

Time Warp Simulation on Multi-core Processors and Clusters

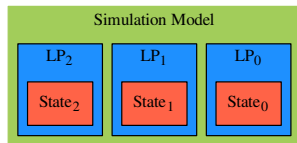
Douglas A. Weber
Masters Thesis Defense
Advisor: Professor Philip A. Wilsey

University of Cincinnati

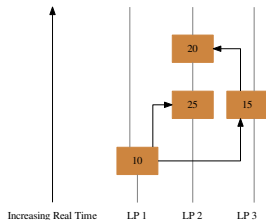
March XX, 2016

- ▶ Three main components
 - ▶ State variables
 - ▶ Simulation clock
 - ▶ Pending event set
- ▶ Unprocessed events stored in pending event set
- ▶ Events processed in time stamp order
- ▶ Simulation clock and state variables updated only when event occurs

- ▶ Model system as a set of Logical Processes (LPs)
- ▶ Events exchanged between LPs

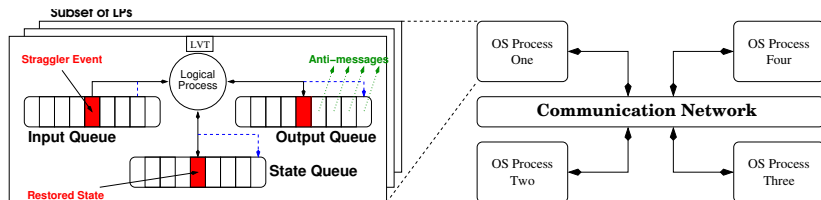


Possible Causality Violations

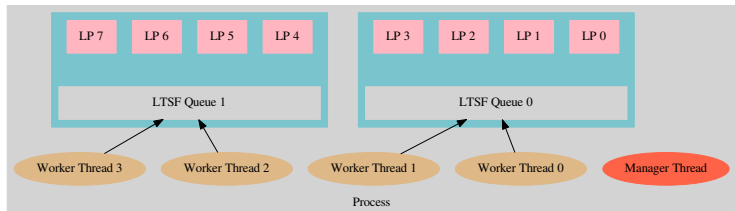


- ▶ Events can be received *and* processed out of order
- ▶ Two solutions
 - ▶ Conservative and Optimistic

Optimistic Mechanism



- ▶ Rollback Mechanism
 - ▶ State Restoration, Anti-Messages
- ▶ Local Virtual Time (LVT) & Global Virtual Time (GVT)
- ▶ Fossil Collection



- ▶ Worker Threads and Manager Thread
- ▶ LTSF Queues/Worker Thread - Contention vs Rollbacks
- ▶ LP Partitioning

Worker Threads

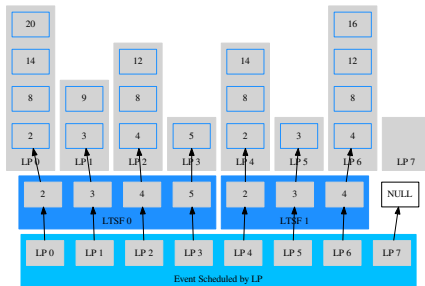
```
while termination not detected do
   $e \leftarrow \text{getNextEvent}()$ 
   $lp \leftarrow \text{receiver of } e$ 

  if  $e < \text{last processed event for } lp$ 
  then
    └ rollback  $lp$ 

  if  $e$  is an anti-message then
    └ cancel event with  $e$ 
    └ schedule new event for  $lp$ 
    └ continue

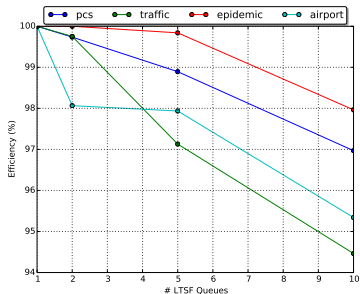
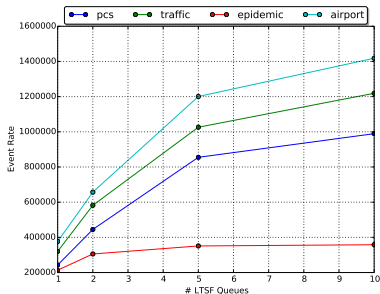
  process event  $e$ 
  save state of  $lp$ 
  send new events

  move  $e$  to processed queue
  replace scheduled event for  $lp$ 
```



- Unprocessed and Processed Queues
- Scheduling to LTSF
- Rollbacks

Sharing LTSF Queues

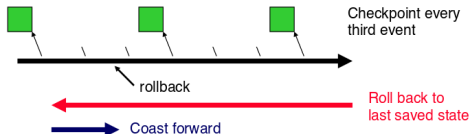


$$EventRate = \frac{CommittedEvents}{Runtime}$$

$$Efficiency = \frac{CommittedEvents}{ProcessedEvents} * 100\%$$

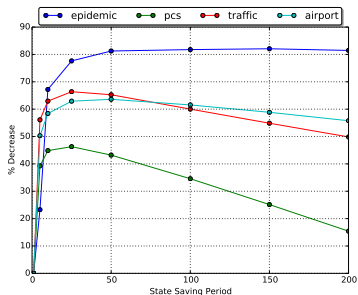
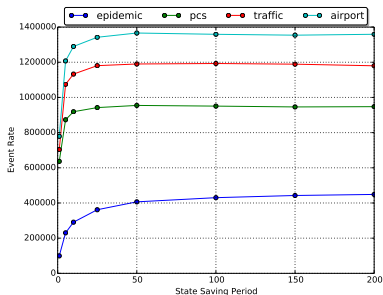
Reducing contention more important than reducing rollbacks

Save state only once every N events

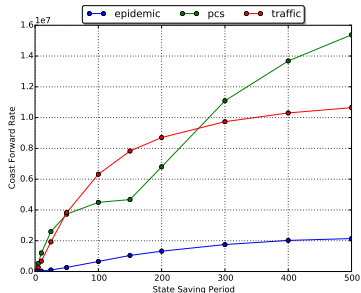
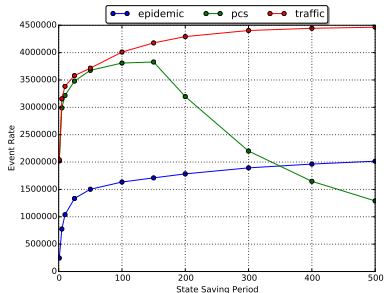


- ▶ Not all states available to roll back to
- ▶ Must "Coast Forward" to reproduce state
- ▶ Increases time to rollback
- ▶ Decrease time to save states and reduce memory footprint

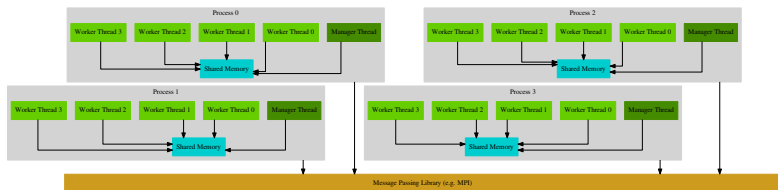
Intel® Xeon® X5675



8 Nodes - Intel® Xeon® E5410

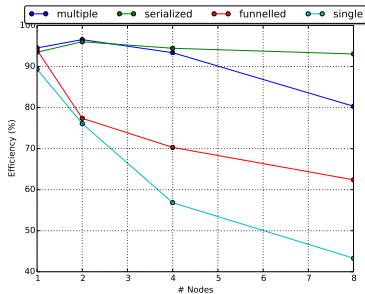
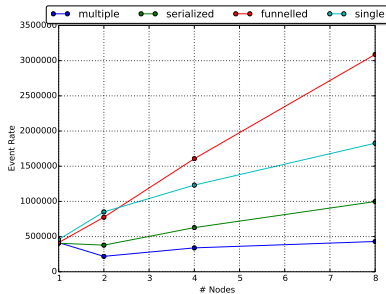


$$\text{CoastForwardRate} = \frac{\text{CoastForwardEvents}}{\text{Rollbacks}} * \text{RollbackRate}$$



Thread Message Passing Models

- ▶ Single threaded
- ▶ Funnelled
- ▶ Serialized
- ▶ Multiple



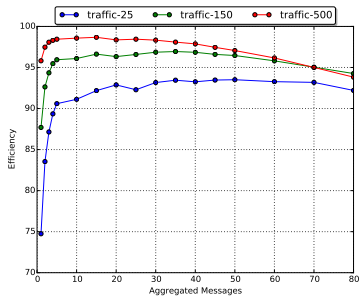
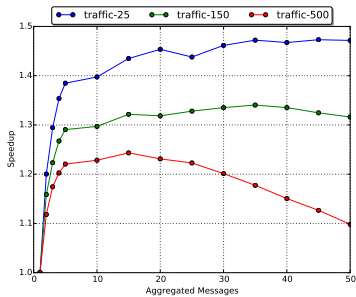
- ▶ Funneled/Single - Minimal Synchronization, High Latency
- ▶ Serialized/Multiple - Lots of Synchronization, Low Latency
- ▶ Performance will vary with model and partitioning.

- ▶ Wait
for N messages with same destination *process* and send together

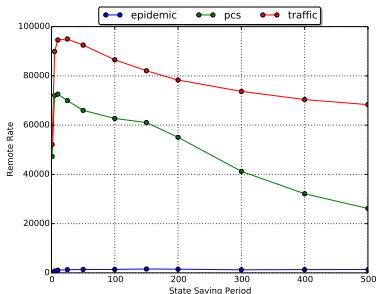
Num Msgs 4 bytes	Length 0 (L[0]) 4 bytes	Length 1 (L[1]) 4 bytes	Length 2 (L[2]) 4 bytes	...	Message 0 L[0]	Message 1 L[1]	Message 2 L[2]	...
---------------------	----------------------------	----------------------------	----------------------------	-----	-------------------	-------------------	-------------------	-----

- ▶ Pros
 - ▶ Single buffer allocated for multiple messages
 - ▶ Latency shared by multiple messages
 - ▶ Events destined for same LP may be sent together
- ▶ Cons
 - ▶ May increase event latency too high

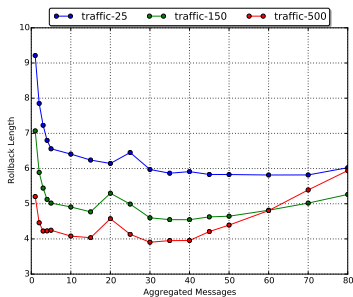
Message Aggregation - Traffic Model



- ▶ Best with small number of aggregated events
- ▶ Different behavior with different state saving periods



- Faster rate of remote events better

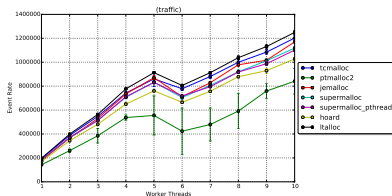


- Can lower rollback length?

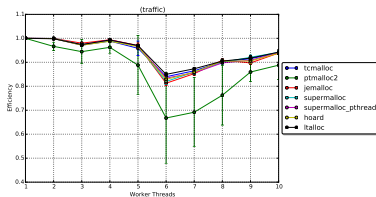
$$\text{AverageRollbackLength} = \text{RolledBackEvents} / \text{Rollbacks}$$

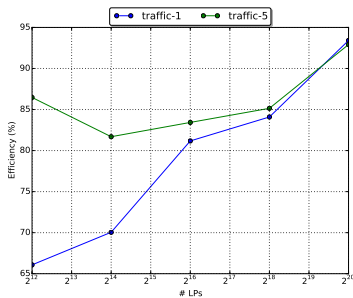
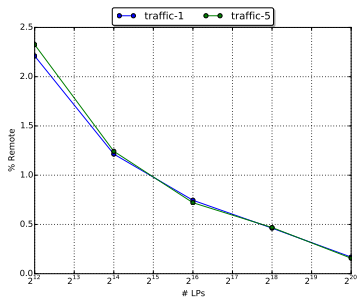
Intel® Xeon® X5675

Hyperthreading - 6 Cores, 12 Threads



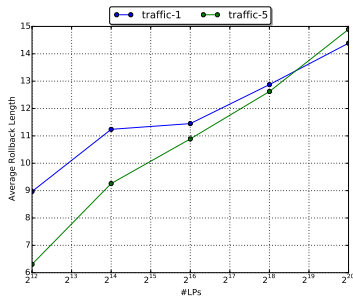
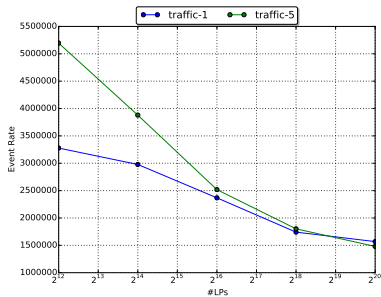
- ▶ Imbalance at 6 worker threads
- ▶ ptmalloc2 default in GLIBC - broken?





- ▶ Coarser Granularity
- ▶ Higher Efficiency

Scaling Simulation Models - Disadvantages



- ▶ More LPs to fossil collect
- ▶ Longer Rollbacks

- ▶ Conclusions
 - ▶ Synchronization should always be avoided
 - ▶ Handle interprocess communication on a single thread
 - ▶ 1 LTSF queue per worker thread
 - ▶
 - ▶ System can handle more rollbacks
 - ▶ Message aggregation works
 - ▶ Memory allocation problem
- ▶ Future Research
 - ▶ Optimistic Fossil Collection
 - ▶ Scaling up