. This imposes new challenges related to fault tolerance.

Checkpointing[1] is one of the most commonly used fault tolerance mechanisms. It introduces an overhead that augments application execution time – the overhead is dependent on checkpoint file size, checkpoint frequency, and system parameters, such as storage bandwidth and network bandwidth. Applications executed on MPP systems have extremely large data sets and, thus, large checkpoint files. As a result, the overhead associated with writing these files can impact application performance and scalability [11]. Since application scalability is an important issue for MPP systems, techniques that reduce checkpoint overhead are valuable. Research in this area focuses on three related topics: improvement of system reliability, i.e., increased mean time to interruption(MTTI), reduction of the total number of checkpoints, and reduction of the latency of the checkpointing operation. While there is much to be done to improve reliability, the focus of this paper is on the latter two topics in the context of periodic checkpointing. Given the checkpointing parameters such as checkpoint latency and MTTI, Daly's model [4], [5], [6], provides a method for computing the optimal checkpoint which is associated with the optimal execution time. The choice of a checkpoint interval influences the number of checkpoint operations performed during an application's execution. Complementary work by Oldfield proposes the use of a combination of a lightweight storage architecture [13] and overlay network [9] to reduce checkpoint latency.

Consider a situation where the checkpoint latency decreases from δ_1 to δ_2. Assuming that Daly's optimal checkpoint interval is used to guide the selection of the periodic checkpoint interval, the decrease in checkpoint latency triggers a series of questions relevant to computational scientists:

1. Is the optimal checkpoint interval, τ_2, that corresponds to the decreased checkpoint latency, δ_2, greater or less than the current optimal checkpoint interval, τ_1? Proposition 1 claims that τ_2 will be less than or equal to τ_1. In general, the proposition claims that the optimal checkpoint interval is a non-decreasing function of checkpoint latency.

2. If we choose not to change the checkpoint interval to the new optimal checkpoint interval, τ_2,can we still achieve a decrease in the expected execution time? Claim 5.1 asserts that this is true and it quantifies the decrease in the expected execution time.

3. What happens if the checkpoint interval is changed from τ_1 to τ_2? Are we always guaranteed to decrease the expected execution time? A computational scientist can find the answer to this question by using Daly's execution time model to compute and compare the expected execution times corresponding to (δ_2, τ_2) and (δ_2, τ_1). Since τ_2 is the optimal checkpoint interval corresponding

[1] An application "checkpoint" is data that represents a consistent state of the application that can be saved and then, in the event of a failure, restored and used to resume execution at the saved state. A checkpoint is generally stored to persistent media (e.g., a file system).

to δ_2, we expect that, for every $\tau_1 \neq \tau_2$, the expected execution time corresponding to (δ_2, τ_2) is less than the expected execution time corresponding to (δ_2, τ_1). But, this is not always true. Surprisingly, there are values of the checkpoint interval, τ_3, which are less than τ_2, for which the expected execution time corresponding to (δ_2, τ_2) is greater than that corresponding to (δ_2, τ_3). As stated by [4], the reason for this is that τ_2 is an approximation of the real optimal checkpoint interval. Section 5.1 details issues concerning execution time of checkpointing applications that arise naturally in this context.

4. The error of approximation of the optimal checkpoint interval is less than or equal to 6% as stated in [4]. With this information, can we identify all values of τ_3 for which this anomalous behavior occurs? To answer this question, we first need to know if the approximation to the optimal checkpoint interval is
   (a) always less than the real optimal checkpoint interval;
   (b) always greater than the real optimal checkpoint interval; or
   (c) unpredictable and could be either greater or less than the real optimal checkpoint interval.
5. Proposition 2 shows that an optimal checkpoint interval, τ_{opt}, computed using the expression derived from Daly's checkpointing model, is always less than the real optimal checkpoint interval, τ_{real}. As illustrated in Figure 4, the values with the aforementioned behavior, τ_3, are in the range $\tau_{opt} < \tau_3 < (\tau_{opt} + \alpha)$.
6. The next natural question is: Referring to point 5, how large can α get? Section 5.2 demonstrates that α can be as large as 12% of τ_{opt}, which is consistent with the error bounds of the model.
7. Is 12% of τ_{opt} a good estimate of α? Do we have an alternate way of computing α? Theorem 1 presents an expression for an approximation of α, which we know is always larger than α. Section 5.3 presents empirical data that compares the two methods of estimating α. Our observation is that, for the sample set presented, the closed form expression for α is better than *the 12% heuristic*.

Thus, as indicated above, this paper answers these questions. Accordingly, it produces the information required by a computational scientist to understand the effect of a decrease in checkpoint latency on the optimal checkpoint interval and application execution time. The paper is organized as follows. In the next section, we present background material. Section 3 presents Daly's function for calculating the optimal periodic checkpoint interval. Section 4 and Section 5 present a thorough mathematical analysis of Daly's function and in doing so determine the impact of a decrease in checkpoint latency on the optimal checkpoint interval and application execution time, respectively. Finally, we present a summary, conclusions, and future work in Section 6.

2 Background and Related Work

Checkpointing is primarily done by application programmers, who base the times at which to checkpoint on their knowledge of their applications. At a checkpoint, enough data is stored to enable the application, in the event of a failure, to resume execution at the saved state.

A recent paper by Oliner, Rudolph, and Sahoo [14] presents the idea of cooperative checkpointing, which uses runtime knowledge of system conditions to skip application-directed checkpoints. System conditions can include application interference, contention at the storage system, and network contention. This permits the flexibility to handle non-exponential failure distributions and to provide scalability with increasing failure rates and checkpointing latencies.

Then there are several techniques that target the reduction of checkpoint overhead, i.e., the time added to application execution time as a result of checkpointing. Some of these techniques are meant to hide some of the checkpoint latency and, thus, reduce checkpoint overhead. Copy-on-write checkpoint algorithms take advantage of the low-latency of memory; they copy checkpoint data to a separate memory address space via virtual-memory, page-protection hardware. Once a memory-to-memory transfer is complete, the checkpoint data are saved to stable storage while application execution continues. Copy-on-write algorithms can be improved by adding a buffering capability to enable the overlapping of memory-to-memory transfers of checkpoint data and the writing of the data to stable storage [10]. Although copy-on-write implementations slightly increase checkpoint latency, they decrease checkpoint overhead [7]. Since applications executing on MPP systems use large fractions of the available memory, copy-on-write and checkpoint-to-memory approaches [17] are not suitable for such systems.

The following techniques explicitly target the reduction of checkpoint latency. The use of RAID techniques has been proposed to store coordinated checkpoint data more efficiently [16]. RAID-inspired techniques, such as checkpoint mirroring, N+1 parity, and Reed-Solomon coding, are aimed at minimizing the impact of checkpointing on shared resources, e.g., I/O and network bandwidth, and on reducing checkpoint latency and recovery time [23]. Incremental checkpointing aims at reducing the size of checkpoint data by saving only the memory that has been touched since the last checkpoint operation. Page-based incremental checkpointing requires paging support from hardware and the operating system. Page-based techniques might not scale well on large MPP systems since even if only one bit in a page changes, the entire page must be saved; also, paging is not implemented on many MPP systems. Hash-based, as opposed to page-based, techniques are able to identify bytes changed in a page. This feature is used in [1] to propose an adaptive incremental checkpointing algorithm that aims at minimizing the amount of checkpoint data saved to stable storage. This algorithm uses a secure hashing function to dynamically identify a block corresponding to the approximate number of bytes changed in memory.

Current “in-practice” implementations of periodic checkpointing, such as “checkpoint to disk”, are becoming impractical on high-end MPP systems due to extremely long checkpoint latencies [12]. The long latencies are due to the shear volume of data that needs to be saved periodically and the limited system I/O bandwidth. In addition, as the number of processors increases (a trend in high-end MPP systems), the probability of a processor failure increases. This, in turn, demands more frequent checkpointing.

Several models that define the optimal checkpoint interval have been proposed in the literature. Young proposed a first-order model that defines the optimal checkpoint interval in terms of checkpoint overhead and mean time to interruption (MTTI). Young’s model does not consider failures during checkpointing and recovery [25], while Daly’s extension of Young’s model, a higher-order approximation, does [6]. In addition to considering checkpoint overhead and MTTI, the model discussed in [21] includes sustainable I/O bandwidth as a parameter and uses Markov processes to model the optimal checkpoint interval. The model described in [15] uses useful work, i.e., computation that contributes to job completion, to measure system performance. The authors claim that Markov models are not sufficient to model useful work and propose the use of Stochastic Activity Networks (SANs) to model coordinated checkpointing for large-scale systems. Their model considers synchronization overhead, failures during checkpointing and recovery, and correlated failures. This model also defines the optimal number of processors that maximize the amount of total useful work. Vaidya models the checkpointing overhead of a uniprocess application. This model also considers failures during checkpointing and recovery [24]. To evaluate the performance and scalability of coordinated checkpointing in future large-scale systems, [8] simulates checkpointing on several configurations of a hypothetical petaflop system. Their simulations consider the node as the unit of failure and assume that the probability of node failure is independent of its size, which is overly optimistic.

We use Daly's model of checkpointing to compute application execution time and the optimal checkpoint interval. This model assumes an exponential failure distribution. There is literature stating that system failures do not generally have an exponential distribution [20, 14, 18]. However, in modeling the exeuction time, the failure distribution considered needs to be valid only for the duration of the application run and not for the lifetime of the system. Therefore, in this context we believe that the assumption of an exponential failure distribution is valid.

3 Daly's Checkpoint Model

John Daly constructed a detailed model of wall clock application execution time on a computer system that exhibits Poisson single component failures [4], [5], [6]. In the model, execution time includes the time to perform checkpoints and the time to redo the work performed between the last checkpoint and a failure, i.e., *rework time*. For long-running applications, the execution time (T) is:

$$T = Me^{R/M}\left(e^{(\tau+\delta)/M} - 1\right)\frac{T_s}{\tau} \quad \text{for} \quad \delta << T_s, \tag{1}$$

where

T_s = Time spent doing actual computation of the application,
τ = Time between checkpoints, i.e., *checkpoint interval*,
δ = Time to output a checkpoint/restart file, i.e., *checkpoint latency*,
M = Mean time to interruption (MTTI) of the system, and
R = Rework time.

By minimizing the application execution time in Equation 1, Daly in [6] derives the following approximation for the optimal checkpoint interval, τ_{opt}, which depends on the time required for a checkpoint operation, i.e., the checkpoint latency, δ, and its relationship to the *mean time to interruption* (MTTI) of the system, M.

$$\tau_{opt} = \begin{cases} \sqrt{2\delta M}\left[1 + \frac{1}{3}\left(\frac{\delta}{2M}\right)^{\frac{1}{2}} + \frac{1}{9}\left(\frac{\delta}{2M}\right)\right] - \delta & \delta < 2M \\ M & \delta \geq 2M \end{cases} \tag{2}$$

Next, Section 4 explores the properties of Daly's equations. In particular, it analyzes the impact of the checkpoint latency and the length of the checkpoint interval on the wall clock execution time of a long-running application.

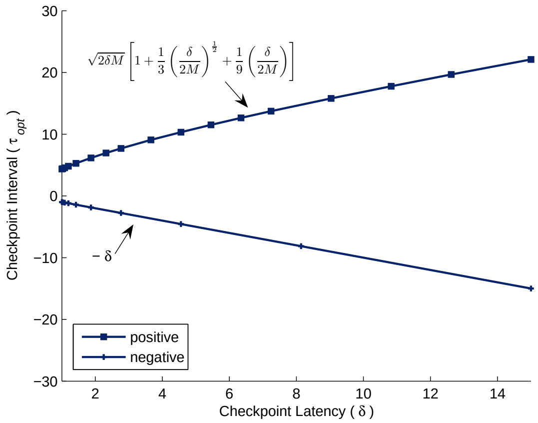

Figure 1. Variation of components of τ_{opt}

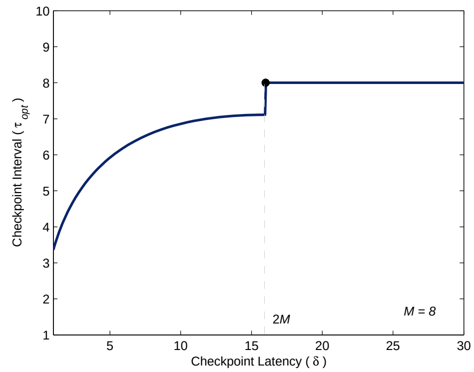

Figure 2. τ_{opt} vs checkpoint latency

4 The Optimal Checkpoint Interval

Examination of Equation 2 does not reveal whether or not the optimal checkpoint interval, τ_{opt}, always decreases with a reduction in checkpoint latency, δ. As illustrated in Figure 1, the equation consists of three terms that are monotonically increasing functions of δ and a single term that is a monotonically decreasing function of δ. From inspection, it is not clear which of these terms dominates in the various subranges of δ. In order to determine this, we examine the plot in Figure 2. This is a plot of τ_{opt} versus δ, i.e., checkpoint latency, for a specific value of MTTI, $M = 8$. In this case, τ_{opt} is a non-decreasing function of checkpoint latency. A follow-on question is: If we were to plot τ_{opt} as a function of δ for a value of M other than 8, would it still be a non-decreasing function? Proposition 1 states that this is indeed the case. Thus, a decrease in the value of checkpoint latency (e.g., using overlay networks and intermediate I/O nodes) results in a reduction of the optimal checkpoint interval.

Proposition 1. *For a given value of M, consider $\delta_1, \delta_2 \geq 0$. Let τ_1 and τ_2 be the corresponding optimal checkpoint intervals computed using Daly's checkpointing model, i.e.,$\tau_1 = \tau_{opt}(\delta_1)$ and $\tau_2 = \tau_{opt}(\delta_2)$. If $\delta_1 < \delta_2$ then $\tau_1 \leq \tau_2$.*

Proof. Since the proposition is conditioned on a fixed value of M, for this proof we can consider τ_{opt} to be a function of a single variable, δ. Proposition 1 is equivalent to the statement "$\tau_{opt}(\delta)$ is a non-decreasing function of δ". Appendix A shows that $\tau_{opt}(\delta)$ is a non-decreasing function of δ thereby proving Proposition 1. $\square$

For values of checkpoint latency, δ, in the range $0 \leq \delta < 2M$, the implications of this result appear counter-intuitive. It may seem strange that the modeled optimal checkpoint interval decreases with a decrease in the checkpoint latency, thus, making the checkpoint process more efficient implies that we need to checkpoint more often. In contrast, it is common to believe that to decrease the checkpoint overhead and, thus, execution time, one must increase the time between checkpoints. This if further explored below.

5 Consequence of Reduction in Checkpoint Latency on Application Execution Time

This section addresses a question that might be of interest to a computational scientist: Given a new technology that reduces the checkpoint latency from δ_1 to δ_2, will using the optimal checkpoint interval associated with δ_2, τ_2, in lieu of the optimal checkpoint interval associated with δ_1, τ_1, lead to improved execution time? To determine whether or not to switch to τ_2, one of the factors that must be considered is its effect on the wall-clock execution time of the application. This issue is addressed in the remainder of this section. Theorem 1 states that the expected application execution time corresponding to τ_2 is less than that corresponding to τ_1, when the difference between τ_1 and τ_2 exceeds a certain threshold. The theorem provides an expression to compute the threshold value. We show that this threshold value can be as large as 12% of τ_{opt}.

Below, the execution time model used is Equation 1. As a reference, let us fix an arbitrary application A with a solution time, T_s such that $T_s >> \delta$ for all relevant values of δ. Accordingly, in the remainder of this section, whenever we refer to execution time, we mean the execution time of application A executing with the specified checkpoint parameters.

5.1 Problem Definition

Consider a system with an MTTI, M, and a checkpoint latency, δ_1, such that $\delta_1 < 2M$. Let the current checkpoint interval, τ_1, be the optimal checkpoint interval corresponding to δ_1, i.e., $\tau_1 = \tau_{opt}(\delta_1)$. Let the execution time of application A under the stated checkpoint conditions be T_1. Assume that a more efficient checkpointing process is introduced and, as a result, the checkpoint latency is reduced from δ_1 to δ_2, where $\delta_2 = \delta_1 - \epsilon$ and $\epsilon > 0$.

Claim 5.1. If T_1' is the expected execution time of application A executing with checkpoint latency δ_2 and checkpoint interval τ_1, then $T_1' < T_1$.

Proof. This claim can be verified by inspecting the expression for application execution time,

$$T = Me^{R/M}\left(e^{(\tau+\delta)/M} - 1\right)\frac{T_s}{\tau}.$$

For a given τ and a given M, reducing δ reduces the the exponent and, therefore, reduces the expected execution time. The difference in the values of the expected execution times is given by

$$\begin{aligned} T_1 - T_1' &= T_1(1+K)(1-e^{(-\epsilon/M)}) \\ &\text{where } K = T_s M \epsilon^{(R/M)}/\tau_1 \end{aligned}$$

□

Now, let τ_2 be the optimal checkpoint interval corresponding to δ_2, i.e., $\tau_2 = \tau_{opt}(\delta_2)$. Both τ_1 and τ_2 are computed using Equation 2. Claim 5.1 showed that assuming the same checkpoint interval for both executions, the expected application execution time decreases with a decrease in checkpoint latency, i.e., from δ_1 to δ_2. The next question is: Can we reduce the execution time further by using a different value of checkpoint interval, e.g., using the new optimal checkpoint interval?

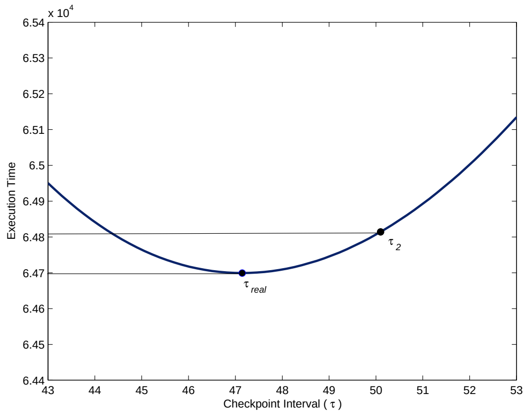

Figure 3. If τ_2 greater than τ_{real}

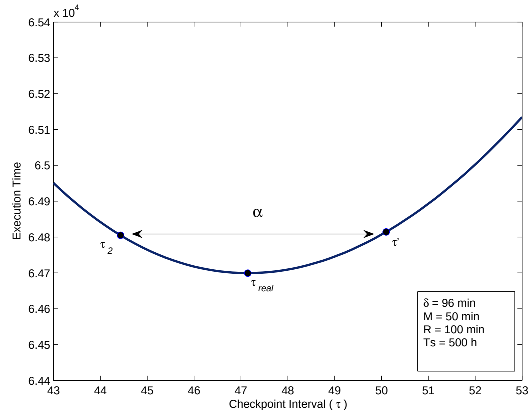

Figure 4. τ_2 smaller than τ_{real}

Suppose τ_2 is the optimal checkpoint interval that minimizes the execution time for δ_2, then for the given set of checkpoint parameters, using a checkpoint interval of τ_2 should result in the minimum achievable value of expected execution time. However, as stated in [6], τ_{opt} is an approximation of the optimal checkpoint interval. In general, an approximate optimal value can be expected to be larger or smaller than the real optimal value as illustrated in Figures **??**. In the context of our discussion, suppose that τ_2 is larger than the real optimal checkpoint interval, τ_{real}, as shown in Figure 3. Then switching the checkpoint interval from τ_1 to τ_2 leads to a decrease in expected execution time. On the other hand, as illustrated in Figure 4, if τ_2 is less than τ_{real}, then for values of checkpoint interval, τ_r, lying in the range $(\tau_2 < \tau_r < (\tau_2 + \alpha))$, the expected execution time increases if we switch the checkpoint interval from τ_r to τ_2. Thus if τ_1, τ_2, and α are such that $(\tau_2 < \tau_1 < (\tau_2 + \alpha))$ then, switching the checkpoint interval from τ_1 to τ_2 increases the expected execution time.

Proposition 2. *For any value of MTTI, M, and δ such that $0.01 \leq \frac{\delta}{2M} < 1$, the value of the optimal checkpoint interval computed using Equation 2 is less than the real optimal checkpoint interval.*

Proof. The proof appears in Appendix A. □

5.2 A First Estimate of the Value of α

The proof of Proposition 2 shows that in reality, as shown in Figure 4, τ_2 is always smaller than τ_{real}. This leads us to the next question: What is the smallest value of checkpoint interval, τ', such that ($\tau' > \tau_{opt}$) and $T(\tau_{opt}) = T(\tau')$), i.e., τ' - $\tau_{opt} = \alpha$? Again, referring to Figure 4, in essence, this question reduces to: What is the value of α?

A first estimate of the value of α is

$$\alpha \simeq 2 * E * \tau_{real},$$

where E is an upper bound on the relative error of the optimal checkpoint interval computed using Equation 2. According to [4], $E \leq 0.059$, i.e.,

$$E = \frac{|\tau_{real} - \tau_{opt}|}{\tau_{real}} < 0.059.$$

When $0.01 \leq \frac{\delta}{2M} < 1$, we know that $\tau_{real} > \tau_{opt}$ and

$$E = \frac{\tau_{real} - \tau_{opt}}{\tau_{real}} < 0.059.$$

Accordingly, our first estimate of the value of α is

$$\alpha \simeq 0.12 * \tau_{real}$$

Since we do not know the value of τ_{real}, we would like to express the value of α as a fraction of τ_{opt}. In the context of our discussion, we know that $\tau_{real} > \tau_{opt}$ and, therefore, $\alpha \simeq E' * \tau_{opt}$, where $E' > 0.12$, i.e., the value of α can get as large as 12 percent of τ_{opt}. Consider the example shown in Figure 5. In this example, the optimal checkpoint interval $\tau_{opt} = 44.43$ and $\alpha = 5.67$, which is 12.6% of τ_{opt}. This example illustrates that 12% of τ_{opt} is an attainable upper bound for the value of α. Thus, $\alpha = 12.5$ percent of τ_{opt} is a simple heuristic for the value of α; let us call this *the 12% heuristic*. This heuristic works well when the approximation error of τ_{opt} is close to 0.059 percent of τ_{opt}.

The value of alpha can be computed using Lambert W function. Although there are mathematical packages like MATLAB and Maple that provide Lambert W function, not all math calculators provide this function as an elementary function. Besides we were curious to see if a closed form expression for an approximation of α in terms of elementary functions of the other parameters could possibly provide additional insights. The following theorem presents a closed form expression for an approximation of α. The value of the estimate thus obtained is always greater than the value of (α) and it is independent of the approximation error of τ_{opt}.

Theorem 1. *For a given value of MTTI, M, consider checkpoint latencies, δ_1 and δ_2, such that $\delta_2 = \delta_1 - \epsilon$ and $\epsilon > 0$. Let τ_1 and τ_2 be the optimal checkpoint intervals corresponding to δ_1 and δ_2, respectively. When the checkpointing latency is δ_2, let T_1 and T_2 be the expected execution times associated with checkpoint intervals τ_1 and τ_2, respectively. Then,*

$$(T_2 < T_1) \quad if \quad \left[(\tau_1 - \tau_2) > \left(\frac{2M}{\tau_2} \left[(M - \tau_2) - Me^{-(\tau_2+\delta_2)/M} \right] \right) \right].$$

Proof. From Equation 1,

$$T_1 = Me^{R/M}(e^{(\tau_1+\delta_2)/M} - 1)\frac{T_s}{\tau_1}$$

and

$$T_2 = Me^{R/M}(e^{(\tau_2+\delta_2)/M} - 1)\frac{T_s}{\tau_2}. \tag{3}$$

Due to Proposition 1, we know that $\tau_1 \geq \tau_2$ and $\tau_1 = \tau_2 + \gamma$, where $0 \leq \gamma < (M - \tau_1)$. We would like to know what values of γ satisfy the inequality $T_2 < T_1$.

$$(T_2 < T_1) \implies \left(\frac{T_1}{T_2} > 1 \right) \implies \left(\frac{Me^{R/M}(e^{(\tau_1+\delta_2)/M} - 1)\frac{T_s}{\tau_1}}{Me^{R/M}(e^{(\tau_2+\delta_2)/M} - 1)\frac{T_s}{\tau_2}} > 1 \right). \tag{4}$$

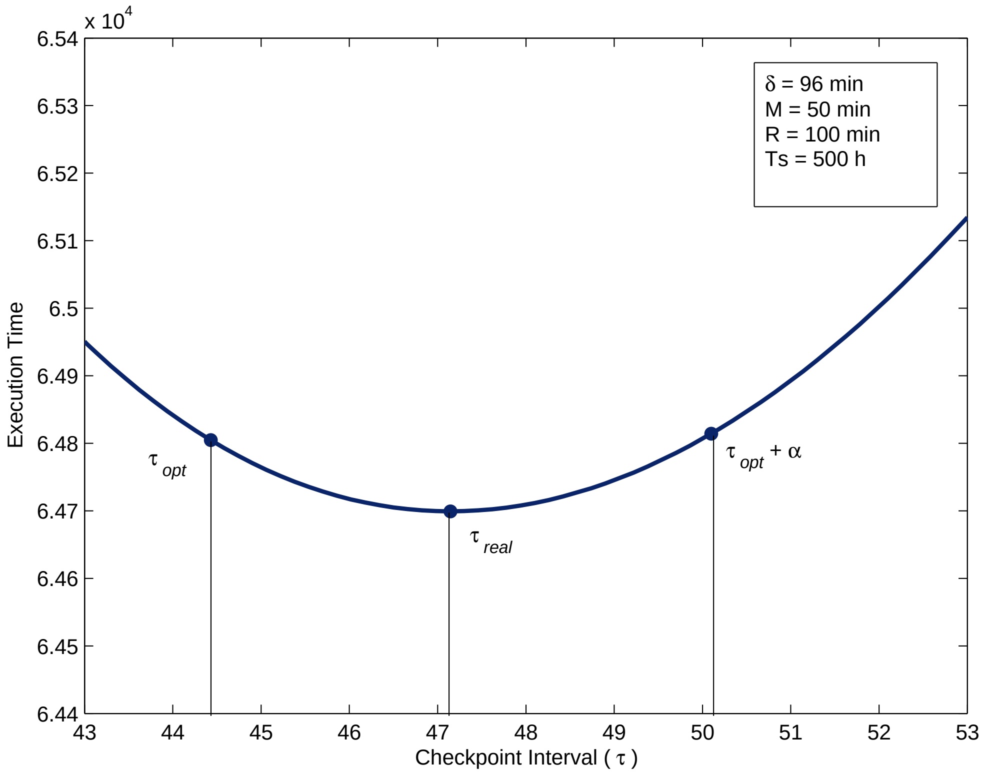

Figure 5. Example with $\alpha = 12.6\%$ of τ_{opt}

In the expression for T_1 in Inequality 4, substituting $\tau_2 + \gamma$ for τ_1 and simplifying, we obtain

$$\tau_2 e^{(\tau_2+\delta_2)/M}(e^{\gamma/M} - 1) > \gamma(e^{(\tau_2+\delta_2)/M} - 1). \tag{5}$$

The Taylor series expansion for $e^{\gamma/M}$ is

$$e^{\gamma/M} = 1 + \frac{1}{1!} \cdot \frac{\gamma}{M} + \frac{1}{2!} \cdot \frac{\gamma^2}{M^2} + \frac{1}{3!} \cdot \frac{\gamma^3}{M^3} + \dots$$

Substituting this series for $e^{\gamma/M}$ in Inequality 5 we get

$$\tau_2 e^{(\tau_2+\delta_2)/M} \left(\frac{1}{1!} \cdot \frac{\gamma}{M} + \frac{1}{2!} \cdot \frac{\gamma^2}{M^2} + \frac{1}{3!} \cdot \frac{\gamma^3}{M^3} + \dots \right) > \gamma \left(e^{(\tau_2+\delta_2)/M} - 1 \right) \tag{6}$$

Since $0 \leq \gamma < (M - \tau_1)$ and $\frac{\gamma}{M} < 1$, the Taylor's series expansion of $e^{\gamma/M}$ converges and is bounded by $\frac{M}{M-\gamma}$. Considering terms up to the quadratic term in the Taylor's series and ignoring higher order terms and simplifying, we find values of γ that satisfy $T_2 < T_1$ are given by

$$\gamma > \frac{2M}{\tau_2} \left[(M - \tau_2) - Me^{-(\tau_2+\delta_2)/M} \right] \tag{7}$$

Thus,

$$(T_2 < T_1) \quad If \quad \left[\gamma > \left(\frac{2M}{\tau_2} \left[(M - \tau_2) - Me^{-(\tau_2+\delta_2)/M} \right] \right) \right].$$

□

In obtaining the expression for γ in the above theorem, a truncation error is introduced becasue we consider the first three terms of taylors series expansion and ignore the rest of the terms. The consequent error in the estimated value of α, is bounded by $((0.0024 * \gamma) + (0.0048 * M))$ as shown in the Appendix.

5.3 Comparison of the Two Estimates of α

So far, we have presented two methods of obtaining approximations of α both of which are known to be greater than the real value;

- *the 12% heuristic*
- the expression given by Theorem 1

For a few sample values of checkpoint latencies and MTTIT, Table 5.3 provides estimates of α obtained using the two methods. In most cases, *the 12% heuristic* yields higher values than the analytical expression. Considering that γ of Theorem 1 is an upper bound of α, it follows that the error in the estimate of *the 12% heuristic* is larger than the error when γ is used as an approximate value of α. In the table, there is one sample with a checkpoint latency of 96 minutes and the value of MTTI of 50 minutes. In this case, the *the 12% heuristic* yields a value of 5.33 and the value of γ is 5.75. This is the only case where *the 12% heuristic* yields a smaller value than γ. In this case, the value of α is 5.8.

Our observation, based on empirical evidence from a larger sample, is that in almost all cases *the 12% heuristic* has a higher error of approximation than γ. The only cases where the approximation error of the *the 12% heuristic* is likely to be small is for values of checkpoint latency less than $2M$ and extremely close to it.

Checkpoint Latency	MTTI	Optimal Checkpoint Interval	γ	Estimate Using 12% Heuristic	Difference in Terms of % of γ
5	10	6.94	0.08	0.83	978
5	20	11.01	0.04	1.32	3345
5	50	19.15	0.02	2.3	15043
5	100	28.38	0.01	3.41	45056
6	3.5	3.10	0.32	0.37	15
10	25	16.19	0.12	1.94	1475
20	15	12.98	0.83	1.56	87
20	50	32.38	0.25	3.89	1475
25	40	29.61	0.48	3.55	633
96	50	44.43	5.75	5.33	-7
45	25	22.18	2.53	2.66	5
70	40	35.44	3.82	4.25	11.34
120	65	57.71	6.91	6.93	0.22

Table 1. Comparison of the two estimates of α, γ, and the $12\% heuristic$

6 Summary, Conclusions, and Future Work

Checkpointing puts a strain on the storage system and the interconnection network, and results in overheads on the application that severely impact performance and scalability. These are critical issues particularly in current and next-generation MPP systems.

Recent work in the area indicates a trend towards adaptation of checkpointing. Just as failure distributions and system conditions may vary over time, so do other factors critical to checkpoint latency – they may change either during an application's execution or among application executions. Such factors include the amount of data to checkpoint (consider mixing full and differential checkpointing) and the capacity of the memory hierarchy available for staging checkpoint data. Given these variables, our future work is focused on dynamically determining the checkpoint interval. This paper provides a foundation for work in that direction.

An important observation related to this work is that although error bound in computing the optimal checkpoint interval is nearly 6%, the resulting execution time is only 1.5% higher than the minimum execution time. So, in terms of just the execution times, the approximation error of the optimal checkpoint interval is not significant. However, we conjecture that the information obtained by the analysis presented in this report could be of significance when we consider other performance metrics of the application, such as network bandwidth consumption and I/O bandwidth consumption. Currently we are exploring this direction of research.

References

[1] Saurabh Agarwal, Rahul Garg, Meeta S. Gupta, and Jose E. Moreira. Adaptive incremental checkpointing for massively parallel systems. In *Proceedings of the 18th Annual International Confer-*

ence on Supercomputing, pages 277–286, New York, NY, 2004. ACM Press.

[2] P Cameron-Smith, J F Lamarque, P Connell, C Chuang, and F Vitt. Toward an earth system model: atmospheric chemistry, coupling, and petascale computing. *Journal of Physics: Conference Series*, 46:343–350, 2006.

[3] William J. Camp and James L. Tomkins. The red storm computer architecture and its implementation. In *The Conference on High-Speed Computing: LANL/LLNL/SNL*, Salishan Lodge, Glenedon Beach, Oregon, April 2003.

[4] John Daly. A model for predicting the optimum checkpoint interval for restart dumps. *Lecture Notes in Computer Science*, 2660:3–12, August 2003.

[5] John Daly. A strategy for running large scale applications based on a model that optimizes the checkpoint interval for restart dumps. In *Proceedings of the 26th International Conference on Software Engineering*, pages 70–74, Edinburgh, Scotland, UK, May 2004.

[6] John Daly. A higher order estimate of the optimum checkpoint interval for restart dumps. *Future Generation Computer Systems*, 22:303–312, 2006.

[7] E. N. Elnozahy, D. B. Johnson, and W. Zwaenepoel. The performance of consistent checkpointing. In *Proceedings of the 11th Symposium on Reliable Distributed Systems*, pages 39–47, Houston, TX, October 1992. IEEE Computer Society Press.

[8] Elmootazbellah N. Elnozahy and James S. Plank. Checkpointing for peta-scale systems: A look into the future of practical rollback-recovery. *IEEE Transactions on Dependable and Secure Computing*, 1(2):97–108, April–June 2004.

[9] Ada Gavrilovska, Karsten Schwan, Ola Nordstrom, and Hailemelekot Seifu. Network processors as building blocks in overlay networks. In *Proceedings of the 11 th Symposium on High Performance Interconnects (HOTI03)*, pages 83–88, August 2003.

[10] Kai Li, Jeffrey S. Naughton, and James S. Plank. Low-latency, concurrent checkpointing for parallel programs. *IEEE Transactions on Parallel and Distributed Systems*, 5(8):874–879, August 1994.

[11] Ron A. Oldfield. Investigating lightweight storage and overlay networks for fault tolerence. In *Proceedings of the High Availability and Performance Computing Workshop*, Santa Fe, NM, October 2006.

[12] Ron A. Oldfield, Sarala Arunagiri, Patricia J. Teller, Seetharami Seelam, Rolf Riesen, Maria Ruiz Varela, and Philip C. Roth. Modeling the impact of checkpoints on next-generation systems. In *Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies*, pages 30–43, San Diego, CA, September 2007.

[13] Ron A. Oldfield, Arthur B. Maccabe, Sarala Arunagiri, Todd Kordenbrock, Rolf Riesen, Lee Ward, and Patrick Widener. Lightweight I/O for scientific applications. In *Proceedings of the IEEE International Conference on Cluster Computing*, Barcelona, Spain, September 2006.

[14] Adam J. Oliner, Larry Rudolph, and Ramendra 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, Cairns, Queensland, Australia, 2006. ACM Press.

[15] Karthik Pattabiraman, Christopher Vick, and Alan Wood. Modeling coordinated checkpointing for large-scale supercomputers. In *Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN'05)*, pages 812–821, Washington, DC, 2005. IEEE Computer Society.

[16] James S. Plank. Improving the performance of coordinated checkpointers on networks of workstations using RAID techniques. In *Proceedings of the Symposium on Reliable Distributed Systems*, pages 76–85, 1996.

[17] James S. Plank, Youngbae Kim, and Jack J. Dongarra. Fault-tolerant matrix operations for networks of workstations using diskless checkpointing. *Journal of Parallel and Distributed Computing*, 43(2):125–138, June 1997.

[18] Ramendra K. Sahoo, Anand Sivasubramaniam, Mark S. Squillante, and Yanyong Zhang. Failure data analysis of a large-scale heterogeneous server environment. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN2004)*, pages 772–781, June 2004.

[19] Ramanan Sankaran, Evatt R. Hawkes, Jacqueline H. Chen, Tianfeng Lu, and Chung K. Law. Direct numerical simulations of turbulent lean premixed combustion. *Journal of Physics: Conference Series*, 46:38–42, 2006.

[20] Bianca Schroeder and Garth A. Gibson. A large-scale study of failures in high-performance computing systems. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN2006)*, Philadelphia, PA, June 2006. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

[21] Rajagopal Subramaniyan, R. Scott Studham, and Eric Grobelny. Optimization of checkpointing-related I/O for high-performance parallel and distributed computing. In *Proceedings of The International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 937–943, 2006.

[22] The BlueGene/L Team. An overview of the BlueGene/L supercomputer. In *Proceedings of SC2002: High Performance Networking and Computing*, Baltimore, MD, November 2002.

[23] Nitin H. Vaidya. A case for two-level distributed recovery schemes. *SIGMETRICS Perform. Eval. Rev.*, 23(1):64–73, 1995.

[24] Nitin H. Vaidya. Impact of checkpoint latency on overhead ratio of a checkpointing scheme. *IEEE Transactions on Computers*, 46(8):942–947, 1997.

[25] John W. Young. A first order approximation to the optimum checkpoint interval. *Communications of the ACM*, 17(9):530–531, 1974.

A Appendix

Theorem 2. *For any given value of $M \in \mathbb{R}^+$, the function $\tau_{opt}(\delta)$ is non-decreasing for $\delta \in \mathbb{R}^+$, where τ_{opt} is computed using Daly's function for the optimal checkpoint interval.*

Proof. Daly's analytical model for the optimal checkpoint interval, τ_{opt}, is a function of two variables, δ and M. Since the theorem states a property of τ_{opt} for a given value of M, we can consider M to be a constant and, thus, τ_{opt} becomes a function of a single variable, δ.

As shown below, Daly's model for the optimal checkpoint interval, τ_{opt}, is a piecewise-defined function in two disjoint intervals, interval $I1 = [0, 2M)$ and interval $I2 = [2M, \infty)$.

$$\tau_{opt}(\delta) = \begin{cases} \tau_{i1}(\delta) & 0 \leq \delta < 2M \\ \tau_{i2}(\delta) & \delta \geq 2M, \end{cases}$$

where τ_{i1} and τ_{i2} are defined as

$$\tau_{i1}(\delta) = \left(\sqrt{2M\delta}\left[1 + \frac{1}{3}\left(\frac{\delta}{2M}\right)^{1/2} + \frac{1}{9}\left(\frac{\delta}{2M}\right)\right] - \delta\right) \qquad 0 \leq \delta < 2M \tag{8}$$

$$\tau_{i2}(\delta) = M \qquad \delta \geq 2M. \tag{9}$$

It can be verified that $\tau_{opt}(\delta)$ is a non-decreasing function of δ if the following conditions are satisfied simultaneously:

1. $\tau_{i1}(\delta)$ is a non-decreasing function of δ.
2. $\tau_{i2}(\delta)$ is a non-decreasing function of δ.
3. $\tau_{i2}(2M)$, the value of $\tau_{i2}(\delta)$ evaluated at $\delta = 2M$, is an upper bound of τ_{i1}.

We use simple calculus to prove that $\tau_{i1}(\delta)$ and $\tau_{i2}(\delta)$ are non-decreasing functions of δ.

Claim A.1. $\tau_{i1}(\delta)$ is a non-decreasing function of δ.

Proof. This can be proven by showing that $\frac{d(\tau_{i1})}{d\delta} \geq 0$. From Equation 8,

$$\frac{d(\tau_{i1})}{d\delta} = \frac{d}{d\delta}\left(\sqrt{2M\delta}\left[1 + \frac{1}{3}\left(\frac{\delta}{2M}\right)^{1/2} + \frac{1}{9}\left(\frac{\delta}{2M}\right)\right] - \delta\right).$$

Expanding the numerator and applying the derivative, we get

$$\frac{d(\tau_{i1})}{d\delta} = \frac{d}{d\delta}\left(\sqrt{2M\delta} + \frac{1}{9}\delta\sqrt{\frac{\delta}{2M}} - \frac{2}{3}\delta\right) = \frac{1}{2}\sqrt{\frac{2M}{\delta}} + \frac{3}{18}\sqrt{\frac{\delta}{2M}} - \frac{2}{3}. \tag{10}$$

Since $(\delta < 2M) \implies \left(\sqrt{\frac{2M}{\delta}} > 1\right)$, we simplify Equation 10 by letting $\sqrt{\frac{2M}{\delta}} = 1 + \epsilon$, where $\epsilon > 0$. Accordingly, Equation 10 becomes

$$\begin{aligned}
\frac{d(\tau_{i1})}{d\delta} &= \frac{1}{2}(1+\epsilon) + \frac{1}{6(1+\epsilon)} - \frac{2}{3} \\
&= \frac{1}{2} + \frac{\epsilon}{2} + \frac{1}{6(1+\epsilon)} - \frac{2}{3} \\
&= \frac{\epsilon}{2} + \frac{1}{6(1+\epsilon)} - \frac{1}{6} \\
&= \frac{3\epsilon(1+\epsilon) + 1 - (1+\epsilon)}{6(1+\epsilon)} \\
&= \frac{3\epsilon^2 + 2\epsilon}{6(1+\epsilon)}.
\end{aligned}$$

Since $\epsilon > 0$ and there are no other negative values in the equation, $\frac{d(\tau_{i1})}{d\delta} > 0$ and, thus, $\tau_{i1}(\delta)$ is a non-decreasing function of δ. $\square$

Claim A.2. $\tau_{i2}(\delta)$ is a non-decreasing function of δ.

Proof.

$$\frac{d(\tau_{i2})}{d\delta} = \frac{d(M)}{d\delta} = 0.$$

Accordingly, $\tau_{i2}(\delta)$ is a non-decreasing function of δ. $\square$

Claim A.3. $\tau_{i2}(2M)$ is an upper bound of τ_{i1}. Accordingly,

Proof. Claim A.1 proves that τ_{i1} is an increasing function of δ. $\tau_{i1}(\delta)$ is defined for δ in the semi-open interval $[0, 2M)$. This implies that the value obtained by extrapolating $\tau_{i1}(\delta)$ to $\delta = 2M$ is an upper bound of τ_{i1}. The extrapolated value is

$$\sqrt{2M * 2M}\left[1 + \frac{1}{3} + \frac{1}{9}\right] - 2M = \frac{8M}{9}.$$

Since $\frac{8M}{9} < M$ and $\tau_{i2}(2M) = M$, $\tau_{i2}(2M)$ is an upper bound of τ_{i1}. $\square$

The above three proven claims together imply that $\tau_{opt}(\delta)$ is a non-decreasing function of δ for any given value of M.

$\square$

Theorem 3. *For every value of MTTI such that $M > 0$ and every value of checkpoint latency, δ, such that ($0.01 \leq \frac{\delta}{2M} < 1$), the value of the optimal checkpoint interval, computed using Equation 2, is less than or equal to the real optimal checkpoint interval.*

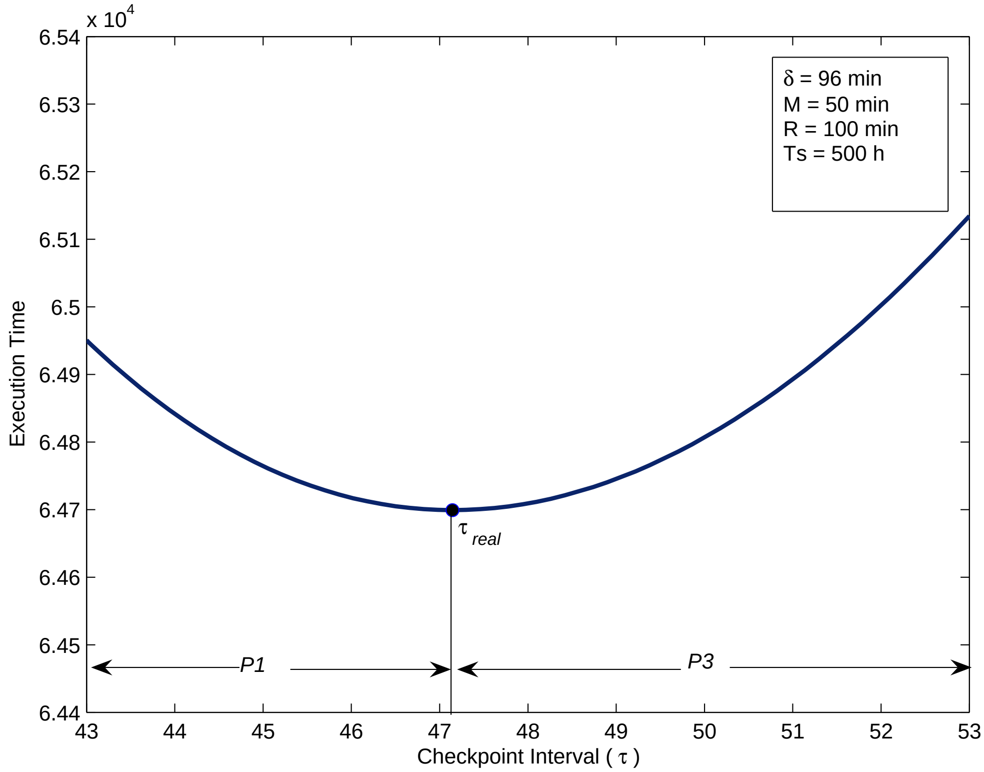

Figure 6. Illustration of a partition

Proof. For ease of representation in the context of this proof, let us define an ordered pair of values $< V1, V2 >$ as a *valid pair* if $0.02 * V1 \leq V2 < 2 * V1$ Given a valid pair of values of MTTI, M, and checkpoint latency, $/delta$, $< M, /delta >$, let us denote the optimal checkpoint interval computed using Equation 2 by τ_{opt} Let τ_{real} represent the real optimal checkpoint interval. Since $\tau_{opt} < M$, it belongs to one of the following partitions;

$$\begin{aligned} P1 &= \{\tau : 0 < \tau < \tau_{real}\} \\ P2 &= \{\tau : \tau = \tau_{real}\} \\ P3 &= \{\tau : \tau_{real} < \tau \leq M\}. \end{aligned}$$

The theorem asserts that $\tau_{opt} \in \{P1 \cup P2\}$

If $\tau_{opt} = \tau_{real}$ then $\tau_{opt} \in \{P2\}$ and the theorem holds. Therefore, we only need to prove the theorem for $\tau_{opt} \neq \tau_{real}$.

Suppose for a value of MTTI, M, and for a value of checkpoint latency, δ, we exhibit a $\triangle t > 0$ such that $[T(\tau_{opt} + \triangle t) < T(\tau_{opt})]$. The existence of such a $\triangle t$, proves that $\tau_{opt} \notin \{P3\}$, which implies that $\tau_{opt} \in \{P1 \cup P2\}$. This proves that for that pair of values of MTTI and checkpoint latency, $\tau_{opt} \in \{P1 \cup P2\}$. If we can provide a method of computing such a $\triangle t$ for any given valid pair of values of MTTI and checkpoint latency, $< M, \delta >$, then, that serves as a proof of Theorem 3. This is indeed the method we use.

In the following lemma, an alternate representation of τ_{opt} is obtained by a simple transformation of δ. It simplifies the proof of the theorem and improves readability.

Lemma 3.1. If δ is represented as $2Ma$, where $0 \leq a < 1$, then, $\tau_{opt} = 2M(B - a)$, where $B = \left(\sqrt{a} + \frac{a}{3} + \frac{a^{\frac{3}{2}}}{9}\right)$.

Proof. For this theorem, the range of values of δ of interest is $0 < \delta < 2M$. Consider the equation for τ_{opt} when $\delta < 2M$,

$$\tau_{opt} = \sqrt{2\delta M}\left[1 + \frac{1}{3}\left(\frac{\delta}{2M}\right)^{\frac{1}{2}} + \frac{1}{9}\left(\frac{\delta}{2M}\right)\right] - \delta.$$

Substituting $\delta = 2Ma$ in the above equation, we obtain

$$\begin{aligned}
\tau_{opt} &= \sqrt{4M^2a} + \frac{2Ma}{3} + \frac{2Ma^{\frac{3}{2}}}{9} - 2Ma \\
&= 2M\sqrt{a} + \frac{2Ma}{3} + \frac{2Ma^{\frac{3}{2}}}{9} - 2Ma \\
&= 2M\left(\sqrt{a} + \frac{a}{3} + \frac{a^{\frac{3}{2}}}{9}\right) - 2Ma \\
&= 2M(B - a) \; where \; B = \left(\sqrt{a} + \frac{a}{3} + \frac{a^{\frac{3}{2}}}{9}\right).
\end{aligned}$$

□

Since $\delta = 2Ma$, $(0.01 \leq \frac{\delta}{2M} < 1) \implies (0.01 \leq a < 1)$.

Example 1. For values of parameters ($M = 1\,min$, $R = 20\,min$, $T_s = 500\,hrs$), and $(0.01 \leq a < 1)$, Figures 7 and 8 are plots of $[T(\tau_{opt}) - T(\tau_{opt} + 10^{-5})]$ and $[log(T(\tau_{opt}) - T(\tau_{opt} + 10^{-5}))]$, respectively, as a function of a. In the range $(0.01 \leq a < 1)$, note that the difference is always positive. This demonstrates that when $M = 1\,min$ and for all values of a in the range $(0.01 \leq a < 1)$,

$$T(\tau_{opt}) > T(\tau_{opt} + 10^{-5}).$$

Thus, for $M = 1\,min$ and for all a such that $(0.01 \leq a < 1)$, we have exhibited a $\triangle t = 10^{-5}$ such that $[T(\tau_{opt}) > T(\tau_{opt} + \triangle t)]$. In order to prove the theorem, we still need to exhibit such a $\triangle t$ for every value of M and every a in the range $(0.01 \leq a < 1)$. The following lemma facilitates this.

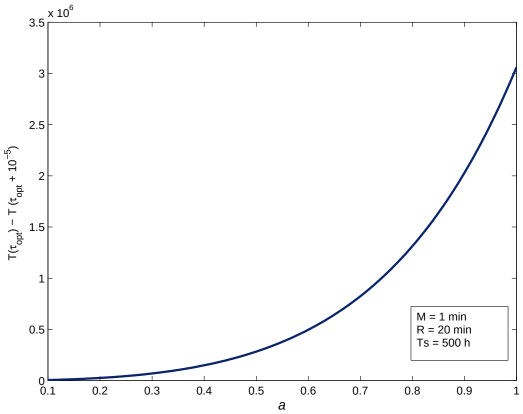

Figure 7. Exhibiting a $\triangle t$ for $M = 1\, min$

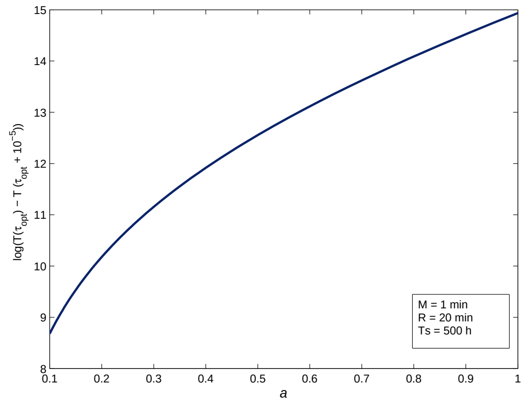

Figure 8. Difference in Execution times plotted on log-lin scale

Lemma 3.2. For a given value of MTTI, M, let $\triangle t$ satisfy the property that for every δ such that $(0.01 \leq \frac{\delta}{2M} < 1)$, the expected execution time corresponding to $[\tau_{opt} + \triangle t]$ is less than the expected execution time corresponding to τ_{opt}. For any other value of MTTI, $M' = f * M$, where $f > 0$, $\triangle t' = f * \triangle t$ satisfies the property that for every δ such that $(0.01 \leq \frac{\delta}{2M} < 1)$, the expected execution time corresponding to $[\tau_{opt} + (\triangle t')]$ is less than the expected execution time corresponding to τ_{opt}.

Proof.

$$Me^{R/M}T_s\left(\frac{e^{(\tau_{opt}+\delta)/M}-1}{\tau_{opt}}\right) > Me^{R/M}T_s\left(\frac{e^{((\tau_{opt}+\delta)/M+\triangle t/M)}-1}{\tau_{opt}+\triangle t}\right)$$

$$\Leftrightarrow \quad \left(\frac{e^{(\tau_{opt}+\delta)/M}-1}{\tau_{opt}}\right) > \left(\frac{e^{((\tau_{opt}+\delta)/M+\triangle t/M)}-1}{\tau_{opt}+\triangle t}\right) \;\; since\ M, e^{R/M},\ and\, T_s > 0$$

From Lemma 3.1, τ_{opt} can be expressed as $2M(B-a)$ and $\frac{\tau_{opt}+\delta}{M} = 2B$, where a and B are defined as in Lemma 3.1.

$$\left(\frac{e^{(\tau_{opt}+\delta)/M}-1}{\tau_{opt}}\right) > \left(\frac{e^{((\tau_{opt}+\delta)/M+\triangle t/M)}-1}{\tau_{opt}+\triangle t}\right)$$

$$\Leftrightarrow \quad \left(\frac{e^{2B}-1}{2M(B-a)}\right) > \left(\frac{e^{2B+\triangle t/M)}-1}{2M(B-a)+\triangle t}\right)$$

$$\Leftrightarrow \quad \left(\frac{e^{2B}-1}{2M(B-a)}\right) > \left(\frac{e^{(2B+t_1)}-1}{2M(B-a)+M*t_1}\right) \;\; where\ t_1 = (\triangle t/M)$$

$$\Leftrightarrow \quad \left(\frac{e^{2B}-1}{2(B-a)}\right) > \left(\frac{e^{(2B+t_1)}-1}{2(B-a)+t_1}\right) \tag{11}$$

Inequality 11 is equivalent to the condition that $(T(\tau_{opt}) - T(\tau_{opt} + (\triangle t))) > 0$ Note that, whether or not Inequality 11 is satisfied, is determined by the values of a and t_1. It can be verified that, if the value of MTTI changes from M to $f * M$, and if $\triangle t$ is substituted by $f * \triangle t$, then the condition $(T(\tau_{opt}) - T(\tau_{opt} + (f * \triangle t))) > 0$ turns out to be identical to Inequality 11.

□

Example 1 demonstrates that when $M = 1\,min$, a value of $\triangle t = 10^{-5}\,min$ satisfies Inequality 11 for all values of a in the range $0.01 \leq a < 1$. From Lemma 3.2, given any other value of MTTI, $M' = f*M$, where $f > 0$ a value of $\triangle t'$ that satisfies $[T(\tau_{opt}) > T(\tau_{opt} + \triangle t')]$ is given by $\triangle t' = \triangle t * f$. Since $M = 1\,min$, $f = M'/M = M'$, where M' is expressed in minutes.

Thus, we have presented a method of computing $\triangle t$ with the desired property for every $M > 0$ and for every δ in the corresponding relevant range. Therefore, as explained before, $\tau_{opt} \in \{P1 \cup P2\}$ is always true.

□

Figure 4 shows τ_2 and an α such that the execution time at checkpoint interval τ_2 and at $\tau_2 + \alpha$ are the same. Given any value of checkpoint interval $\tau < \tau_{real}$, one can find a corresponsing value of $\alpha = \alpha_\tau$ such that the execution time with the checkpoint interval τ is the same as the execution time with checkpoint interval $\tau + \alpha_\tau$. The process of finding an expression for such an α_τ would be similar to the derivation of γ in the proof of Theorem 1. It would be the solution of the following equation instead of Inequality 7.

$$\begin{aligned}\frac{1}{\alpha_\tau} \cdot (e^{\alpha_\tau/M} - 1) &= \left((\frac{1}{\tau * e^{(\tau+\delta)/M})} * (e^{(\tau+\delta)/M} - 1) \right) \\ &= \left(\frac{1}{\tau} * (1 - e^{-(\tau+\delta)/M}) \right)\end{aligned}$$

Solving this for α_τ gives;

$$\alpha_\tau = \left(\frac{2M}{\tau}\left[(M-\tau) - Me^{-(\tau+\delta)/M}\right]\right)$$

In the context of our research we are interested in values of α_τ such that

$$\alpha_\tau + \tau \leq M$$

Consider $\left(\frac{1}{\alpha_\tau} \cdot (e^{\alpha_\tau/M} - 1)\right)$. Using Taylors series expansion of $e^{(\alpha_\tau/M)}$,

$$\begin{aligned}
\frac{1}{\alpha_\tau} \cdot (e^{\alpha_\tau/M} - 1) &= \frac{1}{\alpha_\tau} \cdot \left(\cdot\frac{\alpha_\tau}{M} + \frac{1}{2!} \cdot \frac{\alpha_\tau^2}{M^2} + \frac{1}{3!} \cdot \frac{\alpha_\tau^3}{M^3} + \frac{1}{4!} \cdot \frac{\alpha_\tau^4}{M^4} + \frac{1}{5!} \cdot \frac{\alpha_\tau^5}{M^5} + \dots\right) \\
&= \left[\frac{1}{(\alpha_\tau)} \cdot \left(\cdot\frac{\alpha_\tau}{M} + \frac{1}{2!} \cdot \frac{\alpha_\tau^2}{M^2}\right)\right] + \left[\frac{1}{\alpha_\tau}\left(\frac{1}{3!} \cdot \frac{\alpha_\tau^3}{M^3} + \frac{1}{4!} \cdot \frac{\alpha_\tau^4}{M^4} + \frac{1}{5!} \cdot \frac{\alpha_\tau^5}{M^5} + \dots\right)\right] \\
&= ST_{considered} + ST_{ignored}
\end{aligned}$$

Where

$$\begin{aligned}
ST_{considered} &= \left[\frac{1}{(\alpha_\tau)} \cdot \left(\cdot\frac{\alpha_\tau}{M} + \frac{1}{2!} \cdot \frac{\alpha_\tau^2}{M^2}\right)\right] \\
ST_{ignored} &= \left[\frac{1}{\alpha_\tau}\left(\frac{1}{3!} \cdot \frac{\alpha_\tau^3}{M^3} + \frac{1}{4!} \cdot \frac{\alpha_\tau^4}{M^4} + \frac{1}{5!} \cdot \frac{\alpha_\tau^5}{M^5} + \dots\right)\right]
\end{aligned}$$

In deriving a closed form solution for α_τ, using the method similar to the one in the proof of theorem 1, we consider upto quadratic terms and ignore all the higher order terms. Therefore, $ST_{considered}$ and $ST_{ignored}$ represent the terms considered and terms ignored, respectively.

Proposition 3. *The fraction of* $((\frac{1}{\alpha_\tau}) * (e^{\alpha_\tau/M} - 1))$ *that is ignored is less than* $(\frac{1}{3!} \cdot \frac{\alpha_\tau^2}{M^2})$*, i.e.,*

$$ST_{ignored} < \left[\frac{1}{3!} \cdot \frac{\alpha_\tau^2}{M^2}\right] * \left[\frac{1}{\alpha_\tau} \cdot (e^{\alpha_\tau/M} - 1)\right]$$

Proof.

$$\begin{aligned}
ST_{ignored} &= \frac{1}{\alpha_\tau}\left(\frac{1}{3!} \cdot \frac{\alpha_\tau^3}{M^3} + \frac{1}{4!} \cdot \frac{\alpha_\tau^4}{M^4} + \frac{1}{5!} \cdot \frac{\alpha_\tau^5}{M^5} + \dots\right) \\
&= \frac{1}{3!} \cdot \frac{\alpha_\tau^2}{M^2}\frac{1}{\alpha_\tau}\left(\frac{\alpha_\tau}{M} + \frac{1}{4} \cdot \frac{\alpha_\tau^2}{M^2} + \frac{1}{5*4} \cdot \frac{\alpha_\tau^3}{M^3} + \frac{1}{6*5*4} \cdot \frac{\alpha_\tau^4}{M^4} + \dots\right) \\
&= \frac{1}{3!} \cdot \frac{\alpha_\tau^2}{M^2}\frac{1}{\alpha_\tau}\left(\frac{\alpha_\tau}{M} + \frac{1}{4} \cdot \frac{\alpha_\tau^2}{M^2} + \frac{1}{5*4} \cdot \frac{\alpha_\tau^3}{M^3} + \frac{1}{6*5*4} \cdot \frac{\alpha_\tau^4}{M^4} + \dots\right) \\
&< \left(\frac{1}{3!} \cdot \frac{\alpha_\tau^2}{M^2}\right) \cdot \left(\frac{1}{\alpha_\tau} \cdot (e^{\alpha_\tau/M} - 1)\right)
\end{aligned}$$

□

When $\tau = \tau_{opt}$, we know that the real value of α corresponding to τ_{opt}, denoted by $\alpha_{opt-real}$, is no larger than 12% of τ_{opt} and that $\tau_{opt} \leq M$. Therefore, $\alpha_{opt-real}$ is no greater than 12% of M and the fraction of ignored terms is less than $\frac{0.12*0.12}{6} = 0.0024$. As stated earlier, the closed form expression thus derived is an approximation and is always larger than $\alpha_{opt-real}$ and therefore serves as an upper bound of $\alpha_{opt-real}$. Let us denote this upper bound by α_{opt-ub}. The next prposition gives a lower bound of $\alpha_{opt-real}$, α_{opt-lb}.

Proposition 4. *A lower bound on the value of* $\alpha_{opt-real}$ *is given by* $\alpha_{opt-lb} = (0.9976 * \alpha_{opt-ub}) - (0.0048 * M)$.

Proof. As we already know, $\alpha_{opt-real}$ is a solution of the following equation;

$$\left(\frac{1}{\alpha_{opt-real}} \cdot (e^{\alpha_{opt-real}/M} - 1)\right) = \left(\frac{1}{\tau_{opt}} * (1 - e^{-(\tau_{opt}+\delta)/M})\right)$$

$$\frac{1}{\alpha_{opt-real}}\left(\frac{1}{1!} \cdot \frac{\alpha_{opt-real}}{M} + \frac{1}{2!} \cdot \frac{\alpha_{opt-real}^2}{M^2} + \frac{1}{3!} \cdot \frac{\alpha_{opt-real}^3}{M^3} + \dots\right) = \left(\frac{1}{\tau_{opt}} * (1 - e^{-(\tau_{opt}+\delta)/M})\right)$$

This equation implies the following inequality due to Proposition 3;

$$\left(\frac{1}{M} + \frac{1}{2} \cdot \frac{\alpha_{opt-real}}{M^2}\right) + \left[\left(\frac{1}{3!} \cdot \frac{\alpha_{opt-real}^2}{M^2}\right) \cdot \left(\frac{1}{\alpha_{opt-real}} \cdot (e^{\frac{\alpha_{opt-real}}{M}} - 1)\right)\right] > \left(\frac{1}{\tau_{opt}} * (1 - e^{-\frac{(\tau_{opt}+\delta)}{M}})\right) \tag{12}$$

We also know that,

$$\begin{aligned}\left(\frac{1}{3!} \cdot \frac{\alpha_{opt-real}^2}{M^2} \cdot \frac{1}{\alpha_{opt-real}} \cdot (e^{(\alpha_{opt-real}/M)} - 1)\right) &< (0.0024)\left(\frac{1}{\alpha_{opt-real}} \cdot (e^{\alpha_{opt-real}/M} - 1)\right) \\ &< (0.0024)\left(\frac{1}{\tau_{opt}} * (1 - e^{-(\tau_{opt}+\delta)/M})\right)\end{aligned}$$

Therefore, Inequality 12 becomes

$$\left(\frac{1}{M} + \frac{1}{2} \cdot \frac{\alpha_{opt-real}}{M^2}\right) + \left((0.0024) * (\frac{1}{\tau_{opt}}) * (1 - e^{-(\tau_{opt}+\delta)/M})\right) > \left(\frac{1}{\tau_{opt}} * (1 - e^{-(\tau_{opt}+\delta)/M})\right)$$

$$\left(\frac{1}{M} + \frac{1}{2} \cdot \frac{\alpha_{opt-real}}{M^2}\right) > (0.9976) * \left(\frac{1}{\tau_{opt}} * (1 - e^{-(\tau_{opt}+\delta)/M})\right)$$

We have established that $\alpha_{opt-real}$ satisfies the above inequality. A value of α that satisfies the inequality with equality is therefore a lower bound of $\alpha_{opt-real}$, α_{opt-lb}. Thus,

$$\left(\frac{1}{M} + \frac{1}{2} \cdot \frac{\alpha_{opt-lb}}{M^2}\right) = (0.9976) * \left(\frac{1}{\tau_{opt}} * (1 - e^{-(\tau_{opt}+\delta)/M})\right)$$

$$
\begin{aligned}
\alpha_{opt-lb} &= \frac{2M}{\tau_{opt}}\left(0.9976M(1-e^{-(\frac{\tau_{opt}+\delta}{M})}) - \tau_{opt}\right) \\
&= \frac{2M}{\tau_{opt}}\left(0.9976M(1-e^{-(\frac{\tau_{opt}+\delta}{M})}) - 0.9976\tau_{opt} - 0.0024\tau_{opt}\right) \\
&= 0.9976 * \left[\frac{2M}{\tau_{opt}}\left(1-e^{-(\frac{\tau_{opt}+\delta}{M})}\right) - \tau_{opt}\right] - 0.0024 * \frac{2M}{\tau_{opt}} * \tau_{opt} \\
&= ((0.9976 * \alpha_{opt-ub}) - (0.0048 * M))
\end{aligned}
$$

□

Proposition 5.

$$
\begin{aligned}
|\alpha_{opt-real} - \alpha_{opt-lb}| &< ((0.0024 * \alpha_{opt-ub}) + (0.0048 * M)) \\
|\alpha_{opt-real} - \alpha_{opt-ub}| &< ((0.0024 * \alpha_{opt-ub}) + (0.0048 * M))
\end{aligned}
$$

Proof. Due to Proposition 3

$$
\alpha_{opt-lb} = (0.9976 * \alpha_{opt-ub}) - (0.0048 * M) \implies
$$

$$
\alpha_{opt-ub} - \alpha_{opt-lb} \leq ((0.0024 * \alpha_{opt-ub}) + (0.0048 * M))
$$

Thus, we have bounded $\alpha_{opt-real}$ from above and below and we know that it lies in an interval of length $((0.0024 * \alpha_{opt-ub}) + (0.0048 * M))$. Therefore,

$$
\begin{aligned}
|\alpha_{opt-real} - \alpha_{opt-lb}| &< ((0.0024 * \alpha_{opt-ub}) + (0.0048 * M)) \\
|\alpha_{opt-real} - \alpha_{opt-ub}| &< ((0.0024 * \alpha_{opt-ub}) + (0.0048 * M))
\end{aligned}
$$

□