BigNetSim Tutorial

Presented by
Gengbin Zheng & Eric Bohm

Parallel Programming Laboratory

University of Illinois at Urbana-Champaign

Outline

- Overview
- BigSim Emulator
- Charm++ on the Emulator
- Simulation framework
  - Online mode simulation
  - Post-mortem simulation
  - Network simulation
- Performance analysis/visualization

Postmortem Simulation

- Run application once, get trace logs, and run simulation with logs for a variety of network configurations
- Implemented on POSE simulation framework

How to Obtain Predicted Time

- Use BgPrint(char *) in similar way
  - Each BgPrint() called at execution time in online execution mode is stored in BgLog as a printing event
- In postmortem simulation, strings associated with BgPrint event is printed when the event is committed
- “%f” in the string will be replaced by committed time.

Compile Postmortem Simulator

- Compile Bigsim simulator
- Compile pose
  - Use normal charm++
  - *cd charm/net-linux/tmp*
  - *make pose*
- Obtain simulator
  - svn co https://charm.cs.uiuc.edu/svn/repos/BigNetSim
- Compile BigNetSim simulator
  - fix BigNetSim/trunk/Makefile.common
  - *cd BigNetSim/trunk/BlueGene*
  - *make*

Example (AMPI CJacobi3D cont.)

- **BigNetSim/trunk/tmp/bigsimulator 0 0**

```
bgtrace: totalBGProcs=4 X=2 Y=2 Z=1 #Cth=1 #Wth=1 #Pes=3
Opts: netsim on: 0
Initializing POSE...
POSE initialization complete.
Using Inactivity Detection for termination.
Starting simulation...
256 4 1024 1.750000 9 1000000 0 1 0 0 0 8 16 4
Info> timing factor 1.000000e+08 ...
Info> invoking startup task from proc 0 ...
[0:AMPI_Barrier_END] interation starts at 0.000217
[0:RECV_RESUME] interation starts at 0.000755
[0:RECV_RESUME] interation starts at 0.001292
[0:RECV_RESUME] interation starts at 0.001829
[0:RECV_RESUME] interation starts at 0.002367
[0:RECV_RESUME] interation starts at 0.002904
[0:RECV_RESUME] interation starts at 0.003441
[0:RECV_RESUME] interation starts at 0.003978
[0:RECV_RESUME] interation starts at 0.004516
[0:RECV_RESUME] interation starts at 0.005053
Simulation inactive at time: 587350
Final GVT = 587351
```

Outline

- Overview
- BigSim Emulator
- Charm++ on the Emulator
- Simulation framework
  - Online mode simulation
  - Post-mortem simulation
  - Network simulation
- Performance analysis/visualization

Big Network Simulator

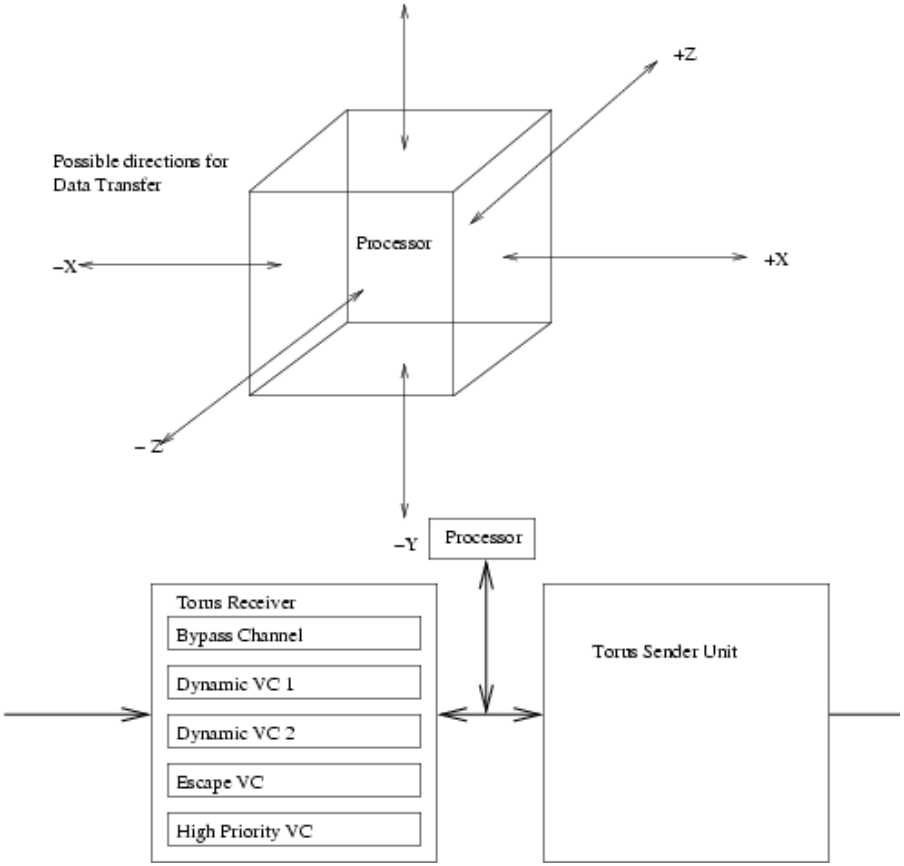
- When message passing performance is critical and strongly affected by network contention

BigNetSim Overview

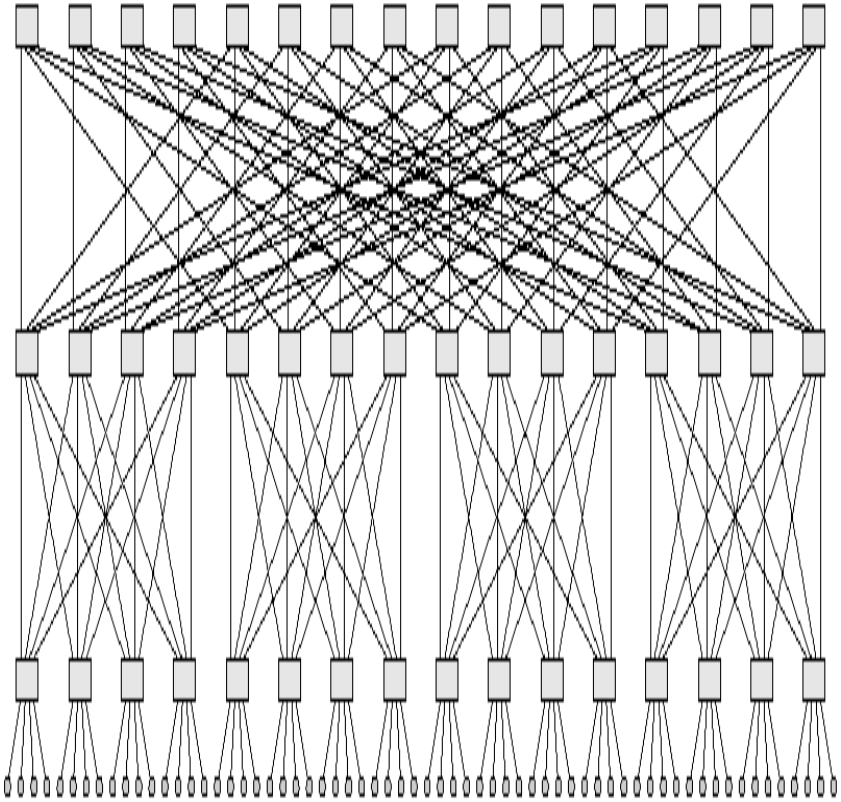
- Networks
- Design
- POSE
- Catalog of Network Simulations
- Building
- Running
- Configuration
- Modular NetSim
  - Mix and match architecture, topology, routing
- Using the Generator
- Extensibility

Networks

Direct Network



Indirect Network

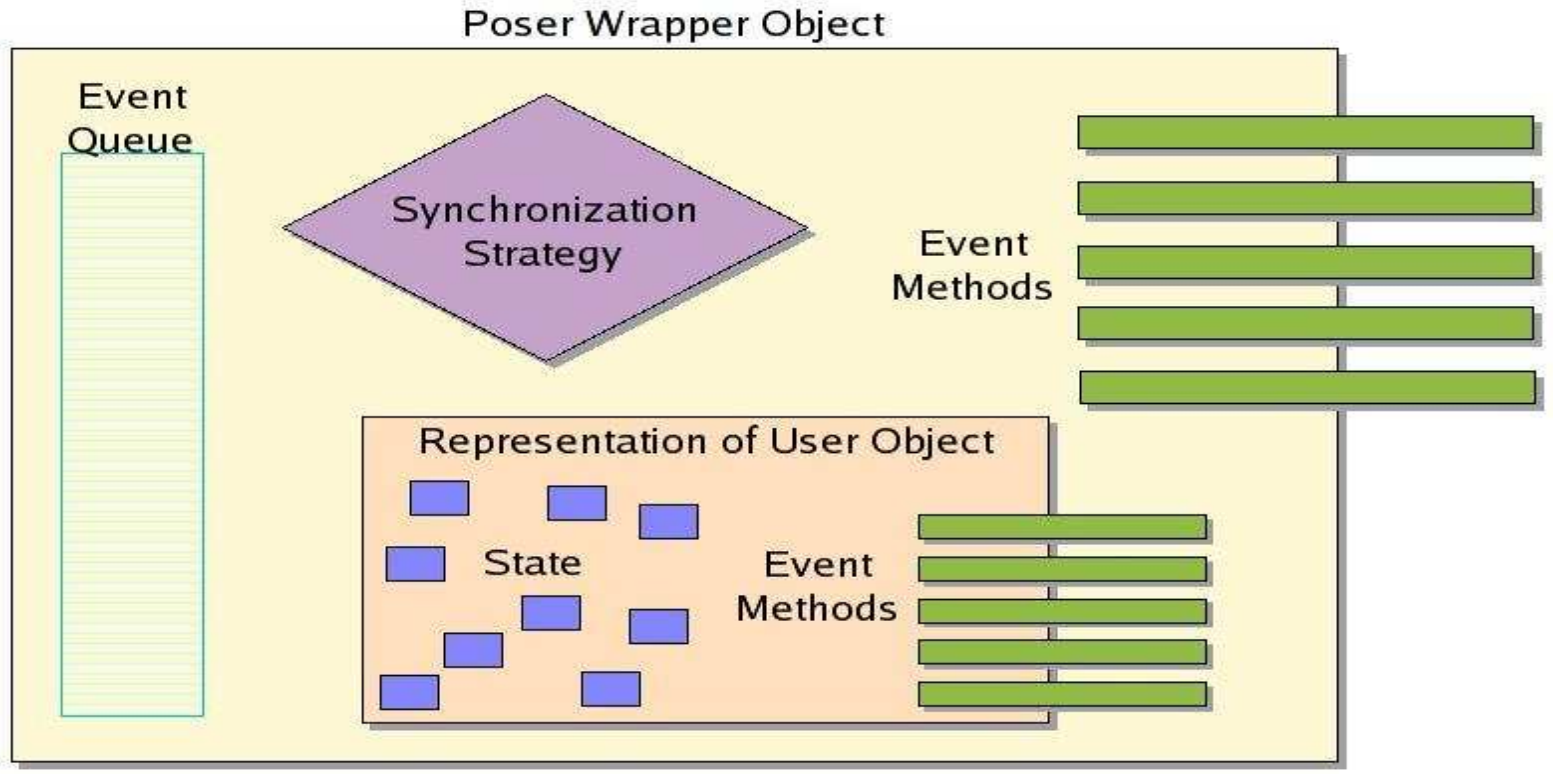


Implementation

- Post-Mortem Network simulators are Parallel Discrete Event Simulations
  - Parallel Object Simulation Environment (POSE)
  - Network layer constructs (NIC, Switch, Node, etc) implemented as poser simulation objects
  - Network data constructs (message, packet, etc) implemented as event methods on simulation objects

POSE

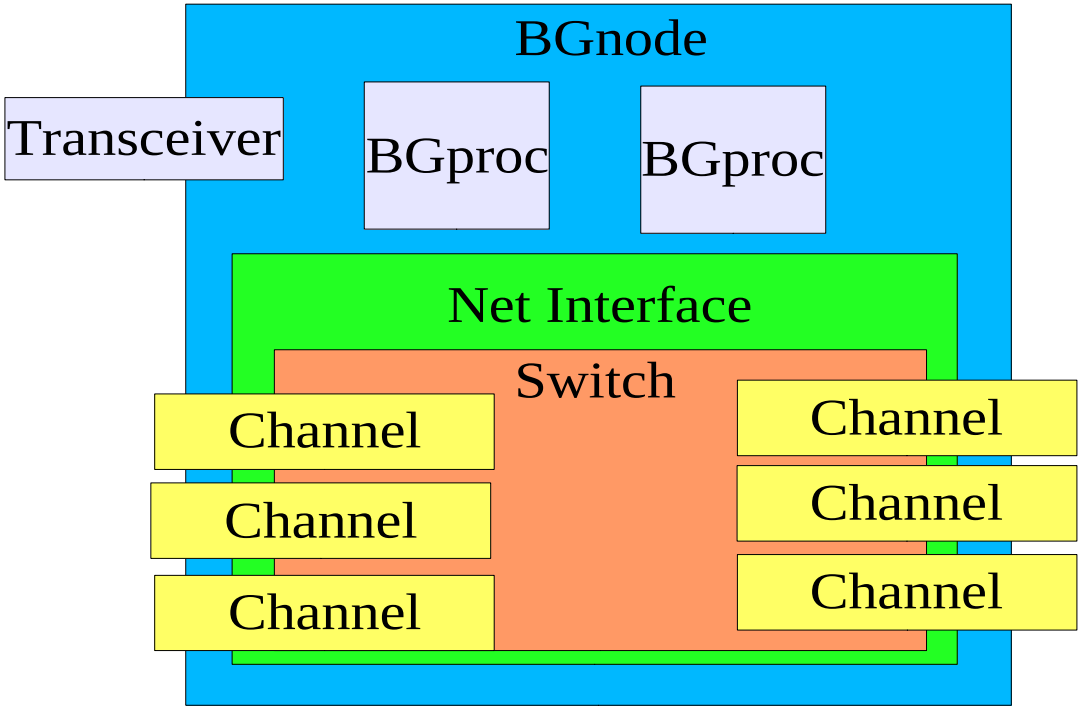
- Each poser is a tiny simulation



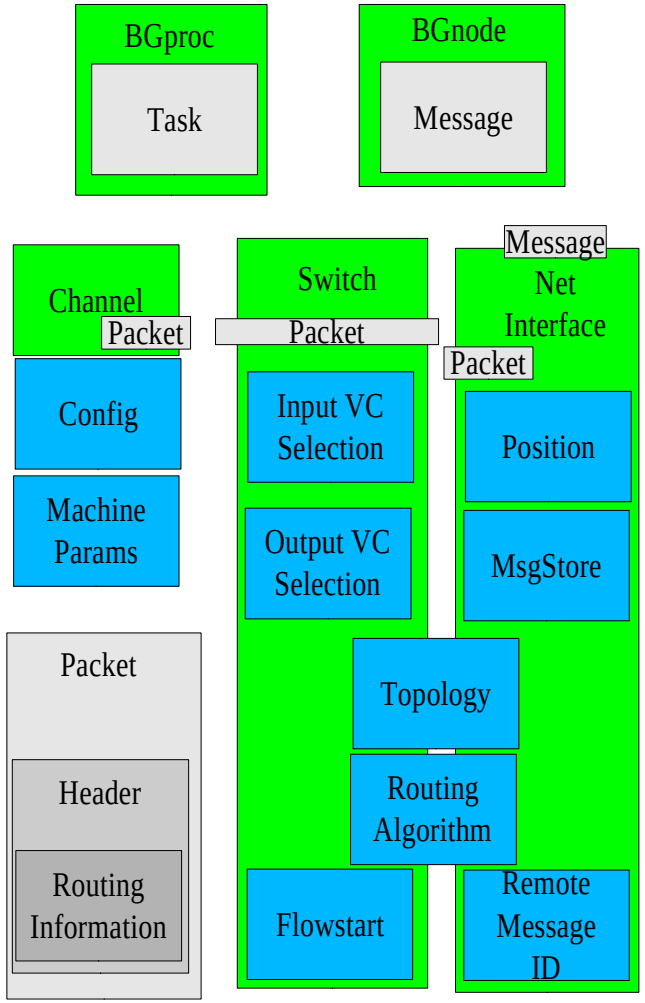
Interconnection Networks

- Flexible Interconnection Network modeling:
  - Choose from a variety of
    - Topologies
    - Routing Algorithms
    - Input Virtual Channel Selection strategies
    - Output Virtual Channel Selection strategies

BigNetSim Design



BigNetSim API: Extensibility



Topology

- Topologies available
  - HyperCube;
  - Mesh; generalized k-ary-n-mesh; n-mesh;
  - Torus; generalized k-ary-n-cube;
  - FatTree; generalized k-ary-n-tree;
  - Low Diameter Regular graphs(LDR)
  - Hybrid topologies
    - HyperCube-Fattree;
    - HyperCube-LDR;

Network Modeling

- Routing models
  - Virtual cut-through routing
- Contention Modeling
  - Port contention at a Switch
  - Load contention: available buffer at next layer of switches
- Adaptive and static Routing algorithms
  - Minimal deadlock-free
  - Non-minimal
  - Fault-tolerant

Routing Algorithms

- K-ary-N-mesh / N-mesh
  - Direction Ordered;
  - Planar Routing;
  - Static Direction Reversal Routing
  - Optimally Fully Adaptive Routing (modified too)
- K-ary-N-tree
  - UpDown (modified, non-minimal)
- HyperCube
  - Hamming
  - P-Cube (modified too)

Input/Output VC selection

- Input Virtual Channel Selection
  - Round Robin;
  - Shortest Length Queue
  - Output Buffer length
- Output Virtual Channel Selection
  - Max. available buffer length
  - Max. available buffer bubble VC
  - Output Buffer length

Building POSE

- POSE
  - cd charm
  - ./build pose net-linux
  - options are set in pose_config.h
    - stats enabled by POSE_STATS_ON=1
    - user event tracing TRACE_DETAIL=1
    - more advanced configuration options
      - speculation
      - checkpoints
      - load balancing

Building BigNetSim

- svn co https://charm.cs.uiuc.edu/svn/repos/BigNetSim
- Build BigNetSim/Bluegene
  - cd BigNetSim/trunk/Bluegene
  - make
  - for sequential simulator
    - make clean; make SEQUENTIAL=1
  - cd ../tmp

Running

- charmrun +p4 bigsimulator 1 1
- Parameters
  - First parameter controls detailed network simulation
    - 1 will use the detailed model
    - 0 will use simple latency
  - Second parameter controls simulation skip
    - 1 will skip forward to the time stamp set during trace creation
    - 0 if not set or network startup interesting

Configuring BigNetSim

USE_TRANSCEIVER 0 For network analysis ignore trace and generate random traffic
NUM_NODES 0 Number of nodes, taken from trace file or set for transceiver
MAX_PACKET_SIZE 256 Maximum packet size
SWITCH_VC 4 The number of switch virtual channels
SWITCH_PORT 8 Number of ports in switch, calculated automatically for direct networks
SWITCH_BUF 1024 Size in memory of each virtual channel
CHANNELBW 1.75 Bandwidth in 100 MB/s
CHANNELDELAY 9 Delay in 10 ns . So 9 => 90ns
RECEPTION_SERIAL 0 Used for direct networks where reception FIFO access has to be serialized
INPUT_SPEEDUP 8 Used to limit simultaneous access by VC in a port. Should be less than or equal to number of VC. Currently used only for bluegene.
ADAPTIVE_ROUTING 1 Additional flag to use adaptive/deterministic routing
COLLECTION_INTERVAL 1000000 Collection * 10ns gives statistics bin size
DISPLAY_LINK_STATS 1 Display statistics for each link
DISPLAY_MESSAGE_DELAY 1 Display message delay statistics

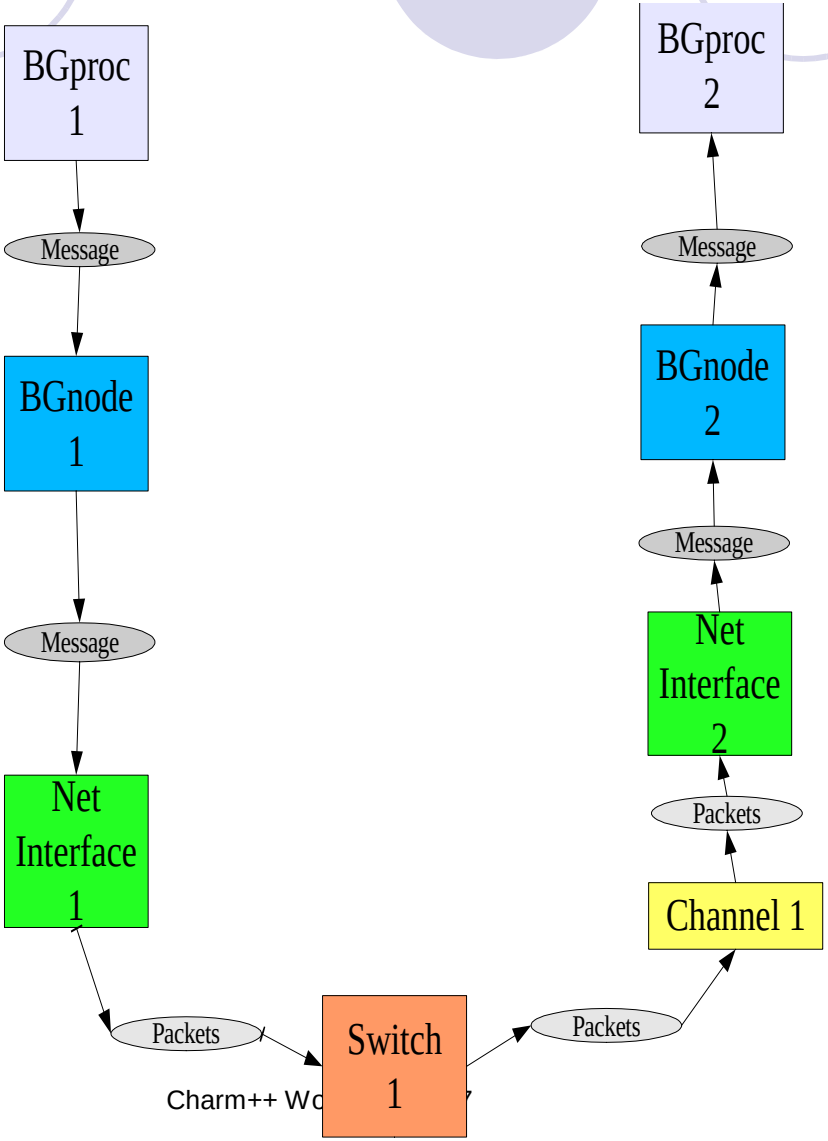
Output

- Completion time for trace run
- Per Link utilization, link contention high water marks
- If trace projections logs for the trace exist, an updated “corrected” copy is created.
- Turn on -tproj to get simple trace of network performance if projections traces from the emulator are not available
- Use -projname YOURAPPNAME to direct bignetsim to your existing tracelogs for updating.

Artificial Network Loads

- Generate traffic patterns instead of using trace files
  - additional command line parameters
    - Pattern
    - Frequency
- Pattern
  - 1 kshift
  - 2 ring
  - 3 bittranspose
  - 4 bitreversal
  - 5 bitcomplement
  - 6 poisson
- Frequency
  - 0 linear
  - 1 uniform
  - 2 exponential

BigNetSim: Data Flow



Adding a Network

- mkdir new subdir in trunk
- copy boilerplate InitNetwork.h
- copy boilerplate Makefile
  - change MACHINE make variable to your dirname
- new InitNetwork.C
  - Define switch, channel, nic mappings
  - Define how switches route and select virtual channels
  - Define topology and default routing

Adding a Topology

- New *.h *.C in trunk/Topology
  - constructor()
  - getNeighbours()
  - getNext()
  - getNextChannel()
  - getStartPort()
  - getStartVC()
  - getStartSwitch()
  - getStartNode()
  - getEndNode()

Adding a Routing Strategy

- New *.h *.C files in trunk/Routing
  - constructor()
  - selectRoute()
  - populateRoute()
  - loadTable()
  - getNextSwitch()
  - sourceToSwitchRoutes()

Adding a VC Selector

- Either Input or Output VC Selector
  - new *.h *C in [Input/Output]VCSelector
  - constructor()
  - select[Input/Output]VC()

Future

- Improved scalability
  - adaptive strategies
  - improved hardware collectives
  - out-of-core loading of tracefiles
  - load balancing
  - network fault simulation
- Ports to BG/L, Cray XT3, etc.
- Representative collection of netconfig files

Case Study - NAMD

- Molecular Dynamics Simulation Applications
- Compile BigSim Charm++:
  - *./build bigsim net-linux bigsim*
- Compile NAMD:
  - Get source code from:
    - *http://charm.cs.uiuc.edu/~gzheng/namd-bg.tar.gz*
  - *./config fftw Linux-i686-g++*

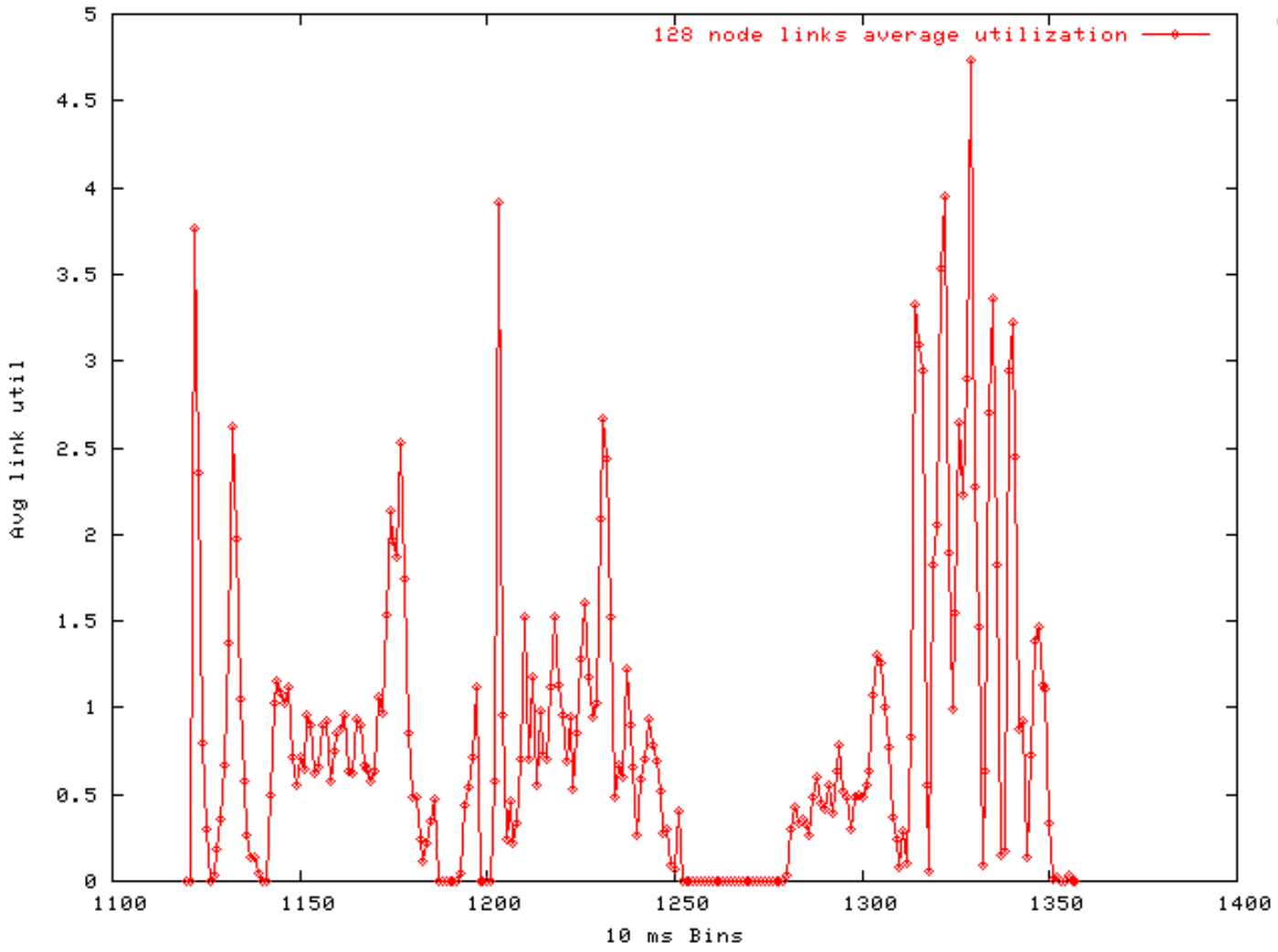
Validation with Simple Network Model

NAMD Apo-Lipoprotein A1 with 92K atom.

Performance simulation using 8 Lemieux processors

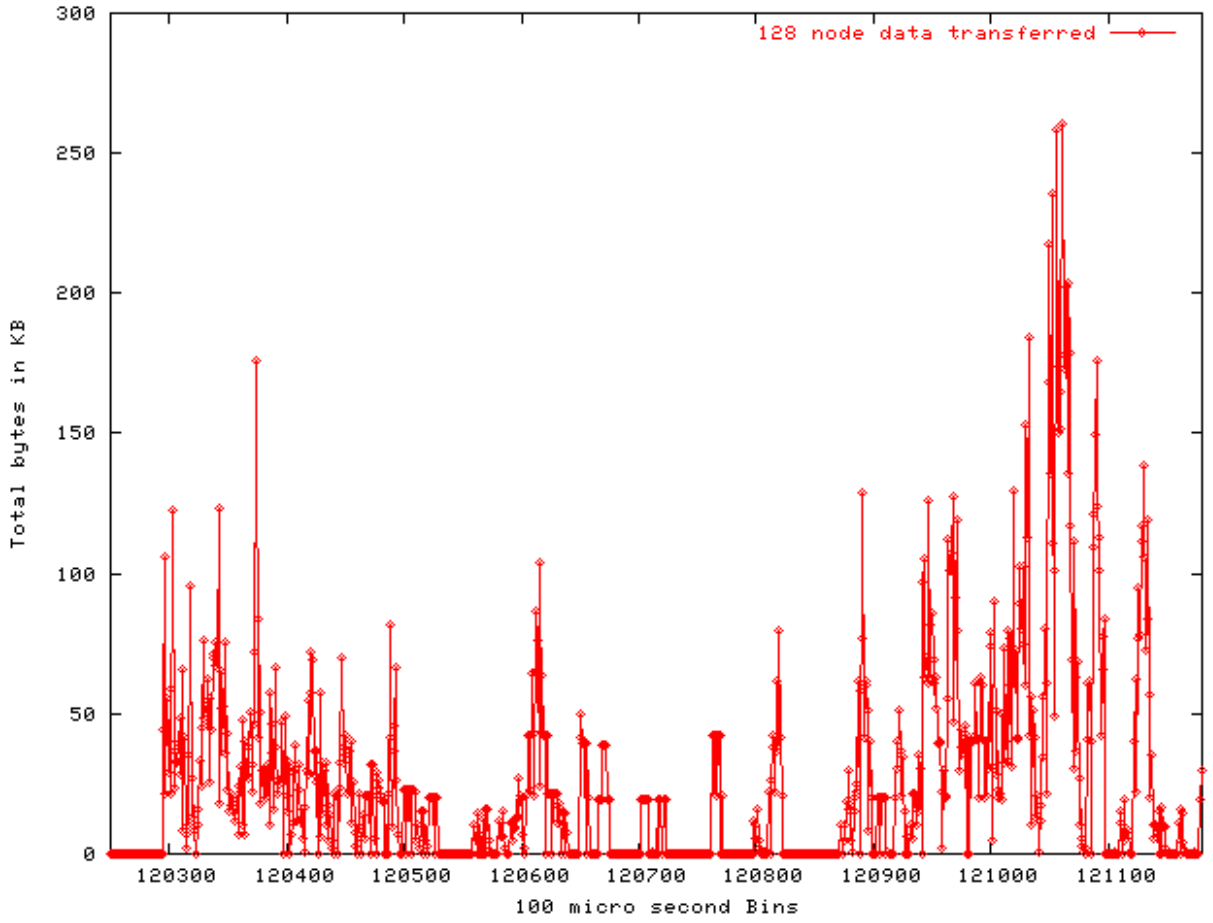
Processors	128	256	512	1024
Actual time (ms)	71.5	40.3	23.9	17.6
Predicted time (ms)	75.8	43.6	25.1	20.8

Network Communication Pattern Analysis



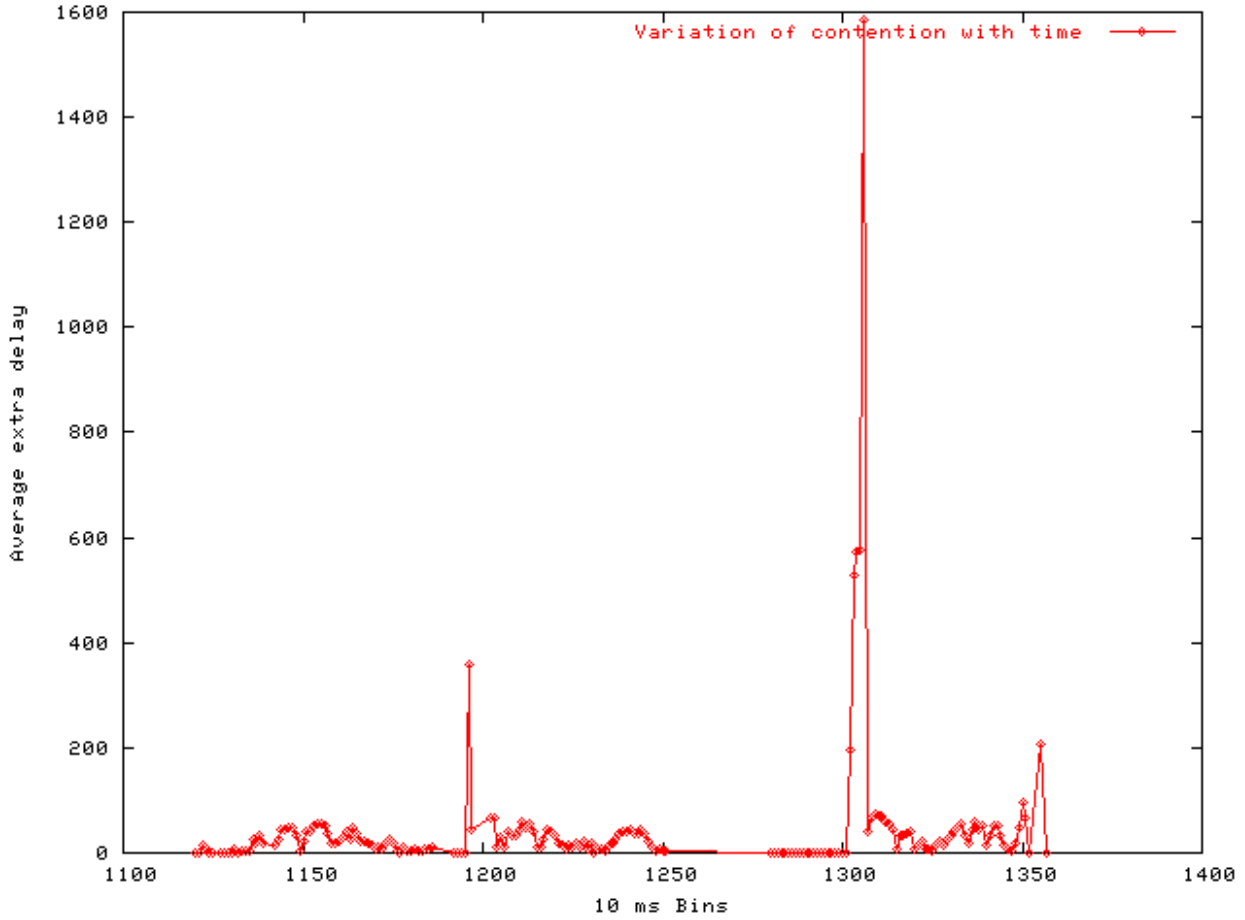
- NAMD with apoa1
- 15 timestep

Network Communication Pattern Analysis



Data transferred (KB) in a single time step

Contention Encountered by Messages



Outline

- Overview
- BigSim Emulator
- Charm++ on the Emulator
- Simulation framework
  - Online mode simulation
  - Post-mortem simulation
  - Network simulation
- Performance analysis/visualization

Performance Analysis/Visualization

- trace-projections is available for BigSim and BigNetSim
- One challenge:
  - Number of log files can be overwhelming

Generate Projections Logs

- Link application with
  - –tracemode projections
- Select subset of processors in bgconfig:

  projections 0-100,2000,3100-3200
- With timestamp correction, two sets of projections logs are generated
  - Before and after timestamp correction

Generate Projections Logs (the hideous secret)

- Problem:
  - Projections tracing function maintains a fix sized buffer for storing projections logs
  - Buffer is flushed to disk when it is filled up, disk I/O can effect predicted time
- Solution:
  - Use +logsize runtime option to provide large projections buffer size
- In fact, in online mode simulation, simulation aborts when disk I/O occurs.

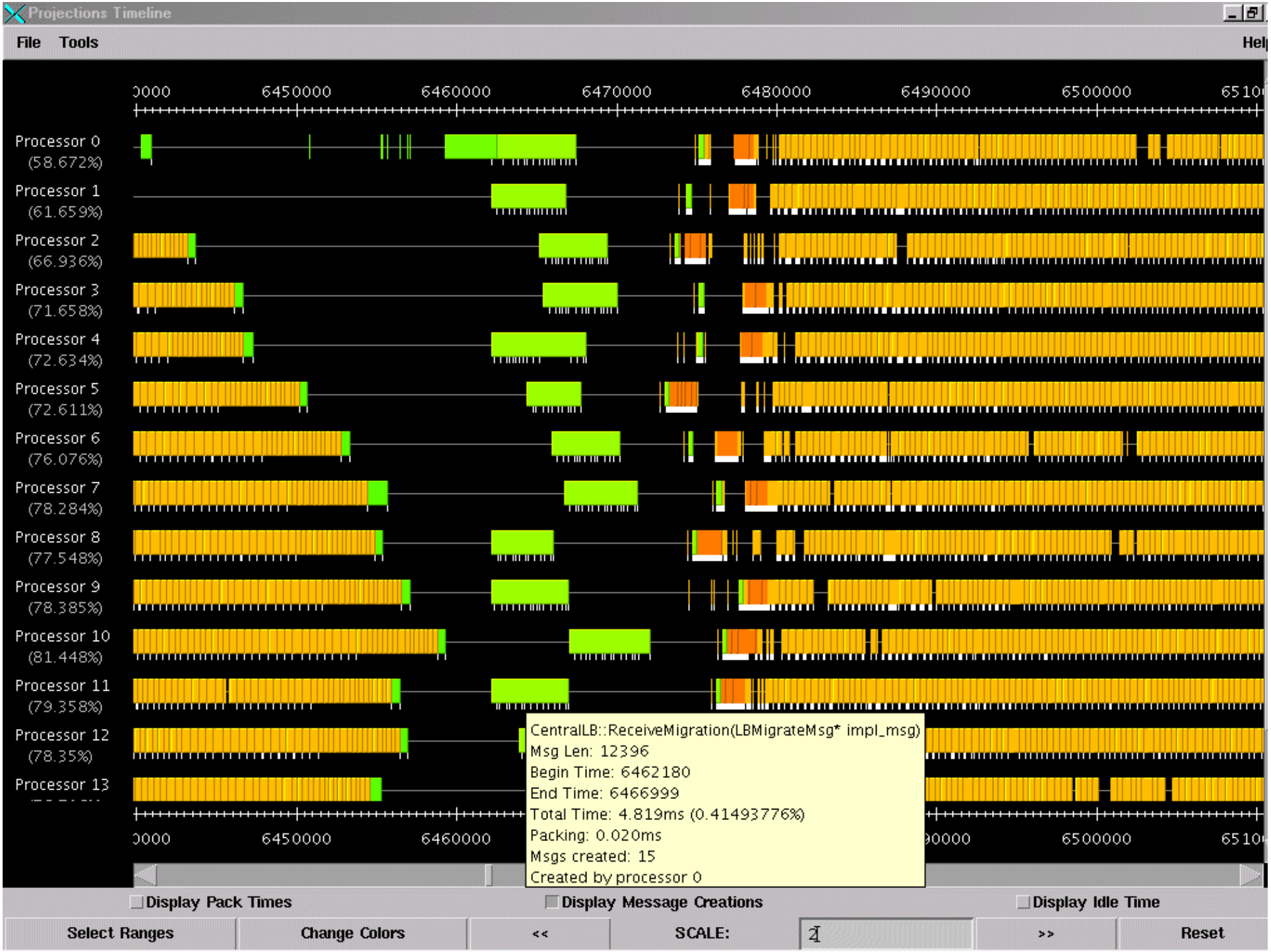
Projections with Jacobi

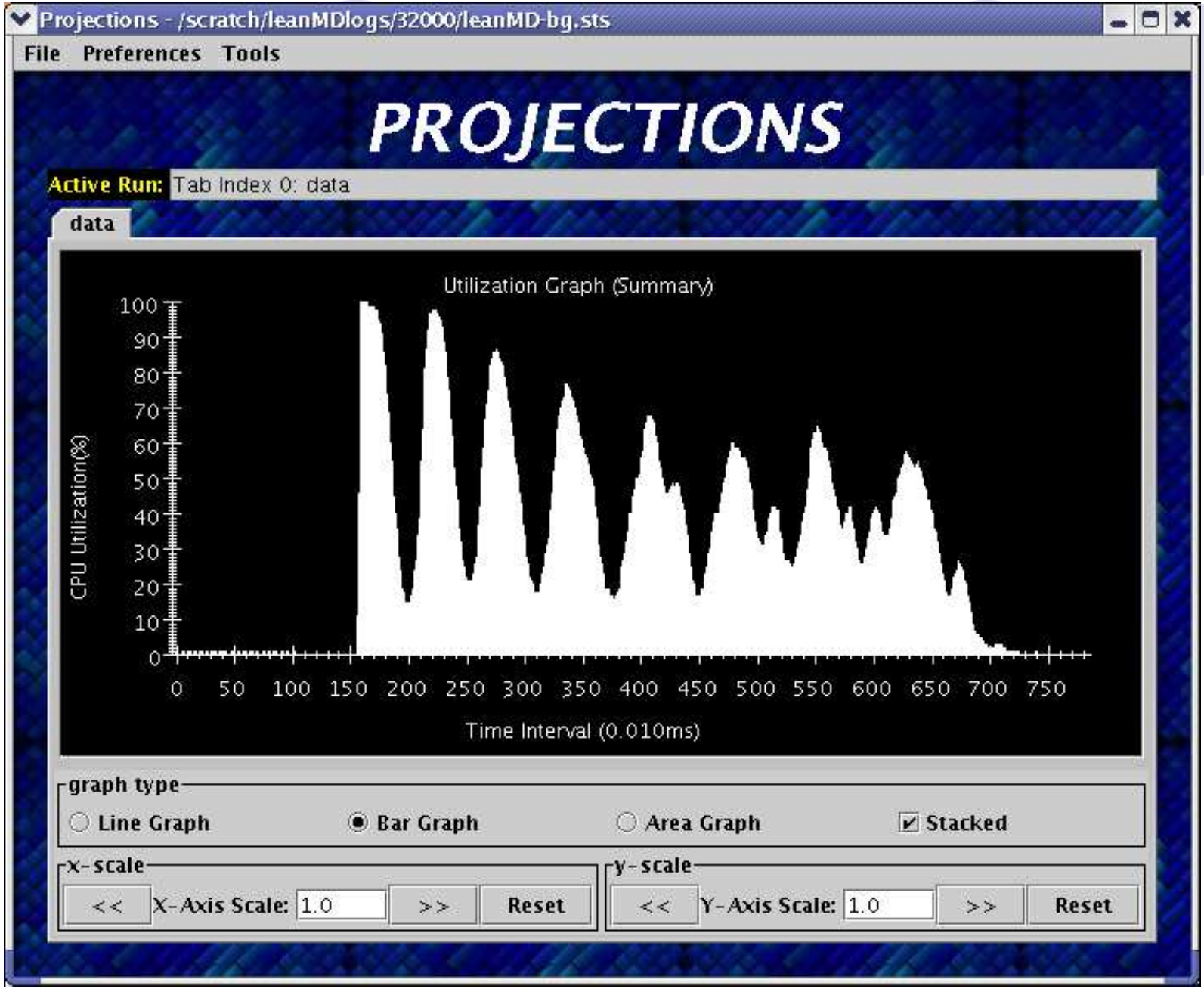
- cd charm/examples/bigsim/sdag/jacobi-no-redn
- ./charmrun +p4 ./jacobi 16384 10 8192 +bgconfig ./bg_config
- Config file:

```
x 32
y 16
z 16
cth 1
wth 1
stacksize  10000
#timing  walltime
timing  bgelapse
#timing  counter
cpufactor   1.0
fpfactor    5e-7
traceroot     .
log      yes
correct  yes
network   lemieux
projections 0,1000,8189-8191
```


Projections Timeline <3>
File Tools Colors
Help
0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950
PE 0
(85.3%)
PE 1000
(84.5%)
PE 8189
(83.7%)
PE 8190
(83.7%)
PE 8191
(83.7%)
Display Pack Times
Display Message Sends
Display Idle Time
View User Events (0)
Select Ranges
Change Colors
<<
SCALE:
1.0
>>
Reset
Highlight Time
Slection Begin Time
Selection End Time
Selection Length
Zoom Selected
Load Selected
Jump to graph:
Usage Profile
Graph
Histogram
Communication
Overview

Make bgtest
With 16 processors


Projections Timeline
File
Tools
Help
0000
6450000
6460000
6470000
6480000
6490000
6500000
6510
Processor 0
(58.672%)
Processor 1
(61.659%)
Processor 2
(66.936%)
Processor 3
(71.658%)
Processor 4
(72.634%)
Processor 5
(72.611%)
Processor 6
(76.076%)
Processor 7
(78.284%)
Processor 8
(77.548%)
Processor 9
(78.385%)
Processor 10
(81.448%)
Processor 11
(79.358%)
Processor 12
(78.35%)
Processor 13
CentralLB::ReceiveMigration(LBMigrateMsg* impl_msg)
Msg Len: 12396
Begin Time: 6462180
End Time: 6466999
Total Time: 4.819ms (0.41493776%)
Packing: 0.020ms
Msgs created: 15
Created by processor 0
Display Pack Times
Display Message Creations
Display Idle Time
Select Ranges
Change Colors
<<
SCALE:
1
>>
Reset


Projections - /scratch/leanMDlogs/32000/leanMD-bg.sts
File
Preferences
Tools
PROJECTIONS
Active Run:
Tab Index 0: data
data
Utilization Graph (Summary)
CPU Utilization(%)
100
90
80
70
60
50
40
30
20
10
0
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
Time Interval (0.010ms)
graph type
Line Graph
Bar Graph
Area Graph
Stacked
x-scale
<<
X-Axis Scale:
1.0
>>
Reset
y-scale
<<
Y-Axis Scale:
1.0
>>
Reset

Thank You!

Free download of Charm++ and BigSim at
http://charm.cs.uiuc.edu
Send comments to ppl@charm.cs.uiuc.edu