Memory Management Techniques for Time Warp on a Distributed Memory Machine

PADS-9, Lake Placid, NY

Bruno R. Preiss and Wayne M. Loucks
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ont.

Introduction

- Goal of Parallel and Distributed Simulation
  - Completion of some task in a timely manner with the available resources.

- Possible Issues
  - Problem characteristics
  - Synchronization techniques
  - Scheduling techniques
  - Implementation Issues
  - **Memory Management techniques.**

Introduction

This paper examines:

- Two memory management techniques for a **distributed memory machine** using Time Warp Synchronization
  - One based on an extension to Artificial Rollback
  - One new scheme (Pruneback).

Memory Management

Why Does the Memory Model Matter?

- Shared memory model:
  - By sequencing execution in a specific manner it is possible to constrain the amount of memory required to the amount of memory required for a sequential simulation of the same problem.
  - Thus it is possible to consider memory optimal (in a Lin sense) operation.

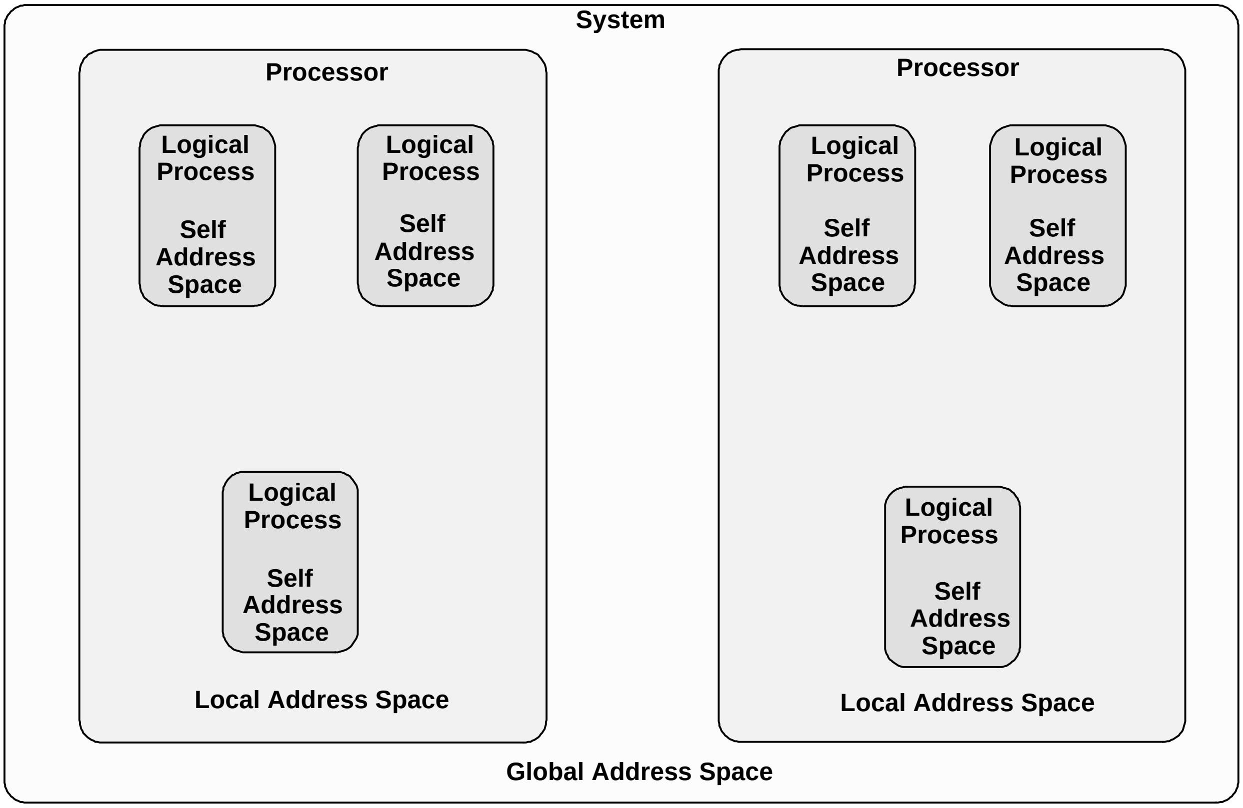
Memory Management

- Distributed Memory Model:
  - Pathological case:
    - * At time τ_1 (Simulation time) all of the active events are pending at processor A. Thus the entire (sequential) state must be stored in the memory at Processor A.
    - * At time τ_2 (Simulation time) all of the active events are pending at processor B. Thus the entire (sequential) state must be stored in the memory at Processor B.
    - * If we assume that the sequential state at times τ_1 and τ_2 is the maximum size of the sequential state, then we can argue that all N processors must have M bytes of storage and thus the system can never be Memory Optimal in the Lin sense.

Memory Stall Recover Techniques

- Recover **unused** memory. (I.e. States and messages with times smaller than GVT). — Most common technique is to advance GVT.
- Use less memory to save states
  - Check Point Intervals greater than 1.
  - Incremental Checkpointing
- Recover memory that is being used.
  - By changing LVT of some/all of the Logical Processes
  - By leaving LVT the same for all of the Logical Processes.

Triggers and Targets



Triggers:

- The region of memory that has been exhausted and cannot provide for the next request.

Target:

- The region of memory that must be used to recover memory from if further simulation is to be enabled.

Recovery Techniques

Rollback Based Techniques

- Message Sendback,
- Gafni's Protocol,
- Cancelback,
- Artificial Rollback

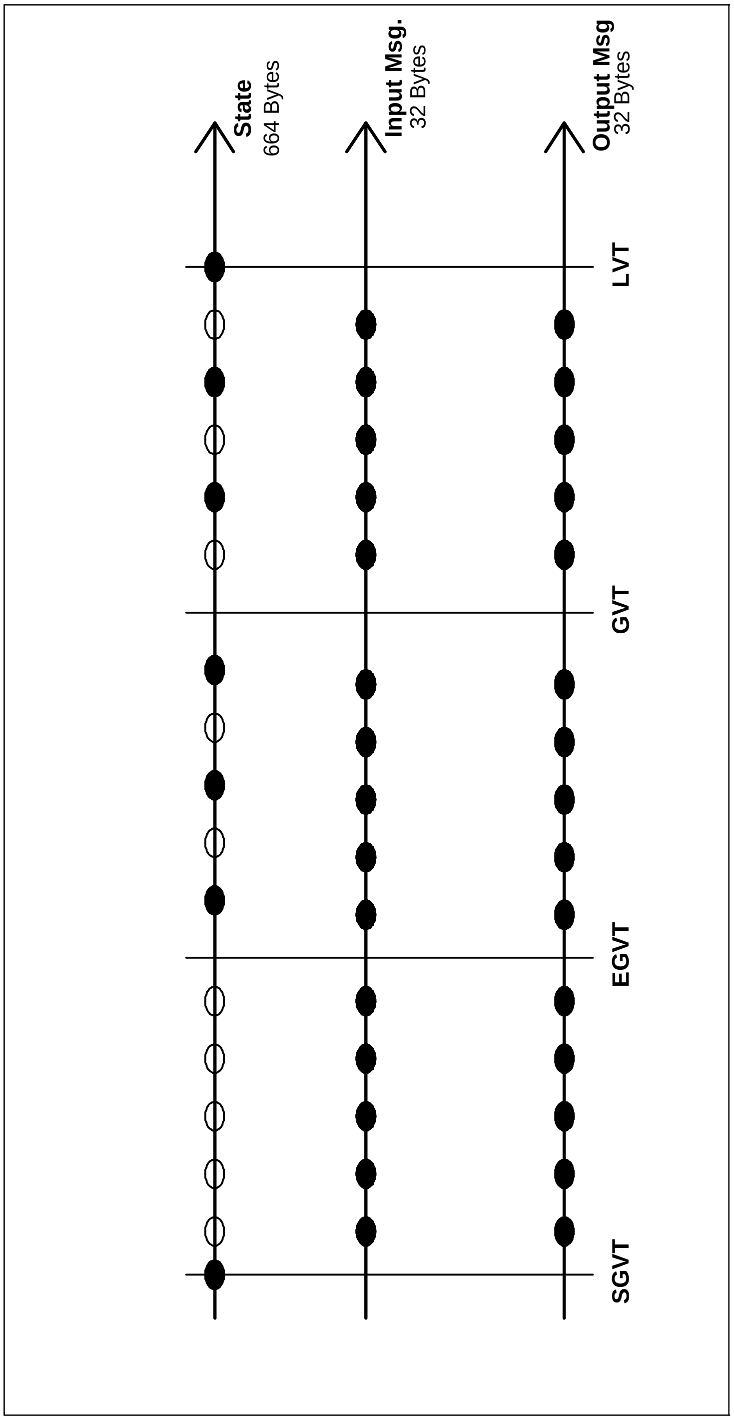
These are **Conservative** Memory Management Schemes, at the first sign of trouble they unwind some (potentially) useful calculations.

Non-Rollback Based Techniques

- Proposed Pruneback Technique
  - Based on the fact that not all of the states between SGT and LVT must be maintained in memory.

This is an **Optimistic** Memory Management Scheme, when trouble appears, discard enough to carry on but do not back up in simulation time until absolutely necessary.

Recovery Techniques



Empirical Results

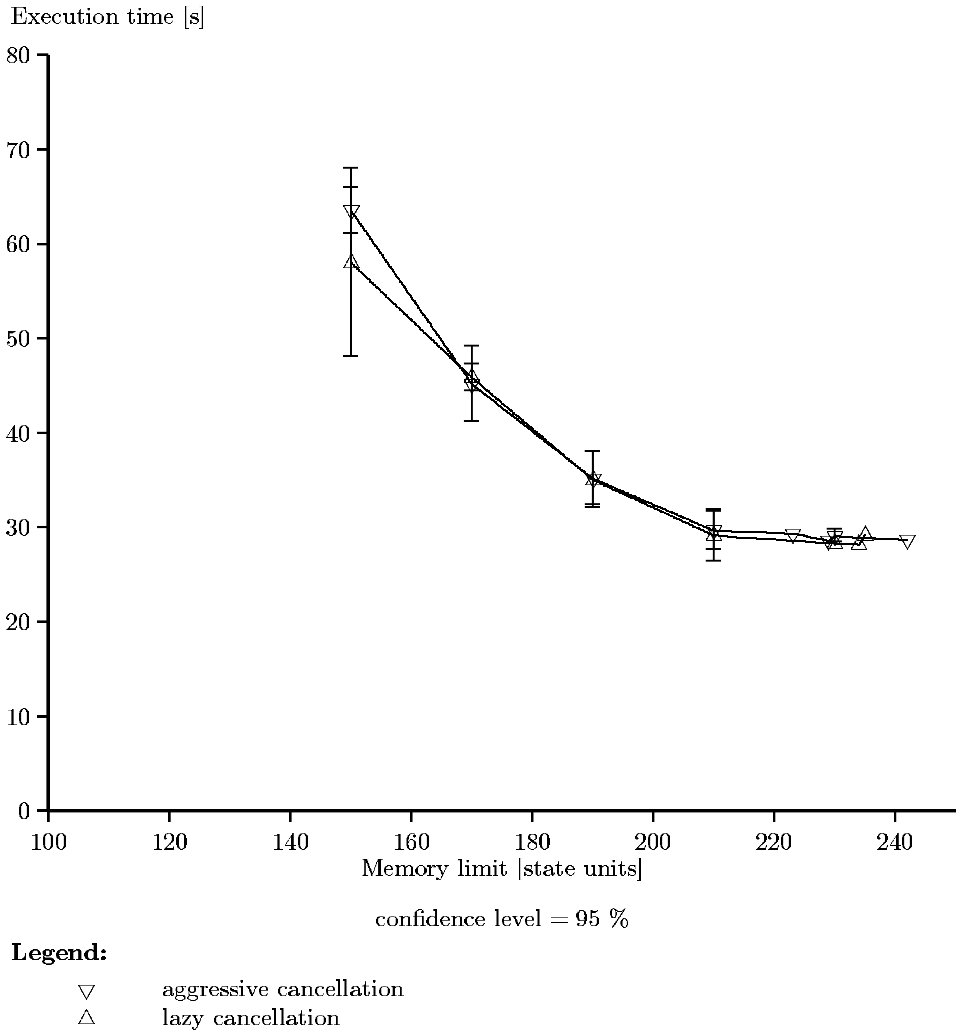


Figure 1: Artificial rollback—Execution time vs. memory limit.