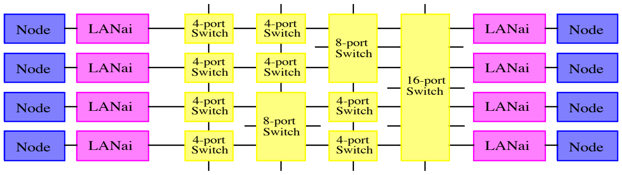
High Performance Scalable Computing (HPSC) Performance Modeling Using Ptolemy

Eric K. Pauer
Sanders, a Lockheed Martin Company
Signal Processing Center
Nashua, NH 03061-0868
pauer@sanders.com

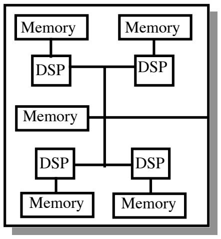
High Performance Scalable Computing (HPSC)

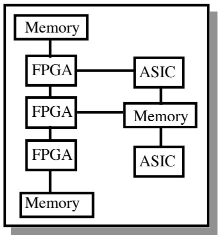
- HPSC architecture provides:
  - high data bandwidth
  - distributed processing
  - real time processing
- Goal is to simplify development by separating:
  - application software implementing algorithm
  - system software passing data among processing nodes
- HPSC comprised of:
  - Processing nodes
  - LANai (network interfaces)
  - Myrinet network of switches

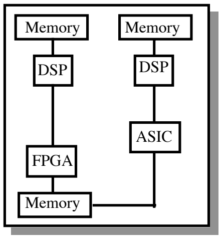


Processing Nodes

- Implement application algorithms
- Consist of
  - one or more digital signal processors and/or RISC processors
  - programmable hardware logic like Field Programmable Gate Arrays (FPGAs) or Application Specific Integrated Circuits (ASICs)
  - a combination of the above







LANai

- acts as the interface between the processing node and the network
- has independent transmit and receive sections
- transmits and receives data at 160 Mbyte/second rate
- LANai has high speed dedicated static RAM to load and store data
- Data synchronization tables are used to route data through network (transmit) or organize incoming data from network (receive)
- LANai transmit side creates packet header

LANAI Transmit DST

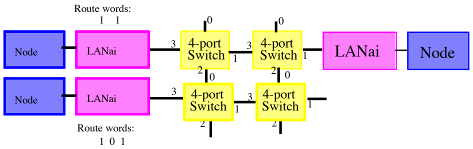
Packet	Address	Size	Route words	Desk Index
0	0x40000000	512	0 4 3 2	4
1	0x40000200	256	1 2 0 3 6	2
:		:	:	:
N-1	0X40001100	2048	517	1

LANAI Receive DST

Packet	Address	Size
0	0x70000000	1024
1	0x70000400	256
:	:	:
M-1	0x70001000	512

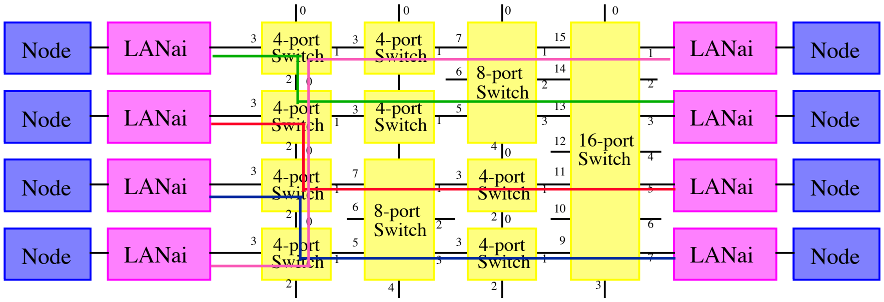
Myrinet Network of Switches

- Myrinet network is comprised of a network of multi-port switches
- Ports have independent transmit and receive ports
- Most common are 4-port, 8-port, and 16-port switches
- Have throughput of 160 Mbytes/second
- Operate by extracting port number from header, and passing data packet through specified transmit port
- Very low latency
- No buffering - packet is transmitted as soon as header is decoded
- Must handle contention when multiple packets from different receive ports are addressed to same transmit port



Myrinet Example

No Contention

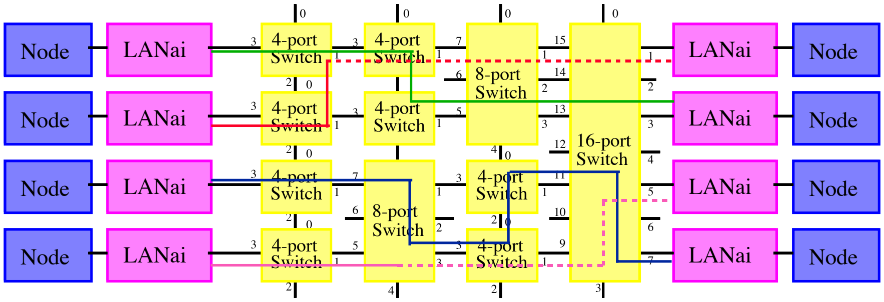


Route Words

2 1 1 3 3
2 1 1 1 5
2 1 3 1 7
0 0 0 1 1 1 1

Myrinet Example

Contention



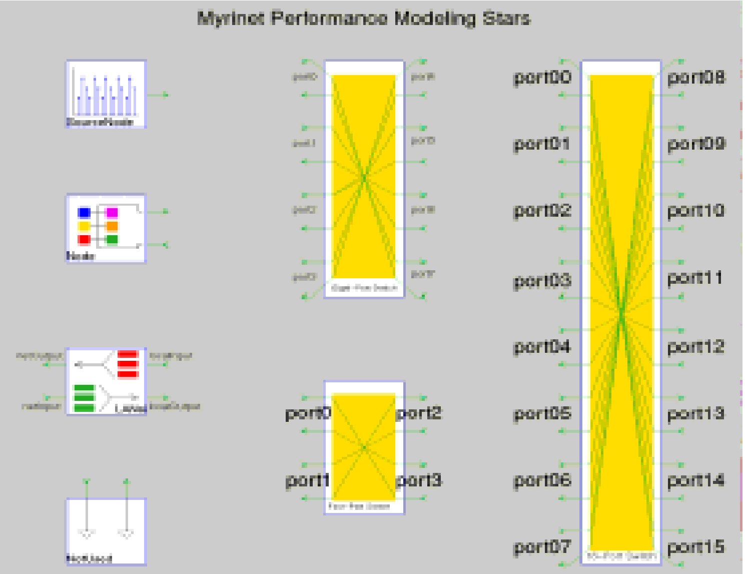
Route Words

1 2 1 3 3
0 1 1 1 1
1 3 0 1 7
1 3 1 5

Performance Modeling Stars

- Discrete Event (DE) Domain: event-driven model of computation
- SourceNode star: creates data blocks at specified rate
- Node star: processes data blocks at specified rate
- LANai star
  - using data blocks from the SourceNode or Node, the transmit side of LANai creates data packets to transmit to the network
  - receive side of LANai receives data packets from the network and reassembles data packets to create data blocks for the Node
  - receive side also receives control packets to suspend or resume transmission of data
- Switch star
  - receives data or control packets on one port and retransmits them on another port
  - must handle contention and send appropriate control packets to suspend or resume data transmission
- NotUsed star: used to terminate unused ports on Switch stars

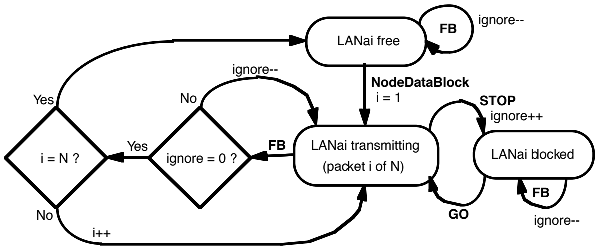
Performance Modeling Stars



New Ptolemy Particles

- NodeDataBlock represents block of data sent to/from SourceNode or Node from/to LANai
- Packet particle
  - serves as pure virtual (abstract) base class for other packets
- DataPacket particle
  - derived from Packet
  - represents typical Myrinet data packet
- ControlPacket particle
  - derived from Packet
  - represents Myrinet control packet
  - STOP or GO control packet
- Feedback particles (modified)
  - used on internal feedback queues of stars to cause the star to be revisited (executed) at a future time

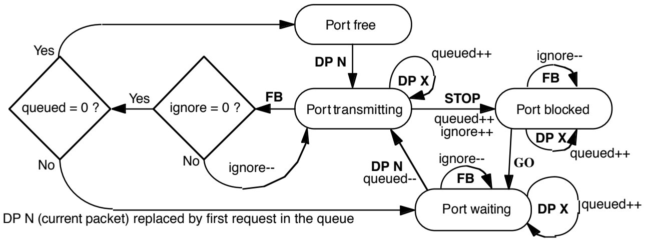
Myrinet LANai State Diagram



State Diagram of Myrinet LANai Behavior

- **illustrates behavior as DataBlock consisting of N data packets is transmitted**
- **i represents packet index**
- **ignore is used as counter for the number of feedback particles to ignore due to incoming STOP messages**

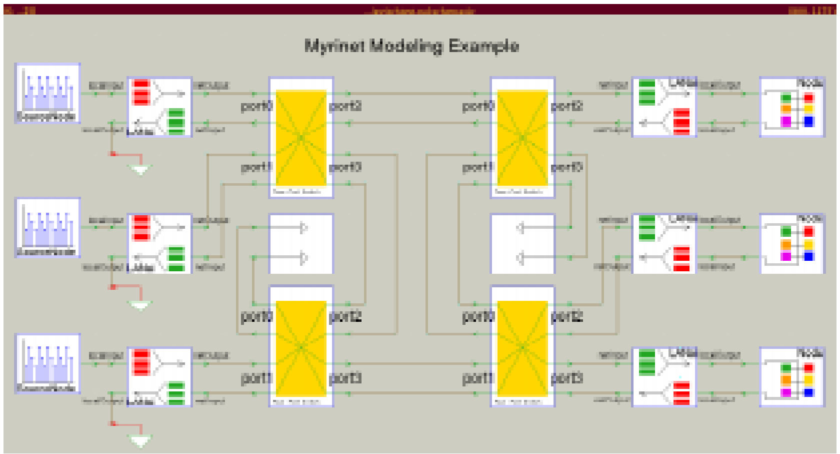
Switch Port State Diagram



State diagram of Myrinet Switch Port Behavior

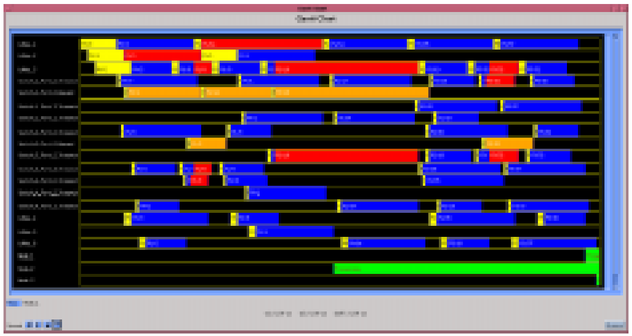
- **state diagram applies to each individual port within a Switch**
- **ignore is used as counter for the number of feedback particles to ignore due to incoming STOP messages**
- **queued is used as counter for the number of data packets queued**
- **DP N represents data packet received on port N (current packet)**
- **DP X represents data packet arriving on other than port N**

Simple Myrinet Modeling Example



Simple Myrinet Modeling Example

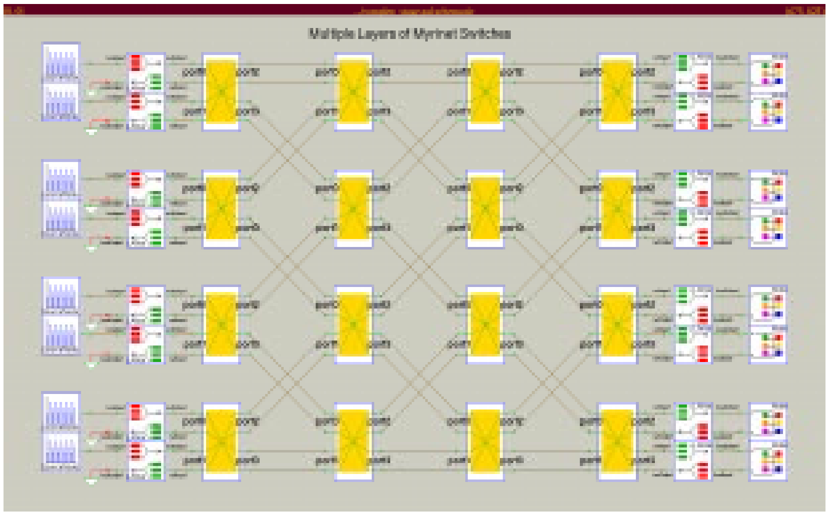
Modeling Results for Simple Example



Gantt Tool Display of Simple Myrinet Modeling Example

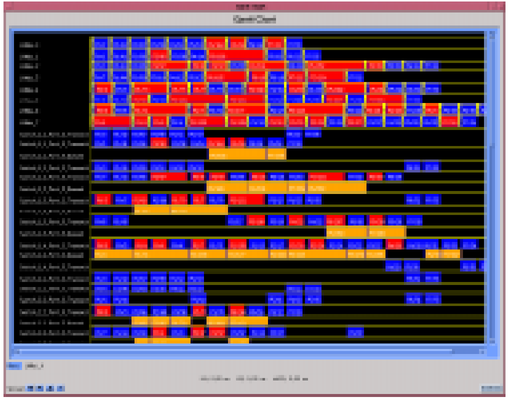
- **Yellow: start-up latency**
- **Blue: normal transmission/reception**
- **Green: processing of data on Node**
- **Orange: origin of contention, one or more packets queued in the switch**
- **Red: propogating effect of switch contention down current data path**

HPSC Example Architecture with Multiple Layers of Switches



HPSC Architecture with Multiple Layers of Switches

Performance Results for Multiple Switch Layer Example

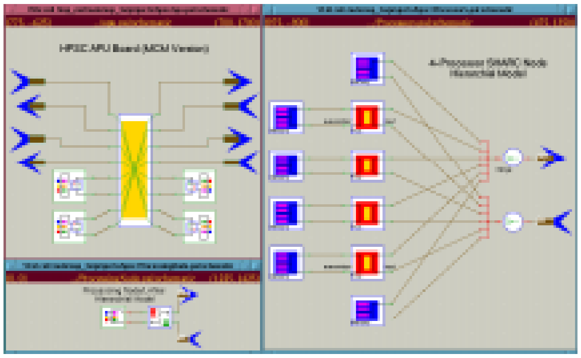


- Yellow: start-up latency
- Blue: normal transmission/reception
- Green: processing of data on Node
- Orange: origin of contention, one or more packets queued in the switch
- Red: propogating effect of switch contention down current data path

Performance Modeling Advantages

- Allows different hardware configurations to be examined without the expense or time of procuring or setting up hardware
- Rapid exploration of many hardware configurations
- Provides both macro and micro view at the behavior of the system
  - Where bottlenecks exist and why
  - Where underutilized capability exists
  - Overall system performance can be predicted (estimated)
- Performance modeling can provide information to hardware
  - Architecture and interconnects
  - DSTs can be reused
- Goal: to have performance models predict performance to within +/- 10% of actual

Role of Hierarchy in Performance Modeling


HPSC APU Board (MCM Version)
4-Processor SHARC Node Hierarchial Model

Examples of Hierarchical Performance Modeling within Ptolemy

- Groups of connected stars can be captured into a single galaxy using Ptolemy’s hierarchical capability
- Useful for capturing logical and/or physical boards or subsystems
- Useful for modeling at different levels of abstraction

Conclusions

- HPSC architecture (http://www.sanders.com/hpc/HPSCS/HPSCS.html)
- Myrinet protocol (http://www.myri.com)
- Ptolemy (http://ptolemy.eecs.berkeley.edu)
- Performance modeling extensions to Ptolemy’s DE domain
  - New stars and associated state models
  - New particles
- Examples of HPSC Performance modeling and Gantt Tool
- Advantages of Performance modeling
- Role of Hierarchy in Performance Modeling
- Short and long papers on this work available at
  - http://www.sanders.com/spard/publish.html