LETTER

Checkpoint Time Arrangement Rotation in Hybrid State Saving with a Limited Number of Periodical Checkpoints

Ryo SUZUKI[†], *Nonmember*, **Mamoru OHARA**[†], **Masayuki ARAI**[††], **Satoshi FUKUMOTO**[††a)], *and* **Kazuhiko IWASAKI**[††], *Members*

SUMMARY This paper discusses hybrid state saving for applications in which processes should create checkpoints at constant intervals and can hold a finite number of checkpoints. We propose a reclamation technique for checkpoint space, that provides effective checkpoint time arrangements for a rollback distance distribution. Numerical examples show that when we cannot use the optimal checkpoint interval due to the system requirements, the proposed technique can achieve lower expected overhead compared to the conventional technique without considering the form of the rollback distance distribution.

key words: *distributed checkpointing, hybrid state saving, checkpoint-space reclamation, time arrangement rotation*

1. Introduction

The hybrid state saving technique [1] is a checkpointing-restart technique for distributed systems [2] that combines uncoordinated checkpointing and logging. In fault recovery mechanisms that include such state savings, one of the main issues has been to find optimal checkpoint intervals [3]. However, we focus on a new aspect of checkpointing and a recovery strategy using a reclamation technique for checkpoint space in storage.

This paper discusses hybrid state saving in distributed applications in which processes cannot always save checkpoint data in a stable storage due to system restrictions. For example, process sharing and using storage exclusively requires adjustment of the timings of saving checkpoints in order to avoid conflicts [4]. In this paper, we assume that processes must create checkpoints at constant intervals determined by the restrictions. An intelligent transport system is an example application, in which cars sense their environmental information and periodically send this information as checkpoints by means of wireless communication via beacons, which are evenly spaced along roads. As another example, imagine that artificial satellites orbiting the Earth can send data only when the satellites pass over their base stations on the ground.

In this paper, we propose an efficient checkpoint reclamation technique under the system restrictions described above. The basic concept of the proposed scheme is to thin out the checkpoint sequence that each process has created up until the new checkpointing time. The proposed technique calculates the recovery overhead considering the form of the rollback distance distribution and discards a checkpoint to obtain the effective time arrangements of checkpoints. The expected total overhead can be reduced even if we have to set the checkpoint interval to a non-optimal interval.

The remainder of this manuscript is organized as follows. Section 2 presents an outline of hybrid state saving. A checkpoint-space reclamation technique that provides a more effective time arrangement of checkpoints is presented in Sect. 3. In addition, we analyzed the expected overhead of the proposed technique in Sect. 3. Numerical examples presented in Sect. 4 reveal that the proposed technique can achieve less overhead than the conventional technique. Finally, Sect. 5 summarizes the paper.

2. Hybrid State Saving and Recovery

Regarding state saving and the recovery operation after error detection, Soliman et al. presented a system model [1], in which uncoordinated checkpointing and logging are combined and used. In other words,

- a process takes a checkpoint periodically at all T event executions [6],
- a process saves the changes in its state after each event execution.

On the other hand, recovery operation after error detection is expressed as shown in Fig. 1. The point in time to which a failed process should return is referred to as the recovery point in the recovery operation. By executing either rollback or rollforward from the closest checkpoint to the recovery point with the bi-directional log, hybrid state sav-

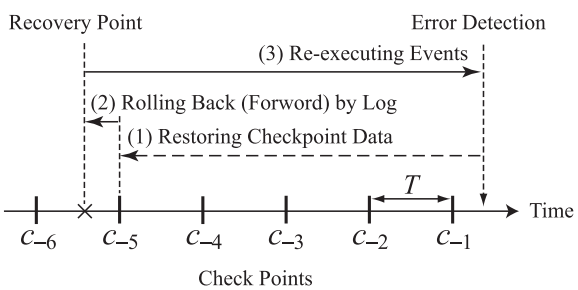

Fig. 1 Recovery operation in hybrid state saving.

Manuscript received May 2, 2012.
Manuscript revised September 3, 2012.

[†]The authors are with the Graduate School of Engineering, Tokyo Metropolitan University, Hachioji-shi, 192–0397 Japan
[††]The authors are with the Faculty of System Design, Tokyo Metropolitan University, Hachioji-shi, 192–0397 Japan
a) E-mail: s-fuku@tmu.ac.jp

DOI: 10.1587/transinf.E96.D.141

ing can achieve efficient recovery [1]. The procedures are as follows:

(1) a failed process restores the checkpoint that is closest to the recovery point (c_{-5} in Fig. 1),
(2) using the bi-directional log the process rolls back (or forward) from the closest checkpoint to the recovery point,
(3) the process re-executes events between the recovery point and the error detection point.

The time interval between the recovery point and the error detection point is referred to as the rollback distance. In a number of applications, the recovery points are not always distributed uniformly. Soliman et al. assumed a geometric distribution for the rollback distance distribution in their analysis of the overhead of hybrid state saving [1]. We also support their assumption.

Soliman et al. further supposed that processes can hold infinite checkpoints for convenience of the analysis. However, processes can hold a limited number of checkpoints because they have only finite size of storage in practical systems. In this paper, we discuss hybrid state saving, in which each process can hold a limited number of checkpoints.

3. Checkpoint Time Arrangement Rotation and Overhead Analysis

In uncoordinated checkpointing, processes periodically create checkpoints and must hold several checkpoints for preparing rollback propagation. In the steady state, processes usually hold as many checkpoints as they can in order to minimize the recovery overhead. Therefore, when a new checkpoint is created, the processes must reclaim the storage space used for existing checkpoints [5].

The simplest way of checkpointing with a finite number of available checkpoints is to generate a new checkpoint and discard the oldest checkpoint at the time point of each checkpoint cycle. Figure 2 (a) illustrates such a scheme with four available checkpoint spaces, where the vertical lines denote checkpoints and × denote possible recovery points obeying a geometric distribution. Time arrangement rotation $\boldsymbol{\alpha}$ is the set of time arrangements obtained after each checkpointing. By taking a new checkpoint while always discarding the oldest checkpoint, c_{-4}, only one type of arrangement appears. Thus, the sequence of available checkpoints is arranged at a fixed interval at all times. However, this conventional scheme cannot always provide the minimized recovery overhead, because the possible recovery points are not uniformly distributed.

We attempt to decrease the expected overhead by thinning out the conventional checkpoint sequence to yield a more effective checkpoint time arrangement. For example, consider time arrangement rotations $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in Figs. 2 (b) and 2 (c), which are the sets of time arrangements obtained after a number of checkpointings. Rotation $\boldsymbol{\beta}$ is obtained by discarding the second to oldest checkpoint, c_{-3}, and the oldest checkpoint, c_{-4}, by turns. Although the time arrangement after checkpointing β_2 is identical to that in rotation $\boldsymbol{\alpha}$, the arrangement after checkpointing β_1 adapts to the rollback distance distribution more effectively. In other words, a lower total overhead is expected in rotation $\boldsymbol{\beta}$. Moreover, by executing the checkpointing that discards checkpoint c_{-3} only once and executing checkpointings that discard c_{-2} and c_{-4} by turns, the time arrangements in rotation $\boldsymbol{\gamma}$ can be obtained. In this case, an even better adaptation of the rollback distance distribution is possible.

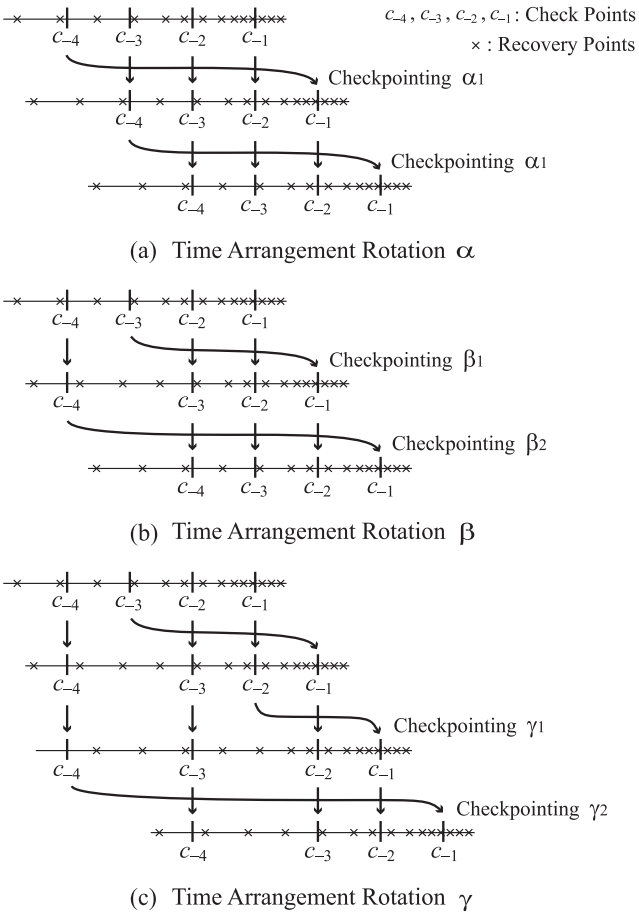

Fig. 2 Illustrations of time arrangement rotations.

We propose a checkpoint-space reclamation technique, which determines the checkpoint to be discarded according to the expected overhead on every checkpointing. Note that the periodicity for the rotation is indispensable in order to maintain the checkpointing scheme in the steady state. The proposed technique calculates the expected overhead for each case, where the checkpoint c_{-i} $(i = 1, 2, \cdots, M)$ would be discarded, and replaces the checkpoint providing the minimum expected overhead with a new checkpoint. Next, we introduce the following example derivation of the expected overhead.

We now assume the number of effective checkpoints M to be finite in each process. In this model, X expresses the time distance measured by the number of events between the error detection point and the recovery point. As mentioned above, we assume a random variable X that obeys a geo-

metric distribution with parameter p. Thus, the probability function is

$$f(x) = Pr\{X = x\} = p(1-p)^x, \tag{1}$$

and the average $E[X]$ is $(1-p)/p$. Moreover, we define T as the checkpoint interval measured by the number of events, C as the overhead for taking/loading a single checkpoint measured by units of time, and δ as the overhead for taking/replaying a log record per event.

First, the overhead needed for a single recovery operation for the conventional method illustrated in Fig. 2 (a) is calculated for an example of deriving the expected overhead of hybrid state saving. When $M \geq 1$, if we assume both the error detection point and the recovery point are at the middle point of the checkpoint interval, then the overhead for rolling back/forward to the recovery point can be approximated as

$$r(X) = \begin{cases} \delta\frac{T}{8} & (0 \leq X < \frac{T}{4}) \\ C + \delta\frac{T}{8} & (T/4 \leq X < \frac{T}{2}) \\ C + \delta\frac{T}{4} & (T/2 \leq X < (M - \frac{1}{2})T) \\ C + \delta\left\{X - (M - \frac{1}{2})T\right\} & \\ \qquad ((M - \frac{1}{2})T \leq X \leq \infty). & \end{cases} \tag{2}$$

For an error, the expected overhead of a return to recovery point can be obtained by

$$R(T) = E[r(X)] = \sum_{x=0}^{\infty} r(x)f(x). \tag{3}$$

Thus, substituting Eq. (1) and (2) into Eq. (3), we obtain

$$R(T) = \frac{\delta T}{8} + C(1-p)^{\frac{T}{4}} + \frac{\delta T}{8}(1-p)^{\frac{T}{2}} + \delta\left(\frac{1-p}{p} - \frac{T}{4}\right)(1-p)^{(M-\frac{1}{2})T}. \tag{4}$$

In addition, the expected event re-execution overhead and the overhead for retaking checkpoints and logs are $E[X]$ and $(C/T + \delta)E[X]$, respectively. Thus, the expected total overhead per event, $H(T)$, which combines overheads for checkpointing, logging, and recovery, is

$$H(T) = \frac{C}{T} + \delta + \lambda\left\{\left(1 + \frac{C}{T} + \delta\right)E[X] + R(T)\right\}, \tag{5}$$

where λ denotes the average number of errors per unit time. As mentioned earlier, the checkpoint interval is given as a constant determined by system restrictions in the applications discussed herein. In other words, the checkpointing overhead is constant. The proposed technique attempts to reduce the expected recovery overhead $R(T)$ by choosing an effective time arrangement rotation.

Next, we concretely describe the proposed checkpoint-space reclamation technique that provides a rotation with an even smaller expected overhead. The time arrangement rotation changes according to how the discarded checkpoint is chosen at the checkpointing time, as in the above-mentioned examples. In the present study, we chose the checkpoint that yielded the lowest expected overhead time arrangement from M available checkpoints c_{-M}, c_{-M+1}, c_{-M+2}, $\cdots$, c_{-2}, c_{-1}. If the expected recovery overhead for an error detection between a checkpointing ψ_i and its successor ψ_{i+1} is denoted by $R_i(T)$, the discarded checkpoint is chosen to satisfy the following equation:

$$R_i(T) = \min_{0 \leq j \leq M} \sum_{x=0}^{\infty} r_{ij}(x)f(x). \tag{6}$$

Here, $r_{ij}(x)$ is the recovery overhead with the new time arrangement obtained by discarding checkpoint $c_{-j}^{<i>}$ from the checkpoint sequence $c_{-M}^{<i>}, c_{-M+1}^{<i>}, \cdots, c_{-2}^{<i>}, c_{-1}^{<i>}$ at the checkpointing ψ_i. We estimate $r_{ij}(x)$ by the same approximation as Eq. (2). Note that $c_0^{<i>}$ means that the process does not take a new checkpoint and no checkpoint is discarded at checkpointing ψ_i. In other words, we perform the following algorithm before generating each checkpoint:

$k \leftarrow 0$; $m \leftarrow \infty$
for $j = 0$ to M **do**
  $R_{ij}(T) \leftarrow \sum_{x=0}^{\infty} r_{ij}(x)f(x)$
  if $R_{ij}(T) < m$ **then**
    $m \leftarrow R_{ij}(T)$
    $k \leftarrow j$
  end if
end for
if $k > 0$ **then**
  discard $c_k^{<i>}$
  generate a new checkpoint
  $\tau[] \leftarrow \{T, \tau[0], \tau[1], \cdots, \tau[k-1], \tau[k+1], \ldots, \tau[m]\}$
else
  $\tau[0] \leftarrow \tau[0] + T$
end if,

where τ is an array in which $\tau[i]$ denotes the checkpoint interval between $c_{-(i+1)}$ and c_{-i}, and $r_{ij}(x)$ is calculated by

function $r_{ij}(x)$
  if $x < \tau[0]/4$ **then return** $\delta\frac{\tau[0]}{8}$
  else if $x < \tau[0]/2$ **then return** $C + \delta\frac{\tau[0]}{8}$
  end if
  $t \leftarrow \tau[0]/2$
  for $i = 1$ to M **do**
    $t \leftarrow t + \tau[i]$
    if $x < t$ **then return** $C + \delta\frac{\tau[i]}{4}$
    end if
  end for
  return $C + \delta\,(x - t)$
end function.

Such checkpoint time arrangements come into a certain regular rotation $\boldsymbol{\phi}$. Each arrangement in $\boldsymbol{\phi}$ is obtained after checkpointing ϕ_i, where $1 \leq i \leq |\boldsymbol{\phi}|$ when $|\boldsymbol{\phi}|$ is the number of arrangements in rotation $\boldsymbol{\phi}$. For all arrangements of $\boldsymbol{\phi}$, the expected overhead to return to a recovery point can be evaluated by

$$R(T) = \frac{1}{|\boldsymbol{\phi}|}\sum_{i=1}^{|\boldsymbol{\phi}|} R_i(T). \tag{7}$$

Table 1 Examples of periodicity of discarded checkpoints.

T	$M = 5$	$M = 10$
250	0, 0, −3, 0, −5, 0, 0, −4	−5, −8, −5, −10
300	0, −3, 0, −5	−6, −8, −6, −10
350	−3, 0, −5	−9, −7, −6, −9, −6, −10
400	−3, −5, −2, 0, −5	−7, −9, −7, −10

The calculation cost for Eq. (7) can be estimated as $O(M)$, whereas that for Eq. (4) is $O(1)$. We search the improved time arrangement rotation in return for the cost increase.

4. Numerical Examples

We provide numerical examples of the proposed technique described in the previous section. The value of parameters were set to $C = 2.7$, $\delta = 0.9$, $\lambda = 0.001$, and $M = 5$ or 10. The rollback distance distribution follows a geometric distribution, and the expected total overheads are calculated for $p = 0.0005$ and $p = 0.001$ ($E[X] = 1999$ and $E[X] = 999$).

Table 1 shows the periodicity of time arrangements of checkpoints in steady states of the proposed technique when $p = 0.0010$. The numbers in the table express discarded checkpoints. For example, when $M = 5$ and $T = 400$ ("-3, -5, -2, 0, -5"), the 3rd checkpoint c_{-3} is first discarded, and the process generates a new checkpoint. Then, c_{-5} and c_{-2} are discarded successively to take new checkpoints. In the next checkpointing operation, the process discards c_0, which means that the process does not create a new checkpoint. Finally, the process does away with the 5th checkpoint c_{-5}. Henceforth, this pattern is repeated perpetually. Finding such periodicity beforehand allows the process to take checkpoints much more effectively.

Figure 3 illustrates the expected total overheads $H(T)$ for the conventionaltechnique and for the proposed technique for $M = 5$ with the changing value of p. Approximately the same optimal checkpoint interval T^* was obtained for both reclamation techniques. For instance, T^* is 800 for $p = 0.0005$ in both techniques. In addition, for $p = 0.001$, $T^* = 400$ in the conventional technique and $T^* = 300$ in the proposed technique. When the checkpoint intervals exceed T^*, these overheads increase gradually and monotonically. There are only slight differences in the expected total overhead, regardless of the reclamation techniques in such a range. On the other hand, in the area in which the checkpoint intervals are smaller than T^*, the proposed technique can lower the expected total overhead significantly. Although the calculation results for $M = 10$ or more have been also obtained, we omit these results because these results denote the same tendency as the results for $M = 5$.

We are not always able to set the optimal checkpoint interval T^* due to certain system requirements. As shown in Fig. 3, $H(T)$ is much more robust with respect to T in the proposed technique. The proposed technique is effective for reducing the overhead when we must use the value of the checkpoint interval other than T^*, especially for small checkpoint intervals.

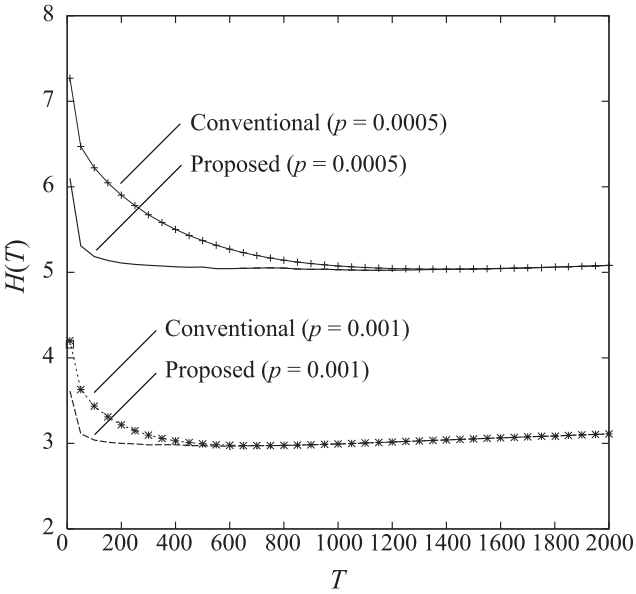

Fig. 3 Expected total overhead of the conventional and proposed techniques for $M = 5$.

5. Conclusion

This paper has discussed hybrid state saving, which is a typical uncoordinated distributed checkpointing technique. We described a discrete time evaluation model with finite available checkpoints. Furthermore, we proposed a reclamation technique for checkpoint space that achieves more effective time arrangements of checkpoints, and numerical examples have been presented.

Treating the case in which the rollback distance distribution dynamically changes in the normal operations is a topic for future study. Moreover, the generalization of the proposed technique to other types of uncoordinated checkpointing without logging remains as a future objective.

Acknowledgements

The present study was supported in part by a Grant-in-Aid for Scientific Research No. 15500046 from the Ministry of Education, Science, Sports and Culture, Japan.

References

[1] H.M. Soliman and A.S. Elmaghraby, "An analytical model for hybrid checkpointing in time warp distributed simulation," IEEE Trans. Parallel Distrib. Syst., vol.9, no.10, pp.947–951, Oct. 1998.
[2] M. Elnozahy, L. Alvisi, Y. Wang, and D. Johnson, "A survey of rollback-recovery protocols in message-passing systems," ACM Comput. Surv., vol.34, no.3, pp.375–408, Sept. 2002.
[3] T. Ozaki, T. Dohi, H. Okamura, and N. Kaio, "Distribution-free checkpoint placement algorithms based on min-max principle," IEEE Trans. Dependable Secure Comput., vol.3, no.2, pp.130–140, April-June 2006.
[4] N.H. Vaidya, "Staggered consistent checkpointing," IEEE Trans. Parallel Distrib. Syst., vol.10, no.7, pp.694–702, July 1999.

[5] Y. Wang, P. Chung, I. Lin, and W. Fuchs, "Checkpoint space reclamation for uncoordinated checkpointing in message-passing systems," IEEE Trans. Parallel Distrib. Syst., vol.6, no.5, pp.546–554, May 1995.

[6] Y. Lin, B. Preiss, W. Loucks, and E. Lazowska, "Selecting the checkpoint interval in time warp simulation," Proc. Workshop on Parallel and Distributed Simulation (PADS) 1993, pp.3–10, 1993.