Optimistic Fossil Collection for Time Warp Simulation*

Christopher H. Young and *Philip A. Wilsey*
Computer Architecture Design Laboratory
Dept. of ECECS, PO Box 210030
Cincinnati, OH 45221–0030

Abstract

Optimistic Fossil Collection is a fully distributed mechanism to reclaim memory from the state and event histories of a Time Warp simulation. Each fossil collector executes with a Logical Process (LP) and operates independently of other fossil collectors. More precisely, each fossil collector examines event arrival times and creates a statistical model of the expected variance from LVT. From this model, it is possible to determine the probability that the LP will, in the future, rollback distance X from LVT. Thus, the fossil collector can examine the time-stamps of items in the state and event histories to find the probability that they will be needed in the future. Comparing this probability against a user-specified risk factor, the fossil collector decides if the item can be marked as a fossil and scavenged. Optimistic fossil collection does, however, allow for the possibility for simulation failure. Consequently, it may be desirable to periodically have complete checkpoints taken and archived during the simulation for possible restart with a smaller risk factor specified.

This method of memory management assumes that there is an underlying stationary distribution for the rollback lengths during a time interval t. This is a reasonable assumption since empirical studies have shown that rollback lengths in Time Warp are relatively constant in length. This assumption can, however, also be relaxed and models that operate without an underlying assumption about the distribution of rollback lengths. This paper reviews the design and implementation of two rollback models for optimistic fossil collection. The first assumes a geometrically distributed rollback length; the second assumes an arbitrary distribution of rollback lengths with fixed mean and variance. An implementation of the mechanism is also reported that describes our experiences with one implementation of optimistic fossil collection.

1 Introduction

A Time Warp synchronized parallel discrete event driven simulator is organized as a collection of asynchronously executing processes that communicate by exchanging time-stamped event messages [12, 14]. The concurrent processes operate as independent distributed discrete event simulators with no explicit synchronization occurring between simulators. Coordination between simulators to maintain the causal relations between events is achieved by sorting event timestamps and rollback. Thus, each simulator maintains state and event histories (sorted in timestamp order) to enable rollback whenever a causality error is discovered.

As global progress (measured in simulation time) is made by the parallel simulation objects, some information in the state and event histories are no longer needed for rollback. The elements of the unneeded histories are called *fossils* and the reclamation of the memory space holding these fossils is called *fossil collection*. Traditionally, Time Warp simulators have implemented fossil collection by comparing history item timestamps to a global simulation time valued called *Global Virtual Time* (GVT). Informally, GVT represents the greatest global time that all of the concurrent simulators have reached (and includes consideration for the time-stamps of messages in transit). Thus, only the history needed to support rollback to GVT is needed; all earlier history information is tagged as fossils. Fossil collection can then be performed to identify additional space for adding new information to the event and state histories.

This paper presents a new, distributed, technique for fossil collection called *Optimistic Fossil Collection* (OFC). OFC operates by establishing a statistical model of rollback behavior and then using this model to establish probabilities that elements from the state and event histories are not fossils. Using this probability and a user defined risk factor, OFC decides if a

*Support for this work was provided in part by the Advanced Research Projects Agency under contract J-FBI-93-116, monitored by the Department of Justice.

specific item can be scavenged (reclaimed). More precisely, when the probability falls below the risk factor, it is available for scavenging. Thus, as new space is required for building additional history information, the oldest members from the histories are examined for possible scavenging. IF none fall below the threshold, then additional (new) space is allocated; otherwise the scavenge-able space is reused. Since fossil identification occurs based on a statistical model, the possibility for failure does occur. Thus, simulators using OFC must also periodically take archival snapshots for possible restart (generally restart will be accompanied by a setting of a lower risk factor). The chief gain with OFC is that it is a fully distributed fossil collection mechanism.

The remainder of this paper is organized as follows. Section 2 we discuss probabilistic simulation. Section 3 discusses fossil collection in Time Warp. A detailed discussion of two statistical models developed for optimistic fossil collection are described in Section 4. Section 5 discusses an implementation of the two models and presents some performance results from a simulation of a parallel (RAID) disk model. Finally, Section 6 contains some concluding remarks.

2 Background

Fossil collection is an integral part of the Time Warp simulation protocol. Fossilized state and event histories need to be removed during the simulation to keep from exhausting memory. The approaches to improve GVT estimation have fallen chiefly along two lines. Those that work to improving the frequency of GVT calculations, thereby reducing the amount of saved history information [2, 8, 9]; and, those that attempt to reduce the number of uncommitted events via cancel-back, bounded time windows, or some other flow control mechanism [6].

D'Souza *et al* [8, 9] use an algorithm called pGVT to improve the frequency and accuracy of GVT estimation. pGVT removes a portion of messaging overhead by letting the LPs decide when to report new GVT information to the GVT Manager. This method allows for a highly accurate GVT estimate that can be used to quickly identify committed events and fossilized state information.

Das and Fujimoto [7] use an adaptive memory management protocol to control the amount of memory used during a Time Warp simulation. The protocol uses fossil collection and cancel-back to keep the distribution of memory between the pool of committed events, uncommitted events, and the free pool at an optimal level. The distribution of memory is automatically adapted using various statistics. An additional technique called "on-the-fly fossil collection" is also proposed by Das, Fujimoto, and others [5]. This technique uses a continuous GVT update mechanism and fossil collects only when new memory space is needed.

Ferscha and Lüthi [10, 11] use adaptive control to reduce the rollback overhead of a Time Warp simulation by limiting the optimism of Time Warp. In several simulations of stochastic petri nets the optimism of Time Warp was limited to improve performance. This was done by creating a probabilistic decision function which takes into accounts the amount of cpu time wasted blocking and the cost of a rollback. A drawback of the method is that an operating system call is required before each message received and executed. This is used to determine the amount of real time elapsed between the receipt of a message and the time it is about to be executed. There is considerable overlap in the approach by Ferscha and optimistic fossil collection. The chief differences (besides the techniques used to build the statistical models) is that Ferscha uses the probabilities to inhibit forward progress and OFC uses the probabilities for fossil identification.

There is also an increasing trend to attempt the dynamic adjustment of simulation parameters at runtime to improve performance. For example, investigators have studied: (i) dynamically sizing checkpoint intervals [19, 23, 24], adaptive bounded time windows [10, 11, 21, 22], optimal memory management [16, 17, 7], and cancellation strategies [25]. These techniques all monitor runtime simulation data (using techniques from non-linear and adaptive control theory [1]) to establish measures on the simulation's performance. Likewise, OFC attempts to monitor past rollback behavior to predict the probability that items from the saved history space are fossils.

3 Fossil Collection

Time Warp simulators require GVT calculations in order to reclaim fossilized state and event histories. This calculation has been required because Time Warp needs to be conservative in its memory reclamation; that is, it needs to ensure that all the state and event information needed to maintain causality constraints (by rollback) is preserved. The GVT bound establishes the lowest time to which an event message can cause an LP to rollback. All events and all states before this time can be reclaimed as long as one state is left before GVT in case a process rollback to GVT.

The calculation of GVT comes at a price. More precisely, virtually all GVT estimation algorithms require the collection of information at a central site [2, 3, 8, 13, 18, 26, 27] which, in some cases, can require considerable messaging overhead. Furthermore, determining the correct frequency for cycling GVT estimation algorithms can be difficult.

In contrast, in optimistic fossil collection, fossil collector independently operates on each LP, independently of the other fossil collectors. Each fossil collector creates a statistical model from the behavior observed and determines a bound on GVT from the statistical model. A user defined risk factor, α, is used to determine how aggressive the fossil collector should be in its collection. A high level of risk implies that the fossil collector will be more optimistic and scavenge more memory, but might have catastrophic rollbacks more often.[1] A low level of risk implies that the fossil collector will be more conservative, but less likely to scavenge a state it may need later.

The risk factor also defines the probability of simulation failure due to collecting a state or event that is needed by the processor. As more memory is collected, the probability of collecting a state that is needed by an LP in the simulation increases. Thus, to maintain causality in the simulation, periodic checkpoints of the entire simulation are needed to allow restarting of the simulation in case of failure.

Optimistic Fossil Collection techniques allow the communication overhead to be reduced by eliminating GVT calculations. The reduction in GVT messages means that the value of the other LP's LVTs must be estimated through their effect on the local LVT. Accordingly, there is a nonzero probability that the estimate is wrong. The reduction in message overhead will come at the price of a close and safe estimate of GVT. There can be an increase in the amount of memory used by the entire simulation since Optimistic Fossil Collection techniques will not bound rollback as closely, or as well, as GVT for all LPs in all simulations. An LP may have a bound for rollbacks that is close to the actual GVT for a high level of risk, but the simulation could be perturbed in such a way that this bound for the LP is too small.[2]

A possible benefit of Optimistic Fossil collection is that some simulations may reduce the amount of memory being used. Basing fossil collection off of the event messages at a LP could allow some LPs to reduce the amount of memory used because fossil collection is

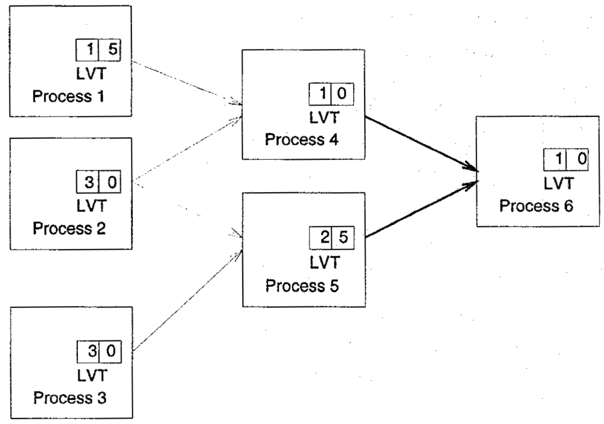

Figure 1: Communicating Time Warp Processes.

based on local message arrival rates. This could reduce the amount of memory for some LPs because there is the possibility of LPs saving less state then if it were saving state in regards to GVT. It is possible for an LP that is doing useful work to be far ahead of GVT to reduce the number of states being saved due to it never being rolled back. Under GVT fossil collection this LP would have to retain unnecessary states because the traditional method of fossil collecting under GVT does not allow an LP to adapt the amount of states saved. For example, assume that there are 6 LPs that communicate as in Figure 1. If the source LPs produce messages with the same timestamp increments and some LPs execute at different rates then it's possible for some LPs to surge ahead and not be rolled back. Consider, if LPs 2, 3, and 5 execute twice as fast as the other LPs then LP 5 will consume more memory than required. It will never be rolled back by any LP other than 2 and 3, but it has to save states based upon LPs 1, 4, and 6. Also all source LPs cannot reclaim memory until GVT passes even though they will never be rolled back.

In these cases, a distributed method of statistically estimating what state and event histories are needed could allow each LP to adapt its fossil collection to the behavior observed at that LP. This could provide the situations where optimistic fossil collection could win over fossil collection via a conservative GVT calculation. Non-communicating LPs would not figure into each other's fossil collection, and source LPs could be less restrained in fossil collection. This could reduce the amount of memory used for some LPs in some simulations.

[1] A catastrophic rollback requires a state, or event, that has been fossil collected, so recovery must be done through an archived checkpoint of the processes event queues and a state.

[2] This would incur a penalty of having to restart the simulation from a checkpoint.

4 Techniques for Optimistic Fossil Collection

An optimistic method for fossil collection allows LPs to reclaim storage as needed on a per LP basis. It provides a mechanism to recover from causality errors in case the estimate is incorrect. Since the correct implementation of GVT algorithms can be one of the hardest part of developing a Time Warp simulator, optimistic methods can shorten the development time of a Time Warp simulator.

The optimistic techniques described in this section assume that the LPs have an underlying stationary distribution for the LVT updates and rollbacks. This assumption is necessary because the parameters estimated at the beginning of the simulation are assumed to remain constant throughout the simulation. These techniques could be modified to operate dynamically by assuming that there is an underlying stationary distribution for time periods $[t, t+\tau]$, and estimating the parameters during these time periods.

The first technique examines only the rollbacks at the particular processor to determine a bound on the rollbacks. The second technique looks at both the LVT advances and rollbacks in order to model the processing behavior between rollbacks, in addition to the rollback behavior.

4.1 Fossil Collection via Rollback Lengths

The first model assumes that the rollbacks at an LP come from a single underlying stationary distribution. Thus, rollbacks can be modeled by estimating the parameters for rollbacks under this distribution. A geometric distribution is used to model the locality of rollbacks, in a Time Warp LP. This distribution was chosen because the probability of different rollback lengths can be derived very easily.

As shown in [28], a bound on the length that an LP can rollback with probability α can be derived after the parameter p is determined. The probability of a rollback of one time unit p is determined using the average rollback length found during the simulation $\bar{\mu}$ since

$$p = \frac{1}{\bar{\mu}}.$$

Thus the probability of a rollback of length l is given by

$$p\{X = l\} = (1-p)^{l-1}p,$$

and the probability that an LP will rollback farther than l is given by

$$p\{X > l\} = (1-p)^{l+1}.$$

Given a user defined probability for failure α, a bound on the length an LP can rollback is calculated by determining the distance where $p\{X > l\} = \alpha$. This is given by

$$l = \frac{\log \alpha}{\log(1-p)} - 1.$$

While this model effectively bounds the distance an LP can rollback, it does not take into account the amount of processing done at the LP between rollbacks. A more robust method would take into account the amount of processing usually done between rollbacks.

4.2 Fossil Collection via LVT updates

The previous model assumed that the length of rollbacks could be modeled by a geometric distribution. The second model described here determines a bound the changes at the LPs LVT.

If an assumption is made that the updates to a LP's LVT come from a single underlying distribution with a finite mean and variance then a bound can be derived by using the Chebyshev inequality. This would bound the changes in the LP's LVT with a known probability, α of being correct.

In the model X denotes an LVT update which is a random variable which comes from a distribution with mean μ and variance σ^2. The probability of how far such a random variable deviates from its mean is given by the Chebyshev inequality as

$$P\{|X-\mu| \geq l\} \leq \frac{\sigma^2}{l^2}. \quad (1)$$

If X represents the change in an LP's LVT and μ and σ represent the mean and variance then equation (1) provides a bound for the possible changes in the LP's LVT.

A bound on the LVT update values can be obtained by choosing an acceptable risk of failure

$$P\{|X-\mu| \geq l\} = \alpha.$$

Using this, the bound, l, can be defined as

$$l \leq \frac{\sigma}{\sqrt{\alpha}}.$$

Since the model assumes knowledge of μ and σ, the bound has to be modified to reflect the fact that these values can only be estimated.

The estimate of μ and σ is calculated based on a sample of LVT changes in the following way. A small collection of independent LVT changes is collected, by sampling based on the outcome of an independent identically distributed sequence of random variables, and an estimate of the true mean for the LVT changes is calculated from the sample $\bar{\mu}$ by using the central limit theorem to determine a confidence interval for the mean μ.

The Central Limit Theorem states that the mean for n independent samples λ_i from a common distribution F can be used to determine an interval estimate for true population mean since as the number of samples increases the deviation of the sample mean from the true population mean becomes normally distributed. Thus, the true mean of the LVT updates can be bounded with a known probability of being correct since

$$P\left\{\frac{\lambda_1+\lambda_2+\cdots+\lambda_n-n\mu}{\sigma\sqrt{n}}\leq a\right\}\rightarrow\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{a}e^{-\frac{x^2}{2}}dx$$

regardless of the distribution of F[3].

After the mean has been calculated the variance is calculated using the collection of sample points using both endpoints of the confidence interval to determine which will give the maximum variance $\bar{\sigma}^2$. This variance is then used to determine a bound on how far the LVT will change.

5 Implementation

The above models were implemented using the WARPED Time Warp simulation kernel developed at the University of Cincinnati [20]. This system implements a Time Warp kernel that parallel applications can use without being aware of the particulars of Time Warp synchronization protocol. The kernel is written in C++, and takes advantage of inheritance to hide implementation details from the users. In addition, it is structured so that investigators of optimizations to Time Warp can implement their optimizations with nominal concerns to the unaffected regions of a Time Warp simulator.

For this effort, OFC, two routines were implemented to collect data for each model. The first routine estimates the average rollback length for the first model by tracking the total length, in time, that the processor has rolled back and number of rollbacks. As the simulation progresses the mean calculated from the ratio of these two values is assumed to converge to the mean of the underlying steady-state distribution.

[3]provided that F has both finite mean and variance

The second routine determines if the current LVT change should be used to update the estimate for the mean and variance. An independent uniform random variable is generated, if the value of the random variable is less than the first decile of its possible range then the current change in LVT is saved in a table. The first decile was chosen arbitrarily so that the samples are taken over a longer period of time. The table of LVT changes is used for determining the confidence interval for the mean and calculating the variance.

Both the Optimistic Fossil Collection algorithms and the pGVT algorithm are implemented as on–the–fly garbage collectors. Each process participates in a GVT calculation as usual, but it also executes code to determine the optimistic bounds for each method. The execution time of this extra code is negligible when compared with the time to send a message from one logical process to another.

The two models were tested against an implementation of the pGVT algorithm on a model of a level 5 Redundant Array of Inexpensive Disks (RAID) [4].

5.1 RAID application model

RAID protects stored information by inducing redundancy into the data stored across the array. RAID level 5 does this by calculating parity information and distributing the parity over the entire array [4]. The redundancy induced by the parity information allows for data to be reconstructed in case of failure. The parity information must be updated on disk writes, but updates are not required for reads. The distribution of the parity information across disks allows for several reads and writes to be processed concurrently.

A queuing model of a level 5 RAID system was created to test the implementation of OFC in the WARPED kernel. The queuing model was constructed using a queuing model library that operates with WARPED. The queuing model can take advantage of the Time Warp synchronization while the RAID model remains oblivious to the details of Time Warp.

The model is a collection of 16 user processes which were derived from a source object in the queuing library. Each process randomly generates a random request for a stripe of random length and location. This request is sent to a fork object. The RAID fork breaks up each request into requests for specific stripe units for disks in the disk array. The fork was derived from a fork primitive which provides general routing capabilities. The fork used for the RAID model required specialized routing capabilities to handle the parity requests for disk writes.

The disks were derived from a server primitive which provides service routines for the user and hand-

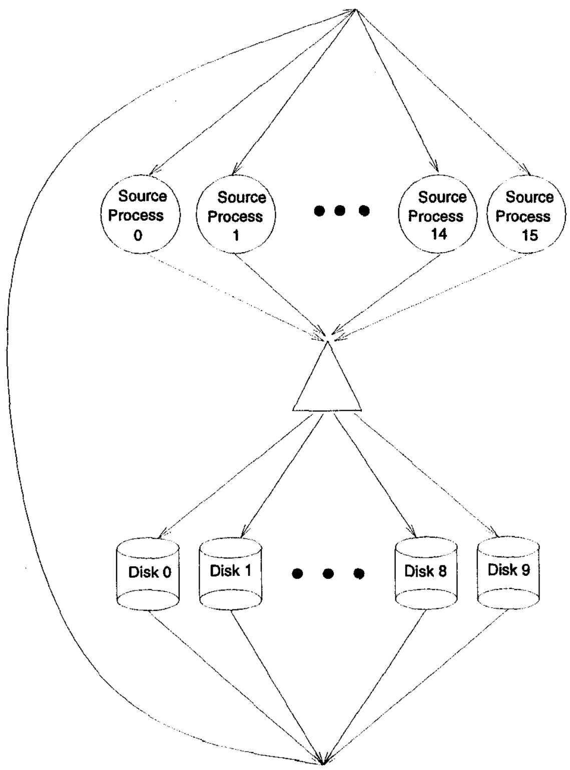

Figure 2: System level model of the RAID simulation.

shaking required for the queue implementation. The RAID servers process the requests in timestamp order. Each server takes a request and calculates the time required to move the heads from the current position using a formula for seek time from [15] where

$$seek\ Time = \begin{cases} 0, & \text{if } x = 0 \\ 0.4623(x-1)^{0.5}+ & \\ \quad 0.0092(x-1)+2 & \text{if } x > 0 \end{cases}$$

This models the time for a disk loosely based upon the IBM 0661 3.25 inch SCSI disk drive.

5.2 Results

Information was gathered on the progression of GVT and the two OFC models for the RAID application on a network of workstations. Studies were done for two different levels of risk: $\alpha = .1$ and $\alpha = .001$ (Figure 3). The graphs are shown from a representative process in the simulation. Since the only concerns are how the models bound GVT and the process' LVT, the data is shown only for each rollback at a process and for each GVT cycle. This also reduces the amount of data on the graph.

The bound derived with the Chebyshev inequality has an initial start up time until enough random samples have been collected to develop the confidence interval. For this run a minimum of 12 samples were required before a confidence interval for the mean the mean was developed, and a maximum of 32 samples were used to calculate the variance.

The bound derived from the geometric model bounds more closely than the Chebyshev bound. The difference in the bounds becomes more pronounced as α decreases. When $\alpha = 0.1$ (Figure 3) the bounds stay very close to GVT, and there are many rollbacks that cross the bounds. As α is decreased in the second study, there is a larger separation between the bounds and GVT (Figure 4). This shows that for larger values of α, or high levels of risk, the bounds calculated by these two techniques approach GVT. This could cause some processes to rollback past the last saved state and cause the simulation to fail. Smaller levels of risk do not bound the process as tightly, so less memory would be collected. Smaller levels of risk also increase the amount of time before the simulation can start collecting fossils. As Figure 4 shows there is a period of time during the start of the simulation where the models give time bounds that are obviously invalid.

The data on the rollbacks and the bounds calculated for the process was examined to determine the number of catastrophic rollbacks each model would have encountered during fossil collection. The table

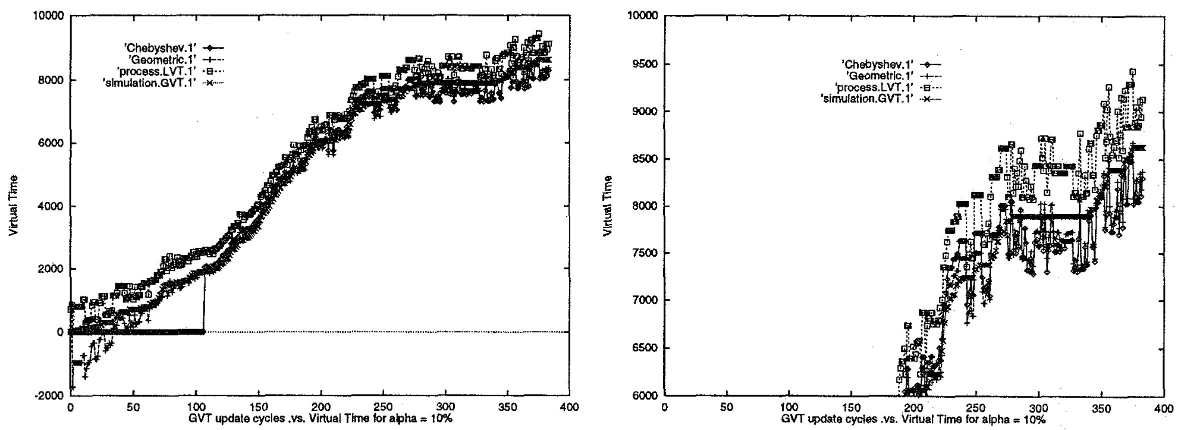

Figure 3: Graph of Process LVT, System GVT, and OFC Bounds Derived for $\alpha = 0.1$.

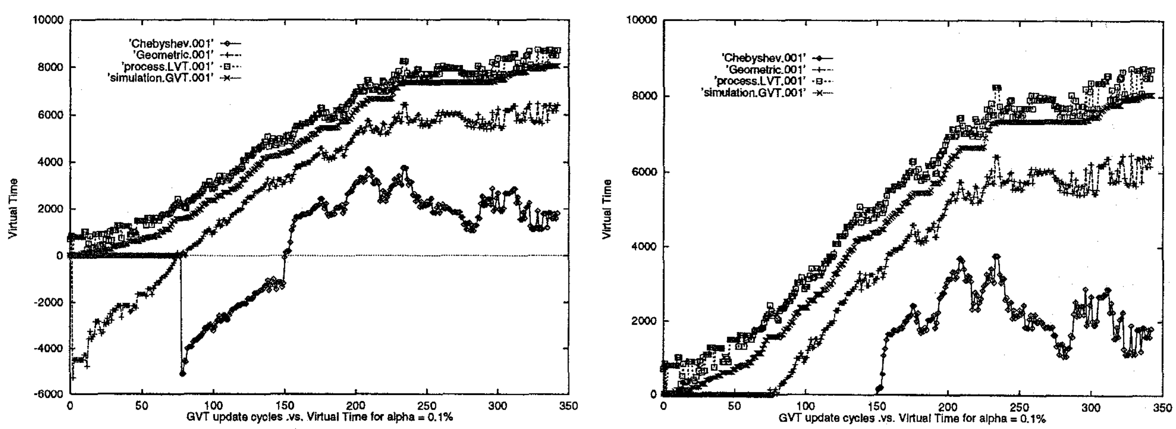

Figure 4: Graph of Process LVT, System GVT, and OFC Bounds Derived for $\alpha = 0.001$.

Geometric		Chebyshev	
alpha	rollbacks	alpha	rollbacks
0.1	4	0.1	3
0.001	0	0.001	0

Table 1: Catastrophic rollbacks experienced for different levels of risk.

below shows number of catastrophic rollbacks experienced for each model at the process. For $\alpha = 0.001$ the process never rolled back before either bound. For $\alpha = 0.1$ the process experienced catastrophic rollbacks for both bounds. This number of rollbacks is summarized in Table 1. Graphs of fossil collection under the different bounds is shown in Figure 5.

This shows that the bounds are responsive to value of α, and that the geometric bounds should experience more catastrophic rollbacks. This is due to the way the data is collected for this implementation, and the generality of the Chebyshev bound. Using both rollbacks and forward execution to calculate the variance takes into account the processing done between rollbacks, but it can also increase the estimated variance. This increase in variance also increases the bound of the Chebyshev model. Sampling only the rollbacks to determine a Chebyshev bound could decrease the variance and the bound.

6 Conclusions

This paper presented the design and implementation of two models for the Optimistic Fossil Collection technique for Time Warp. These two models which have been shown to statically bound rollback lengths in VHDL simulation [28] were tested for optimistically collecting fossils during a simulation of a RAID level 5 model with random delays and a high degree of irregular interaction. These models bounded GVT with a level of aggressiveness that depends upon a user defined parameter.

These models were implemented as an application layer for the WARPED kernel. Both models assumed there was an underlying stationary distribution for the rollbacks, or LVT updates at a process. Once the necessary parameters for these distribution were estimated, fossils can be collected.

These models effectively bound the rollbacks at a process, and in some cases, can allow for some processes to reclaim more memory than under GVT calculations. This can counter balance how other processes' might reclaim less memory quickly under these models than under GVT.

Acknowledgments

The authors would like to thank Nael Abu-Ghazaleh for his comments on earlier versions of this paper.

References

[1] ASTROM, K. J., AND WITTENMARK, B. *Adaptive Control.* Addison Wesley, Reading, MA, 1989.

[2] BAUER, H., AND SPORRER, C. Distributed logic simulation and an approach to asynchronous GVT-calculation. In *6th Workshop on Parallel and Distributed Simulation* (January 1992), Society for Computer Simulation, pp. 205–208.

[3] BELLENOT, S. Global virtual time algorithms. In *Distributed Simulation* (January 1990), Society for Computer Simulation, pp. 122–127.

[4] CHEN, S., AND TOWSLEY, D. The design and evaluation of RAID 5 and parity striping disk array architectures. *Journal of Parallel and Distributed Computing 17*, 1 (January 1993), 58–74.

[5] DAS, S., FUJIMOTO, R., PANESAR, K., ALLISON, D., AND HYBINETTE, M. GTW: a time warp system for shared memory multiprocessors. In *Proceedings of the 1994 Winter Simulation Conference* (December 1994), J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, Eds., pp. 1332–1339.

[6] DAS, S. R., AND FUJIMOTO, R. M. A performance study of the cancelback protocol for time warp. In *Proc. of the 7th Workshop on Parallel and Distributed Simulation (PADS)* (July 1993), Society for Computer Simulation, pp. 135–142.

[7] DAS, S. R., AND FUJIMOTO, R. M. An adaptive memory management protocol for time warp parallel simulation. In *Sigmetrics* (May 1994), pp. 201–210.

[8] D'SOUZA, L. M., FAN, X., AND WILSEY, P. A. pGVT: An algorithm for accurate GVT estimation. In *Proc. of the 8th Workshop on Parallel and Distributed Simulation (PADS 94)* (July 1994), Society for Computer Simulation, pp. 102–109.

[9] D'SOUZA, L. M., KRISHNASWAMY, V., AND WILSEY, P. A. Accurate estimation of global virtual time in time warp simulators. In *Proceedings of the 1995 Winter Simulation Conference* (December 1995), Society for Computer Simulation. (submitted).

[10] FERSCHA, A. Probabilistic adaptive direct optimism control in time warp. In *Proc. of the 9th Workshop on Parallel and Distributed Simulation (PADS 95)* (June 1995), pp. 120–129.

[11] FERSCHA, A., AND LÜTHI, J. Estimating rollback overhead for optimism control in time warp. In *Annual Simulation Symposium* (April 1995).

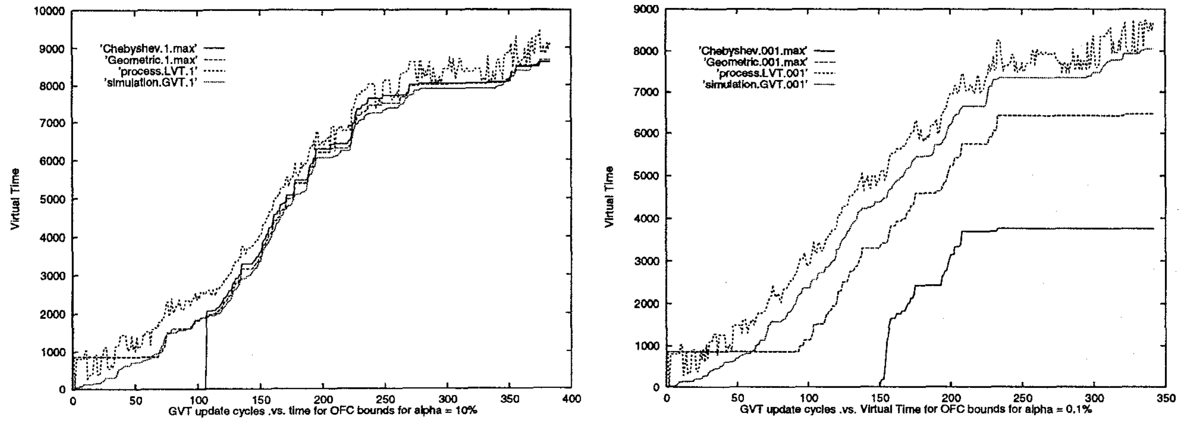

Figure 5: Progression of Process LVT, Simulation GVT, and Last State Saved by OFC Bounds.

[12] FUJIMOTO, R. Parallel discrete event simulation. *Communications of the ACM 33*, 10 (October 1990), 30–53.

[13] FUJIMOTO, R. M., AND HYBINETTE, M. Computing global virtual time in shared-memory multiprocessors, 1995. (submitted for publication).

[14] JEFFERSON, D. Virtual time. *ACM Transactions on Programming Languages and Systems 7*, 3 (July 1985), 405–425.

[15] LEE, E. K., AND KATZ, R. H. Performance consequences of parity placement in disk arrays. In *4th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IV)* (April 1991), pp. 190–199.

[16] LIN, Y., AND LAZOWSKA, E. D. A study of time warp rollback mechanism. *ACM Transactions on Modeling and Computer Simulation 1*, 1 (January 1991), 51–72.

[17] LIN, Y.-B. Memory management algorithms for optimistic parallel simulation. In *6th Workshop on Parallel and Distributed Simulation* (January 1992), Society for Computer Simulation, pp. 43–52.

[18] LIN, Y.-B., AND LAZOWSKA, E. Determining the global virtual time in a distributed simulation. In *1990 International Conference on Parallel Processing* (1990), pp. III–201–III–209.

[19] LIN, Y.-B., PREISS, B. R., LOUCKS, W. M., AND LAZOWSKA, E. D. Selecting the checkpoint interval in time warp simulation. In *Proc of the 7th Workshop on Parallel and Distributed Simulation (PADS)* (July 1993), Society for Computer Simulation, pp. 3–10.

[20] MARTIN, D. E., MCBRAYER, T., AND WILSEY, P. A. WARPED: A time warp simulation kernel for analysis and application development, 1995. (available on the www at `http://www.ece.uc.edu/~paw/warped/`).

[21] MATSUMOTO, Y., AND TAKI, K. Adaptive time-ceiling for efficient parallel discrete event simulation. In *Object-Oriented Simulation Conference (OOS '93)* (January 1993), T. Beaumariage and C. Roberts, Eds., Society for Computer Simulation, pp. 101–106.

[22] PALANISWAMY, A., AND WILSEY, P. A. Adaptive bounded time windows in an optimistically synchronized simulator. In *Third Great Lakes Symposium on VLSI* (1993), pp. 114–118.

[23] PALANISWAMY, A., AND WILSEY, P. A. Adaptive checkpoint intervals in an optimistically synchronized parallel digital system simulator. In *VLSI 93* (September 1993), pp. 353–362.

[24] PREISS, B. R., MACINTRYE, I. D., AND LOUCKS, W. M. On the trade-off between time and space in optimistic parallel discrete-event simulation. In *6th Workshop on Parallel and Distributed Simulation* (January 1992), Society for Computer Simulation, pp. 33–42.

[25] RAJAN, R., AND WILSEY, P. A. Dynamically switching between lazy and aggressive cancellation in a time warp parallel simulator. In *Proc. of the 28th Annual Simulation Symposium* (April 1995), IEEE Computer Society Press, pp. 22–30.

[26] TOMLINSON, A. I., AND GARG, V. K. An algorithm for minimally latent global virtual time. In *Proc of the 7th Workshop on Parallel and Distributed Simulation (PADS)* (July 1993), Society for Computer Simulation, pp. 35–42.

[27] VARGHESE, G., CHAMBERLAIN, R., AND WEIHL, W. E. The pessimism behind optimistic simulation. In *Proc. of the 8th Workshop on Parallel and Distributed Simulation (PADS 94)* (July 1994), Society for Computer Simulation.

[28] YOUNG, C., AND WILSEY, P. A. A distributed method to bound rollback lengths for fossil collection in time warp simulators. *Information Processing Letters*. (submitted).