
CAISSON
Interconnect Network Simulator

Paul Springer
June 10, 2006

- History
- CAISSON Architecture
- Parallel Discrete Event Simulation
- WarpIV
- CAISSON Performance



BG/L

- Model 3D torus interconnect
- Sponsored by LLNL for pre-purchase evaluation
- Highly scalable
- Built on SPEEDES PDES engine
- Run on Origin 2000

Cascade

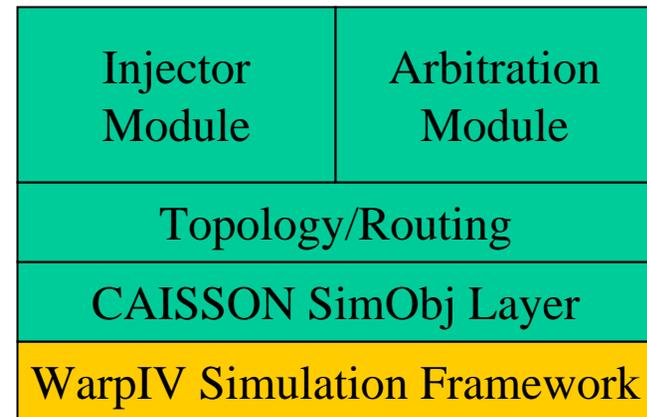
- Cray response to HPCS initiative
- Model future petaflop computer interconnect
- Parallel discrete event simulation techniques for large scale network simulation
- Built on WarpIV engine
- Run on laptop and Altix 3000

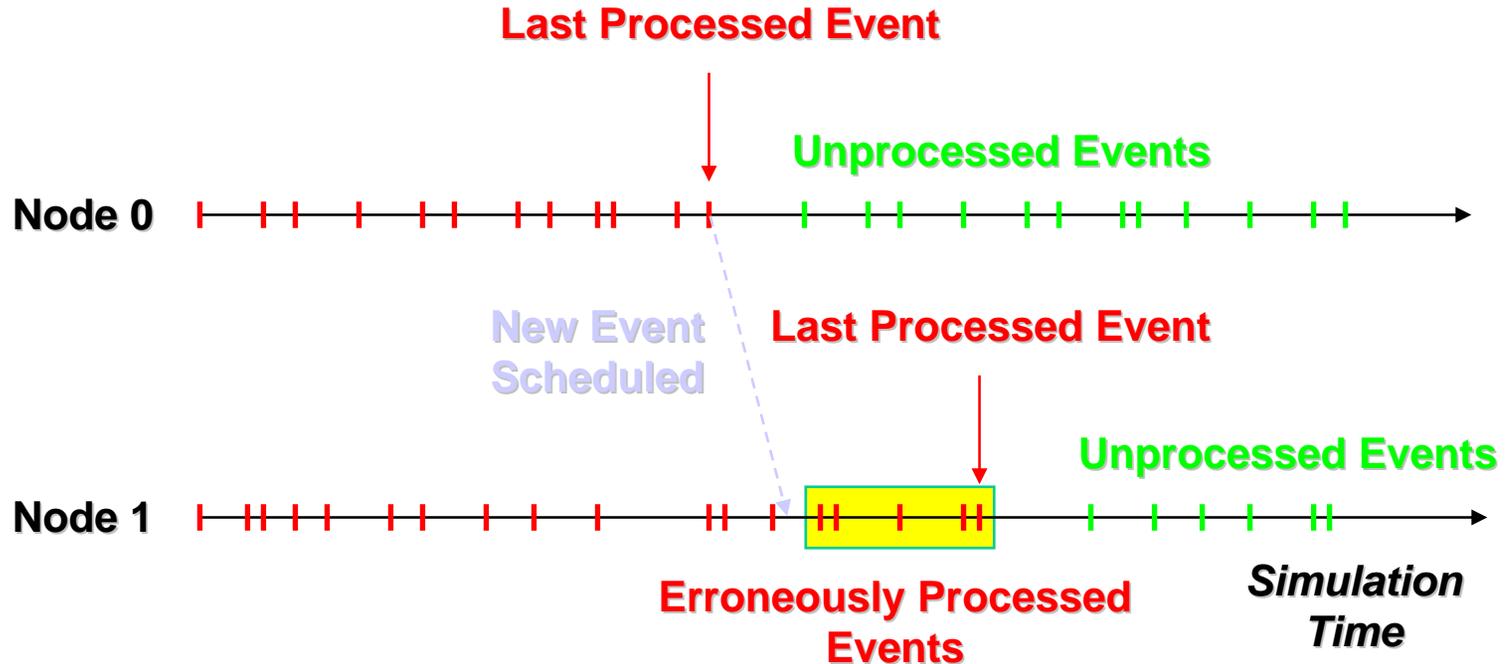
CAISSON Strengths

- Can be sized up to 1000 simulated nodes per host node
- Good parallel scaling characteristics
- Flexible: multiple injectors, arbitration strategies, queue iterators, network topologies

Features

- Flexible modules
- Highly scalable
- Built on WarpIV PDES engine
- Runs on Altix 3000



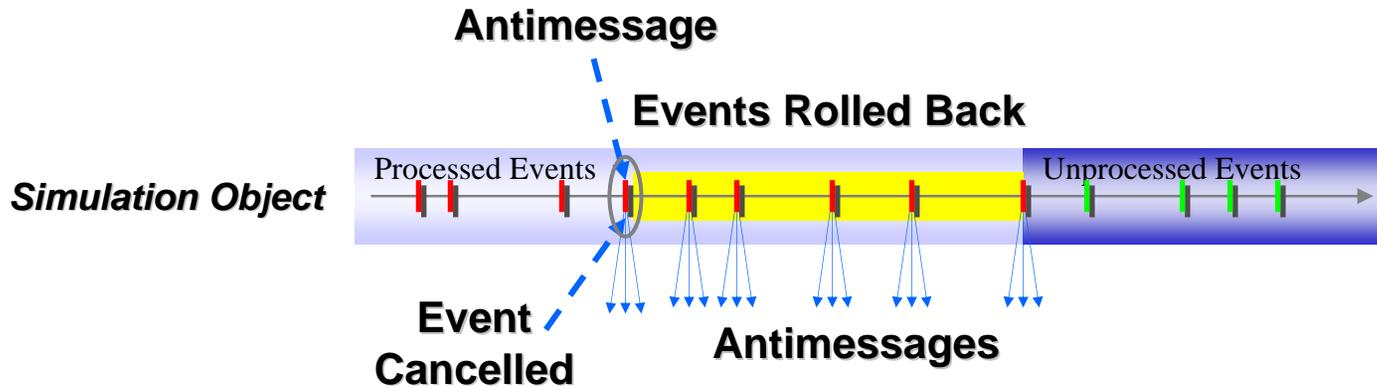
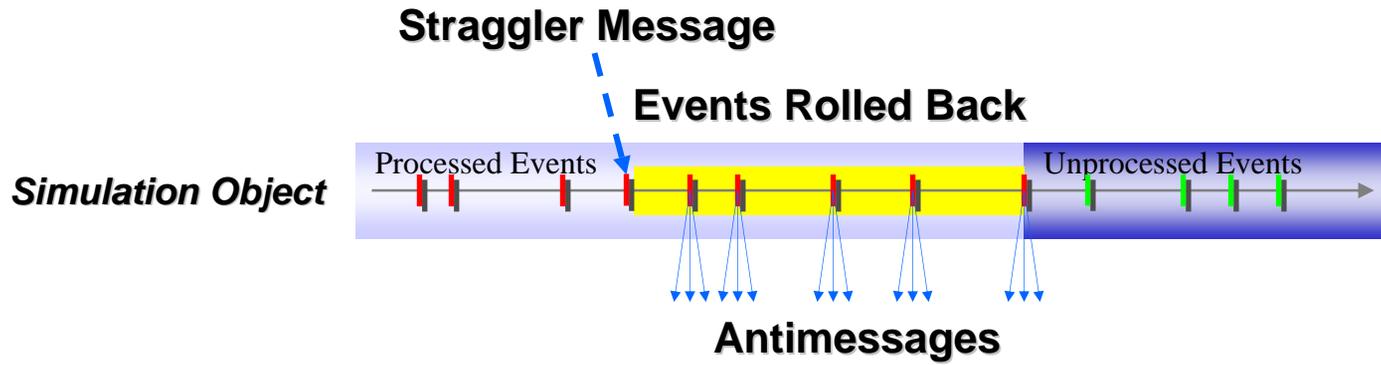


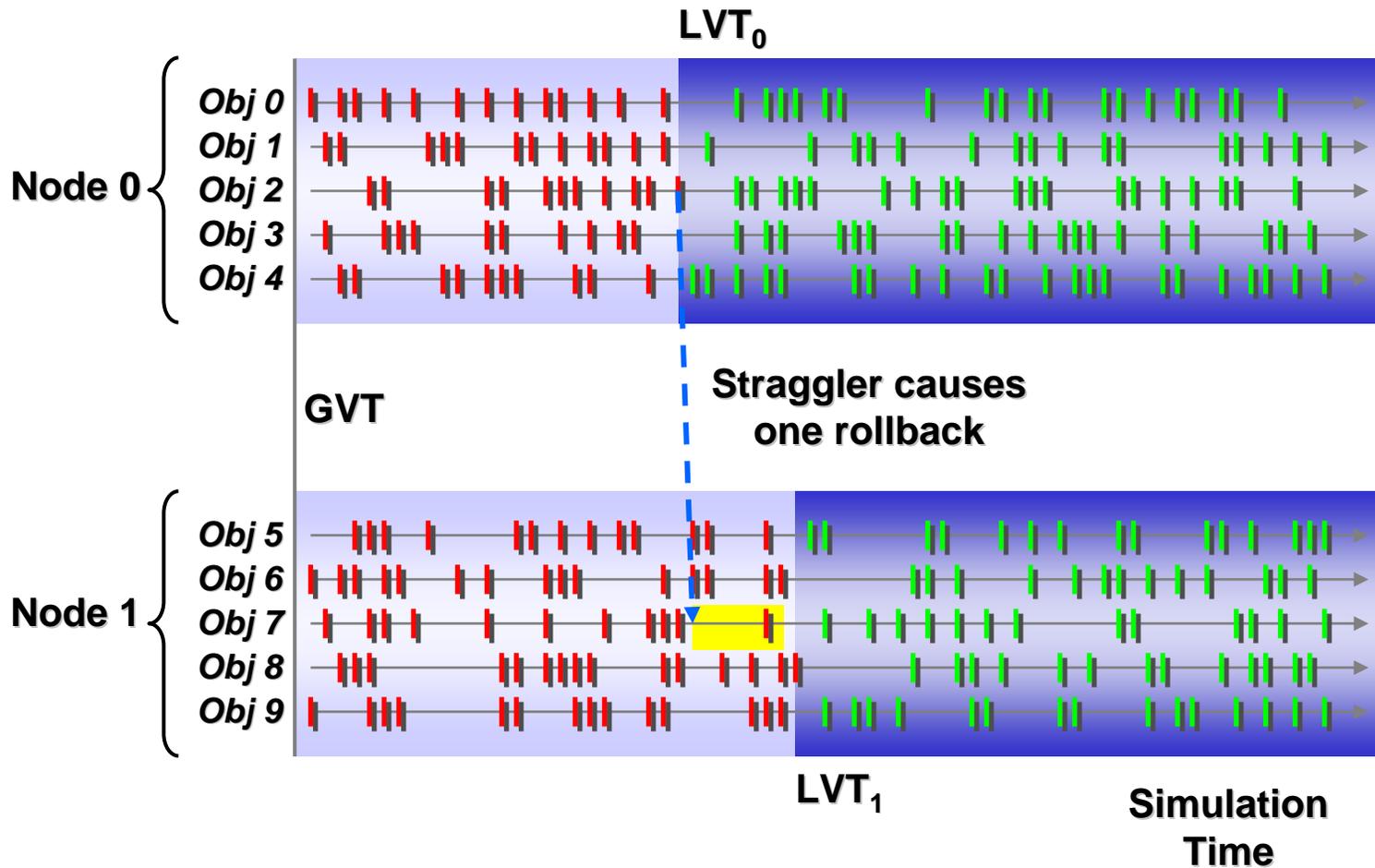
Conservative approach: Never allows events to be processed if it is possible for “straggler” messages to arrive from other nodes

Optimistic approach: Fixes straggler message problem by rolling back state and canceling generated events

<u>Discrete Event Simulation</u>	<u>Time-stepped Simulation</u>
Events plucked from event queue for execution	“for” loop processing
Time resolution varies for different subsystems and times	Uniform time intervals
Event processing only occurs if and when necessary	Processing occurs every time step

- Time Warp mechanism
 - Straggler message arrives
 - Causality error detected
 - State rolled back to values it had at time of straggler's time stamp
 - Messages unsent
 - Previous work re-executed
- Sophisticated engine required
 - Support for checkpointing, rollbacks, message cancellation
- Strengths
 - Particularly scalable for simulations with spatial and temporal inhomogeneities (e.g. network simulations)

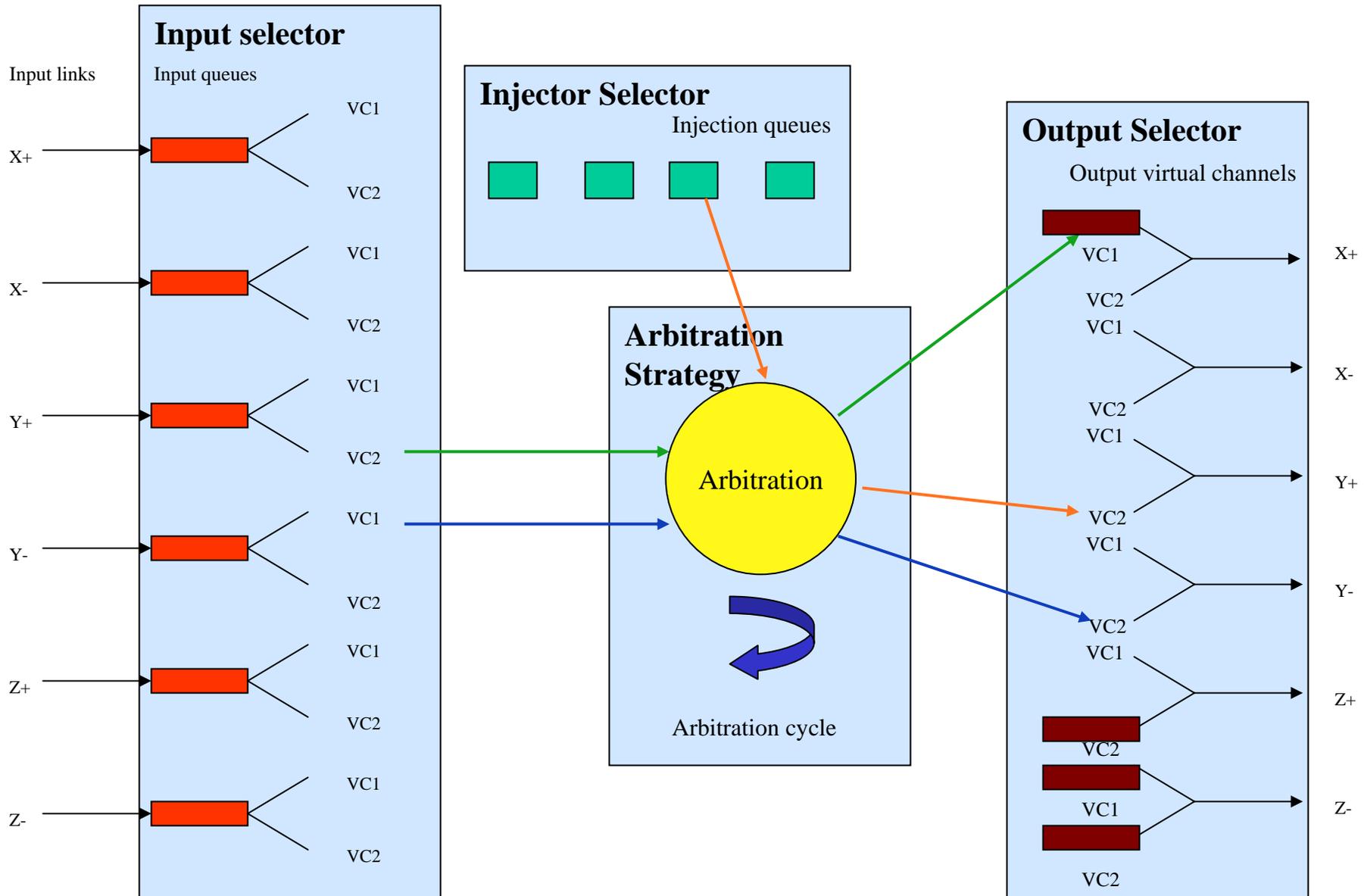




- Time Step Simulation
 - Barrier calls at each step
 - Load imbalance at one node delays all
- Optimistic Simulation
 - Rollbacks
 - Temporal load balancing

- “Rollbackable” equivalents of common types
 - RB_int, RB_double, RB_ptr, etc.
- Rollbackable memory allocation, deallocation
- Rollbackable container classes
 - List class, tree class, etc.
- RB_cout for committed output
 - printf output can be helpful and/or puzzling
- WAIT macro for support of persistent processes

Sample Node Packet Scheduling Model



- Optional Strategies

- Queue Order

- Each arbitration cycle starts at queue 0, checking in sequential order (good for baseline benchmark)

- Round Robin

- Each arbitration cycle remembers where it left off in the previous cycle

- Least Recently Served Priority

- Priority given to least recently served queue

- Interfaces

- first()

- return first queue to check

- next()

- get next queue to check

- isDone()

- return true if queue check cycle has completed

- successful()

- tell selector that this queue had a packet sent out

- unsuccessful()

- tell selector this queue could not send its packet

Input Selector

Queue Order

first()
next() •
isdone()
successful()
unsuccessful()

Round Robin

first()
next() •
isdone()
successful()
unsuccessful()

LRS Priority

first()
next()
isdone()
successful() •
unsuccessful()

Arbitration Strategy

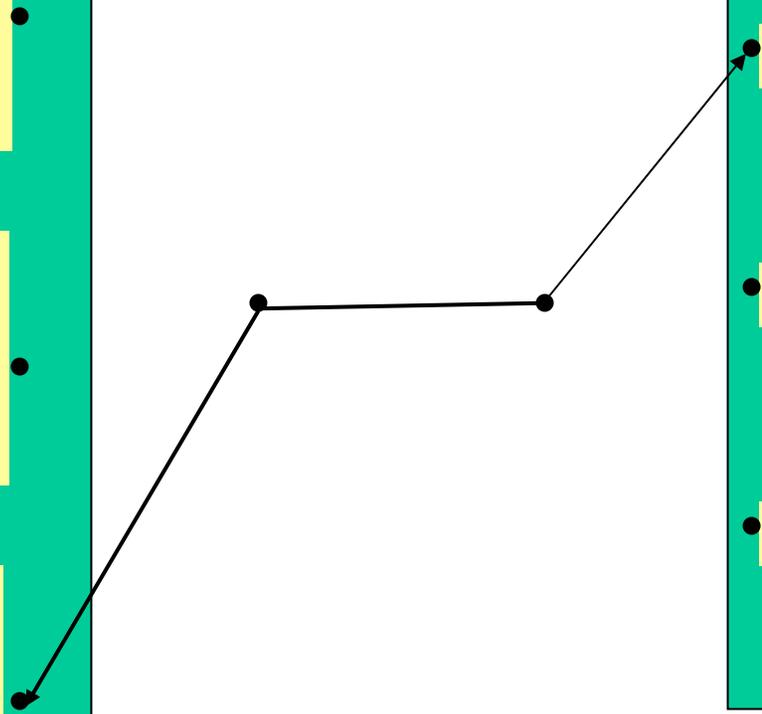
Credit-Based Flow Control

 •

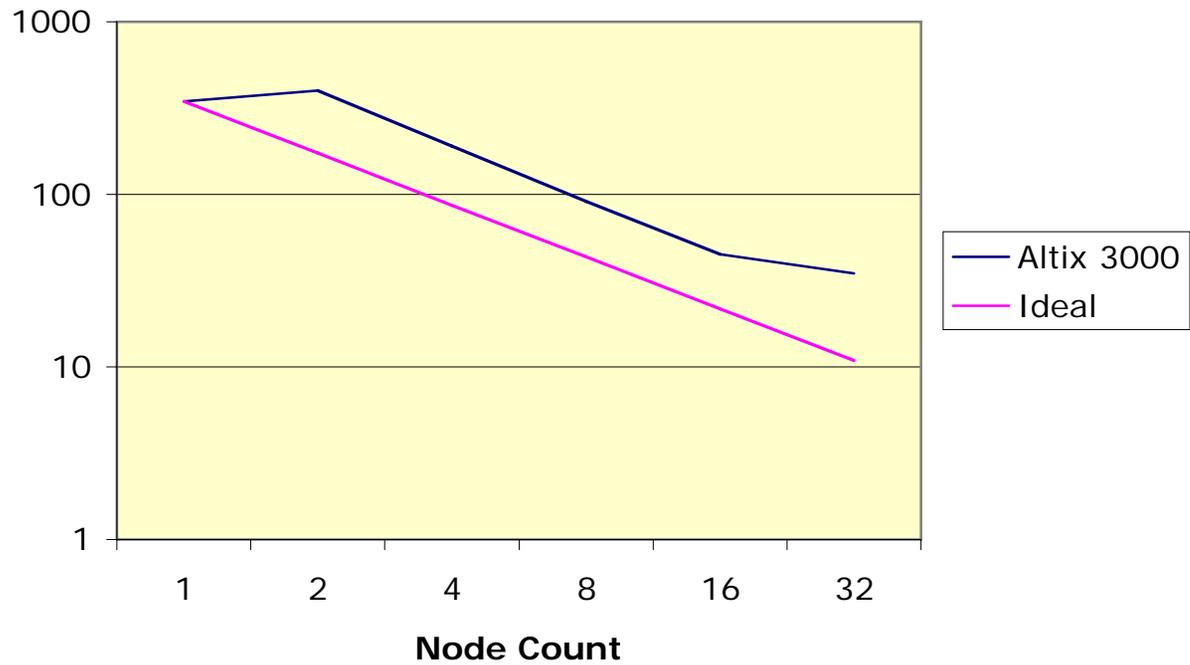
On/Off Flow Control

 •

Ack/Nack Flow Control

 •

Parallel CAISSON Run Times



Configuration	BG/L nodes	Physical Processors	Total injected packets	Average hops / pkt
16x16x16	4K	8	400K	12
16x16x32	8K	16	800K	16
16x32x32	16K	32	1.6M	20
32x32x32	32K	64	3.2M	24
32x32x64	64K	128	6.4M	32

- This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The funding for this research was provided for by the Defense Advanced Research Projects Agency under task order number NM0715612, under the NASA prime contract number NAS7-03001.