Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

CAUSALITY INFORMATION AND FOSSIL COLLECTION IN TIMEWARP SIMULATIONS

Malolan Chetlur

AT&T
Cincinnati, OH 45202, U.S.A.

Philip A. Wilsey

Experimental Computing Laboratory
Department of ECECS
P.O. Box 210030
Cincinnati, OH 45219, U.S.A.

ABSTRACT

This paper presents a Time Warp fossil collection mechanism that functions without need for a GVT estimation algorithm. Effectively each Logical Process (LP) collects causality information during normal event execution and then each LP utilizes this information to identify fossils. In this mechanism, LPs use constant size vectors (that are independent of the total number of parallel simulation objects) as timestamps called *Plausible Total Clocks* to disseminate causality information. For proper operation, this mechanism requires that the communication layer preserves a FIFO ordering on messages. A detailed description of this new fossil collection mechanism and its proof of correctness is presented in this paper.

1 INTRODUCTION

Time Warp simulators store event histories (consisting of input events, output events and state vectors) in anticipation of causality errors (Jefferson 1985). These event histories increase memory usage over time and they must be periodically reclaimed to reduce the runtime memory requirement of simulations (Jefferson 1990). This reclamation process is commonly known as fossil collection and it should reclaim only those event histories that are not referenced in the future.

Conventionally, Global Virtual Time (GVT) estimation is used to set a lower time boundary against which fossil collection occurs (Fujimoto 1990), where GVT is defined as the minimum simulation time among individual simulation objects and the messages in transit (Mattern 1993). We can see from the above definition that GVT imposes a strict upper bound on simulation time among the identified fossils. Additionally, event histories that are already fossils (with simulation time greater than GVT) will be reclaimed only when GVT sweeps past their simulation time.

In this paper, we present a fossil collection mechanism that operates independently of GVT estimation. Currently this mechanism functions only for simulation models with static inter-connection topology. This mechanism operates by collecting causality information disseminated during normal event execution and utilizing this information in identifying fossils. Thus, the fossil identification is reduced to a local computation.

The remainder of this paper is organized as follows: Section 2 presents related work in the area of fossil collection in optimistic simulations. Section 3 presents an overview of Time Warp, logical time representations, and assumptions necessary to understand this paper. Section 4 discuss the motivation for fossil identification independent of GVT estimation and Section 5 outlines the fossil collection mechanism approach. Section 6 presents *Plausible Total Clocks*, its properties and the fossil collection mechanism in detail. Section 7 presents the proof of correctness of this mechanism, and Section 8 discuss scenarios where such a technique will be useful. Section 9 concludes this paper with objective assessment of the technique presented.

2 RELATED WORK

In Time Warp optimistic simulation, fossil reclamation through GVT estimation is a prominent and well proven technique (Jefferson 1985). There are several GVT algorithms proposed in the literature (Lin and Lazowska 1990; Bellenot 1990; D'Souza, Fan, and Wilsey 1994; Mattern 1993; Fujimoto and Hybinette 1994; Tomlinson and Garg 1993). These algorithms either measure the rate of virtual time progress (D'Souza, Fan, and Wilsey 1994) or identify consistent snapshots (Mattern 1993) or keeps track of the peak and valley messages (Lin and Lazowska 1990). On the other hand, Optimistic Fossil Collection (OFC) designed by Young (Young, Abu-Ghazaleh, and Wilsey 1998) identifies fossils by predicting future rollback lengths based on the past rollback behavior. OFC is a novel and powerful technique to identify fossils based on local computation and this does not involve any distributed operation. However, this technique can suffer from catastrophic rollbacks resulting in degraded performance.

3 BACKGROUND

3.1 Time Warp and WARPED

Time Warp optimistic simulation is an implementation of *Virtual Time* paradigm proposed by Jefferson (Fujimoto 1990). Virtual time is a single dimensional view of time providing total ordering among events generated during simulation (Jefferson 1985). In Time Warp, simulation objects optimistically execute events sent by causally preceding simulation objects. Simulation objects process their events in non-decreasing virtual time order, thereby obeying local causality constraint (Fujimoto 1990). They maintain their own clock to track their Local Virtual Time (LVT) and an input event queue to store events sent by its causal predecessor objects. A simulation object maintains a history of states that are created while executing input events and corresponding history of output events. These histories are used to recover from causality errors. This history can grow over time and have to be reclaimed (fossil collected) in a timely manner to reduce memory overheads.

The WARPED kernel contains an implementation of Jefferson's virtual time paradigm (Radhakrishnan et al. 1996). In this implementation, several simulation objects are mapped onto a single Logical Process (LP) to share various Time Warp components such as event-list management, scheduling, and communication optimizations. Figure 1 shows the structure of an LP in WARPED implementation. An LP schedules an event with lowest timestamp among its simulation objects (LTSF scheduling), thereby obeying local causality constraint among events within the same LP. The minimum LVT among its simulation objects is the LVT ($lp_i.LVT$) of an LP lp_i. Communication among simulation objects within the same LP are reduced to direct memory operations whereas MPI communication layer is used for communication between objects in different LPs. Fossil collection technique proposed in this paper assumes similar grouping of simulation objects in an LP with LTSF scheduling among its events.

3.2 Logical Time

Logical time representation in distributed systems range from single element scalar clocks to vector clocks and matrix clocks (Raynal and Singhal 1996). Scalar clocks representation establish "happens-before" (Lamport 1978) relation among two events but cannot always determine causality relation between two events from two different processes; whereas vector clocks can provide such insights by comparing two vector clock values (Raynal and Singhal 1996). Given two vector clocks v_1, v_2, $v_1 < v_2 \equiv v_1$ happened-before v_2. We can see that causality relation between any two events with vector timestamps can be easily determined from the relationship between their vector clocks. On the other hand, plausible clocks are constant size clocks that capture weak causality relation between any two events (Torres-Rojas and Ahamad 1996). Given two plausible clocks p_1, p_2, and p_1 happened-before $p_2 \Rightarrow p_1 < p_2$. Also, $p_1 < p_2 \Rightarrow p_1$ happened-before $p_2 \vee p_1$ is concurrent to p_2. This ambiguity is due to the weak causality relation between two plausible clocks. Total clocks proposed by Chetlur *et al* is a two component timestamp representation and is used in avoiding cascading rollback in Time Warp simulation (Chetlur 2001). Total clocks are similar to vector clocks in capturing strong causality relation between events and they exploit this property to identify and remove causally related events to incorrect computations during cancellation.

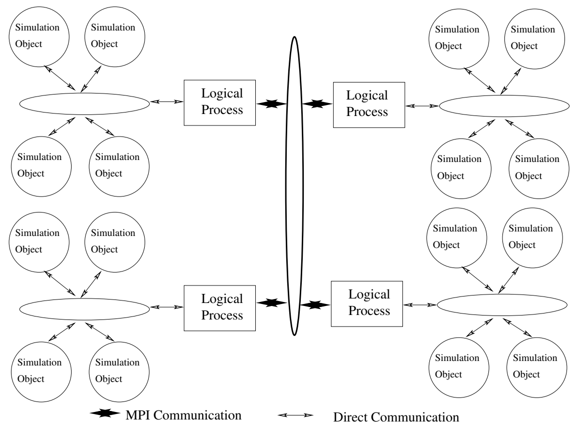

Figure 1: Structure of LPs and Simulation Objects in WARPED Implementation

3.3 Safe Time

In this paper, an LP disseminates its *Safe-Time* estimate to other LPs. Safe-time (ST_i) of an LP (lp_i) is defined as the guaranteed simulation time reached by lp_i and no simulation object can rollback lp_i beyond ST_i. An LP disseminates its safe-time value to its output channel objects. This safe-time value establishes a lower bound on the simulation time of future events to receiver objects. GVT in a optimistic simulation or lookahead specified in a conservative simulation can be readily used as safe-time. Individual LPs can provide their safe-time estimates based on the knowledge of its inter-connection topology and the progress of simulation time across its input channels. *Safe-Time* described in this paper is similar to safe-time described in PDMS and other conservative simulation protocols (Nicol 1996, Xiao et al. 1999).

4 MOTIVATION

In an ultra-large simulation or in a federated mode simulation, GVT imposes a strict and unnecessary constraint on fossil identification. For instance, an LP with smaller LVT slows the progress of GVT and hence limits the identification of fossils in another causally unrelated (concurrent) LP. This results in either increased memory overheads or LPs sacrificing temporal concurrency available in the model for a smaller memory footprint. Therefore, to avoid such scenarios, the knowledge of inter-connection topology among simulation objects along with the knowledge of causality relation between event histories can be used to identify fossils independently of GVT estimation. In addition, GVT estimation belongs to the class of distributed infimum approximation (DIA) algorithms that necessitates the involvement of every process in the distributed operation (Tel 1991). A non-DIA solution to identify fossils will be useful both as a peephole and divide-and-conquer technique within an ultra-large simulation framework.

5 FOSSIL IDENTIFICATION APPROACH

Logical Process can be categorized into four types namely: *source LP*, *sink LP*, *feed-forward LP* and *feedback LP* as shown in Figure 2. A source LP such as *A* is essentially a sequential simulator with several objects simulating a specific part of a large simulation model and sending events to other LPs. This LP can treat its local virtual time (LVT) as safe-time guarantee to other LPs (due to LTSF scheduling policy), and event histories below this time can be reclaimed as fossils. Sink-LP (LP *B*) consists of simulation objects that do not send events to other LPs but only receive events from them. An LP such as *B* can identify fossils from the minimum safe-time among its input channels. Feed-forward LP (LP *C*) consists of simulation objects that do not directly or indirectly send and receive events from the same object in another LP. In the case of feed-forward LP, such as *C*, the minimum among its LVT and safe-time of input channels determines the fossil boundary. LP *D* consists of simulation objects that transitively send and receive events from the same object in another LP (feedback LP). In the case of LP *D*, mere knowledge of the safe-time of input channels do not suffice in identifying fossils. This feedback LP also has to determine the lower bound of the simulation time of messages to be sent by all the simulation objects in its feedback path. This lower bound can be determined from the causality relation between its input events, output events and its states. Therefore, in the case of LP *D*, availability of causality information and safe-time of input channels can be used to identify the fossils locally.

In this paper, we present a new fossil identification technique by disseminating the causality information during normal event execution and then having each LP make

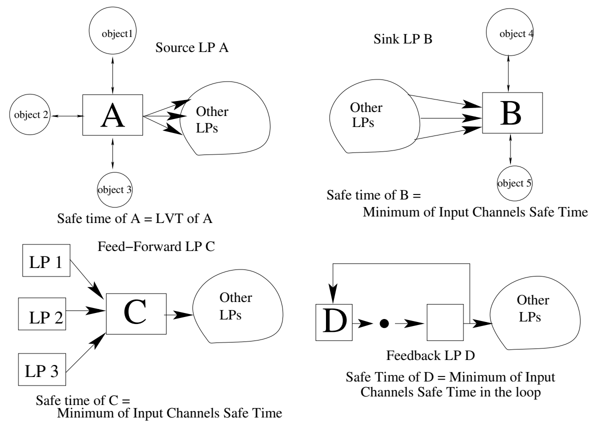
object1
Source LP A
object 2
A
Other LPs
object 3
Safe time of A = LVT of A
Sink LP B
object 4
Other LPs
B
object 5
Safe time of B = Minimum of Input Channels Safe Time
Feed–Forward LP C
LP 1
LP 2
LP 3
C
Other LPs
Safe time of C = Minimum of Input Channels Safe Time
D
Other LPs
Feedback LP D
Safe Time of D = Minimum of Input Channels Safe Time in the loop

Figure 2: Inter-connection Topology and Safe-Time Estimation

local decisions to identify fossils based on the knowledge of its inter-connection topology. The fossil collection mechanism is essentially determining the safe-time of an LP similar to that of a conservative simulation (Nicol 1996). This safe-time is estimated using the connection topology and causal order among events as opposed to lookahead values in conservative simulations. In the proposed mechanism, the events are processed optimistically using virtual time paradigm and event histories are fossil collected conservatively using safe-time. The underlying communication layer is assumed to be FIFO. It is also assumed that simulation objects do not schedule events to other simulation objects with zero length time increments. Finally, simulation objects have no knowledge of predicting any lookahead information.

6 FOSSIL COLLECTION MECHANISM

6.1 Plausible Total Clocks

In this paper, we use a timestamp representation called *plausible total clocks (PTC)* containing a constant size vector to disseminate causality information. PTC are plausible version of the *total clocks* described in (Chetlur 2001). Plausible total clocks consists of a *virtual time* (VT) component and a vector counter component (VC). Virtual time component measures virtual time as proposed by Jefferson (Jefferson 1985). The second component of plausible total clocks is a vector of event counter values. Every LP maintains its own *event counter* (EC). All simulation objects in an LP i share the same event counter and update $VC(i)$ during its event processing. PTC are similar in structure to total clocks, however its size is proportional to the number of LPs instead of the number of simulation objects. Therefore, PTC can capture only weak causality relation between any two events as opposed to strong causality relation captured by total clocks (Charron-Bost 1991).

The event counter of an LP is incremented in every event execution. Additionally, each simulation object keeps track of the latest event counters of causally preceding simulation objects from the *vector counter* (VC) of its input event's timestamp. In an LP lp_i, $TC(P_{(i,a)})$ refers to the plausible total clock value of simulation object $P_{(i,a)}$, $TC(P_{(i,a)}).VT$ refers to its virtual time component and $TC(P_{(i,a)}).VC$ refers to the vector counter component. $TC(e)$ refers to the timestamp of an event e. The size of the vector is equal to the number of LPs (n) and is independent of the number of simulation objects (m) in the simulation model. Generally, n is far less than m ($n << m$).

During simulation, lp_i schedules events in the order of its VT value. A simulation object update its VC value to the maximum value of individual elements of vector counter seen among its input events. Therefore, a simulation object $P_{(i,a)}$ learns about $VC[k], (i \neq k)$ by receiving events from simulation objects in lp_k or from objects that received events from lp_k. Simulation object $P_{(i,a)}$ assigns a future time at which a newly generated event has to be processed as its VT component. Vector component of the simulation object is assigned as the VC value of the output event to disseminate causality information. The VC component of two events can be used to determine weak causal relationship among them. We can see from Theorem 6.1 that $TC(e_1).VC < TC(e_2).VC$ implies $e_1 \longrightarrow e_2$ or e_1 is concurrent to e_2(ambiguity exists). We will show in later proofs that, in spite of only capturing weak causality relation between any two events or states, plausible total clocks will be sufficient to determine fossils in an LP.

Definition 6.1 $TC(e_1).VC \ll TC(e_2).VC$
$\equiv \forall i, 1 \leq i \leq n, TC(e_1).VC[i] < TC(e_2).VC[i]$.

Definition 6.2 $TC.VC = sup(TC_1.VC, TC_2.VC)$
$\equiv \forall i, 1 \leq i \leq n, TC.VC[i]$
$= max(TC_1.VC[i], TC_2.VC[i])$.

Definition 6.3 $TC.VC = min(TC_1.VC, TC_2.VC)$
$\equiv \forall i, 1 \leq i \leq n, TC.VC[i]$
$= min(TC_1.VC[i], TC_2.VC[i])$.

Theorem 6.1 *If $\longrightarrow$ denotes causally precedes relation and $\|$ denotes causally concurrent relation then*

$$TC(e_1).VC < TC(e_2).VC \Rightarrow e_1 \longrightarrow e_2 \vee e_1 \parallel e_2.$$

6.2 Fossil Collection

Procedure Simulate (Procedure 1) describes the simulation cycle in an LP containing a group of simulation objects. Least time-stamped event is chosen for execution. Procedure Process (Procedure 2) shows the normal processing of an event. This also performs the computations necessary for future fossil identification. The event counter maintained by an LP monotonically increases with event execution and all objects in an LP share the same event counter. If an LP is a source LP, then the safe-time guarantees are determined during normal event processing and can be disseminated as part of normal event execution.

Procedure 1 Simulate()

```
1: while Simulation is Not Complete do
2:    new_event=Check_For_New_Message()
3:    Receive(new_event)
4:    event_to_be_processed = Schedule LTSF event
5:    Process(event_to_be_processed)
6:    Fossil_Collect(event_to_be_processed)
7: end while
```

Procedure 2 Process(e, $P_{(i,a)}$)

```
1: $TC(P_{(i,a)}).VT = TC(e).VT$
2: $TC(P_{(i,a)}).VC = sup(TC(P_{(i,a)}).VC, TC(e).VC)$ {this
   total clock value is denoted by $TC((P_{(i,a)})_{before\ e})$}
3: $TC(P_{(i,a)}).VC[i] = lp_i.EC$ {this Total Clock value is
   denoted by $TC((P_{(i,a)})_{after\ e})$}
4: Process_Fossil_Collection_Information(e)
5: if $lp_i$ is a Source Object then
6:    $ST_i = TC(P_{(i,a)}).VT$
7:    $FC_i = lp_i.EC$
8: end if
9: execute_event e {update, save state; send events to other
   simulation objects}
10: $lp_i.EC++$
```

A cancellation mechanism using causality information to prevent cascading rollbacks is detailed in (Chetlur 2001). In this paper, we assume, such a cancellation mechanism is not available. However, during rollback, an LP keeps track of the range of event counter values determined from the rolled-back events in a data-structure namely *CANCEL_RANGE_LIST*. This list is necessary to determine if an input event is causally related to the rolled-back events.

An LP maintains several values to identify the fossils locally. They are: 1) $ST_{ch(e)}$, safe-time guarantee specified across input channel $ch(e)$, 2) $IpFC_{ch(e)}$, the maximum event counter value seen by an LP across its input channel $ch(e)$, 3) FC_i, the maximum event-counter value seen by all its causal predecessors (referred as *fossil-counter*), 4) FCV_i, vector of fossil-counter of all its causal predecessors, also referred as *fossil-counter vector*, 5) $IpFCV_{ch(e)}$, the fossil counter vector received across channel $ch(e)$, 6) FT_i, the simulation time below which lp_i will not be rolled-back (also referred as *fossil time*), and 7) ST_i, safe-time of lp_i calculated from the safe-time of input channels and FT_i.

The above mentioned values are determined from information disseminated as part of the event timestamp (constant size vector), and from the piggy-backed values of safe-time and fossil counter vector. An LP determines its own fossil boundary from the information disseminated by its causal predecessors. The newly determined values are piggybacked to its causal successors. Procedure Process_Fossil_Collection_Iinformation (Procedure 3) performs the determination of fossil boundary during normal event computations and is invoked within procedures Process and Cancel. Procedure Fossil_Collect (described in Procedure 4) is invoked in the main simulation cycle to identify the fossils.

Procedure 3 Process_Fossil_Collection_Information(e)

```
1: if Fossil Collection Information is PiggyBacked then
2:   if $lp_i$ has Feedback Channels then
3:     if $TC(e).VC[i]$ NOT IN $lp_i^{'}s$ CANCEL_RANGE_LIST then
4:       $IpFC_{ch(e)} = sup(TC(e).VC[i], IpFC_{ch(e)})$;
5:     end if
6:     $IpST_{ch(e)} = sup(e.ST, IpST_{ch(e)})$
7:     $FC_i = min(IpFC_{fbch})$ $\forall$ fbch, where fbch is a feedback input channel to $lp_i$
8:     $IpFCV_{ch(e)} = sup(e.FCV, IpFCV_{ch(e)})$
9:     $FCV_i = Min(IpFCV_{ch})$ $\forall$ ch, where ch is an input channel to $lp_i$
10:    $FCV_i[i] = FC_i$
11:  else
12:    $IpST_{ch(e)} = sup(e.ST, IpST_{ch(e)})$
13:    $IpFCV_{ch(e)} = sup(e.FCV, IpFCV_{ch(e)})$
14:    $FCV_i = Min(IpFCV_{ch})$ $\forall$ ch, where ch is an input channel to $lp_i$
15:    $ST_i = min(IpST_{nfch}, lp_i.LVT)$ $\forall$ $nfch$ where nfch is channel with no feedback
16:  end if
17: end if
```

Procedure 4 Fossil_Collect(e)

```
1: if LP $lp_i$ has Feedback Channels then
2:   $FT_{P_{(i,a)}} = st_{P_{(i,a)}}.VT$ where $st_{P_{(i,a)}} << FCV_i$ with maximum VT value in $P_{(i,a)}$
3:   $FT_i = min(FT_{P_{(i,a)}})$ $\forall$ active $P_{(i,a)} \in lp_i$
4:   $ST_i = max(min(IpST_{ch}, lp_i.LVT), FT_i)$ $\forall$ input channel ch
5: end if
6: $\forall$ $st$, If st.VT $\leq$ $ST_i$ Then st is a fossil, where st is the saved state
7: $\forall$ $ipe$ If ipe.VT $\leq$ $ST_i$ Then ipe is a fossil, where ipe is an input event
8: $\forall$ $ope$ If ope.VT $\leq$ $ST_i$ Then ope is a fossil, where ope is an output event
9: if LP $lp_i$ has no Feedback Channels then
10:  $FC_i = fossilstate.VC[i]$, where fossilstate.VT $\leq$ $ST_i$ with maximum VC[i] value
11:  $FCV_i[i] = FC_i$
12: end if
```

We can see that fossil collection procedures (Procedures 3 and 4) determine an LP's safe-time and fossil-counter value based on its inter-connection topology. For an LP without any feedback channels, its safe-time is determined from the safe-time of its input channels and its LVT. This safe-time also forms the boundary between the fossils and its active event histories. Based on the changes to its safe-time, this LP can reclaim fossils and inform its causal successors of its new safe-time. Additionally, it disseminates fossil-counter as part of the fossil-counter vector. In the case of a source-lp, an LP reclaim events as fossils whose simulation time is lower than its LVT and send appropriate values of safe-time and fossil-counter to succeeding objects as shown in procedure Process (refer Procedure 2:lines 5-8).

On the other hand, if an LP lp_i has feedback paths, then the safe-time is not merely dependent on the safe-time of its input channels but also on the simulation time of events sent by lp_i. We can see that, in a feedback-lp, the value of vector counter at index i of the input event specifies the latest event counter value seen by its causal predecessors. This is a valuable information for lp_i to speculate on the simulation time reached by the objects in its feedback path. The value of input channel fossil counter $IpFC_{ch(e)}$ is the latest event counter value of lp_i seen in that specific causal path. However, this does not ensure that all the objects in that causal path have seen an event with that event counter value since more than one simulation object is mapped onto the same LP. However, due to LTSF scheduling, any event with a lower timestamp is given preference over already existing events and therefore effects of a straggler to any object are observed immediately in an LP. In an LP, the safe-time sent through a channel a is set to $IpST_a$ (refer Procedure 3:line 6,12). FC_i is set to the minimum of $IpFC_a$ of all (feedback) input channel a (refer Procedure 3:line 7). This implies that the effect of event e_{fc} with event counter value of FC_i from lp_i is seen by causal predecessors in all its feedback paths. This also implies that all the causal predecessors among the feedback channels have optimistically reached the simulation time $TC(e_{fc}).VT$ provided those optimistic executions are not rolled-back. Fossil Counter Vector (FCV_i) is determined from Min (definition 6.3) function of all input channel fossil counter vector $IpFCV_a$(refer Procedure 3:line 9,14). The minimum of individual elements of vector is applied only to elements that have values defined in them. For instance, element at index k of input channel j might not be defined if lp_j does not receive events from lp_k. This undefined value is different from value 0 $or -\infty$ initialized at index k when lp_k is causally related to lp_j. The newly determined FCV_i at lp_i denotes the vector of fossil counter values (FC_k) of all causally related LP k such that the causal predecessors of lp_i have optimistically reached the simulation time $TC(e_{fc_k}).VT$. Therefore, this FCV_i forms the boundary between the fossils and active event histories.

The new fossil-counter values are checked against the CANCEL_RANGE_LIST in Procedure 3 (refer line 4) to ensure event-counter values within the cancel-ranges are ignored. A formal proof of correctness and liveliness of this fossil collection mechanism is presented in the following section.

7 PROOF OF CORRECTNESS

In this section, we will present the proof of correctness of fossil collection mechanism designed using plausible total clocks. We will present theorems and prove them to establish that procedure Fossil_Collect identifies fossils correctly and newer fossils are identified with the progress of the simulation.

Axiom 7.1 *Given an LP lp_i with no feedback channels then*

$$ST_i = \ \forall\ ch(e),\ min(LVT, ST_{ch(e)})$$

where ch(e) is an input channel to lp_i.

Theorem 7.1 *Given an LP lp_i with no feedback channels then input events, output events, state with $VT \leq ST_i$ determined in procedure Fossil_Collect is a fossil.*

Theorem 7.2 *Given FCV_i is the fossil counter vector of LP lp_i, e_{fc_i} is the event sent from simulation object $P_{(i,a)}$ in lp_i with $TC(e_{fc_i}).VC[i] = FC_i$, and st_{fc_i} is the state saved after sending event e_{fc_i}. Then $e_{straggler}$ is a straggler event to simulation object $P_{(i,a)}$ in $lp_i \Rightarrow \exists$ index k such that $TC(e_{straggler}).VC[k] \geq FCV_i[k]$.*

Proof: Let us assume that there exists an arbitrary index k for $e_{straggler}$ to $P_{(i,a)}$ such that $TC(e_{straggler}).VC[k] < FCV_i[k]$. We know from Procedure 3 (refer line 9,14) that $FCV_i = Min(IpFCV_{ch})$ of all input channels. This implies $FCV_i[k] = min(IpFCV_{ch}[k])$. Additionally, we know that $IpFCV$ is disseminated by the causal predecessors of lp_i. Therefore, $FCV_i[k]$ is equal to the minimum event counter value seen among all feedback channels of i and its causal predecessors and hence the lower bound on the value at index k among all LPs in its feedback path. This implies, lp_i has seen events with $VC[k] \geq IpFCV[k]$ from all input channels. Therefore, $e_{straggler}$ to object $P_{(i,a)}$ has value greater than $FCV_i[k]$ and this contradicts the initial assumption and hence the proof. □

Theorem 7.3 *Given an LP lp_i has feedback channels, and $e_{straggler}$ is a straggler event arriving at lp_i. Then $TC(e_{straggler}).VT > FT_i$.*

Proof: Let us assume that $FT_i \geq TC(e_{straggler}).VT$. We know from Procedure Fossil_Collect that FT_i is the minimum VT among the maximal fossil state of all the simulation objects in lp_i (states with maximum VT and whose $VC << FCV_i$) (refer Procedure 4:line 2). Let the state corresponding to FT_i be st_{FT_i} saved after processing event e_{FT_i} in simulation object $P_{(i,a)}$. Therefore, $TC(e_{FT_i}).VT = FT_i$. Additionally, since $TC(st_{FT_i}).VC << FCV_i$, we can infer from Theorem 7.2 that $TC(e_{FT_i}).VC[k] < TC(e_{straggler}).VC[k]$ and $TC(e_{FT_i}).VT \geq TC(e_{straggler}).VT$. This inference is possible, only if there exists another straggler $e_{straggler2}$ to lp_i such that $TC(e_{straggler2}).VC < FCV_i \wedge TC(e_{straggler2}).VT \geq FT_i$. From theorem 7.2, we know that $TC(e_{straggler2}).VC < FCV_i$ does not exist and therefore the initial assumption that $FT_i \geq TC(e_{straggler})$ results in a contradiction and hence the proof. □

We can infer from Theorems 7.1 and 7.3 that lp_i identifies fossils based on the ST_i calculated in procedure Fossil_Collect (refer line 4).

Theorem 7.4 *The Safe-time calculated in Procedure Fossil_Collect progresses with the simulation.*

Proof: An LP lp_i can fall under four scenarios: (a) lp_i is a source lp, (b) lp_i is feed-forward lp, (c) lp_i is a feedback lp, and (d) lp_i is sink lp.

Case a: From Procedure Process (refer line 6), it is trivial that ST_i progresses with the simulation time of lp_i.

Case b: We can see from Procedure Process_Fossil_Collection_Information (refer line 15) that in the case of feed-forward lp, the safe-time value is the minimum of input channel safe-times and its LVT. Additionally, the progress in safe-time of its causal predecessors is independent of its progress in simulation. Therefore, progress of safe-times of causally preceding LPs of lp_i result in the progress of its safe-time.

Case c: We can see from Procedure Fossil_Collect that ST_i is calculated from three different values namely (refer line 3-4): (1) The minimum safe-time among all input channels, (2) The minimum $FT_{P_{(i,a)}}$ among all active objects denoted by FT_i, and (3) The LVT of lp_i denoted by $lp_i.LVT$. In the case of an LP with slowly progressing simulation objects, $lp_i.LVT$ can be less than $IpST_{ch(e)}$ and *min* operation in Fossil_Collect takes care of this specific scenario. FT_i determined in Fossil_Collect is essentially the simulation time of the maximal state that is seen in all feedback paths. In addition, only active objects (non-idle or non-terminated) are considered in the calculation of FT_i, thereby enabling progress of FT_i with changes in FCV_i. In scenarios where event with timestamp of FT_i from lp_i is the minimal event among all LPs in the feedback path, the max operation ensures progress in ST_i. The fossil counter FC_i progresses as newer events are processed and eventually an lp_i sees the effect of its own events in their feedback channels. This progress implies the progress of FCV_i and hence FT_i. We assume that the change in ST_i, FCV_i are piggy-backed to all the output channels; if there are no events to piggy-back, then mechanism such as *null messages* is employed to disseminate ST_i and FCV_i. This ensures progress of FCV in causally succeeding LPs, thereby progress of ST_i in lp_i.

Case d: Same as case (b).

From cases (a), (b), (c) and (d), we can infer that ST_i progresses with the simulation and hence the proof. □

8 APPLICATIONS

In the previous sections we presented a fossil identification technique that is independent of GVT estimation. This technique assumes static inter-connection topology among simulation objects. The interconnection topology of the physical processes is known beforehand in several simulation models in the area of digital logic design, computer system simulation, telecommunication networks, transportation and manufacturing simulation (Fujimoto 1993). Therefore, this fossil collection mechanism can be applied in such simulation models.

The fossil collection mechanism presented in this paper does not involve all the objects in the simulation. This technique enables a divide-and-conquer approach to fossil identification. A simulation model is partitioned into set of smaller models based on communication topology and fossils are identified independently of the overall progress of GVT. Therefore, this technique can be useful in integrating different simulator implementations within an optimistic framework. However, this technique requires special partitioning and compiler tools to partition a large model onto a subset of LPs with necessary inter-connection topology information gathered before simulation execution. We can also see that, fossils identified in an LP using this technique can have time greater than GVT; this enables performing irrevocable operations such as I/O earlier than the existing techniques. Additionally, fossil identification technique independent of GVT estimation are scalable since they do not involve all the objects in the simulation.

In this technique, events send causality information in a constant size vector. Transmitting vectors as part of event execution incur performance overhead. Although the overhead is lesser than a vector of size equal to the number of simulation objects, the timestamp overhead has to be reduced for faster event execution. Depending on the communication topology, it is possible to maintain a smaller size of vector counter component. We can see that, in this fossil collection mechanism, if lp_i is in feed-forward path to lp_j, then it is sufficient for lp_i to disseminate only ST_i, FC_i and not the whole vector FCV_i to enable fossil collection mechanism. This is due to the fact that, lp_i needs the fossil counter values of causally preceding LPs only in the feedback paths. Therefore, large fossil counter vector sent by a simulation object can be discarded and only a scalar value instead of a vector can be disseminated to causal successors in feed-forward path.

9 CONCLUSION

This paper presents a fossil collection mechanism that exploits causality information and inter-connection topology information. This paper presents *Plausible Total Clocks* timestamp representation consisting of constant size vectors and a new fossil collection mechanism within this framework. Conventionally, fossils are identified after GVT estimation; resulting in a global simulation time reached by all the objects in a simulation. The set of fossils identified using GVT technique is smaller than the actual number of active histories that has turned into a fossil. The fossil collection mechanism proposed in this paper determines safe-time value for every LP during event execution and identifies fossils based on the newly determined safe-time value. Therefore, in this technique, an LP can reclaim memory once a specific safe-time is reached irrespective of other LPs reaching this safe-time and this technique is independent of GVT estimation. A proof of correctness and liveliness of this fossil collection mechanism is presented. The goal of our research is to discover useful insights provided by causality information and present new algorithms that exploit this information during the execution of the simulation. Fossil collection mechanism presented is one such algorithm exploiting causality information.

REFERENCES

Bellenot, S. 1990, January. Global virtual time algorithms. In *Distributed Simulation*, 122–127. Society for Computer Simulation.

Charron-Bost, B. 1991, July. Concerning the size of logical clocks in distributed systems. *Information Processing Letters* 39 (1): 11–16.

Chetlur, M. 2001, May. Causality representation and cancellation mechanism in timewarp simulations. In *12th Workshop on Parallel and Distributed Simulation*. Society for Computer Simulation.

D'Souza, L. M., X. Fan, and P. A. Wilsey. 1994, July. pGVT: An algorithm for accurate GVT estimation. In *Proceedings of the 8th Workshop on Parallel and Distributed Simulation (PADS 94)*, 102–109. Society for Computer Simulation.

Fujimoto, R. 1990, October. Parallel discrete event simulation. *Communications of the ACM* 33 (10): 30–53.

Fujimoto, R. 1993. Parallel and distributed discrete event simulation: algorithms and applications. In *Proceedings of the 1993 Winter Simulation Conference*, 106–114.

Fujimoto, R. M., and M. Hybinette. 1994, August. Computing global virtual time in shared-memory multiprocessors. *ACM Transactions on Modeling and Computer Simulation* 7(4):425–446.

Jefferson, D. 1985, July. Virtual time. *ACM Transactions on Programming Languages and Systems* 7 (3): 405–425.

Jefferson, D. 1990. Virtual time II: storage management in conservative and optimistic systems. In *Proceedings of the 9th Annual Symposium on Principles of Distributed Computing*, 75–89. ACM Press, New York, NY.

Lamport, L. 1978, July. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM* 21 (7): 558–565.

Lin, Y.-B., and E. Lazowska. 1990. Determining the global virtual time in a distributed simulation. In *1990 International Conference on Parallel Processing*, III–201–III–209.

Mattern, F. 1993, August. Efficient algorithms for distributed snapshots and global virtual time approximation. *Journal of Parallel and Distributed Computing* 18 (4): 423–434.

Nicol, D. 1996, December. Principles of conservative parallel simulation. In *Proceedings of the 1996 Winter Simulation Conference (WSC 96)*, 128–135.

Radhakrishnan, R., T. J. McBrayer, K. Subramani, M. Chetlur, V. Balakrishnan, and P. A. Wilsey. 1996, March. A comparative analysis of various time warp algorithms implemented in the WARPED simulation kernel. In *Proceedings of the 29th Annual Simulation Symposium*, 107–116.

Raynal, M., and M. Singhal. 1996, February. Logical time: Capturing causality in distributed systems. *IEEE Computer*:49–56.

Tel, G. 1991. *Topics in distributed algorithms*. Cambridge Universiy Press.

Tomlinson, A. I., and V. K. Garg. 1993, July. An algorithm for minimally latent global virtual time. In *Proceedings of the 7th Workshop on Parallel and Distributed Simulation (PADS)*, 35–42. Society for Computer Simulation.

Torres-Rojas, F. J., and M. Ahamad. 1996. Plausible clocks: constant size logical clocks for distributed systems. In *Workshop on Distributed Algorithms*, 71–88.

Xiao, Z., B. Unger, R. Simmonds, and J. Cleary. 1999. Scheduling critical channels in conservative parallel discrete event simulation. In *Proceedings of the Thirteenth Workshop on Parallel and Distributed Simulation (PADS '99)*, 20–28. Washington, DC, USA: IEEE Computer Society.

Young, C. H., N. B. Abu-Ghazaleh, and P. A. Wilsey. 1998, September. OFC: a distributed fossil-collection algorithm for time-warp. In *12th International Symposium on Distributed Computing, (DISC'98, formerly WDAG)*.

AUTHOR BIOGRAPHIES

MALOLAN CHETLUR is a senior technical specialist in AT&T, USA. He is currently pursuing his PhD at the University of Cincinnati. He is a member of ACM and IEEE computer Society. His interests include parallel and distributed simulation, distributed computing, and software architecture. His email address is `<mal@ececs.uc.edu>`.

PHILIP A. WILSEY is a professor in the Department of ECECS, University of Cincinnati. He is also the associate editor of IEEE Potentials. He is currently studying the application of feed-back control techniques to distributed systems. His interests include parallel and distributed simulation, distributed computing, and VHDL-CAD. His email address is `<philip.wilsey@ieee.org>`.