Modeling the BlueGene/L 64K Node Network - Statistical Simulator

Jim Pool,Thomas Sterling, Dan Meiron,
Sharon Brunett, Maciej Brodowicz,
Tom Gottschalk, Paul Springer,
Ed Upchurch

California Institute of Technology
and
NASA Jet Propulsion Laboratory

BlueGene/L Workshop
13 August 2002

Objectives

- Understand implications of BG/L network architecture & Drive results from **real-world ASCI applications**
- Develop **statistical models** of: applications, processors as message generators, and the network
  - **BUT keep:**
    - **application communications distribution**
    - **Network contention as function of load/size – adaptive routing**
- Represent **64K Nodes Explicitly** in Statistical Model
- Create trace analysis tools to **characterize applications**

Strategy

- Develop **“Rapid Prototype”** Statistical Model Using Commercial Graphical Modeling Tool (SES/workbench)
- Implement 64K node statistical network simulator – **parallel version**
  - **SPEEDES - Paul Springer**
  - **FPGA – Maciej Brodowicz**
  - **Our Own – Tom Gottschalk**
- Perform application driven experiments **(bottleneck/sensitivity analyses)**
- Validate against cycle-level simulations for **small systems**

Applications

- RM3D/AMR3D
  - Science: compressible turbulence
  - Uniform & Adaptive Mesh
- Magnetic Hydro Dynamics (MHD)
  - Science: magnetic reconnection in two dimensions solves hydrodynamics and resistive Maxwell's equations
  - Data exchanges - nearest neighbor non blocking send and receive
  - global reduction, MPI_Allreduce of the minimum time step
- Gyrokinetic Toroidal Code (GTC)
  - Science: GTC is a Particle in Cell (PIC) - calculates micro-turbulence in a tokamak
  - a few MPI_allreduce, almost all MPI calls are nearest neighbor
  - communications done in a circular fashion, and using MPI_sendrecv
- Quantum Monte Carlo
  - Science: obtains electronic structure of molecules and materials
  - manager - MPI_isend msgs directly to each worker to gather statistics
  - Workers check the incoming buffer with a polling, MPI_Iprobe; MPI_Reduce for further statistics gathering

SPEEDES Background

- Synchronous Parallel Environment for Emulation and Discrete-Event Simulation (SPEEDES)
- **Parallel discrete event simulation framework**, developed at JPL by Jeff Steinman, early 90's
- Used for large-scale military simulations – SPAWAR: 100 node SMP to **simulate 1,000,000 objects**
- **Optimistic** approach, "breathing time warp"
- Uses time windows to prevent runaway objects from triggering excessive rollbacks
- Uses **shared memory** for message passing on SMP computers

SPEEDES Performance Test

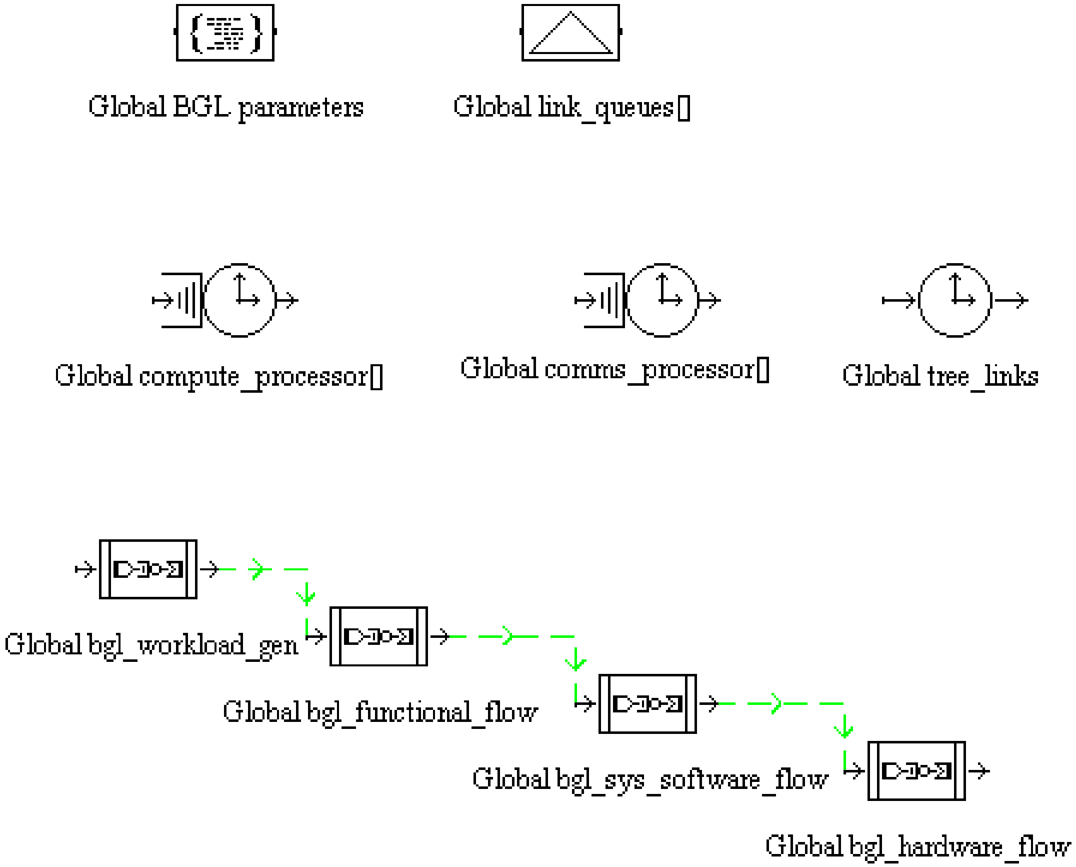
- Ported SPEEDES to JPL’s 128-processor SGI 2000
- Ran approx 1/10th scale performance test
- 10 simulation seconds
- Randomly chosen destination
- Total elapsed time: 15 seconds
- Speedup Approx 8x over single node

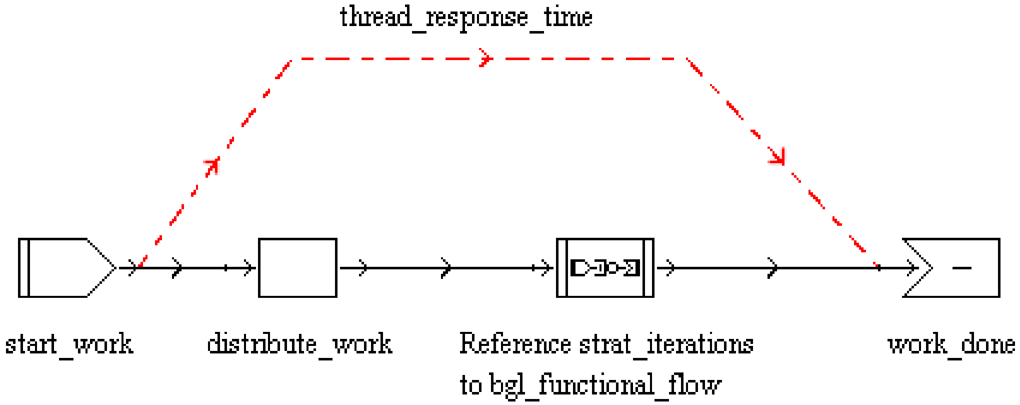
Hardware-Accelerated Network Simulator

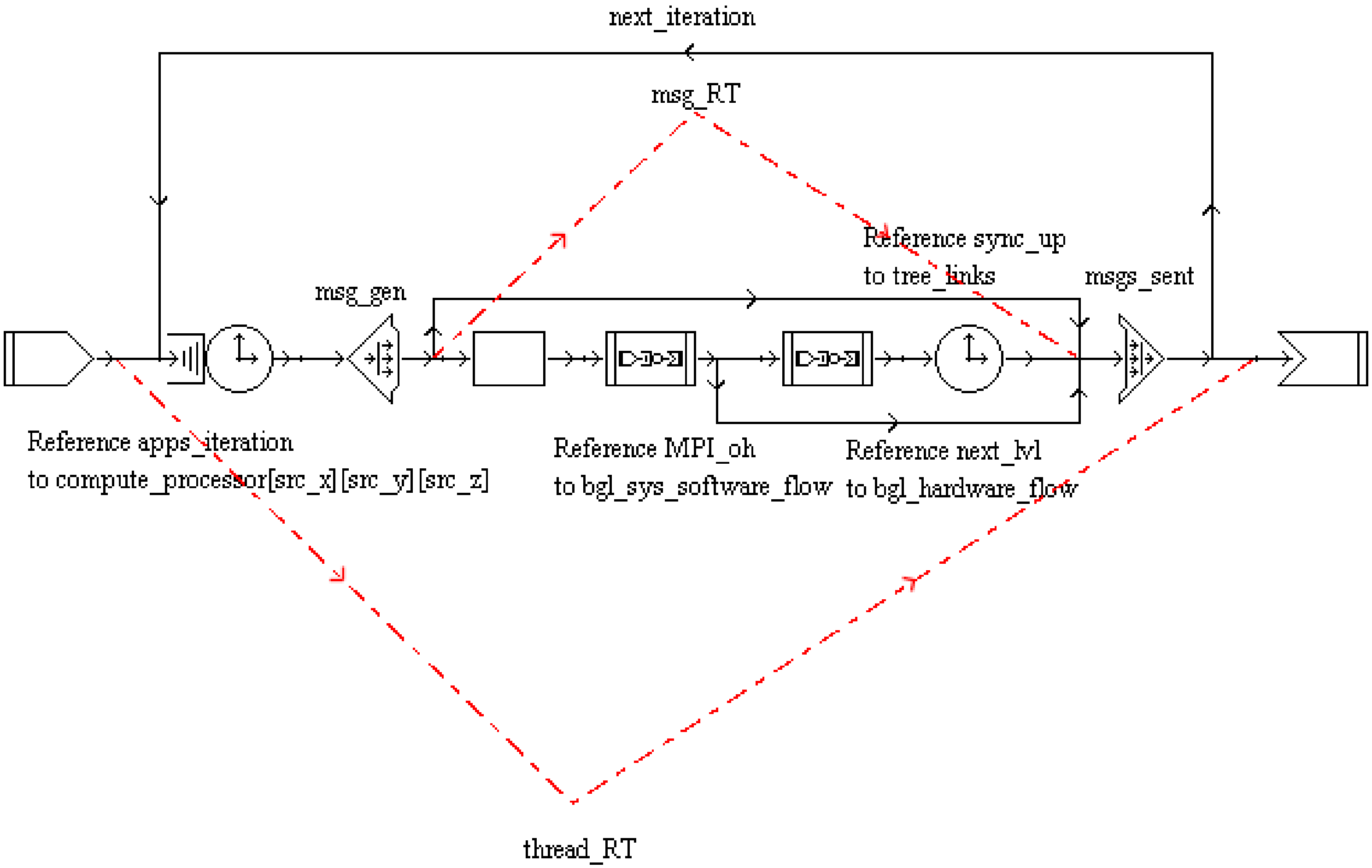
- **FPGA** fast simulation of 8x8x8 torus network (scalable to larger networks)
- We have 2 Xilinx XCV600E FPGAs (Nallatech – UK)
  - 985,000 system gates each
  - Nearly 300kbits of dual-ported on-chip memory
- **Over 1 billion events/sec at 50 MHZ clock rate**
  - SES/workbench prototype **100,000 events/sec** on 700 MHZ PC
- Routing and buffering algorithms translated directly to FPGA logic
- Each of 512 emulated communication nodes can use up to 50 logic cells and 1kbit of memory (queuing)
- Could act as a testbed for various communication scenarios (with test datasets supplied on the fly by the driver code running on a host PC)

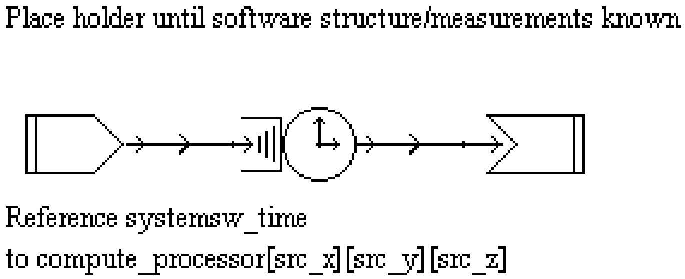
Prototype Model

- SES/workbench
  - Torus network topology parameterized (x,y,z)
  - Flexible workload generator
  - Thread – message- packet level
  - Can handle 8x8x8 – memory limitations
  - Most 8x8x8 runs minutes – depends on workload


Global BGL parameters
Global link_queues[]
Global compute_processor[]
Global comms_processor[]
Global tree_links
Global bgl_workload_gen
Global bgl_functional_flow
Global bgl_sys_software_flow
Global bgl_hardware_flow

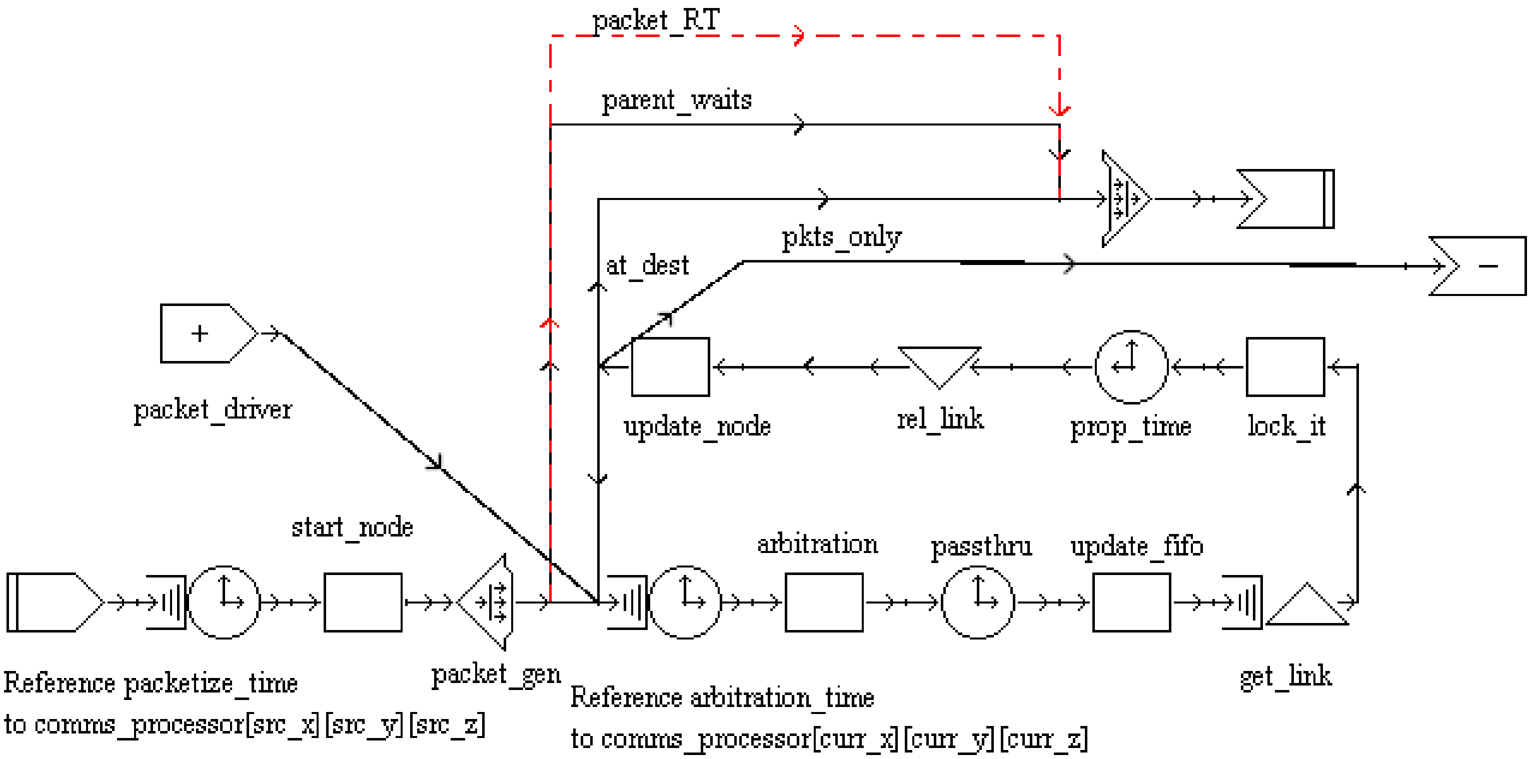

thread_response_time
start_work
distribute_work
Reference strat_iterations
to bgl_functional_flow
work_done


next_iteration
msg_RT
Reference sync_up
to tree_links
msgs_sent
msg_gen
Reference apps_iteration
to compute_processor[src_x] [src_y] [src_z]
Reference MPI_oh
to bgl_sys_software_flow
Reference next_lvl
to bgl_hardware_flow
thread_RT



Place holder until software structure/measurements known

Reference systemsw_time
to compute_processor[src_x] [src_y] [src_z]


packet_RT
parent_waits
at_dest
pkts_only
+
packet_driver
−
update_node
rel_link
prop_time
lock_it
start_node
arbitration
passthru
update_fifo
packet_gen
get_link
Reference packetize_time
to comms_processor[src_x] [src_y] [src_z]
Reference arbitration_time
to comms_processor[curr_x] [curr_y] [curr_z]

Task Definition

- Scalable Simulation Of Messages For Large Parallel Machines

- Open/Unknown: Adequate Fidelity
  - Seek Guidance In Usual Cost/Benefits/Fidelity Trade Space

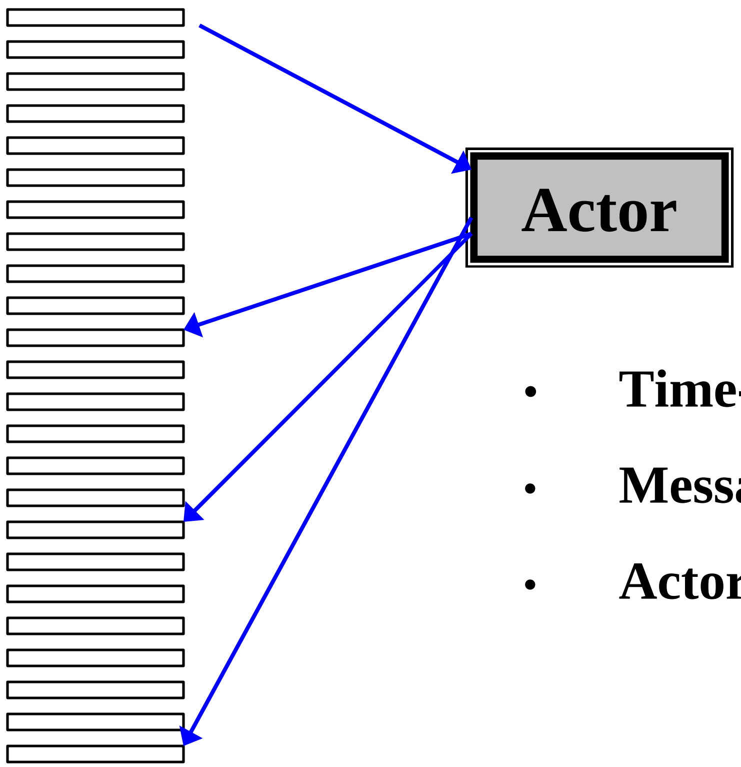
Design Assumptions

- Highest Priority
  - Adequate Packet Modeling
  - Routing Procedures (Cut Through, etc.)
- Approximations, First Pass
  - Statistical, “Semi-Correlated” Message Generation
  - Receipt: Timing Statistics, Nothing More
  - Collectives (e.g., Barriers) Ignored

Basic Simulation Formalism

- (Distributed) DIS, Messages Among Actors
- Messages: User or Packet-Level Data Representations
- “Actors”: Message Producers/Consumers
  - Apps Processor: Generation
  - Outgoing Packetizer
  - Routers
  - Packet Collectors, Message Receipt

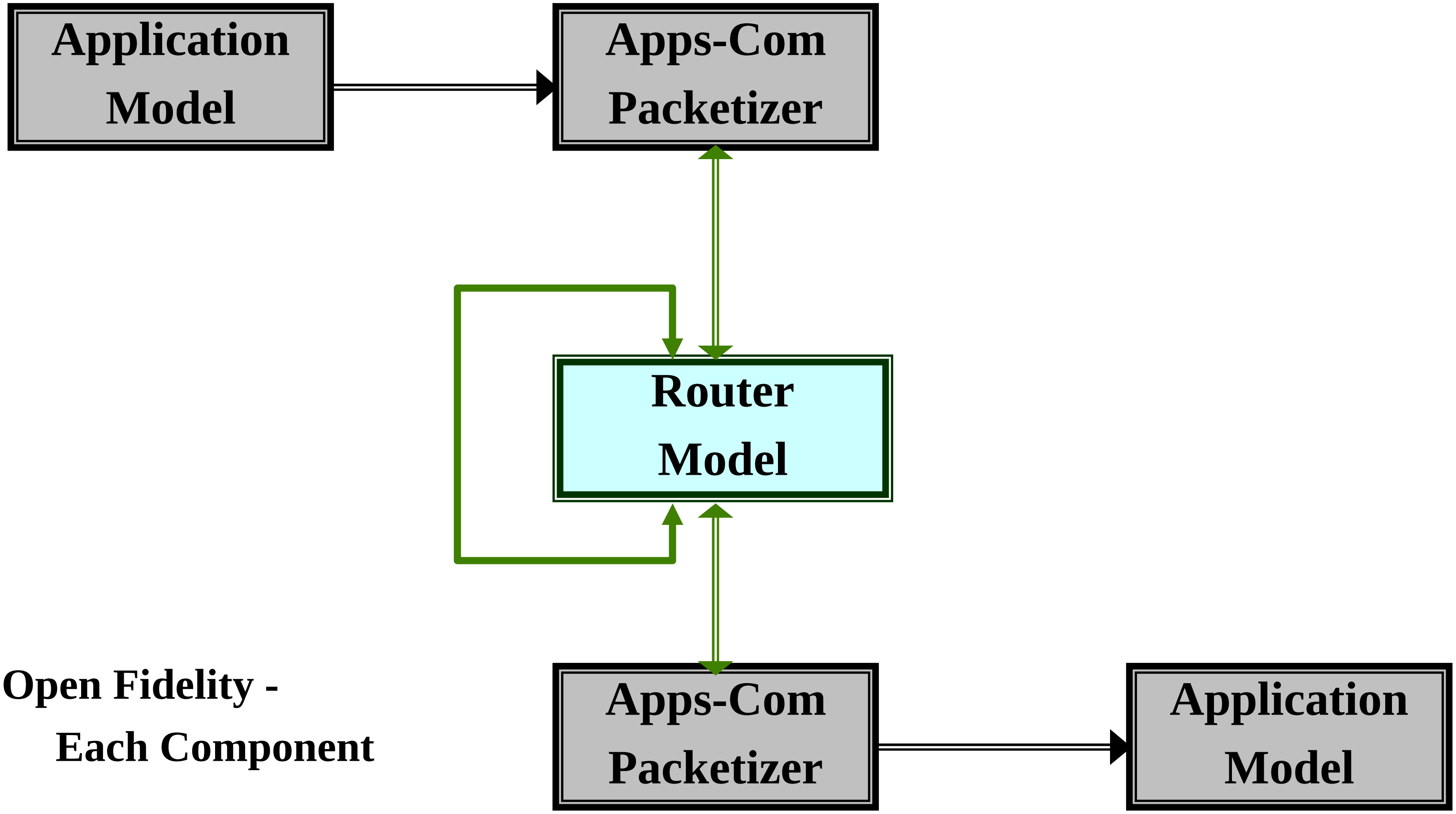
Basic Framework



- Time-Tagged Event Queue
- Messages Tied To Receiving Actors
- Actors Spawn New Messages For Queue

Event (Message) Queue

Simulation Objects



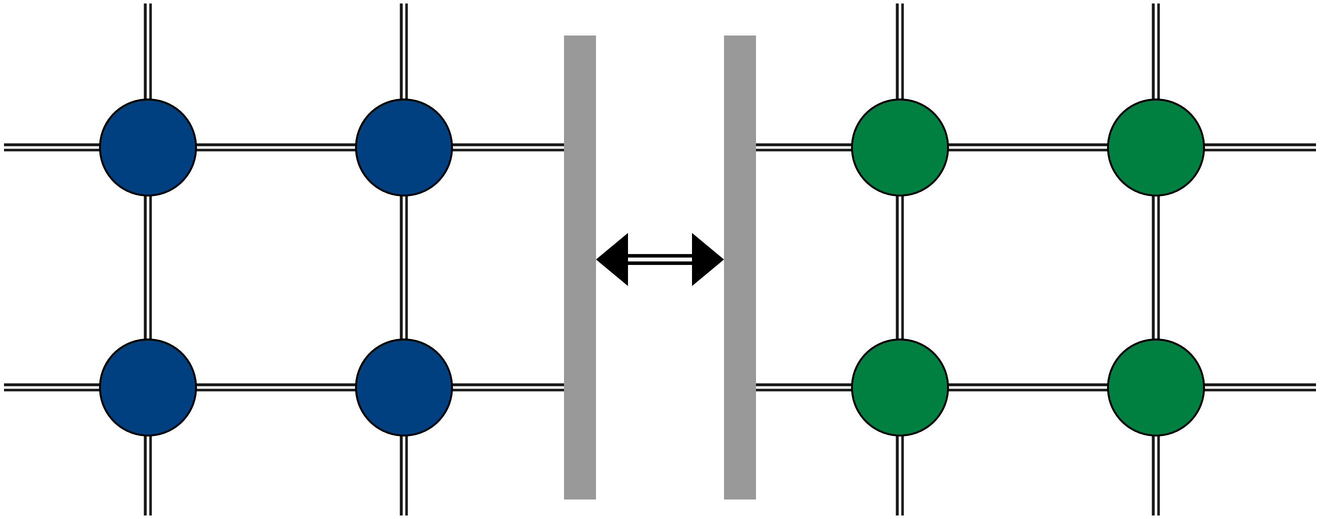
Open Fidelity -

Each Component

Simulation Objects II

- Applications Objects
  - Message Sources, Statistical
- Packetizers
  - Message <=>Packet Translator
  - Applications-Communications Interface
- Communications (Router)
  - Packet Communications. FIFO, Cut-Through, Tokens, etc. As Needed

Scalable Extensions



“Soft” Event Queue Management Across Simulator Nodes: Time Delayed/Shifted Packets Across Boundaries

Scalable Extensions II

- Messages "Through" Boundary Accumulated Into Time-Stamped Set
- Periodic, Scheduled Swaps At Boundaries
  - New Events Within Simulation Queue
- Swapped Messages To Simulation Queues
  - Time Stamps Sifted By Accumulation Time
  - Adequate For "Near-Steady-State" Modeling

Current Activities

- Framework
  - Development/Testing Of Distributed DIS Approximation With Toy Actors
- Router Modeling
  - Abstract Lessons/Fidelity From WorkBench Studies
- Message Generation
  - Statistical Representations Of Traces
    - Types, Sizes, Hop-Counts For Messages
    - Sequence Correlations Within Single Processor