T H E   N A T I O N A L   U N I V E R S I T Y
o f   S I N G A P O R E

Founded 1905

S c h o o l *o f* C o m p u t i n g
Lower Kent Ridge Road, Singapore 119260

TR10/99

A Cost-Effective State Saving Scheme for Optimistic Parallel Simulation

Seng Chuan TAY, Yong Meng TEO
and Chin Hock NG

October 1999

T e c h n i c a l   R e p o r t

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

T S CHUA
Acting Dean of School

A Cost-Effective State Saving Scheme for Optimistic Parallel Simulation

Seng Chuan Tay[*] *Yong Meng Teo*[†] *and Chin Hock Ng*[‡]
National University of Singapore
Lower Kent Ridge Road
Singapore 119260
email: phytaysc@nus.edu.sg

Abstract

Unlike the conservative approach, Time Warp (TW) executes simulation events greedily and uses a rollback mechanism to recover from causality errors. The TW protocol has the potential to exploit a higher degree of parallelism in the simulated system but it is realized with an overhead. For the simulator to carry out a rollback, the system state must be checkpointed. While increasing the checkpointing frequency increases the state saving cost, an infrequent scheme also escalates the *coast forward effort* when a large number of executed events are *redone*. Such a paradox indicates the need for a cost model to decide if a system state should be saved. This paper uses a probabilistic approach to weigh the performance gain and loss of each checkpointing. Logical processes of the TW simulation and their processing elements are assumed to be homogeneous. By the use of exponential distribution on inter-arrival time and service time, we can derive the rollback probability, thereby calculate the expected coast forward effort if a state is not saved. Based on the derived expectation, a state vector is saved only if the expected coast forward effort is larger than the state saving cost and vice versa. Our experiments show that the cost model reduces the simulation elapsed time by close to 30% as compared to saving the system state after each event execution, and saving the system state at a predefined interval.

Keywords: performance optimization, optimistic simulation, state saving, rollback, cost model

1 Introduction

Time Warp (TW) is an optimistic mechanism [8] used to manage the event execution in parallel discrete-event simulation (PDES). As compared to the conservative mechanism [1, 2, 14], the events processed in TW logical processes (LPs) may violate the causality constraint [5]. An out-of-sequence event message ($\mathcal{M}$), also called straggler, is identified if the local virtual time (LVT) of the destined LP is greater than the timestamp of the arriving straggler (TS($\mathcal{M}$)). When such a causality error occurs, a rollback procedure annuls those events simulated ahead of time. The destined LP performs the recovery by sending notifications to its successors to cancel the messages it has erroneously sent, restoring itself to a latest state before TS($\mathcal{M}$), and re-executing its simulation from thereon. Hence, each LP of TW has to maintain a timestamped state queue to allow for recovering a correct past state[1]. Checkpointing the simulated system can be a costly overhead in TW mechanism [17]. The conventional approach is to save the state whenever an event is executed (or the checkpointing interval $w = 1$). As in the conventional scheme every event can be a potential recovery point, it

[*]Centre for Information Technology and Applications, Faculty of Science

[†]School of Computing

[‡]Department of Computational Science

[1]The LP also has to maintain an input queue and an output queue for storing the incoming and outgoing event messages respectively.

allows rollback to be carried out efficiently, i.e, $\mathcal{M}$ can be immediately executed after the closest state of timestamp less than TS($\mathcal{M}$) is restored. Nonetheless, such a checkpointing scheme can become expensive in terms of wall-clock time especially when the size of state vector is huge[2].

Other schemes are *incremental*, *infrequent*, *adaptive* or *hybrid*. Instead of saving the entire vector, the incremental approach [4, 23, 28] saves only those changes to the state. When memory consumption is concerned, the incremental approach is useful if the size of state vector is large and only a small portion is modified after an event has been executed. However, the incremental approach requires additional processor time to reconstruct the desired state from the incremental changes thereby incurring a performance penalty. The infrequent approach [10, 18] reduces the frequency of state saving, i.e., $w > 1$. As a result, the state saving cost is also decreased proportionally. However, the infrequent state saving approach also has drawbacks. Suppose a state $\mathcal{S}$ of timestamp TS($\mathcal{S}$) is restored after a straggler is detected. All the events in the time interval from TS($\mathcal{S}$) to TS($\mathcal{M}$) will need to be redone before $\mathcal{M}$ can be executed. Such a performance penalty is actually a repeated effort and proportional to the size of checkpointing interval. Adaptive schemes use the dynamic of the simulator at runtime, and allows LPs to adjust their checkpointing interval on the fly with respect to the simulation advancement. Variation of such schemes depends on the parameters used, such as memory usage [3], time spent in saving state and event and restoration time [22], and rollback behavior [11]. Such an adaptiveness depends on the characteristics of statistical data collected, thus the decision to save a state or not is also based on the extrapolation of the runtime history. The prediction is accurate provided the system is stable throughout the whole simulation run. Otherwise, the extrapolation may not be appropriate and can produce adverse effect. Recently, hybrid approaches such as combining periodic (or infrequent) approach and probabilistic approach [21], combining event history and incremental approach [20], embedding the incremental state saving mode on a sparse state saving basis [19], multiplexing the incremental approach and infrequent approach at fixed interval [7], and switching automatically from periodic approach and incremental approach based on the cost model constructed by runtime statistics [24] have been proposed.

Instead of using the incremental, infrequent, adaptive or hybrid approaches, we weigh the loss and gain in wall-clock time before a state is saved (or not saved). The *rollback probability* is derived using mathematical convolution, and applied to a back-tracking algorithm for computing the *expected coast forward effort*. In the proposed cost model, a state vector is saved only if the *expected* coast forward effort is larger than the state saving cost and vice versa. The advantages of the cost model are:

1. It does not need to collect statistical data at runtime so no extrapolation is performed. Instead, the cost model is based on a strong mathematical foundation.
2. It ensures that the decision to save a state or not is cost effective, thus the overall simulation time can be reduced.

The rest of this paper is organized as follows. Section 2 adopts a probabilistic approach to model the LVT advancement in LPs. By the use of convolution technique, we derive the rollback probability due to the occurrence of stragglers. Section 3 constructs the cost model for checkpointing decision based on the probability derived. We compute the coast forward effort using a back-tracking algorithm and compare it with the state saving time before a checkpointing decision is made. Such a comparison ensures that the system is checkpointed only if it is cost effective. Section 4 investigates the effectiveness of the proposed cost model in reducing the overall simulation elapsed

[2]Saving the state vector after each event execution is also expensive in terms of memory space but the details are beyond the scope of this paper.

time against two checkpointing schemes. We also evaluate the average number of coasted forward events incurred in a rollback, the hit ratio where coasting forward is not needed, and the ratio of the number of states saved with respect to the number of events executed. The aggregate effect of these factors to the simulation elapsed time is also analyzed. Finally, section 5 contains our concluding remarks and some discussions on future work.

2 Mathematical Formulation of Causality

The following assumptions are made:

1. a simulator contains p homogeneous LPs and p homogeneous processing elements (PEs)
2. the placement of LPs on PEs is one-to-one
3. simulator executes two types of events: *arrival* and *departure*
4. each arrival event has a corresponding departure event in the same LP and the execution of each departure event will in turn schedule an arrival event in one of its succeeding LPs[3]
5. inter-arrival time and service time are exponentially distributed[4]
6. same granularity for arrival and departure routines
7. same amount of access time for transmit buffer and receive buffer
8. memory space is sufficient to complete the simulation
9. GVT is calculated after a constant number of events are executed, and the GVT window also serves as the barrier for LPs to be synchronized[5]

Table 1 contains a list of parameters used to derive the rollback probability. Wall clock time (or CPU time equivalently) is used in the measurements represented by T. Based on its timestamp order, each executed event is dynamically assigned an increasing event number, or *event index* synonymously. The distribution of the current event numbers in LPs does not exhibit large fluctuation due to the homogeneity in LPs and the homogeneity in PEs. We model the distribution of the current event numbers by a normalized discrete probability density function (pdf) with a parameter χ representing the deviation of the event numbers[6] (refer to Appendix A). Normal distribution is chosen due to its clustered density on the mean and median where more than 95% of the occurrences are included within two standard deviations on either sides of the mean. χ is also regarded as the spread of the current event numbers in LPs. As the current event numbers also represents the latest LVTs of their LPs, the normal function also represents the distribution of LP advancement during a simulation run. In our formulation, event numbers are re-used after a rollback is activated but not after a fossil collection.

Message passing among the simulation processes (see figure 1) is modeled by communication time consisting of *buffer access time* (T_{buffer}) and *transit time* ($T_{transit}$), where T_{buffer} refers to the time duration required to pack data into an output buffer or unpack a message into an input buffer, and $T_{transit}$ refers to the time duration required for the message to travel from the source to destination. In this paper we assume constant for $T_{transit}$. More complicated formulation based

[3]This corresponds to the queuing model where each customer will arrive to and depart from a service counter, and a customer after leaving the counter will enter one of the succeeding counters. Consequently, each customer will generate two events in the queuing model.

[4]This assumption is commonly used in literatures for mathematical tractability.

[5]While this assumption constrain the optimism of TW mechanism, the LPs are still executing asynchronously within the GVT window. The window size is a constant but its value is not assumed in our analysis.

[6]χ is similar to the σ in the continuous normal function.

PARAMETER		DESCRIPTION
system	λ	arrival rate (per simulated time) of each LP
	μ	service rate (per simulated time) of each LP
	β	LVT advancement rate (per simulated time)
	p	number of PEs
	N_s	number of events processed in sequential simulation
	N_p	number of true events processed by an LP ($N_p = \frac{N_s}{p}$)
	c	communication delay (in terms of number of events processed)
	a	lower bound of GVT window (in terms of event index)
	u	number of events processed during an LP visit
	d	diameter (longest path) of LP interconnections
	$f(d)$	number of events spreading on the longest path of LP inter-connections
	$\widehat{GVT}$	number of events processed (less number of events rolled back) before a GVT computation is activated
measured	T_{event}	event (arrival or departure) execution time
	T_{state}	state saving time
	T_{buffer}	buffer (receive or transmit) access time
	$T_{transit}$	message transmission time
derived	**PROBABILISTIC ROLLBACK**	
	$rb(I_0, J_0)$	probability (prob.) that an event message sent at index I_0 will cause a rollback when it is processed at index J_0
	$RB(J_0)^+$	prob. that a straggler is processed at index J_0
	$halt^{I_k}_{J_k}(d_k)$	prob. that a rollback caused by a straggler sent at index I_k of the source LP and processed at index J_k of the destined LP will stop after d_k events are undone
	CHECKPOINTING OVERHEAD	
	$T_{CCF}(J_0)$	coast forward effort required to re-execute the events when the state at index J_0 is not saved
	$\overline{T_{CCF}(J_0)}$	expectation of $T_{CCF}(J_0)$

Table 1: Parameters and Measures of Probabilistic Checkpointing Cost Model

on the connection topology of processors can be further studied. The reception of data is modeled by buffer access time only, and transit time is excluded to prevent double accounting. In the abstraction the duration for a sender to transmit a message to the receiver includes three time intervals in:

1. construction of message in the transmission buffer (T_{buffer})
2. message transmission on communication link ($T_{transit}$)
3. reception of message in the reception buffer (T_{buffer})

Thus, the number of events processed during this communication delay is $c = \lceil \frac{2 \times T_{buffer} + T_{transit}}{T_{event}} \rceil$.

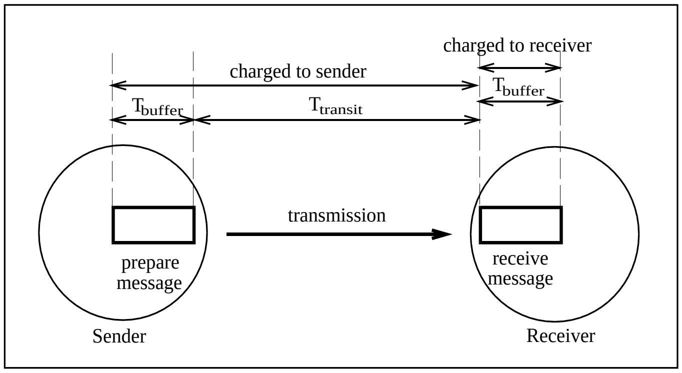

Figure 1: Communication Time Accounting

2.1 Characterization of Rollback Probability

Let a denote the lower bound (in terms of event index) of the the GVT window. We denote the window by $[a, a + \widehat{GVT}]$, where $\widehat{GVT}$ is the number of events executed and not rolled back before the next GVT computation is activated. For the ease of illustration, we let $a = 0$ in the following characterization. General case of the rollback abstraction is obtained by sliding the GVT window on the simulation time scale, i.e., by changing the value of a where $0 \leq a \leq N_p - 1$.

2.1.1 Proximity of Current Event Numbers in LPs

If the difference of LVTs in LPs is small so is the rollback probability. Such a difference in LVTs can also be translated into the deviation of current event numbers since each event execution in the LPs will cause an advancement in their LVT. Let u be the number of events processed when one message is sent across an LP, and d the diameter (or the longest path) of the LP interconnection. The number of LPs on the longest path is $d - 1$. In our assumption we let $u = 2$, i.e., a message will generate an arrival event and a departure event on each LP visit. Suppose LP_a is the first LP, and LP_b the last LP on the diameter (figure 2). As the longer the diameter, the later LP_b will start its event execution. Consequently, the deviation of the current event indices in LP_a and LP_b is increased. For the normalized discrete pdf, we therefore approximate the deviation of current event numbers in LPs based on the diameter of LP interconnections.

Since c events can be executed during each inter-LP transmission, and u events during each LP visit, the total number of events executed for the duration in traversing the longest path is $f(d) = d \times c + (d-1) \times u$. This number of events is used to approximate the deviation of current event numbers in LPs, i.e.,

$$\chi \approx f(d) \tag{1}$$

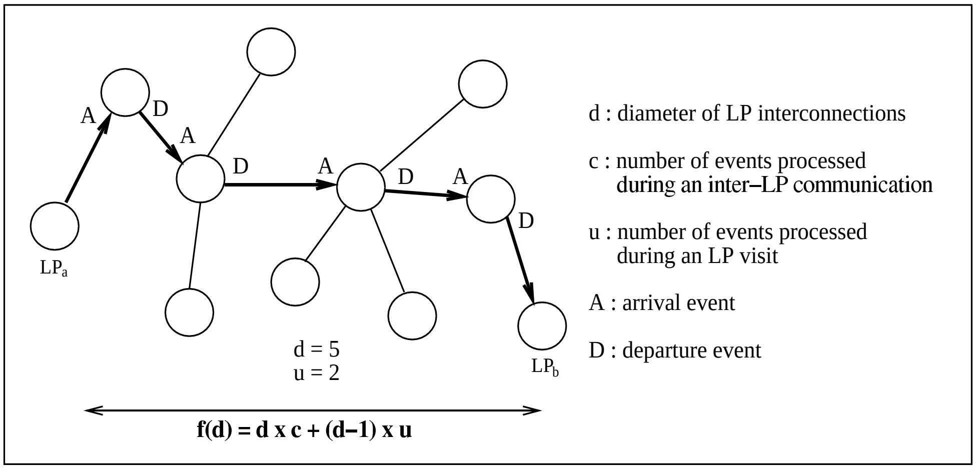

Figure 2: Spread of Current Event Numbers in LPs

2.1.2 Abstraction of Causality Error

We assume that the time interval of consecutive LVTs is exponentially distributed with a mean of $\frac{1}{\beta}$, where β is the LVT advancement rate defined as follows:

$$\beta = \begin{cases} 2\lambda & \text{if } \lambda < \mu \\ \lambda + \mu & \text{otherwise} \end{cases}$$

where λ is the arrival rate, and μ the service rate. The LVT after the n-th event is executed, denoted by LVT_n, is modeled based on the following observations.

- The first event processed by an LP is an arrival event. Otherwise, the causality constraint is violated.
- An LP cannot advance its LVT until the first arrival event is processed.
- An LP with LVT_n has advanced its LVT $n-1$ times after the first arrival event is executed.

Assume that the inter-arrival time and service time are identically and independently distributed (IID). We can parameterize LVT_n (see figure 3) as a sum of two random variables R_1 and R_2,

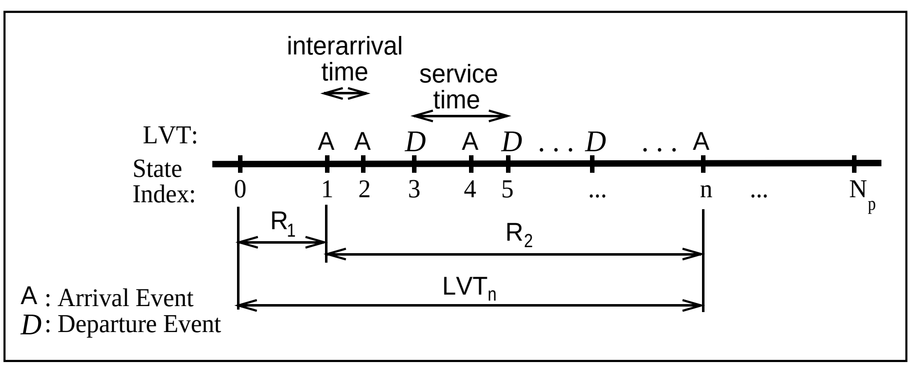

Figure 3: LVT Advancement

where $R_1 \sim \exp(\lambda)$, and $R_2 \sim$ gamma $(\beta,\, n-1)$.

Let random variable Z represent the LVT of an LP after the n-th event is executed. The probability density function of Z can be derived by convolution technique (refer to [25]), and is given as follows:

$$g(z) = \begin{cases} \lambda T^{n-1} \times \left(e^{-\lambda z} - e^{-\beta z} \sum_{k=0}^{n-2} \frac{(\theta z)^k}{k!} \right) & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\theta = \beta - \lambda$, and $T = \frac{\beta}{\theta}$.

Suppose the straggler $\mathcal{M}$ is generated immediately after the p-th event is executed in the sending LP, and processed after the q-th event is executed in the receiving LP. Let the timestamp of $\mathcal{M}$ be $LVT_{p,send}$, and the LVT of the receiving LP be $LVT_{q,recv}$, and modeled by random variables X and Y respectively. Similarly, the probability density functions, denoted by $g_1(x)$ and $g_2(y)$ respectively are given as follows:

$$g_1(x) = \begin{cases} \lambda T^{p-1} \times \left(e^{-\lambda x} - e^{-\beta x} \sum_{k=0}^{p-2} \frac{(\theta x)^k}{k!} \right) & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$g_2(y) = \begin{cases} \lambda T^{q-1} \times \left(e^{-\lambda y} - e^{-\beta y} \sum_{l=0}^{q-2} \frac{(\theta y)^l}{l!} \right) & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

We want to compute $\Pr(LVT_{p,send} < LVT_{q,recv})$, which is the probability for $\mathcal{M}$ to create a causality error in the receiving LP. Let $g_{X,Y}(x,y)$ be the joint density function of X and Y. To simplify the mathematical expressions we let $G_n(x) = \sum_{i=0}^{n-2} x^i = \frac{1-x^{n-1}}{1-x}$ in the following derivations. We also assume that the summation expression returns a zero if their lower index is greater than their upper index. Suppose the timestamp of $\mathcal{M}$ and the LVT of the destined LP are statistically independent, we have

$$\begin{aligned} Pr(LVT_{p,send} < LVT_{q,recv}) &= \int_0^\infty \int_0^y g_{X,Y}(x,y)\, dx\, dy \\ &= \int_0^\infty \int_0^y g_1(x) \times g_2(y)\, dx\, dy \\ &= \int_0^\infty \int_0^y \lambda T^{p-1} \times \left(e^{-\lambda x} - e^{-\beta x} \sum_{k=0}^{p-2} \frac{(\theta x)^k}{k!} \right) \times \\ &\qquad \lambda T^{q-1} \times \left(e^{-\lambda y} - e^{-\beta y} \sum_{l=0}^{q-2} \frac{(\theta y)^l}{l!} \right) dx\, dy \end{aligned} \tag{2}$$

From equation (9) in Appendix B, we have $Pr(LVT_{p,send} < LVT_{q,recv})$

$$\begin{aligned} &= \lambda^2 T^{p+q-2} \times [\frac{1}{2\lambda^2} - \frac{G_p\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda(\lambda+\beta)} - \frac{G_q(\frac{\theta}{\beta})}{\lambda\beta} + \frac{G_q\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda(\lambda+\beta)} + \frac{G_p(\frac{\theta}{\beta}) \times G_q(\frac{\theta}{\beta})}{\beta^2} \\ &\quad - \frac{1}{2\beta^2} \sum_{l=0}^{q-2} \frac{(\frac{\theta}{2\beta})^l}{l!} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k \sum_{i=0}^{k} \left(\frac{\left(\frac{1}{2}\right)^i \times (l+i)!}{i!} \right)] \end{aligned} \tag{3}$$

2.2 Rollback Probability

Suppose an LP has executed the J_0-th event, we want to know the probability ($RB(J_0)^+$) that the next event message received will cause a rollback in the LP. Equation (3) can compute the rollback probability provided the event indices in both source and destined LPs are known. In practice, however, the TW LPs execute events asynchronously so the event index in the source LP is unknown. The only information available is that the advancement in the source LP is confined within the GVT window[7]. For the ease of discussion, we let LP_0 send an event message $\mathcal{M}$ to LP_1 (see figure 4). Let the index[8] of LP_0 be I_0 when $\mathcal{M}$ is generated, and the index of LP_1 be J_0 when

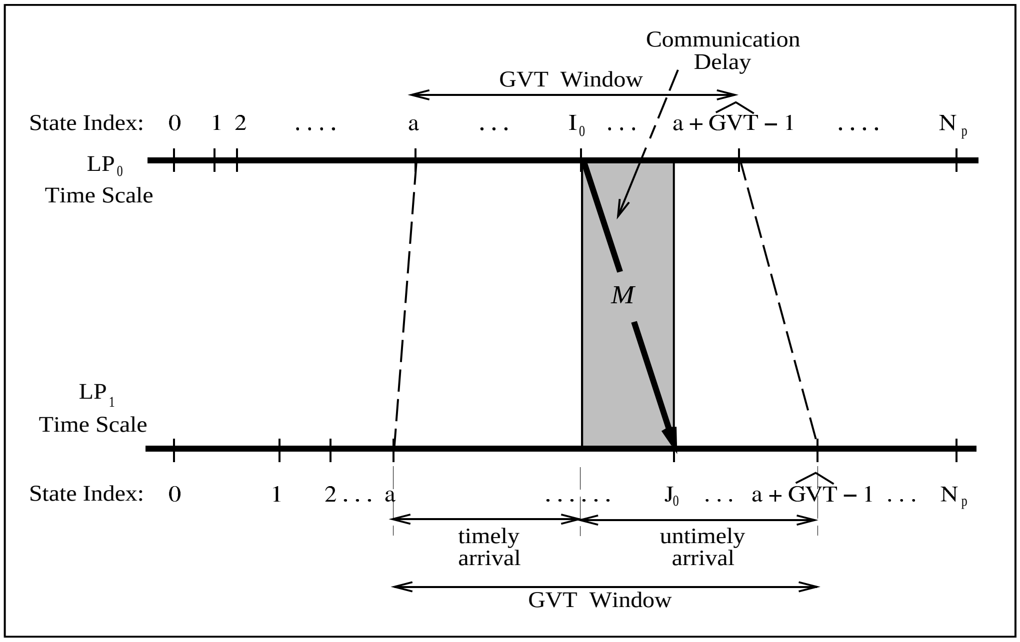

Figure 4: Causality Error

$\mathcal{M}$ is executed, where $0 \leq I_0 \leq N_p$, and $0 \leq J_0 \leq N_p$. We observe that $\mathcal{M}$ will become a straggler in LP_1 provided the timestamp of $\mathcal{M}$ (denoted by LVT_{I_0,LP_0}) is less than the LVT of the receiving LP at index J_0 (denoted by LVT_{J_0,LP_1}). Since LP_1 processes $\mathcal{M}$ at index J_0, the probability for the message to be generated when LP_0 is at index $J_0 - c$, where c is the number of events processed by LP_1 during the communication delay, is the highest among other event numbers in LP_0. We therefore assign a normalized discrete pdf which is peaked at $max(J_0 - c, 0)$ to I_0.

Let $rb(I_0, J_0) = \Pr(LVT_{I_0,LP_0} < LVT_{J_0,LP_1})$. The rollback probability (due to straggler) after the J_0-th state is executed is

$$RB(J_0)^+ \quad = \quad \sum_{I_0=0}^{\widehat{GVT}-1} f_{N_{max(J_0-c,0)}}(I_0) \times rb(I_0, J_0) \tag{4}$$

2.3 Halt Probability

Suppose an LP at index J_0 receives a straggler $\mathcal{M}$ sent by the preceding LP at index I_0 (see figure 5), and $\mathcal{M}$ causes the destined LP to rollback d_0 events, where $d_0 \leq J_0$. During the restoration, the ideal state vector to be restored should be the one saved at index $(J_0 - d_0)$ so that the simulator does not need to coast forward. In the cost model we have to make use of a halt probability $(halt\,^{I_0}_{J_0}(d_0))$, which is the probability that the rollback will stop after d_0 events are undone. The following observations are made for the abstraction of $halt\,^{I_0}_{J_0}(d_0)$:

- $LVT_{I_0,LP_0} > LVT_{J_0-d_0,LP_1}$. Otherwise the rollback will still continue after d_0 events are undone.

[7]We assume that all LPs cannot proceed beyond the GVT window until the next GVT is computed.

[8]The index refers to the current event index of LP from hereafter.

- $LVT_{I_0,LP_0} < LVT_{J_0-d_0+1,LP_1}$. Otherwise the number of events undone is less than d_0.

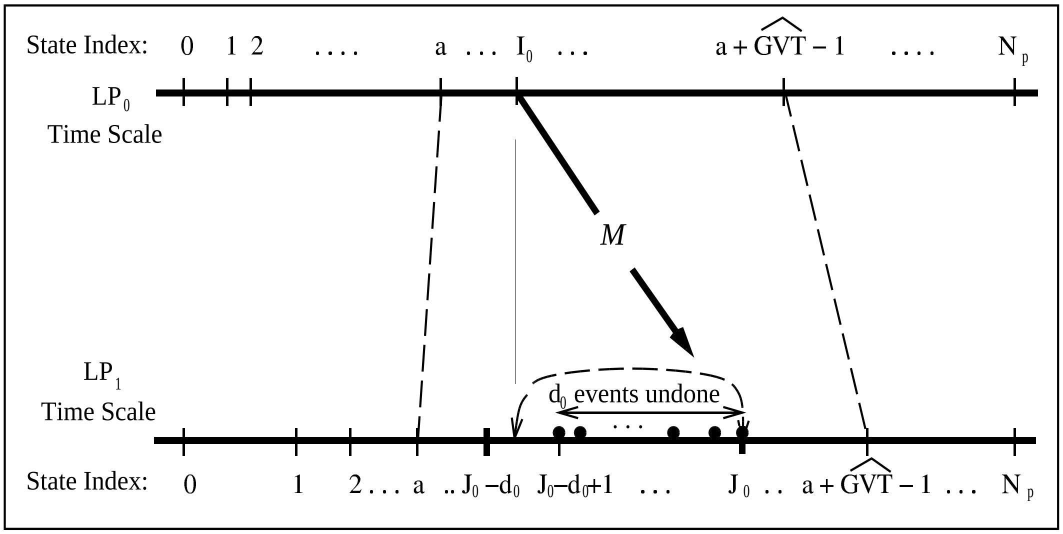

Figure 5: Rollback Events

Since a rollback cannot go below the lower bound of the GVT window, we impose the total probability constraint on $halt_{J_0}^{I_0}(d_0)$. In general, we ensure the condition

$$\sum_{d_k=1}^{J_k} halt_{J_k}^{I_k}(d_k) \quad = \quad 1$$

for $k \geq 0$. This is done by normalizing the *halt* probability with respect to its sum as follows:

$$halt_{J_k}^{I_k}(d_k) \quad = \quad \frac{(1 - rb(I_k, J_k - d_k)) \times rb(I_k, J_k - d_k - 1)}{R\_SUM_{J_k}^{I_k}}$$

$$\text{where} \quad R\_SUM_{J_k}^{I_k} \quad = \quad \sum_{d_k=1}^{J_k} (1 - rb(I_k, J_k - d_k)) \times rb(I_k, J_k - d_k + 1) \tag{5}$$

3 The Checkpointing Cost Model

The proposed cost model is parameterized by two categories of wall clock time, namely the coast forward effort (T_{CCF}) and the state saving cost (T_{state}). Consider a scenario where the J_0-th event, $0 \leq J_0 \leq \widehat{GVT} - 1$, is executed in an LP, and subsequently it receives a straggler $\mathcal{M}$ after the J_1-th event is executed, where $J_0 + 1 \leq J_1 \leq \widehat{GVT} - 1$ (see figure 6). If $\mathcal{M}$ undo (or rolls back) $J_1 - J_0$ events and the state vector at J_0 has been saved, such a state will be restored and $\mathcal{M}$ can be executed immediately. Otherwise, the straggler will continue to undo the executed events until a saved state is found. Suppose the state restored corresponds to the J_s-th event, where $O \leq J_s \leq J_0 - 1$. The TW simulator will have to redo (or re-execute) from the $(J_s + 1)$-th event to the J_0-th event before $\mathcal{M}$ can be executed[9].

Let $T_{CCF}(J_0)$ be the coast forward effort to re-execute from the $(J_s + 1)$-th event to the J_0-th event and $\overline{T_{CCF}(J_0)}$ be the expectation. In our checkpointing scheme, the system state is saved at index J_0 provided $\overline{T_{CCF}(J_0)} > T_{state}$. To compute the expected coast forward effort, we also have to compute $Pr(J_0, J_1)$, which is the probability for a rollback to undo from the J_1-th event to the $(J_0 + 1)$-th event. This is derived based on the following observations:

[9]This is also called the coast forward phase.

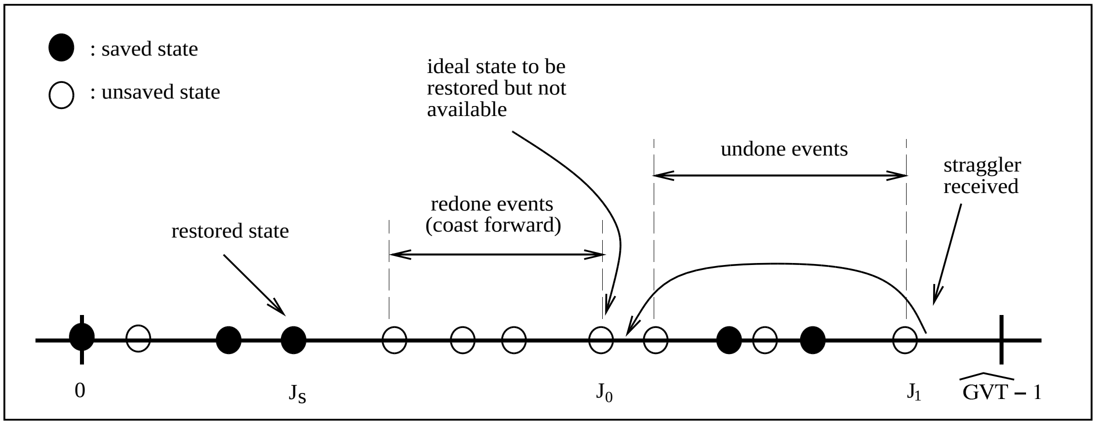

Figure 6: Coasting Forward After a State is Restored

- The straggler arrives immediately after the J_1-th event is executed, thereby causing the LP to rollback.
- The rollback halts after $J_1 - J_0$ events are undone provided the state at index J_0 has been saved.

By equations (4) and (5), we have

$$Pr(J_0, J_1) = RB(J_1)^+ \times halt_{J_0}^{J_1}(J_1 - J_0) \tag{6}$$

However, if the state at index J_0 is not saved, the rollback will continue until a saved state is found. In this case a coast forward effort is required. Since the rollback can occur only when $J_1 > J_0$, we derive the expected coast forward effort based on the following summation:

$$\overline{T_{CCF}(J_0)} = \begin{cases} \sum_{J_1=J_0+1}^{\widehat{GVT}-1} Pr(J_0, J_1) \times T_{CCF}(J_0) & \text{if } J_0 > 0 \\ 0 & \text{if } J_0 = 0 \end{cases}$$

where the value of coast forward effort $T_{CCF}(J_0)$ is a multiple of T_{event}. Figure (7) shows the back-tracking algorithm used in the computation. Given an index J_0 and the state is not saved, the coast forward effort required is repeatedly computed based on the expected coast forward effort at

if $(\overline{T_{CCF}(J_0 - 1)} > T_{state})$ $\quad T_{CCF}(J_0) = T_{event}$

else if $(\overline{T_{CCF}(J_0 - 2)} > T_{state})$ $\quad T_{CCF}(J_0) = 2 \times T_{event}$

  else if $(\overline{T_{CCF}(J_0 - 3)} > T_{state})$ $\quad T_{CCF}(J_0) = 3 \times T_{event}$

  :

  :

    else if $(\overline{T_{CCF}(1)} > T_{state})$ $\quad T_{CCF}(J_0) = J_0 \times T_{event}$

     else $T_{CCF}(J_0) = (J_0 + 1) \times T_{event}$

Figure 7: Back-Tracking Algorithm for Computing the Coast Forward Effort

the lower indices until one of the expected value is greater than the state saving cost. Subsequently, the actual coast forward effort is computed by the product of the number of redone events and the event granularity in wall clock time. Such a back-tracking algorithm implements the proposed checkpointing scheme where a state vector is saved provided the expected gain is larger than the expected loss in wall-clock time.

4 Model Validation and Performance Analysis

We implemented three checkpointing schemes, including the conventional (or frequent) approach, infrequent approach and the proposed cost model on the Fujitsu AP3000 distributed-memory parallel computer using the simulation workbench called SPaDES/C++ (Structured Parallel Discrete-Event Simulation) [27]. The modular design of SPaDES/C++ supports experimental research in synchronization protocols, and ease of parallel simulator development without dealing with the intricacies of simulation synchronization and parallelism. To handle the spawning, communication, and synchronization of processes, the PVM (Parallel Virtual Machine) library [6] is adopted.

Four parameter values used in the cost model are obtained by taking measurements (see table 2) on the implementation platform, Fujitsu AP3000 distributed-memory parallel computer. The values for computation costs (T_{event} and T_{state}) in table 2 are obtained by timing the execution time of the respective code segments in the simulation program over 1000 iterations and taking their average. The buffer access time (T_{buffer}) is obtained by clocking the elapsed time of PVM code

parameter	time (μsec)
T_{event}	1200
T_{state}	990
T_{buffer}	2750
$T_{transit}$	1290

Table 2: Granularity of Parameter

segment for packing and unpacking the message and taking their average over 1000 iterations. As additional protocols are required by the PVM to allocate memory space for the transmission and reception buffers, T_{buffer} has a high value as compared to the other measurements. Transmission time ($T_{transit}$) is also obtained by clocking the elapsed time of two ping-pong programs over 1000 iterations and taking their average. The communication delays in number of processed events is $c = \lceil \frac{2\times 2750+1290}{1200} \rceil = 6$. The GVT interval chosen is $\widehat{GVT} = 50$ after a series of sample runs to get the least elapsed time. The rollback probability and the expected coast forward cost are computed in advance, and implemented by a look-out table in the simulation program. Performance figures presented below have been averaged over 50 replicated simulation runs.

4.1 Application Examples

Exponential distribution is assumed for arrival time and service time. Figure 8 consists of (i) MIN (Feed-Forward configuration) and (ii) Torus (Feedback configuration). As for the 8×8 Omega MIN, the diameter of the LP interconnection is 3, and the deviation of the current state number is $\chi \approx 3 \times 6 + (3-1) \times 2 = 22$. The mean inter-arrival time used in the packet generator and the mean service time used in each switching element are $\frac{1}{100}$ second. The 4×4 torus consists of 16 nodes each with the same mean service time of $\frac{1}{60}$ second. For the $n \times n$ torus network, the diameter is $d = n$ due to the feedback connection so $\chi \approx 4 \times 6 + (4-1) \times 2 = 30$. The routing on the torus network is uniformly distributed on the four directions.

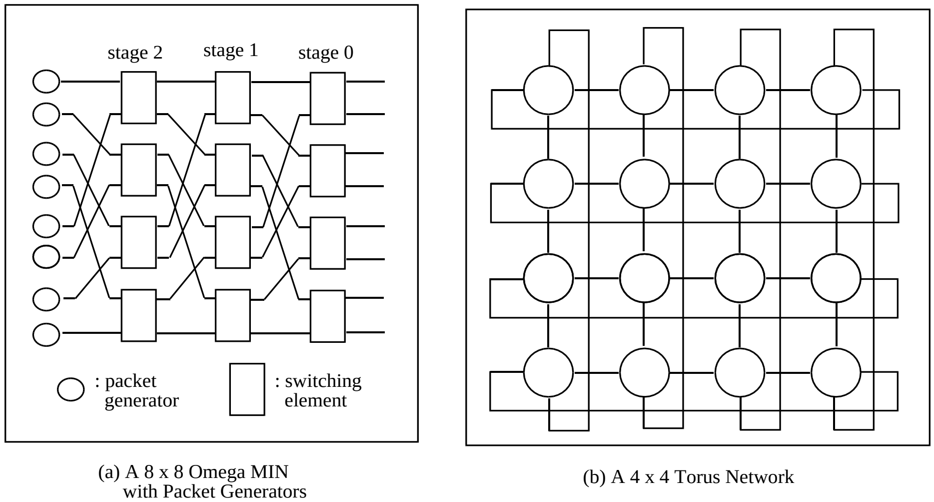

Figure 8: Application Examples

4.2 Checkpointing and Coasting Forward Overheads

Let $\frac{EvCF}{RB} = \frac{total\ no.\ of\ coasted\ forward\ events}{total\ no.\ of\ rollback\ occurrences}$ be the average number of events coasted forward for a rollback, and $hit\ ratio = \frac{no.\ of\ rollback\ occurrences\ where\ the\ restored\ state\ corresponds\ to\ index\ J_0}{total\ no.\ of\ rollback\ occurrences}$ be the percentage of rollback occurrences where coasting forward is not needed (refer to figure 6), and $\frac{State}{EvExe} = \frac{total\ no.\ of\ states\ saved}{total\ no.\ of\ executed\ events}$ be the percentage of the number of states saved with respect to the number of events executed. Tables 3 and 4 compares the effectiveness of the cost model against two checkpointing schemes. As observed, the conventional scheme ($w = 1$) has a 100% hit ratio and

scheme	$\frac{EvCF}{RB}$	$hit\ ratio$	$\frac{State}{EvExe}$
frequent ($w = 1$)	0	100%	100%
infrequent ($w = 20$)	6.5	7.3%	5%
probabilistic cost-based	3.4	87.5%	16%

Table 3: Comparison of Overheads - 8×8 MIN

scheme	$\frac{EvCF}{RB}$	$hit\ ratio$	$\frac{State}{EvExe}$
frequent ($w = 1$)	0	100%	100%
infrequent ($w = 20$)	6.9	7.1%	5%
probabilistic cost-based	3.6	83.4%	19%

Table 4: Comparison of Overheads - 4×4 Torus

coasting forward is not needed because the state vector is saved whenever an event is executed. The overheads incurred by the infrequent approach vary with the size of the checkpointing interval (w) and we present the best experimental result when $w = 20$. On the average, the number of states saved for the infrequent scheme is inversely proportional to w, and the number of coasted forward events is proportional to the interval. The probabilistic cost approach outperforms the infrequent approach for the number of coasted forward events and hit ratio, but it saves more states than the infrequent scheme. The aggregate effect of these factors to the simulation elapsed time is analyzed in the next section.

4.3 Elapsed Time

Figures 9 and 10 show that the elapsed time of the proposed cost model is better than that of the other two schemes in application examples. This effect is due to the rollback probability and cost-effectiveness considerations which ensure that the decision to save (or not to save) a state vector will lead to a net gain in wall clock time. Although the conventional scheme has a 100% hit ratio and does not incur any overhead to coast forward the simulator (refer to tables 3 and 4), its

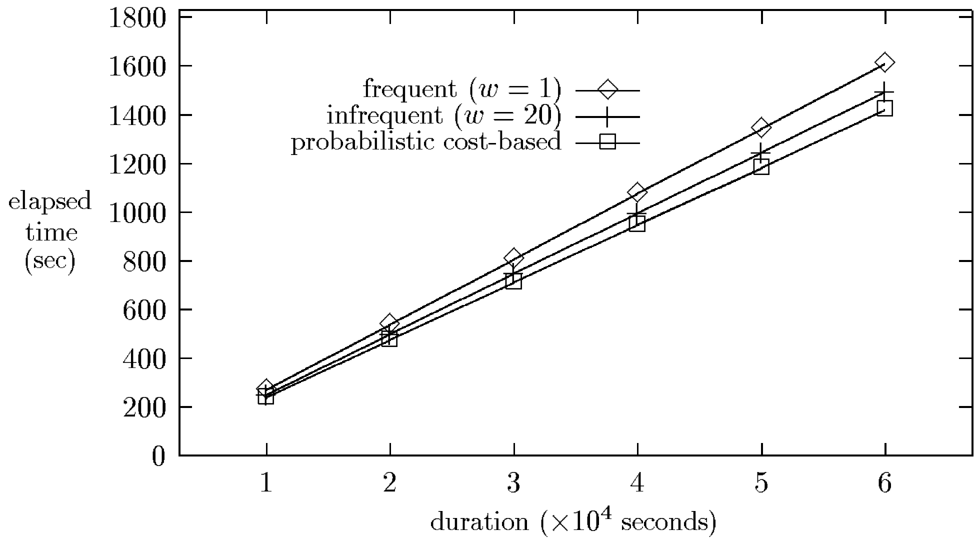

Figure 9: Elapsed Time of MIN Simulation

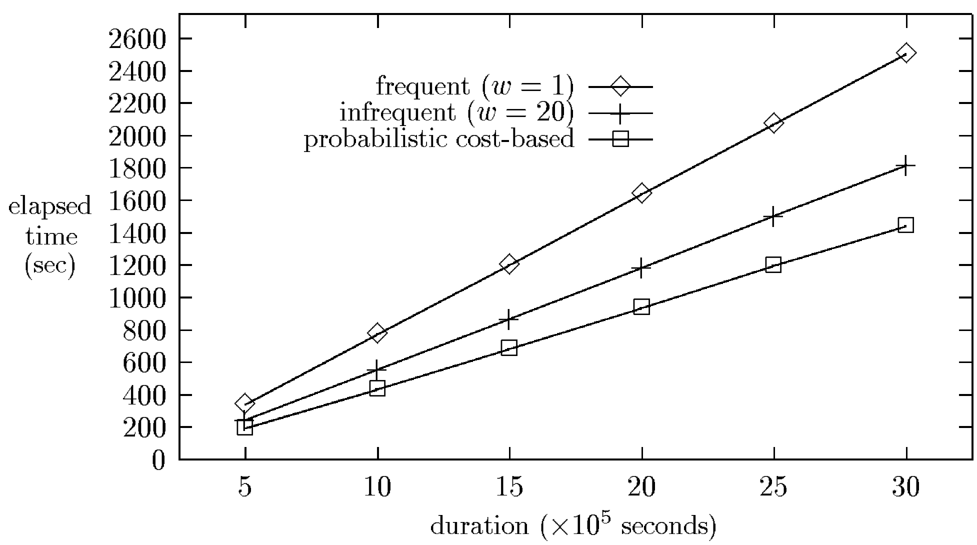

Figure 10: Elapsed Time of Torus Simulation

elapsed time does not outperform the other two schemes due to the huge amount of time incurred in saving the state vectors. On the other hand, the infrequent scheme has reduced the state saving overhead but the saved states are statically selected without any consideration for their usefulness or vulnerability to rollback risk. As such, the infrequent scheme has incurred additional effort to coast forward the simulator. As compared to the state saving overhead, the gain in not saving the state vectors in the infrequent scheme outweighs its loss in coasting forward the simulator, thus it yields a net gain in overall elapsed time as compared to the conventional approach. Out of the three checkpointing schemes, the probabilistic cost model has the best performance. Although the cost model incurs a larger state saving overhead as compared to that of infrequent scheme (see tables 3 and 4), its hit ratio is substantially higher, i.e., a higher probability for the TW simulator not to incur the coast forward overhead when a causality error occurs. Even when coasting forward is need, the number of coasted events in the cost model is also smaller as compared to that of infrequent scheme. On the average, the cost model reduces the elapsed time by 12% as compared to the conventional checkpointing scheme and 15% as compared to the infrequent scheme for MIN simulation, and 42% and 20% respectively for the torus simulation.

5 Conclusions and Future Work

While the frequent checkpointing scheme incurs a substantial overhead in saving the system states, the infrequent approach also introduces a coast forward risk in redoing the executed events. Thus, a cost model is necessary to ensure that the decision to save a state or not will result in a net gain. This paper uses a probabilistic cost model and considers two factors, namely coast forward effort and state saving overhead so that the checkpointing decision is cost effective. The proposed model considers a homogeneous system (both LPs and PEs are assumed to be homogeneous) and derives the rollback probability due to the arrival of straggler. A back-tracking algorithm is used to compute the coast forward effort, and the system state is saved only if the expected coast forward effort is larger than the state saving cost. As the rollback probability and the expect coast forward effort are computed in advance and implemented as a look-out table, the cost model does not incur substantial overhead. Our implementation results as compared to two checkpointing schemes show that the probabilistic cost model approach is effective in reducing the overall elapsed time in both feed-forward and feedback configurations. We are extending the cost model to cover heterogeneous simulation and platform through different parameterizations of LVT advancement rates and communication delays respectively, and to consider the impact of cascading rollbacks on the coast forward effort.

References

[1] R.E. Bryant, *"Simulation of Packet Communications Architecture Computer Systems"*, MIT-LCS-TR-188, Massachusetts Institute of Technology, 1977.

[2] K.M. Chandy and J. Misra, *"Distributed Simulation: A Case Study in Design and Verification of Distributed Program"*, IEEE Trans. on Software Engineering, Vol. SE-5, No. 5, pp. 440-452, September 1979.

[3] S. R. Das and R. M. Fujimoto, *"Adaptive Memory Management and Optimism Control in Time Warp"*, ACM Trans. On Modeling and Computer Simulation, Vol. 7. No. 2, pp. 239-271, April 1997.

[4] S. Franks, F. Gomes, B. Unger and J. Cleary. *"State Saving for Interactive Optimistic Simulation"*, Proc. of 11th Workshop on Parallel and Distributed Simulation, pp. 72-79, June 1997.

[5] R.M. Fujimoto, *"Parallel Discrete Event Simulation"*, Comm. of ACM, Vol. 33, No. 10, pp. 31-53, October 1990.

[6] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderan, *"PVM 3 User's Guide and Reference Manual"*, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, May 1993.

[7] F. Gomes, S. Franks, B. Unger, J. Cleary, *"Multiplexed State Saving for Bounded Rollback"*, Winter Simulation Conference, 1997.

[8] D. R. Jefferson, *"Virtual Time"*, ACM Transactions on Programming Languages and Systems, Vol. 7, No. 3, pp. 404-425, July 1985.

[9] D. R. Jefferson, *"Virtual Time II: Storage Management in Distributed Simulation"*, Proc. of the 9th Annual ACM Symposium on Principles of Distributed Computing, pp. 75-89, 1990.

[10] Y. B. Lin and E. D. Lazowska, *"Reducing the State Saving Overhead for Time Warp Parallel Simulation"*. Technical Report 90-02-03, Dept. of Computer Science, University of Washington, Seattle, Washington, February 1990.

[11] Y. B. Lin and E. Lazowska, *"Optimality Consideration for Time Warp Parallel Simulation"*, in Proc. of 1990 SCS Multiconference on Distributed Simulation, pp. 29-34, 1990.

[12] Y. B. Lin, B. R. Preiss, W.M. Loucks and E. D. Lazowska, *"Selecting the Checkpoint Interval in Time Warp Simulation"*, Proc. 7th Workshop on Parallel and Distributed Simulation, pp. 3-10, May 1993.

[13] Y. B. Lin and B. R. Preiss, *"Optimal Memory Management for Time Warp Parallel Simulation"*, ACM Trans. on Modelling and Computer Simulation, Vol. 1, No. 4, pp. 283-307, October 1991.

[14] J. Misra and K. M. Chandy, *"Asynchronous Distributed Simulation via a Sequence of Parallel Computations"*, Communication of the ACM, Vol. 24, No. 4, pp. 198-206, April 1981.

[15] J. Misra, *"Distributed Discrete-Event Simulation"*, Computing Surveys, Vol. 18, No. 1, pp. 39-65, March 1986.

[16] D. Nicol and R. Fujimoto, *"Parallel Simulation Today"*, in Annals of Operations Research: Simulation and Modeling, edited by O. Balci, Vol. 53, pp. 249-286, 1994.

[17] D. Nicol and X. Liu, *"The Dark Side of Risk"*, Proc. of 11th Workshop on Parallel and Distributed Simulation (PADS'97), Lockenhaus, Austria, IEEE Computer Society Press, pp. 188-195, June 10-13, 1997.

[18] B. R. Preiss, I. D. MacIntyre, W. M. Loucks, *"On the Trade-Off between Time and Space in Optimistic Parallel Discrete-Event Simulation"*, Proc. of the SCS Multiconference on Distributed Simulation, Vol. 24, No. 3, pp. 33-42, January 1992.

[19] F. Quaglia and V.Cortellessa, *"Rollback-Based Parallel Discrete Event Simulation by Using Hybrid State Saving"*, Proc. 9th European Simulation Symposium, pp. 275-279, October 1997.

[20] F. Quaglia, *"Event History Based Sparse State Saving in Time Warp"*, Proc. of 12th Workshop on Parallel and Distributed Simulation (PADS'98), Alberta, Canada, IEEE Computer Press, pp. 72-79, May 26-29, 1998.

[21] F. Quaglia, *"Combining Period and Probabilistic Checkpointing in Optimistic Simulation"*, Proc. of 13th Workshop on Parallel and Distributed Simulation (PADS'99), Georgia, USA, IEEE Computer Press, pp. 109- 116, May 1-4, 1999.

[22] R. Ronngren and R. Ayani, *"Adaptive Checkpointing in Time Warp"*, Proc. of the 8th Workshop on Parallel and Distributed Simulation (PADS'94), ACM Vol. 24, No. 1, pp. 110-117, July 1994.

[23] S. Skold and R. Ronngren, *"Event Sensitive State Saving in Time Warp Parallel Discrete Event Simulation"*, Proc. of 1996 Winter Simulation Conference, December 1996.

[24] H. M. Soliman, *"On the Selection of the State Saving Strategy In Time Warp Parallel Simulations"*, Trans. of The Society for Computer Simulation International, March 1999.

[25] S.C. Tay Y.M. Teo and Rassul Ayani, *"Performance Prediction of Optimistic Simulation with Rollback Thrashing"*, Technical Report TR41/97, Department of Information Systems and Computer Science, National University of Singapore, pp. 24, November 1997.

[26] S.C. Tay Y.M. Teo and R. Ayani, *"Performance Analysis of Time Warp Simulation with Cascading Rollbacks"*, Proc. of 12th Workshop on Parallel and Distributed Simulation (PADS'98), pp.30-37, IEEE Computer Society Press, Canada, May 1998.

[27] Y.M. Teo, S.C. Tay and K.T. Kong, *"Structured Parallel Simulation Modeling and Programming"*, Proc. of the 31st Annual Simulation Symposium, Boston, Massachusetts, USA, IEEE Computer Society Press, pp. 135-142, April 5-9, 1998.

[28] D. West and K. Panesar, *"Automatic Incremental State Saving"*, Proc 10th Workshop on Parallel Simulation, pp. 78-85, May 1996.

[29] C. H. Young, N. B. Abu-Ghazaleh, P. A. Wisley, *"OFC:A Distributed Fossil-Collection Algorithm for Time Warp"*, 12th International Symposium on Distributed Computing, 1998.

[30] C. H. Young, N. B. Abu-Ghazaleh, P. A. Wisley, R. Radharamanan, *"Performance Benefits of Optimism in Fossil Collection"*, Hawaii International Conference on System Sciences, 1999.

Appendix A

We construct a normalized discrete pdf f_{N_w} (see figure 11) as follows. Let χ denote the spread of f_{N_w}. In this pdf χ is approximately equal to the standard deviation of the continuous normal distribution. Let

$$\begin{aligned} coef\_N_w(x) &= \frac{1}{\sqrt{2\pi}\chi} e^{-\frac{(x-w)^2}{2\chi^2}} \\ \text{and} \quad N\_SUM_w &= \sum_{x=0}^{\widehat{GVT}-1} coef\_N_w(x) \end{aligned}$$

For each $w \in \{0, 1, 2, \ldots, \widehat{GVT}-1\}$, the corresponding pdf is defined as

$$f_{N_w}(x) = \begin{cases} \frac{coef\_N_w(x)}{N\_SUM_w} & \text{if } x \in \{0,1,2,\ldots,\widehat{GVT}-1\} \\ 0 & \text{otherwise} \end{cases}$$

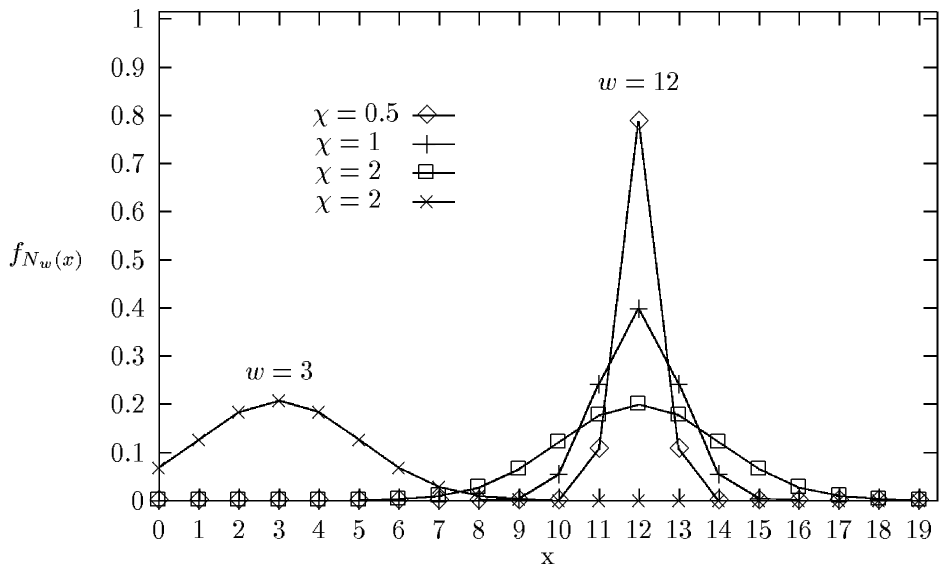

Figure 11: Normalized Distribution with different Peak (w) and Spread (χ)

Note that $f_{N_w}(x)$ has a peak value when $x = w$, and $f_{N_w}(w+i) = f_{N_w}(w-i)$ (symmetrical with respect to w), $i > 1$, if they exist. We also observe that $\forall\ w$

$$\begin{aligned} \sum_{x=0}^{\widehat{GVT}-1} f_{N_w}(x) &= \sum_{x=0}^{\widehat{GVT}-1} \frac{coef\_N_w(x)}{N\_SUM_w} \\ &= \frac{\sum_{x=0}^{\widehat{GVT}-1} coef\_N_w(x)}{N\_SUM_w} \\ &= \frac{N\_SUM_w}{N\_SUM_w} \\ &= 1 \qquad \text{(Total Probability Constraint)} \qquad \square \end{aligned}$$

Appendix B

In the following derivations we use the integrals from [25] directly. From equation (2), we have

$$\int_0^\infty \int_0^y \lambda T^{p-1} \left(e^{-\lambda x} - e^{-\beta x} \sum_{k=0}^{p-2} \frac{(\theta x)^k}{k!} \right) \times \lambda T^{q-1} \left(e^{-\lambda y} - e^{-\beta y} \sum_{l=0}^{q-2} \frac{(\theta y)^l}{l!} \right) dx\, dy$$

$$= \quad \lambda^2\, T^{p+q-2} \times \int_0^\infty \left(e^{-\lambda y} - \sum_{l=0}^{q-2} \frac{(\theta\, y)^l\, e^{-\beta\, y}}{l!} \right) \int_0^y \left(e^{-\lambda x} - \sum_{k=0}^{p-2} \frac{(\theta\, x)^k e^{-\beta\, x}}{k!} \right) dx\, dy \tag{7}$$

To simplify the mathematical expression, we denote the geometric sum $\sum_{i=0}^{n-2} x^i$ by

$$G_n(x) \quad = \quad \frac{1 - x^{n-1}}{1-x}.$$

$$\begin{aligned}
\int_0^y \left(e^{-\lambda x} - \sum_{k=0}^{p-2} \frac{(\theta\, x)^k e^{-\beta\, x}}{k!} \right) dx \quad &= \quad \int_0^y e^{-\lambda x}\, dx - \sum_{k=0}^{p-2} \frac{\theta^k}{k!} \int_0^y x^k e^{-\beta\, x}\, dx \\
&= \quad \frac{1 - e^{-\lambda\, y}}{\lambda} - \sum_{k=0}^{p-2} \theta^k \times \left(\frac{1}{\beta^{k+1}} - \sum_{i=0}^{k} \frac{y^i\, e^{-\beta y}}{\beta^{k+1-i} \times i!} \right) \\
&= \quad \frac{1 - e^{-\lambda\, y}}{\lambda} - \frac{1}{\beta} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k + \sum_{k=0}^{p-2} \sum_{i=0}^{k} \frac{\theta^k\, y^i\, e^{-\beta y}}{\beta^{k+1-i} \times i!} \\
&= \quad \frac{1 - e^{-\lambda\, y}}{\lambda} - \frac{G_p(\frac{\theta}{\beta})}{\beta} + \frac{1}{\beta} \sum_{k=0}^{p-2} (\frac{\theta}{\beta})^k \sum_{i=0}^{k} \frac{\beta^i}{i!} y^i\, e^{-\beta y}
\end{aligned} \tag{8}$$

Substitute equation (8) to (7), we have

$$\int_0^\infty \left(e^{-\lambda y} - \sum_{l=0}^{q-2} \frac{(\theta\, y)^l\, e^{-\beta\, y}}{l!} \right) \left(\frac{1 - e^{-\lambda\, y}}{\lambda} - \frac{G_p(\frac{\theta}{\beta})}{\beta} + \frac{1}{\beta} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k \sum_{i=0}^{k} \frac{\beta^i}{i!} y^i\, e^{-\beta y} \right) dy$$

$$\begin{aligned}
= \quad & \int_0^\infty [\, \frac{e^{-\lambda\, y} - e^{-2\lambda\, y}}{\lambda} - \frac{G_p(\frac{\theta}{\beta})}{\beta} e^{-\lambda\, y} + \underbrace{\frac{1}{\beta} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k \sum_{i=0}^{k} \frac{\beta^i}{i!} y^i\, e^{-(\lambda+\beta) y}}_{H_1} \\
& \quad \underbrace{- \frac{1}{\lambda} \sum_{l=0}^{q-2} \frac{(\theta\, y)^l\, e^{-\beta\, y}}{l!}}_{H_2} + \underbrace{\frac{1}{\lambda} \sum_{l=0}^{q-2} \frac{(\theta\, y)^l\, e^{-(\lambda+\beta)\, y}}{l!}}_{H_3} + \underbrace{\frac{G_p(\frac{\theta}{\beta})}{\beta} \sum_{l=0}^{q-2} \frac{(\theta\, y)^l\, e^{-\beta\, y}}{l!}}_{H_4} \\
& \quad \underbrace{- \frac{1}{\beta} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k \sum_{i=0}^{k} \frac{\beta^i}{i!}\, y^{l+i}\, e^{-2\beta y}}_{H_5} \,]\, dy \\
= \quad & \frac{1}{2\lambda^2} - \frac{G_p\left(\frac{\theta}{\beta}\right)}{\lambda\, \beta} + \frac{G_p\left(\frac{\theta}{\beta}\right) - \frac{\beta}{\lambda+\beta} G_p\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda\, \beta} - \frac{G_q(\frac{\theta}{\beta})}{\lambda\, \beta} + \frac{G_q\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda(\lambda+\beta)} \\
& + \frac{G_p(\frac{\theta}{\beta}) \times G_q(\frac{\theta}{\beta})}{\beta^2} - \frac{1}{2\beta^2} \sum_{l=0}^{q-2} \frac{(\frac{\theta}{2\beta})^l}{l!} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k \sum_{i=0}^{k} \frac{\left(\frac{1}{2}\right)^i \times (l+i)!}{i!} \\
= \quad & \frac{1}{2\lambda^2} - \frac{G_p\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda(\lambda+\beta)} - \frac{G_q(\frac{\theta}{\beta})}{\lambda\, \beta} + \frac{G_q\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda(\lambda+\beta)} + \frac{G_p(\frac{\theta}{\beta}) \times G_q(\frac{\theta}{\beta})}{\beta^2} \\
& - \frac{1}{2\beta^2} \sum_{l=0}^{q-2} \frac{(\frac{\theta}{2\beta})^l}{l!} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k \sum_{i=0}^{k} \left(\frac{\left(\frac{1}{2}\right)^i \times (l+i)!}{i!} \right)
\end{aligned} \tag{9}$$

We evaluate H_1 to H_5 as follows.

$$\begin{aligned}
H_1 \quad &= \quad \frac{1}{\beta} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k \sum_{i=0}^{k} \frac{\beta^i}{i!} \int_0^\infty y^i\, e^{-(\lambda+\beta) y}\, dy \\
&= \quad \frac{1}{\beta} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \right)^k \sum_{i=0}^{k} \frac{\beta^i}{i!} \times \frac{i!}{(\lambda+\beta)^{i+1}}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\beta(\lambda+\beta)} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k \sum_{i=0}^{k} \left(\frac{\beta}{\lambda+\beta}\right)^i \\
&= \frac{1}{\beta(\lambda+\beta)} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k \frac{1-(\frac{\beta}{\lambda+\beta})^{k+1}}{1-\frac{\beta}{\lambda+\beta}} \\
&= \frac{1}{\beta(\lambda+\beta)(\frac{\lambda+\beta-\beta}{\lambda+\beta})} \left(\sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k - \frac{\beta}{\lambda+\beta} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta} \times \frac{\beta}{\lambda+\beta}\right)^k \right) \\
&= \frac{1}{\beta\,\lambda} \left(\sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k - \frac{\beta}{\lambda+\beta} \sum_{k=0}^{p-2} \left(\frac{\theta}{\lambda+\beta}\right)^k \right) \\
&= \frac{G_p\left(\frac{\theta}{\beta}\right) - \frac{\beta}{\lambda+\beta} G_p\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda\,\beta} \qquad (10)
\end{aligned}
$$

$$
\begin{aligned}
H_2 &= \frac{1}{\lambda} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \int_0^\infty y^l\, e^{-\beta\, y}\, dy \\
&= \frac{1}{\lambda} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \times \frac{l!}{\beta^{l+1}} \\
&= \frac{1}{\lambda\,\beta} \sum_{l=0}^{q-2} \left(\frac{\theta}{\beta}\right)^l \\
&= \frac{G_q(\frac{\theta}{\beta})}{\lambda\,\beta} \qquad (11)
\end{aligned}
$$

$$
\begin{aligned}
H_3 &= \frac{1}{\lambda} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \int_0^\infty y^l\, e^{-(\lambda+\beta)\, y}\, dy \\
&= \frac{1}{\lambda} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \times \frac{l!}{(\lambda+\beta)^{l+1}} \\
&= \frac{1}{\lambda(\lambda+\beta)} \sum_{l=0}^{q-2} \left(\frac{\theta}{\lambda+\beta}\right)^l \\
&= \frac{G_q\left(\frac{\theta}{\lambda+\beta}\right)}{\lambda(\lambda+\beta)} \qquad (12)
\end{aligned}
$$

$$
\begin{aligned}
H_4 &= \frac{G_p(\frac{\theta}{\beta})}{\beta} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \int_0^\infty y^l\, e^{-\beta\, y}\, dy \\
&= \frac{G_p(\frac{\theta}{\beta})}{\beta} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \times \frac{l!}{\beta^{l+1}} \\
&= \frac{G_p(\frac{\theta}{\beta})}{\beta^2} \sum_{l=0}^{q-2} \left(\frac{\theta}{\beta}\right)^l \\
&= \frac{G_p(\frac{\theta}{\beta}) \times G_q(\frac{\theta}{\beta})}{\beta^2} \qquad (13)
\end{aligned}
$$

$$
\begin{aligned}
H_5 &= \frac{1}{\beta} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k \sum_{i=0}^{k} \frac{\beta^i}{i!} \int_0^\infty y^{l+i}\, e^{-2\beta y}\, dy \\
&= \frac{1}{\beta} \sum_{l=0}^{q-2} \frac{\theta^l}{l!} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k \sum_{i=0}^{k} \frac{\beta^i}{i!} \frac{(l+i)!}{(2\beta)^{l+i+1}} \\
&= \frac{1}{2\beta^2} \sum_{l=0}^{q-2} \frac{(\frac{\theta}{2\beta})^l}{l!} \sum_{k=0}^{p-2} \left(\frac{\theta}{\beta}\right)^k \sum_{i=0}^{k} \left(\frac{\left(\frac{1}{2}\right)^i \times (l+i)!}{i!} \right) \qquad (14)
\end{aligned}
$$