ISSN:0975-9646

Mallikarjuna Shastry P.M. et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , 2011, 2025-2032

A New Optimal Checkpoint Restart Model

Mallikarjuna Shastry P.M. #1, K. Venkatesh#2

#1Dept. of Computer Science and Engg., Sapthagiri College of Engg.,

#2Research Centre, M. S. Ramaiah Institute of Technology,

#Affiliated to Visvesvaraya Technological University,

Bangalore, Karnataka, India.,

***Abstract*— In this paper, we have developed a New Optimal Checkpoint Restart Model (NOCRM) by using Weibull's distribution and Poisson's process with Exponential distribution to obtain an optimal checkpoint interval to reduce the total time lost (which includes checkpoint cost, rollback cost and restart cost) due to checkpointing of MPI applications.**

The different parameters like scale parameter and the shape parameter, β of Weibull's distribution and the checkpoint cost of an application are required for developing our NOCRM. For Exponential distribution, we set β to 1.

We have shown that, the NOCRM developed by us for determining optimal checkpoint interval reduces a large amount of waste time in comparison to the models developed by Yudan Liu et al., Bouguerra Mohammed Slim et al. and Young.

The optimal checkpoint interval determined by our NOCRM using scale parameter, α and the shape parameter, β of Weibull's distribution and the checkpoint cost, T_S can be effectively used to provide the fault tolerance to any application. Our NOCRM leads to considerable reduction in waste time.

***Keywords*—Fault Tolerance, Checkpointing, Optimal Checkpoint Interval, Total Waste Time, Mean Time Between Failures.**

I. INTRODUCTION

By knowing the mean time between failures (MTBF) from historical data of the system, one can use the Poisson's process for the arrival of failures with exponential distribution for MTBF [1] or the Weibull's distribution for MTBF for the analysis of expectation value of the total time lost (total waste time) during the execution of an application [2, 3].

Fault tolerance is achieved by using one of the checkpointing protocols such as blocking coordinated checkpointing protocol in which an initiator takes a checkpoint by synchronizing with all other processes of an MPI application [4].

Checkpoints can be taken using either fixed checkpoint interval method or variable checkpoint interval method [5]. In case of fixed checkpoint interval method, the checkpoint interval size remains same between any two successive checkpoints and the checkpoint interval size need not be uniform between two successive checkpoints in incremental/variable checkpoint interval method [5].

II. LITERATURE SURVEY

A lot of research work has been carried out on

i) the development of a checkpoint/restart model to obtain optimal checkpoint interval using Poisson's process with Exponential distributions and Weibull's distribution and

ii) selection of fixed / variable checkpoint interval.

A. Poisson's Process with Exponential Distribution

Young [1] presented a model using first order approximation to obtain an optimal checkpoint interval using Poisson's process for the occurrence of failures. This model [1] shows that, the total waste time due to checkpointing can be reduced using fixed checkpoint interval.

A.J. Oliner et al. [6] presented an approach based on Young's [1] periodic checkpointing model. This model [6] reduces checkpoint cost but increases the rollback cost.

The model developed by Young [1] has been improved by J.T. Daly [7, 8] to reduce the checkpoint overheads using higher order approximation. The cost function developed by [7] was used to estimate the total wall clock time required for completion of the execution of an application and an optimal checkpoint interval was determined to reduce this total wall clock time. This cost function is a sum of solve time (execution time of application without checkpointing), dump time (time required to write a checkpoint onto a local disk), rework time (time required to rollback the application to most recent checkpoint after a failure) and restart time.

R.A. Oldfield et al. [9] proposed a refined analytical model that exhibits Poisson single component failures [7, 8, 10]. This model [9] has been incorporated into Daly's equation [7] to obtain an optimal checkpoint interval. In this model [9], average completion 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. This model [9] reduces the checkpoint overheads by using lightweight storage architectures and overlay networks.

N. Naksinehaboon et al. [11] used Poisson's process to develop an incremental checkpoint restart model in HPC environments. This model [11] takes a sequence of incremental checkpoints between any two full checkpoints. A full checkpoint saves the entire data section and the stack of the application. But, an incremental checkpoint saves only the address spaces that have changed since the previous checkpoint. The recovery cost is decided by the number of incremental checkpoints. After performing m incremental checkpoints, either another incremental checkpoint or a full checkpoint is performed. A full checkpoint is chosen if the cost of performing a full checkpoint is cheaper than the recovery cost for an incremental checkpoint. If a process fails after taking incremental checkpoints and before taking a full checkpoint, the rollback cost will be very high as the incremental checkpoints do not save the complete state of the running process.

The Markov process model, developed by R. Subramaniyan et al. [12] using Exponential distribution, uses a sustainable I/O bandwidth as a parameter to determine an optimal checkpoint interval.

In the failure process model developed using Poisson's distribution by M. Wu et. al. [13] proposed to use an M/G/1 to describe the failure of a system. In this model [13], the recovery time follows a general distribution.

The reliability aware optimal k node allocation algorithm proposed by R. Narasimha et al. [14, 15] minimizes the average completion time and waste time. The algorithm of [15] is based on the expected completion time of application and reliability of a system of k nodes.

The markov availability models developed using Exponential distribution by K. Wong et al. [16], J.S. Plank et al. [17] and R. Geist et al. [18], obtained an optimal checkpoint interval to maximize the availability of system.

B. Weibull's Distribution

Yudan liu et al. [2] presented an optimal checkpoint restart model using Weibull's distribution for obtaining an optimal checkpoint interval. A stochastic renewal reward process has been used to derive optimal checkpoint interval. They [2] use variable checkpoint interval method to checkpoint the MPI applications. In variable checkpoint interval method, the size of checkpoint interval varies from one checkpoint to another checkpoint.

The model of [2] also determines the rollback coefficient for each checkpoint interval. When the checkpointed application fails during its execution, the application is rolled back to the previous checkpoint and resumes its execution from there. The amount of time by which the application has to rollback is decided by the value of rollback coefficient depending upon the checkpoint interval size.

Bouguerra Mohammed Slim et al. [3] used both Weibull's and Poisson distributions to develop a new flexible checkpoint/restart model. This model [3] showed the improvement over the model developed by Yudan liu et al [2]. This model [3] determines the expected completion time by using Poisson's and Weibull's distribution laws of failures. This model [3] also uses the variable checkpoint interval method to checkpoint the MPI applications.

We have developed a model for obtaining optimal checkpoint interval by minimizing the total time lost (total waste time) which includes the overheads of checkpointing mechanism like checkpoint cost, rollback cost and restart cost.

We have compared the results obtained by our model with case studies of the models developed by Yudan liu et al [2], Bouguerra Mohammed Slim et al. [3] and Young [1], as we have also used Weibull's and Poisson\'s process with Exponential distribution for developing the New Optimal Checkpoint Restart Model (NOCRM).

C. Selection of Type of Checkpoint Interval

M.Shastry et al. [5] shown that the fixed checkpoint interval reduces the rollback cost and total waste time due to checkpointing of fault tolerant MPI applications in comparison to the incremental / variable checkpoint interval. Hence, we have used the fixed checkpoint interval in our further analysis.

III CHECKPOINTING AND RESTARTING OF MPI APPLICATIONS

One of the coordinated checkpointing protocols, such as Blocking coordinated checkpointing protocol is considered in the development of our model, NOCRM. This protocol works in two phases for checkpointing an MPI application [19].

When a failure occurs during the execution of a periodically checkpointed MPI application, the application rolls back to the most recent checkpoint and resumes the execution from there. So, recovering from a failure involves two steps. One is, the application has to be rolled back to the most recent checkpoint and second one is, restarting (resuming) the MPI application from the most recent checkpoint [4, 27-3].

A new process of checkpointing begins after the MPI application resumes its execution after each failure. The pictorial representation of checkpointing and restarting of an MPI application during its execution is shown in Fig. 1.

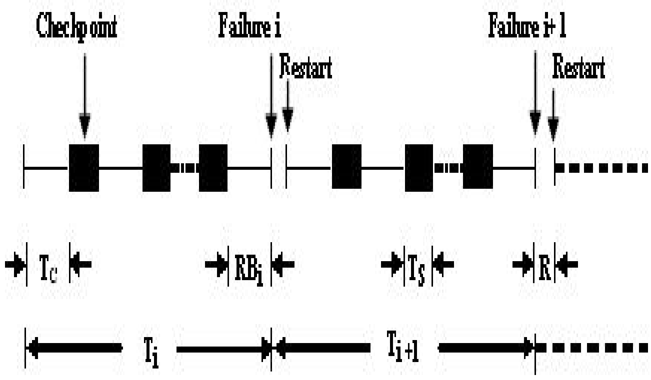

Fig. 1 Checkpointing and Restarting of MPI Application as a Stochastic System

IV ASSUMPTIONS MADE AND NOTATIONS USED IN NOCRM

A. Assumptions Made in NOCRM

We have made the following assumptions which are similar (except the 3rd assumption) to the assumptions made in [2] to develop our New Optimal Checkpoint Restart Model (NOCRM).

1. A series of F failures may interrupt the execution of MPI application.
2. A separate monitoring software system is used to monitor continuously the failure of a checkpointed MPI application.
3. Checkpoint interval T_C is fixed and each checkpoint is taken periodically after the time T_S.
4. When a failure occurs during the execution of MPI application, MPI application is rolled back to the most recent checkpoint and restarted from there.
5. Time required for writing a checkpoint, T_S, onto a local disk is a constant.
6. Time required for resuming / restarting (restart cost R) the MPI application from the most recent checkpoint is a constant and is considered to be negligible as has been discussed by J.T. Daly [7].

B. Notations Used in NOCRM

The notations used in this paper are presented in Table I.

TABLE I. NOTATIONS

Parameter	*Meaning*
T_C	Optimal Checkpoint Interval
T_S	Time required to save the checkpoint onto a local disk
β	Shape Parameter of Weibull's distribution
α	Scale Parameter of Weibull's distribution (MTBF)
T_i	Execution time till a failure occurs in i^{th} cycle
N_i	Number of checkpoints taken till a failure occurs in i^{th} cycle
RB_i	Rollback cost due to a failure in i^{th} cycle
CC_i	Checkpoint cost in i^{th} cycle
R	Restart cost (time required to resume the execution of the application after a failure)
TL_i	Total time lost in i^{th} cycle
F	Number of failures
TL_e	Total time lost due to F failures during the execution of MPI application in exponential distribution
TL	Total time lost (total waste time) due to *F* failures during the execution of an application
ACT	Average Completion Time of an Application
ET	Execution Time of an application without checkpointing
λ	Failure Rate

V. A NEW OPTIMAL CHECKPOINT RESTART MODEL (NOCRM)

A. Determination of Waste Time (time lost) Due to Checkpointing

If the fault tolerant application undergoes *F* failures, the execution of fault tolerant application will have *F* cycles. As we have considered a cycle to be the execution time interval of the application between two successive failures, $Ti = \alpha$ (MTBF).

The number of checkpoints to be taken in cycle, i before the occurrence of a failure can be determined [1] by

$$N_i = \lfloor T_i / (T_C + T_S) \rfloor \qquad (1)$$

Then, the cost of checkpoint in i^{th} cycle, is computed [1] as follows.

$$CC_i = N_i \, T_S \qquad (2)$$

The cost of rollback in i^{th} cycle, is then computed [1] as follows.

$$Rb_i = (T_i - N_i \, (T_C + T_S)) \qquad (3)$$

The time lost (waste time) in i^{th} cycle TL_i due to a failure [4, 59] can be obtained by adding checkpoint cost, rollback cost, restart cost together as follows.

$$TL_i = CC_i + Rb_i + R \qquad (4)$$

The time lost in i^{th} cycle can also be computed as follows by substituting Eq. (2) and Eq. (3) in Eq. (4).

$$TL_i = T_i - N_i \, T_C + R \qquad (5)$$

We put $t = T_i$ and $n = N_i$ for general variables in each cycle and integrate over all cycles to estimate the total time lost (total waste time).

B. Development of NOCRM Using Weibull's Distribution

In Eq. (5), we determine the total time lost (total waste time) by integrating over all the checkpoints taken during the execution of the entire application and also the time lost due to reruns caused by *F* failures as follows.

$$TL = \sum_{n=0}^{\infty} \int_{n\,(T_C+T_S)}^{(n+1)\,(T_C+T_S)} [t - n\,T_C] \; p(t) \, dt + R\,F \qquad (6)$$

Where, *R* is the restart cost which is a constant.

We have used Weibull's distribution for MTBF to estimate the total time lost (total waste time) in the Eq. (6).

Hence, we put

$$p(t) = (\beta/\alpha)\,(t/\alpha)^{(\beta-1)}\; e^{-(t/\alpha)^{\beta}} \qquad (7)$$

After substituting Eq. (7) for $p(t)$ in Eq. (6), we get

$$TL = \sum_{n=0}^{\infty} \int_{n\,(T_C+T_S)}^{(n+1)\,(T_C+T_S)} [t - n\,T_C] \; (\beta/\alpha) \; (t/\alpha)^{(\beta-1)}\; e^{-(t/\alpha)^{\beta}} \; dt + R\,F \qquad 8)$$

On further evaluation, we get

$$TL = \int_0^{\infty} t\;(\beta/\alpha)(t/\alpha)^{\beta-1} e^{-(t/\alpha)^{\beta}} \, dt \; - \; T_C \sum_{n=1}^{\infty} n \int_{n(T_C+T_S)}^{(n+1)\,(T_C+T_S)} (\beta/\alpha)(t/\alpha)^{(\beta-1)} e^{-(t/\alpha)^{\beta}} \, dt + RF \qquad (9)$$

The first term in the Eq. (9) is simply the expectation value of t, hence, the first term becomes

$$\alpha \;\; \Gamma\,(1 + 1/\beta) \qquad (10)$$

Second term in Eq. (9) is evaluated as follows:

1. We use the substitution, $(t/\alpha)^{\beta} = x$, and $dx = ((\beta\,(t)^{\,\beta-1})/\alpha^{\beta}) \; dt$ and obtain

$$T_C \left[\sum_{n=1}^{\infty} n \left\{ e^{-((n+1)(T_C+T_S)/\alpha)^{\beta}} \; - \; e^{-(n\,(T_C+T_S)/\alpha)^{\beta}} \right\} \right] \qquad (11)$$

2. Upon expansion and simplification of the Eq. (11), we get,

$$-\; T_C \left[\sum_{n=1}^{\infty} e^{(-n\,(T_C + T_S)/\alpha)^{\beta}} \right] \qquad (12\;)$$

After substituting the Eq. (10) and Eq. (12) in the Eq. (9) for first and second term respectively, we get the following equation for total time lost, TL.

$$TL = \alpha\,\Gamma(1+1/\beta) - T_C\left[\sum_{n=1}^{\infty} e^{(-n\,(T_C+T_S)/\alpha)^{\beta}}\right] + R\,F \qquad (13)$$

To obtain optimal value of T_C which minimizes TL, we differentiate TL w.r.t T_C and equate the result to zero. Since, the first and third terms in Eq. (13) are constants, they become zero upon differentiation.

i.e. $d\,(TL)\,/\,dT_C = 0$, which leads to the condition

$$-d/dT_C\left[T_C\sum_{n=1}^{\infty} e^{-k\,n^{\beta}}\right] = 0 \qquad (14)$$

Where, $K = ((T_C + T_S)/\alpha)^{\beta}$

Differentiating the term within the square bracket of Eq. (14) and equating it to zero, we get

$$\beta\,(T_C/\alpha)\,((T_C+T_S)/\alpha)^{\beta-1}\sum_{n=1}^{\infty} n^{\beta}\,e^{-k\,n^{\beta}} = \sum_{n=1}^{\infty} e^{-k\,n^{\beta}} \qquad (15)$$

We have used MATHEMATICA software to evaluate the left hand side (LHS) and right hand side (RHS) of the Eq. (15) for the given value of scale parameter α, shape parameter β of Weibull's distribution and the checkpoint cost T_S of an MPI application. We have developed the following algorithm to determine the optimal value of T_C using the Eq. (15) for the given inputs such as scale parameter α, shape parameter β of Weibull's distribution and the checkpoint cost T_S of an MPI application.

Algorithm: Estimate-T_C ( )

The variable ITC stores initial value of T_C and can be set in the range of (α - T_S) /10 to (α - T_S) /10. It means that atleast 10 to 30 checkpoints should be taken before the failure of an application. The variable FTC stores final value of T_C and is set to (α - T_S). This indicates that, final checkpoint can be taken just before the failure of the application as α is MTBF and T_S is the time required to save the checkpoint on a local disk. The variables X, Y1, and Y2 are floating point arrays. The array X stores the T_C values, array Y1 stores LHS values and Y2 stores RHS values.

1. Set i to zero.
2. Initialize T_C to ITC.
3. While (T_C <= FTC) do
   Begin
   a) Evaluate LHS and RHS of the Eq. (15) using MATHEMATICA software for the values of T_C, α, β and T_S.
   b) Set X[i] = T_C, Y1[i] = LHS and Y2[i] = RHS.
   c) Increment i by one.
   d) Increment T_C by ITC.
   End
4. Plot the graph G1 using (X, Y1) and the graph G2 using (X, Y2).
5. Determine the value of T_C from the point of intersection of G1 and G2 at which, the Eq. (15) is satisfied.

The first point of intersection of G1 and G2 is considered as the optimal value of T_C, if G1 and G2 interest at only one value of T_C. If G1 and G2 intersect at more than one point, the optimal checkpoint interval T_C is obtained by computing the total waste time (total time lost) for each value of T_C and choosing the T_C for which the waste time is minimum.

C. Development of NOCRM Using Poisson's Process with Exponential Distribution

The total waste time (total time lost) is evaluated for Exponential distribution, by putting β = 1 in the Eq. (13).

After substituting β = 1 in Eq. (13), we get

$$TL_e = \alpha - T_C\sum_{n=1}^{\infty} e^{-n\,K_e} + R\,F \qquad (16)$$

Where, $K_e = ((T_C+T_S)/\alpha)$ for exponential distribution for mean time between failures.

The second term in the Eq. (16) becomes

$$-T_C\left[e^{-K_e} + e^{-2K_e} + e^{-3K_e} + ..\right]$$

which is equivalent to

$$-T_C\left[e^{-K_e} + (e^{-K_e})^2 + (e^{-K_e})^3 + ..\right] \qquad (17)$$

The series in the square bracket of the Eq. (17) is a geometric series with the first term as e^{-ke} and the common ration also as e^{-ke} and it's sum up to infinity is equal to 1/ (e^{-ke} – 1), so that the equation (16) can be written as

$$TL_e = \alpha - T_C\,(1/(e^{(T_C+T_S)/\alpha} - 1)) + R\,F \qquad (18)$$

The Eq. (18) can also be written as

$$TL_e = \alpha + T_C\,(1/(1 - e^{(T_C+T_S)/\alpha})) + R\,F \qquad (19)$$

Where, α = (1/λ) (MTBF) and the Eq. (19) obtained by us is same as the Young's model for Exponential distribution [1] except the third term "*R F*" (Restart cost due to *F* failures). In [1], the above Eq. (19) is differentiated and equated to zero to obtain the condition for minimizing TL_e.

Further, the equation

$$T_C = \sqrt{2\,T_S\,T_F} \qquad (20)$$

Where, T_F = 1 / λ, is obtained in second order approximation [1].

However, we have used our algorithm Estimate T_C () to determine the optimal value of T_C from Eq. (15) for Exponential distribution with β = 1. The checkpoint cost, T_S, the scale parameter, α (MTBF) and the shape parameter, β are used as the inputs to the algorithm Estimate-T_C ().

We shall compare the results obtained by our NOCRM with Young's model in section 6.3.

The total waste time (total time lost), TL due to checkpointing of MPI applications is computed using optimal checkpoint interval obtained from our NOCRM as follows.

$$TL = \sum_{i=1}^{F} TL_i \qquad (21)$$

Where, the waste time in each cycle i, TL_i is computed using the Eq. (5). The average completion time of the application due to checkpointing is computed as follows.

$$ACT = ET + TL \qquad (22)$$

Where, ET, is the execution time (completion time) of the application.

VI RESULTS AND DISCUSSION

A. Comparison of our NOCRM with Yudan Liu's Model using Weibull's Distribution

In this case study, we have compared the results obtained by our NOCRM with the case studies discussed by Yudan liu et al. [2] using the shape parameter, β = 0.509, scale parameter, α = 1235 minutes (20.58 Hrs) of Weibull's Distribution (where α is MTBF). The total waste time is computed using the Eq. (21).

1). For Checkpoint Cost, T_S = 10 Minutes

When checkpoint cost, T_S is 10 minutes, shape parameter, β is 0.509, and scale parameter, α is 1235 minutes, we have obtained the optimal checkpoint interval, T_C as 400 minutes for Weibull's distribution using our algorithm Estimate T_C as shown in Fig. 2. Since, both LHS and RHS of the Eq. (15) tend to zero on $T_C \rightarrow \infty$, the first intersection of graphs G1 and G2 is considered for the optimal value of T_C as shown in Fig. 2.

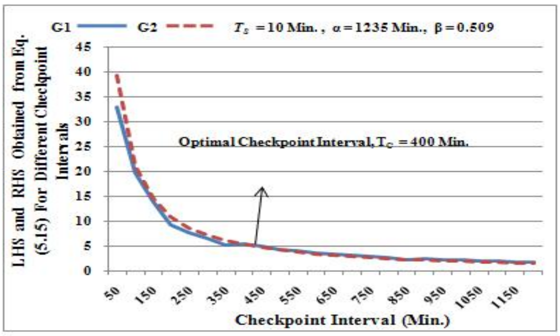

Fig. 2 Determination of Optimal Checkpoint Interval from NOCRM (T_S = 10 Min.)

The comparison of waste time produced by our NOCRM using optimal checkpoint interval of 400 minutes with the waste time produced by the model of [2] using the checkpoint interval of 60 minutes is shown in the Fig. 3. In Fig. 3, it clear that, the waste time from our NOCRM is significantly very less in comparison to the waste time of the model developed by [2].

As the requested checkpoint interval, T_C, used by [2] is 60 minutes, it is quite expected that the number of checkpoints to be taken will be more and hence, total checkpoint cost and the total waste time are very high. Our NOCRM saves an average of 72% of waste time due to checkpointing of MPI applications in comparison to the model [2].

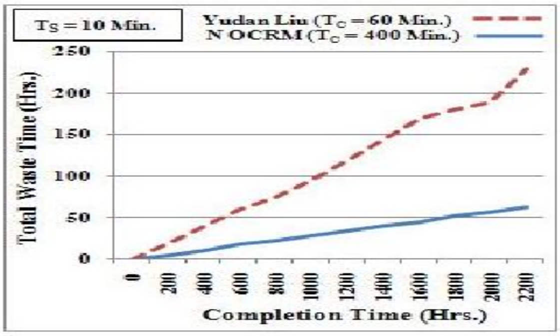

Fig. 3 Waste Time of Yudan Liu's Model and NOCRM (T_S = 10 Min.)

2). For Checkpoint Cost, T_S = 5 minutes

When checkpoint cost, $\boldsymbol{T_S}$ is 5 minutes, shape parameter, β is 0.509, and scale parameter, α is 1235 minutes, we have obtained the optimal checkpoint interval, *TC* as 200 minutes for Weibull's distribution using our algorithm Estimate $\boldsymbol{T_C}$ ( ) as shown in Fig. 4. Since, both LHS and RHS of the Eq. (15) tends to zero on $T_C \rightarrow \infty$, the first intersection of graphs G1 and G2 is considered for the optimal value of T_C as shown in Fig. 4.

The comparison of waste time produced by our NOCRM using optimal checkpoint interval of 200 minutes with the waste time produced by the model of [2] using the checkpoint interval of 60 minutes is shown in the Fig. 5. In Fig. 5, it is clear that, our NOCRM saves an average of 64% of waste time in comparison to the model [2].

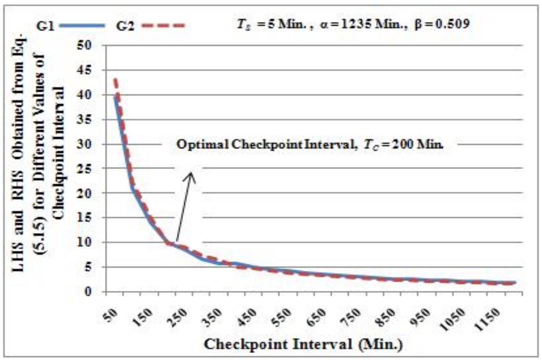

Fig. 4 Determination of Optimal Checkpoint Interval from NOCRM (T_S = 5 Min.)

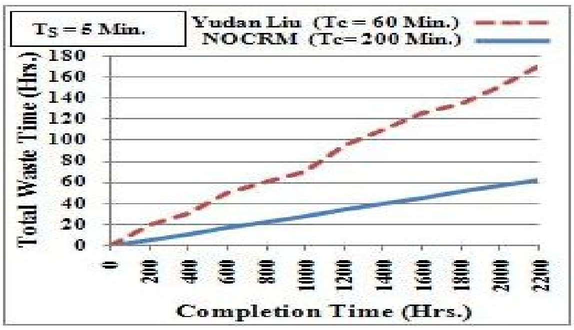

Fig. 5 Waste Time of NOCRM and Yudan Liu's Model (T_S = 5 Min.)

3). For Checkpoint Cost, T_S = 2 Minutes

When checkpoint cost, $\boldsymbol{T_S}$ is 2 minutes, shape parameter, β is 0.509, and scale parameter, α is 1235 minutes, we have obtained the optimal checkpoint interval, T_C as 110 minutes for Weibull's distribution using our algorithm Estimate $\boldsymbol{T_C}$ ( ).

The comparison of waste time produced by our NOCRM using optimal checkpoint interval of 110 minutes with the waste time produced by the model of [2] using the checkpoint interval of 120 minutes is shown in the Fig. 6.

In this case, the requested checkpoint interval of 120 minutes used by [2] is almost close to the value of optimal checkpoint interval, T_C, obtained by our NOCRM, the comparison clearly shows that, our NOCRM is advantageous over the model developed by [2]. Our NOCRM saves an average of 56% of waste time in comparison to the model [2] as shown in Fig. 6.

The comparison of waste time produced by our NOCRM using optimal checkpoint interval of 110 minutes with the waste time produced by the model of [2] using the checkpoint interval of 150 minutes is shown in the Fig. 7. In Fig. 7, it is clear that, our NOCRM saves an average of 55 % of waste time in comparison to the model [2].

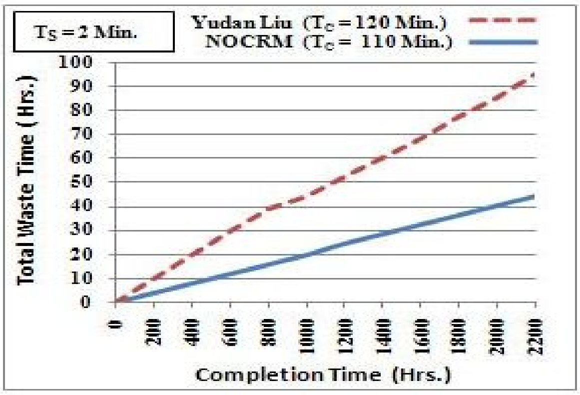

Fig. 6 Waste Time of Yudan Liu's Model With T_C = 120 Min. and NOCRM With T_C = 110 Min., (T_S = 2 Min.)

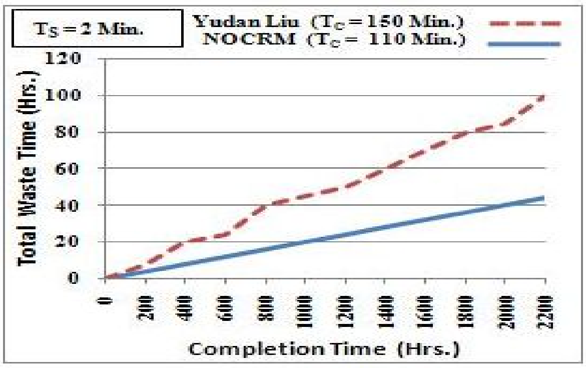

Fig. 7 Waste Time of Yudan Liu's Model with T_C = 150 Min. and NOCRM with T_C = 110 Min., (T_S = 2 Min.)

4). For Checkpoint Cost, T_S = 1 Minute

When checkpoint cost, $\boldsymbol{T_S}$ is 1 minute, shape parameter, β is 0.509, and scale parameter, α is 1235 minutes, we have obtained the optimal checkpoint interval, T_C as 75 minutes for Weibull's distribution using our algorithm Estimate - T_C ( ).

The comparison of our NOCRM using optimal checkpoint interval of 75 minutes with the model developed by [2] for the requested checkpoint interval of 60 minutes is shown in the Fig. 8. The comparison clearly shows that our NOCRM saves an average of 20% of waste time in comparison to the model [2] as shown in Fig. 8.

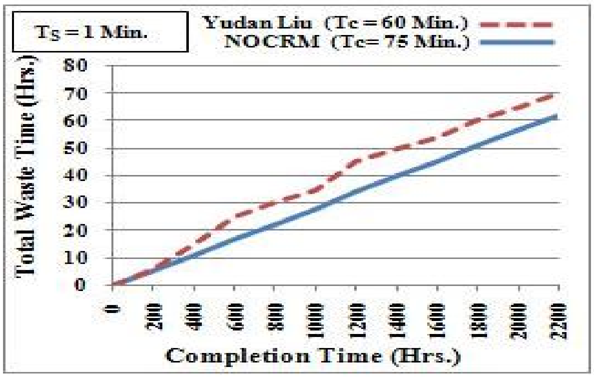

Fig. 8 Waste Time of NOCRM and Yudan Liu's Model (T_S = 1 Min.)

B. Comparison of NOCRM with FCM Using Poisson's Process with Exponential Distribution

In this case study, we have compared the results obtained by our NOCRM using the shape parameter, β= 1 with the case study discussed by Bouguerra Mohammed Slim et al. [3] for Exponential Distribution. We have used the same values of λ (i.e λ = (1 / α)) and T_S as used in FCM [3] to compare our NOCRM with FCM [3]. The total waste time and the average completion time of applications due to checkpointing are computed using the equations Eq. (21) and Eq. (22) respectively.

1). Variation in Average Completion Time with Respect to Checkpoint Cost

When shape parameter, β is 1.0, and scale parameter, α is 2880 minutes (λ = 0.5 failures per day), we have obtained the optimal checkpoint intervals for the different checkpoint cost, varying from 0 to 95 minutes using our algorithm Estimate - T_C (). These optimal checkpoint intervals are then used to compute the average completion time of the fault tolerant MPI application with the execution time of 7 days.

Fig. 9 shows a comparison of the average completion time taken by NOCRM using optimal checkpoint intervals with the average completion time taken by FCM [3]. In Fig. 9, it is clear that, our NOCRM saves 10% of average completion time of the above MPI application in comparison to FCM [3].

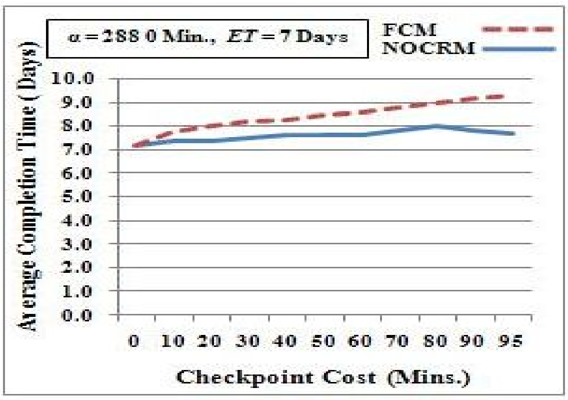

Fig. 9 Variation in Average Completion Time With Respect to Checkpoint Cost

2). Variation in Average Completion Time with Respect to Failure Rate Per Day

When checkpoint cost, T_S is 10 minutes, shape parameter, β= 1 and the failure rate, λ per day vary from 0.5 to 95, we have obtained the optimal checkpoint intervals using our algorithm Estimate-T_C (). These optimal checkpoint intervals are then used to compute the average completion time of the fault tolerant MPI application with the execution time of 10 days.

Fig. 10 shows a comparison of the average completion time taken by NOCRM using optimal checkpoint intervals with the average completion time taken by FCM [3].

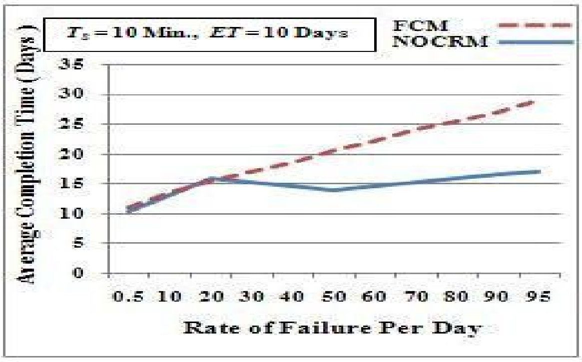

Fig. 10 Variation in Average completion Time w.r.t Rate of Failure Per Day

In Fig. 10, it is clear that, the average completion time taken by our NOCRM is almost close to FCM, when the rate of failure per day, λ is between 0.5 and 2.0 and our NOCRM is advantageous when the rate of failure per day, λ is above 2.0. Our NOCRM saves 24% of average completion time of the above MPI application in comparison to FCM [3].

3). Variation in Checkpoint Numbers with Respect to Failure Rate per Day

When checkpoint cost, T_S is 10 minutes, shape parameter, β is 1 and the failure rate, λ per day vary from 0.5 to 95, we have obtained the optimal checkpoint intervals using our algorithm Estimate-T_C (). These optimal checkpoint intervals are then used to compute the number of checkpoints taken during the execution of a fault tolerant MPI application with the execution time of 10 days.

Fig. 11 shows a comparison of the number of checkpoints taken by NOCRM using optimal checkpoint intervals with the number of checkpoints taken by FCM [3].

In Fig. 11, it is clear that, NOCRM saves an average of 49% of total number of checkpoints taken during the execution of the above MPI application in comparison to FCM [3].

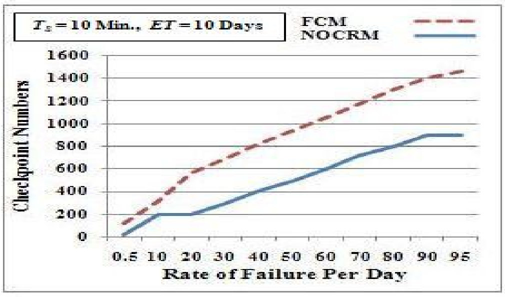

Fig. 11 Variation in Optimal Checkpoint Numbers

C. Comparison of NOCRM with Young's Model Using Poisson's Process with Exponential Distribution

In this case study, we compare the results obtained from our NOCRM with the results obtained from Young's model [1] using Exponential Distribution for MTBF.

1). Variation in Checkpoint Interval with Respect to Checkpoint Cost

Fig. 12 shows a comparison of the checkpoint intervals obtained from our algorithm Estimate-T_C (), when shape parameter β is 1.0, and scale parameter α is 2880 minutes (λ= 0.5 failures per day), with the checkpoint intervals obtained from Young's model [1] for the checkpoint costs varying from 10 to 95 minutes.

The Eq. (20) is used by the Young's model [1] to compute the optimal checkpoint intervals for the above values of checkpoint cost and α (α= T_F = (1/λ). In Fig. 12, it is clear that, the value of checkpoint interval increases gradually with the increase in checkpoint cost in case of Young's model [1], but, in NOCRM checkpoint interval does not increase uniformly with the increase in checkpoint cost.

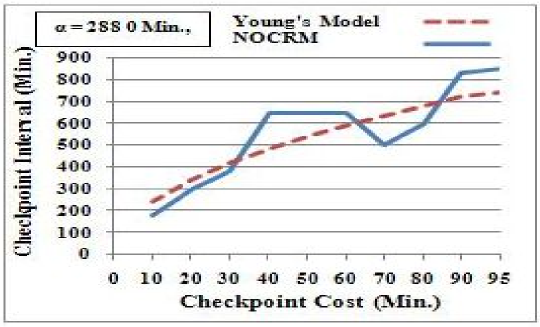

Fig. 12 Variation in Checkpoint Intervals

2). Variation in Total Waste Time with Respect to Checkpoint Cost

When shape parameter, β is 1.0, and scale parameter, α is 2880 minutes (λ= 0.5 failures per day), we have obtained the optimal checkpoint intervals for the checkpoint cost varying from 0 to 95 minutes using our algorithm Estimate T_C ().

The Eq. (20) is used by the Young's model [1] to compute the optimal checkpoint intervals for the above values of checkpoint cost and α (α = T_F = (1/λ). These optimal checkpoint intervals are then used to compute the average completion time of the fault tolerant MPI application with execution time of 7 days (168 Hours).

Fig. 13 shows a comparison of the waste time produced by optimal checkpoint intervals obtained from NOCRM with the waste time produced by checkpoint intervals obtained from Young's model for checkpoints costs varying from 10 to 95 minutes and α = 2880 minutes.

In Fig. 13, it is clear that, our NOCRM reduces 34% of the waste time due to checkpointing of the above MPI application in comparison to Young's model [1].

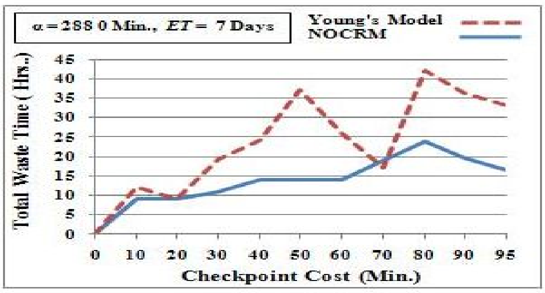

Fig. 13 Variation in Checkpoint Cost

3). Variation in Total Waste Time with Respect to Completion Time When shape parameter, β is 1.0, and scale parameter, α is 2880 minutes (λ= 0.5 failures per day), we have obtained the optimal checkpoint interval as 180 minutes for the checkpoint cost of 10 minutes using our algorithm Estimate-T_C (). The checkpoint interval, T_C, obtained from Young's model by using the Eq. (20), when checkpoint cost, T_S = 10 minutes and α = T_F = 2880 minutes, is 240 minutes.

Fig. 14 shows a comparison of the waste time produced by optimal checkpoint interval of 180 minutes obtained from our NOCRM with the waste time produced by checkpoint interval of 240 minutes obtained from Young's model [1] for the MPI applications of execution time (completion time) varying from 100 hours to 2500 hours.

In Fig. 14, it is clear that, our NOCRM reduces 25% of waste time due to checkpointing in comparison to Young's model [1] for the MPI applications of execution time (completion time) varying from 100 hours to 2500 hours.

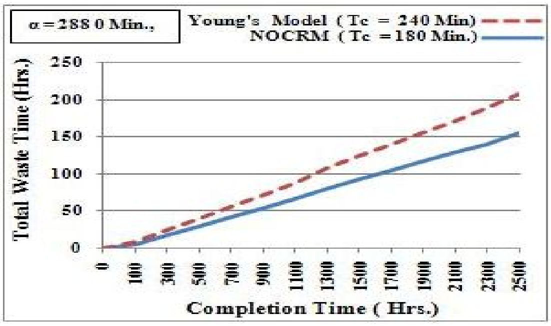

Fig. 14 Waste time of NOCRM and Young's Model

VII. CONCLUSIONS

The NOCRM developed by us using Weibull's and Exponential distribution for determining optimal checkpoint interval saves a large amount of waste time in comparison to [1-3] when the different parameters like scale parameter α and the shape parameter β of Weibull's distribution and the checkpoint cost T_C are known. For Exponential distribution, we set β to 1.

Thus, our NOCRM can be used to determine the optimal checkpoint interval for any fault tolerant MPI application when the checkpoint cost of an application is known along with the parameters of Weibull's Distribution.

The optimal checkpoint interval obtained from our NOCRM can be effectively used to provide the fault tolerance to an application. Our NOCRM leads to considerable reduction in total waste time. The scale parameter, α, the shape parameter, β and the checkpoint cost, T_S are system dependent.

REFERENCES

[1] J.W. Young. A first order approximation to the optimum checkpoint interval. *Communications of the ACM*, 17 (9): 530-52, 1974.

[2] Yudan Liu, R. Nassar, C. Leangsuksun, N. Naksinehaboon, M. Paun and SL. Scott. An optimal checkpoint/restart model for a large scale high performance computing system. In *Parallel and Distributed Processing, 2008. IEEE International Symposium.* pages 1-9. IEEE, 2008.

[3] M.S. Bouguerra, D. Trystram, T. Gautier and J.M. Vincent. A new flexible Checkpoint/Restart model. 2008.

[4] A. Bouteiller, P. Lemarinier, G. Krawezik,and F. Cappello. Coordinated checkpoint versus message log for fault tolerant MPI. *IEEE Computer Society*, 2003.

[5] P.M. Mallikarjuna Shastry and K. Venkatesh. Selection of a Checkpoint Interval in Coordinated Checkpointing Protocol for Fault Tolerant Open MPI. *International Journal on Computer Science and Engg., (IJCSE)*, 2(6): 2064-2070, 2010.

[6] A.J. Oliner, and L. Rudolph and R.K. Sahoo. Cooperative checkpointing: a robust approach to large-scale systems reliability. In *Proceedings of the 20th annual international conference on Supercomputing*, pages 14-23. ACM, 2006.

[7] J. T. Daly. A higher order estimate of the optimum checkpoint interval for restart dumps. *Future Generation Computer Systems*, 22(3):303--22, 2006.

[8] J. T. Daly. A model for predicting the optimum checkpoint interval for restart dumps. *Computational Science—ICCS 2003*, pages 724-724, 2003.

[9] R. A. Oldfield, S. Arunagiri, P. J. Teller, S. Seelam, M.R. Varela, R. Riesen, P.C. Roth. Modeling the impact of checkpoints on next-generation systems. 2007.

[10] J.T. Daly. A strategy for running large scale applications based on a model that optimizes the checkpoint interval for restart dumps. In *Proceedings of I International Workshop on Software Engineering High Performance Computing System Applications*, pages 70-74. Citeseer, 2004.

[11] N. Naksinehaboon, Y. Liu, et al. Reliability-aware approach: An incremental checkpoint/restart model in HPC environments. In *Eighth IEEE International Symposium on Cluster Computing and the Grid*. pages 783-788. IEEE, 2008.

[12] R. Subramaniyan, E. Grobelny,S. Studham and A.D. George. Optimization of checkpointing-related I/O for high-performance parallel and distributed computing. *The Journal of Supercomputing*, 46(2):150-180, 2008.

[13] M. Wu, X.H. Sun and H. Jin. Performance under failures of high-end computing. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*. page 48.ACM, 2007.

[14] N.R. Gottumukkala.Failure analysis and reliability-aware resource allocation of parallel applications in High Performance Computing systems. 2008.

[15] N.R. Gottumukkala,C.B. Leangsuksun, R. Nassar, M. Paun, D. Sule and S.L. Scott.Reliability-aware Optimal K Node Allocation of Parallel Applications in Large Scale High Performance Computing Systems.

[16] K. Wong and M. Franklin. Distributed computing systems and checkpointing. In *High Performance Distributed Computing, 1993, Proceedings the 2nd International Symposium on*. pages 224-233.IEEE, 1993.

[17] J.S. Plank and M.G. Thomason. The average availability of parallel checkpointing systems and its importance in selecting runtime parameters. In *Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposiums on*. pages 250-257. IEEE, 1999.

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

[19] P.M. Mallikarjuna Shastry and K. Venkatesh. Performance Evaluation of Coordinated Checkpointing Protocol using MPI Point to Point and Collective Communications. *International Journal on Advanced Computing and Communications.* Vol. 3 Issue 1, Page No: 12-18, Feb-2011

AUTHORS PROFILE

MR. Mallikarjuna Shastry P.M has received his B.E. and M.Tech in Computer Science and Engineering from Karnataka University Dharwar, and Vishweshwaraiah Technological University, Belgaum, Karnataka, India respectively. He is a Professor in the department of Computer Science and Engg. at Sapthagiri College of Engg, Bangalore, Karnataka, India. He has totally 19 years of teaching experience in Computer Science and Engg.

Prof. Dr. K. Venkatesh has received his M.Sc. in Physics from Mysore University in 1973 and MS from BITS Pilani in 2001 respectively. He has received Ph.D in 1980 from Mysore University. He is a professor at M.S.Ramaiah Institute of Technology, Bangalore-54 and has totally 32 years of teaching experience.