
Scaling analysis of a neocortex inspired cognitive model on the Cray XD1

Kenneth L. Rice

Tarek M. Taha

Christopher N. Vutsinas

Clemson University

December 13, 2008

Motivation

- Examined a Neocortex Inspired cognitive model
 - Geared towards pattern recognition
 - Large scale would be useful
 - Hardware acceleration needed for large scale

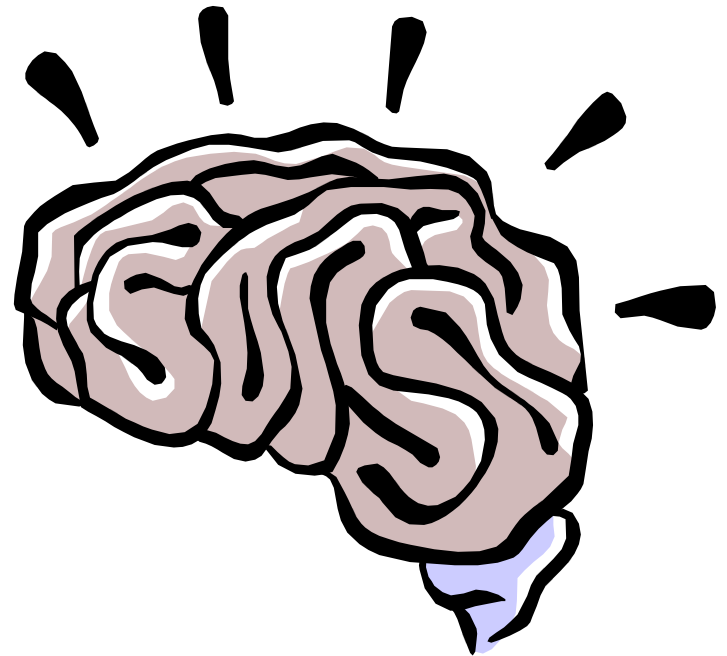
Platform

- Implemented model on a Cray XD1
- XD1 contained:
 - 144 Xilinx Virtex II Pro XCVP50 FPGAs
 - 864 2.0 GHz AMD Opteron cores
- Two implementations for the model:
 - Software (C)
 - Hardware-Accelerated (FPGA/C)



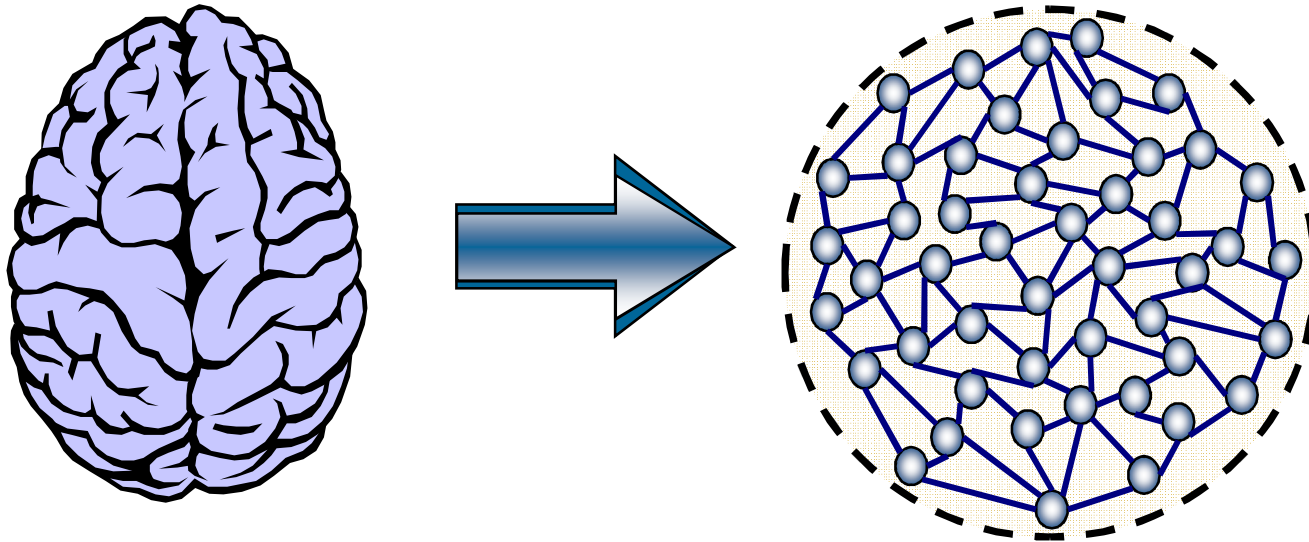
What is the Neocortex?

- Outer layer of the brain
- Site of cognitive processing
- Structure:
 - Large, Flat
 - Uniform processing
 - Hierarchical organization



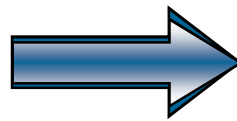
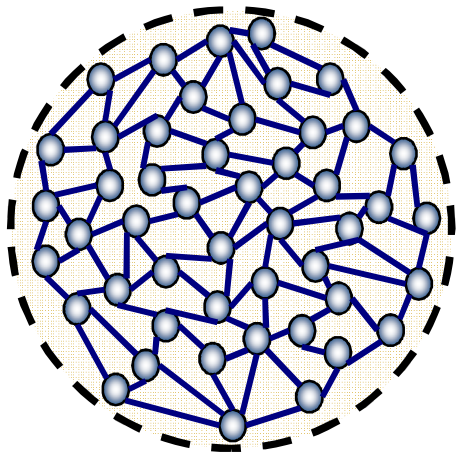
Cortical Models

- Several new models (< 5 yrs)
- Based on new observations
- Useful for real world applications



Hardware Implementation

- Models have high parallelism:
 - Large collection of nodes in parallel
 - Uniform computations
- Large set of simple processing elements (PEs)
 - FPGAs implementations are attractive

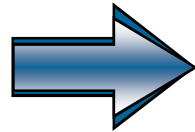


Motivation for Scaling Study

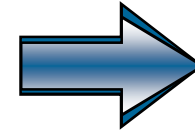
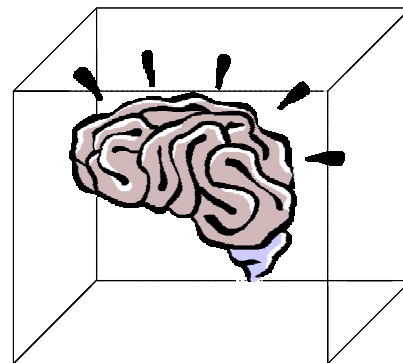
- Biological scale implementations of models are useful.
- Want to evaluate performance of a large scale implementation on a Cray XD1.

Real World Applications

Speech Recognition
Robotic Control
Data Mining
Pattern Recognition
Computer Vision



Specific Neocortex Model



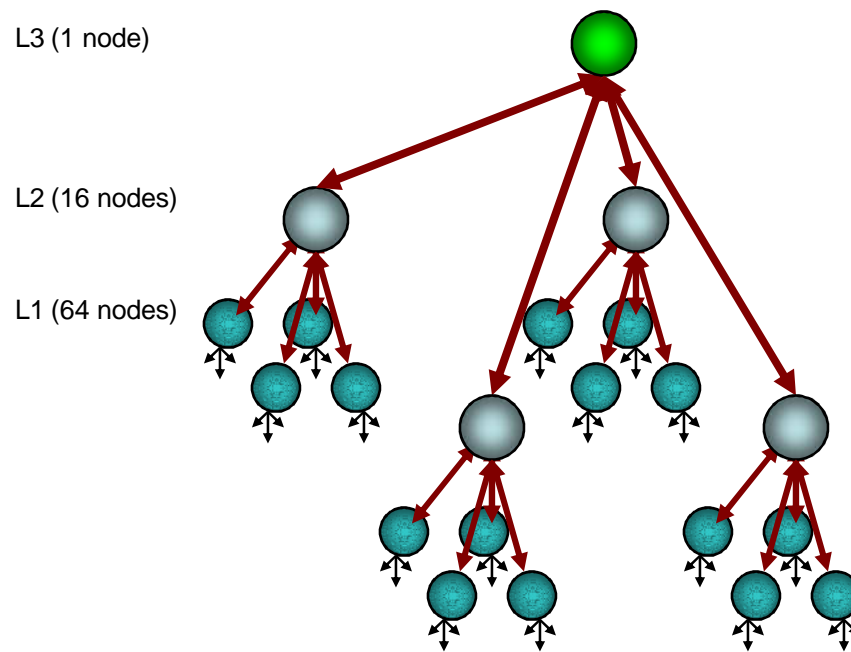
Cray XD1



Model Background

George and Hawkins Model

- Models invariant pattern recognition based on Hawkins' models of neo-cortex
- Recognizes images under various transformations
- This model is currently being developed by Numenta, Inc for use in cognitive applications



Belief propagation

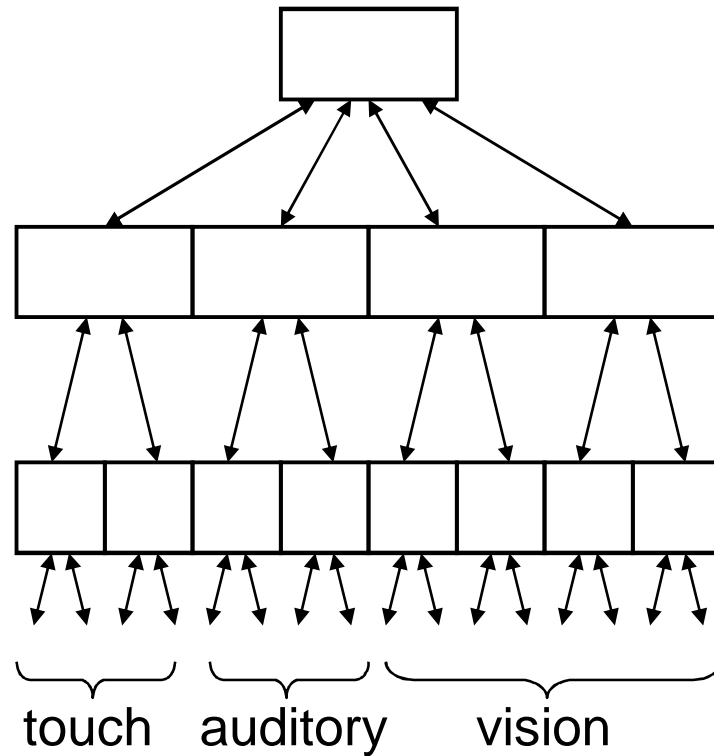


Figure adapted from On Intelligence

Belief propagation

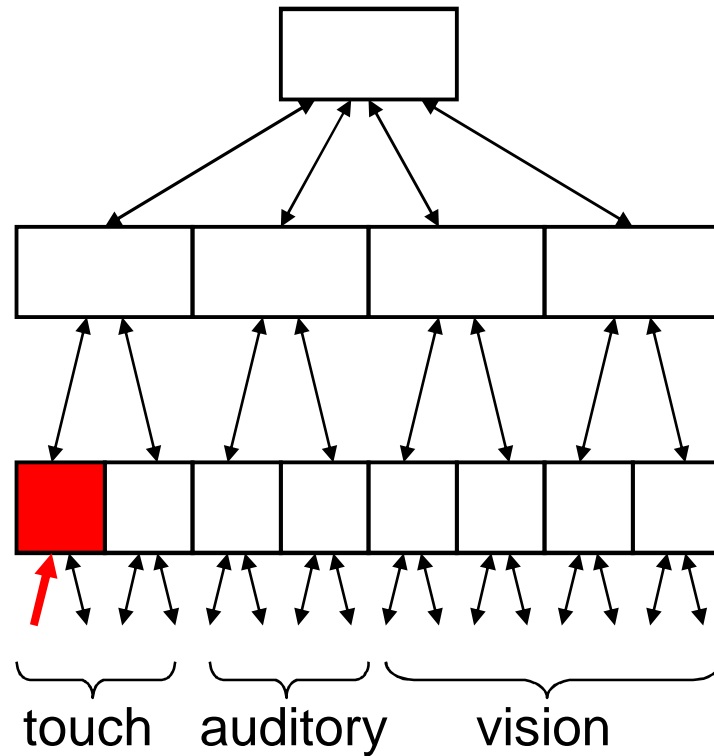


Figure adapted from On Intelligence

Belief propagation

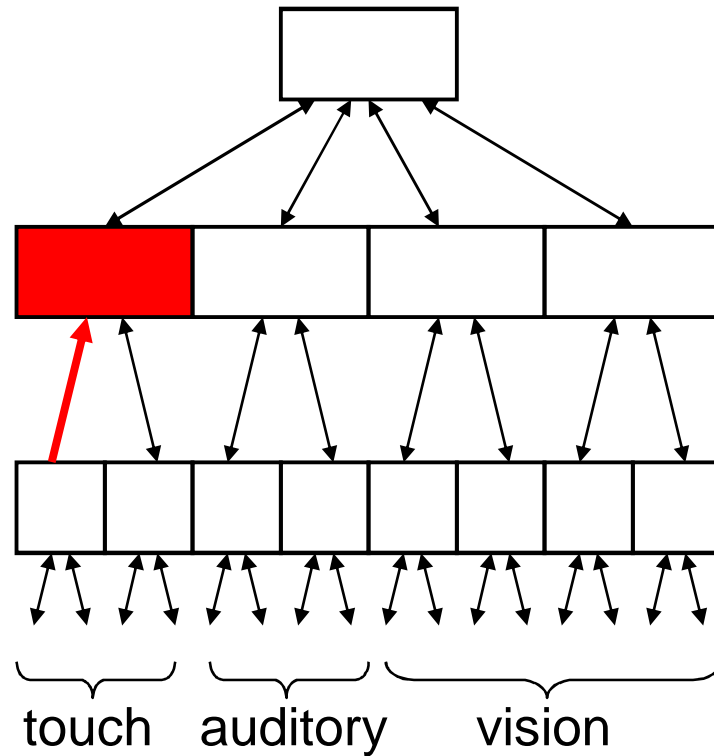


Figure adapted from On Intelligence

Belief propagation

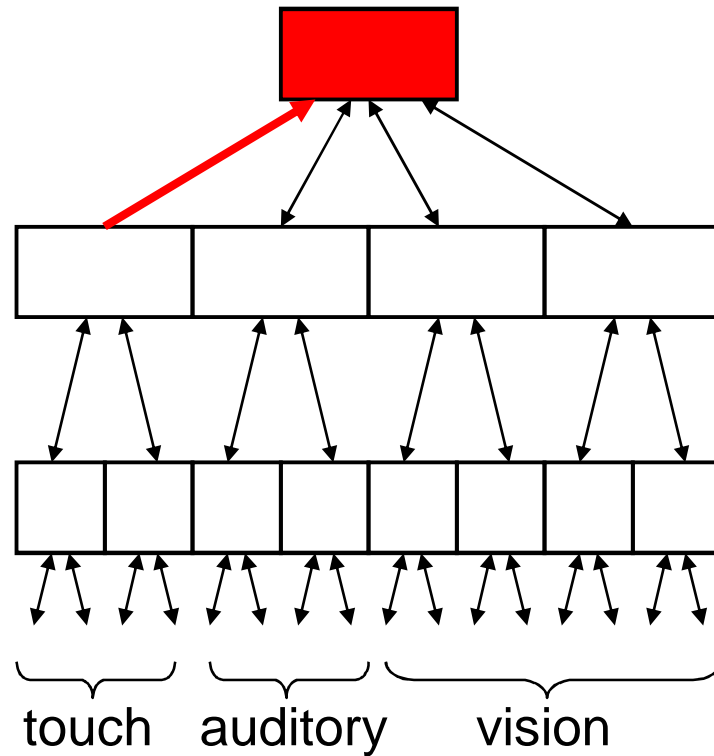


Figure adapted from On Intelligence

Belief propagation

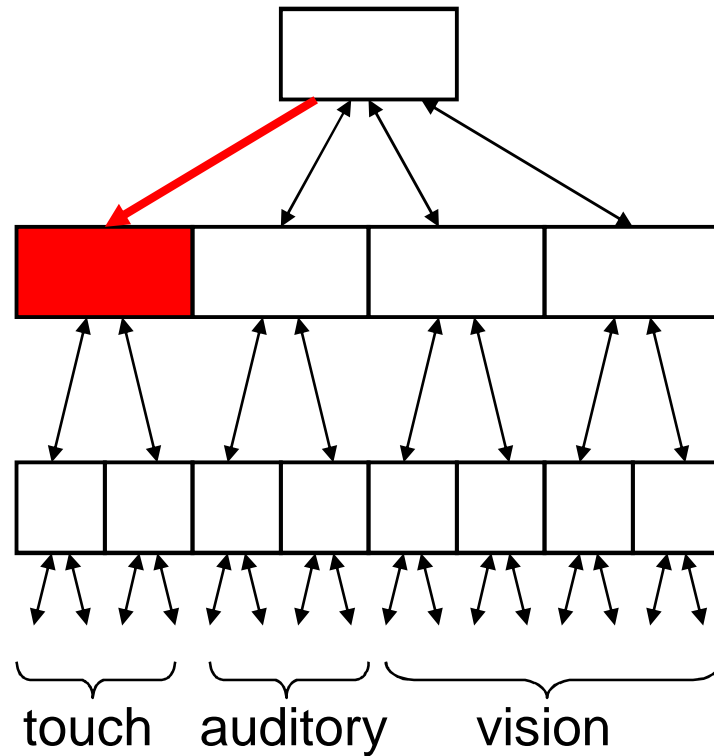


Figure adapted from On Intelligence

Belief propagation

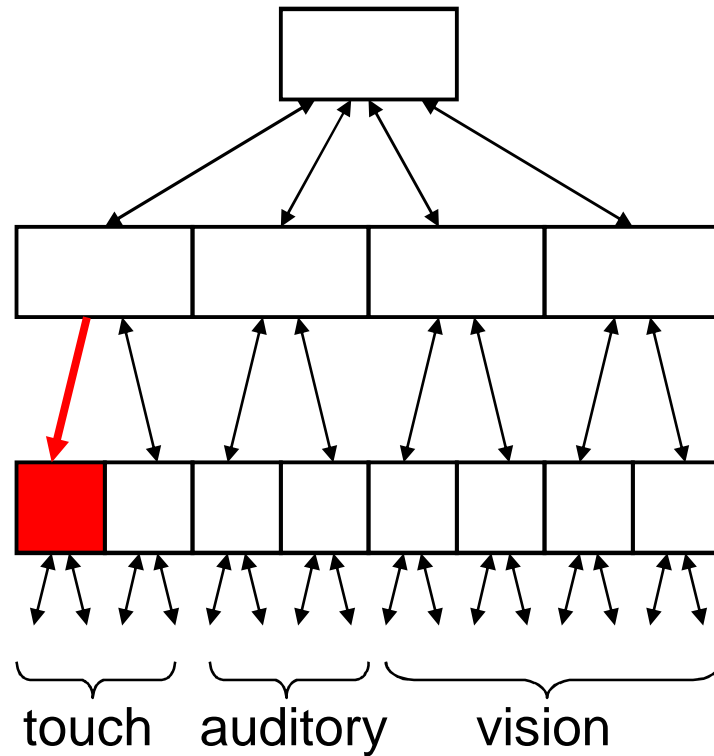
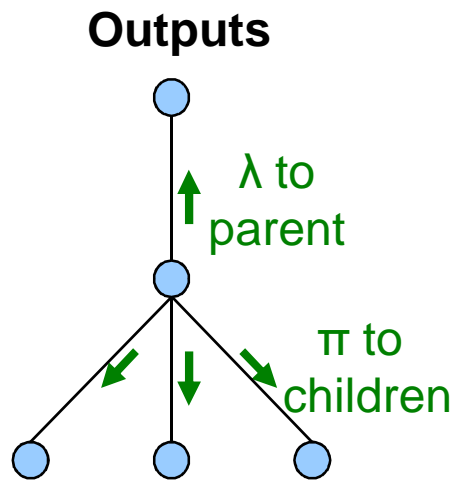
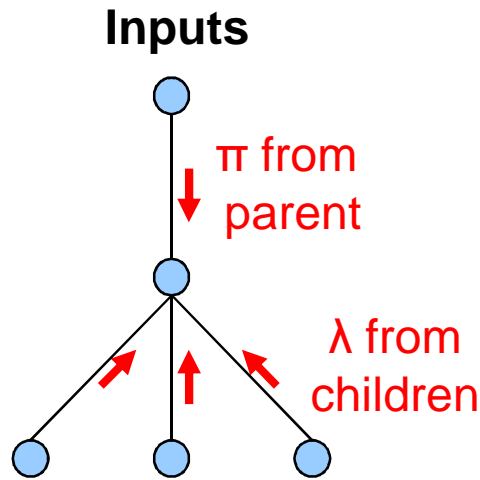


Figure adapted from On Intelligence

Node Computations



Bayesian Belief Propagation:

$$\lambda_{product}[i] = \prod_{child} \lambda_{in}[child][i] \quad (1)$$

$$F_{xu}[j][k] = \pi_{in}[j] \times P_{xu}[j][k] \times \lambda_{product}[k] \quad (2)$$

$$m_{row}[j] = \max(m_{row}[j], F_{xu}[j][k]) \quad (3)$$

$$m_{col}[k] = \max(m_{col}[k], F_{xu}[j][k]) \quad (4)$$

$$\lambda_{out}[j] = m_{row}[j] / \pi_{in}[j] \quad (5)$$

$$\pi_{out}[child][k] = m_{col}[k] / \lambda_{in}[child][k] \quad (6)$$

All matrix operations are element-by-element:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \times \begin{bmatrix} p & q \\ r & s \end{bmatrix} = \begin{bmatrix} ap & bq \\ cr & ds \end{bmatrix}$$

Data Optimizations

- Logarithmic conversion:
 - Change multiplies and divides to additions and subtractions
 - Reduce range of possible values
- Fixed point representation
 - Simpler hardware
- Data compression:
 - Compressed strings of zeros
 - Highly compressible (>90%)

$$\lambda_{product}[i] = \prod_{child} \lambda_{in}[child][i] \quad (1)$$

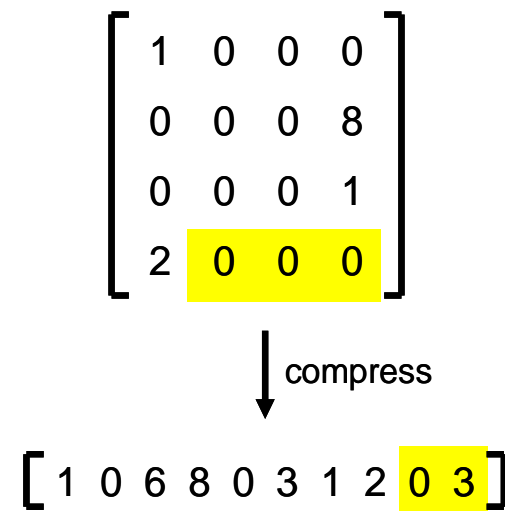
$$F_{xu}[j][k] = \pi_{in}[j] \times P_{xu}[j][k] \times \lambda_{product}[k] \quad (2)$$

$$m_{row}[j] = \max(m_{row}[j], F_{xu}[j][k]) \quad (3)$$

$$m_{col}[k] = \max(m_{col}[k], F_{xu}[j][k]) \quad (4)$$

$$\lambda_{out}[j] = m_{row}[j] / \pi_{in}[j] \quad (5)$$

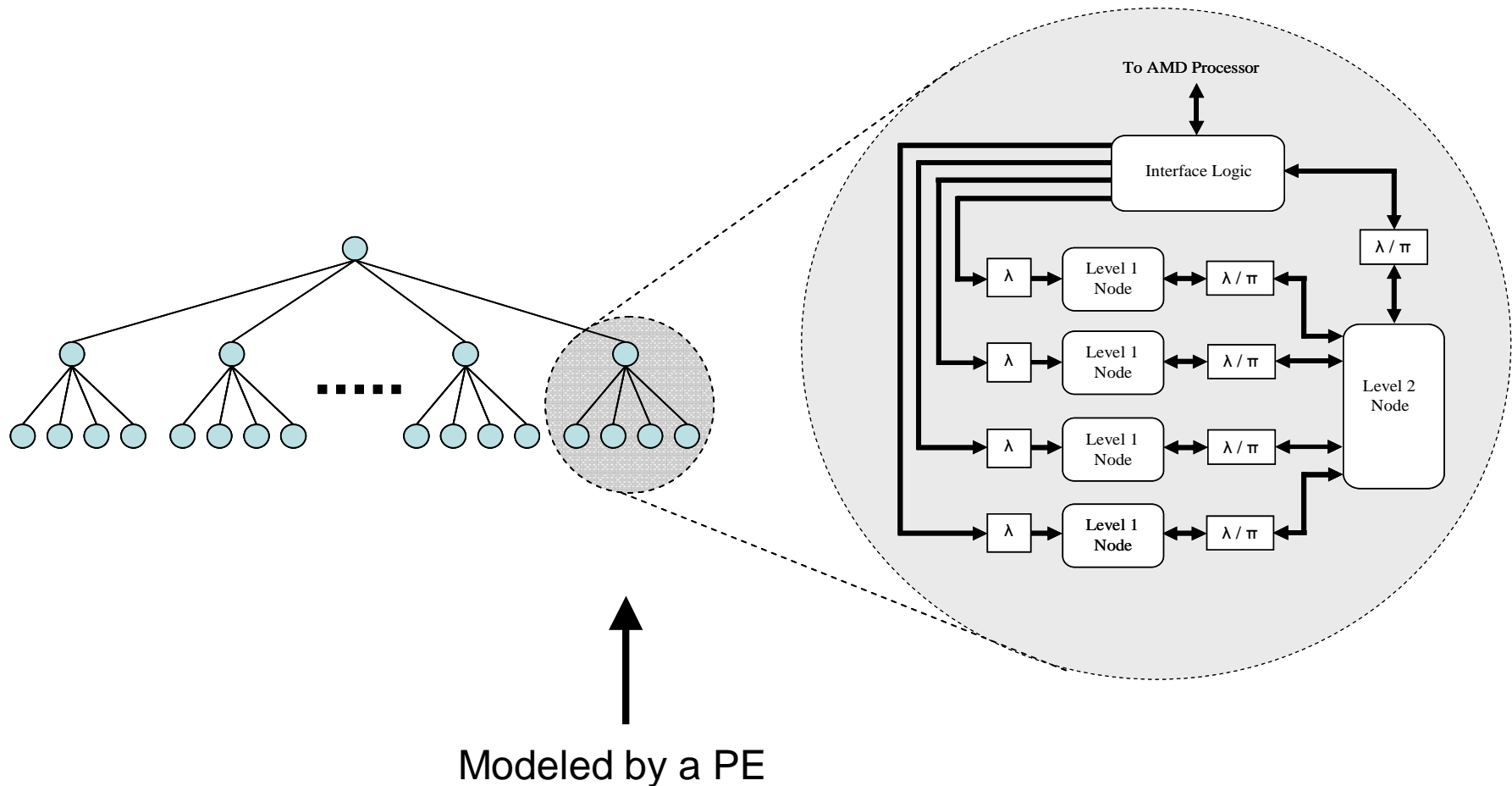
$$\pi_{out}[child][k] = m_{col}[k] / \lambda_{in}[child][k] \quad (6)$$



Cray XD1 Implementation

Processing Element Design

- Hardware design based on processing elements (PE)
- Each PE implements a collection of locally connected nodes



State Machine for Node

Three phases:

- Initialization
- Computation
- Generate output beliefs

$$\lambda_{product}[i] = \prod_{child} \lambda_{in}[child][i] \quad (1)$$

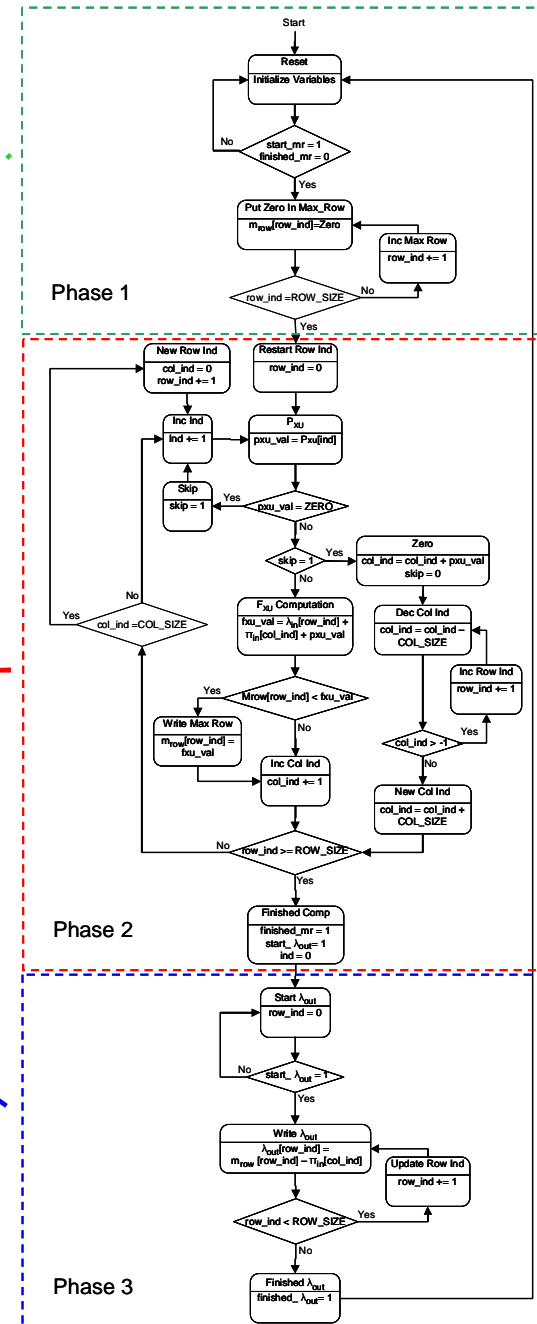
$$F_{xu}[j][k] = \pi_{in}[j] \times P_{xu}[j][k] \times \lambda_{product}[k] \quad (2)$$

$$m_{row}[j] = \max(m_{row}[j], F_{xu}[j][k]) \quad (3)$$

$$m_{col}[k] = \max(m_{col}[k], F_{xu}[j][k]) \quad (4)$$

$$\lambda_{out}[j] = m_{row}[j] / \pi_{in}[j] \quad (5)$$

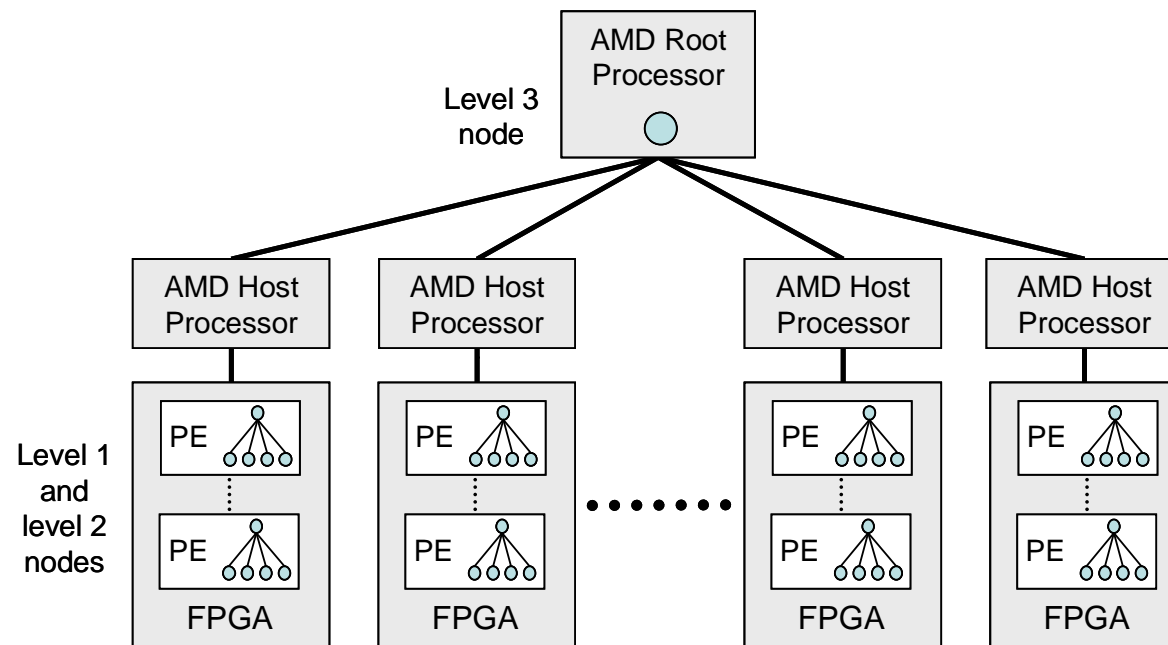
$$\pi_{out}[child][k] = m_{col}[k] / \lambda_{in}[child][k] \quad (6)$$



Network Implementation

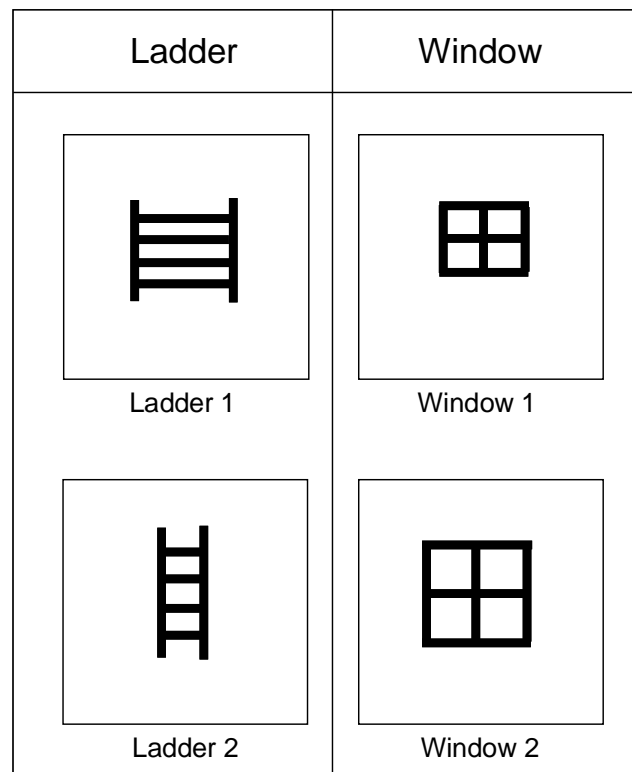
- Design contained multiple PEs per FPGA depending on network type
- Top level node was implemented in C
- Lower level nodes in FPGAs or AMD cores
- Communication was through MPI

Hardware-Accelerated Network



Networks and Test Images

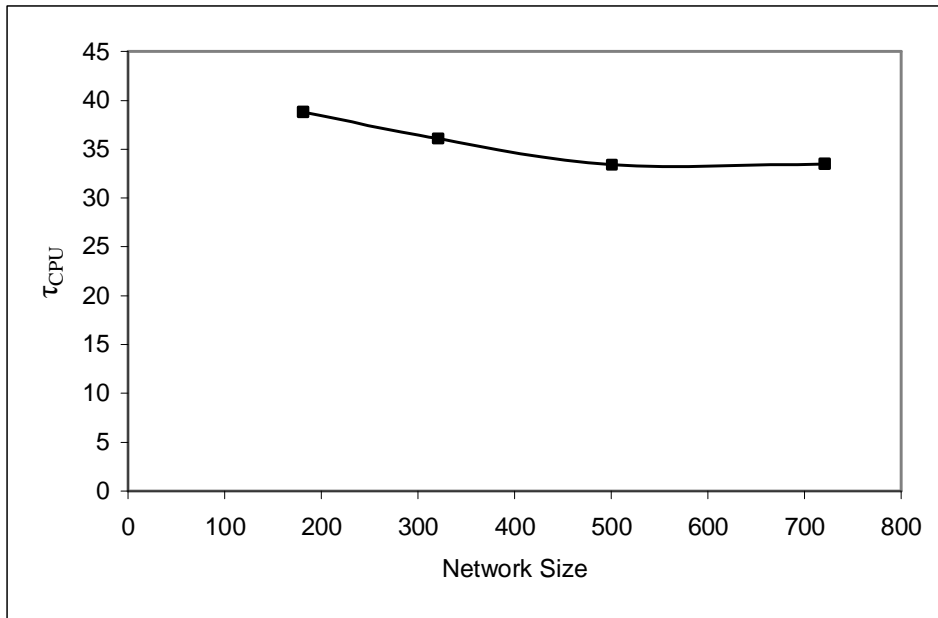
- Trained Network of 181, 321, 501, 721 nodes
- Used 76 Binary image categories
 - Subset of images used by Dileep George to train prototype design



Results

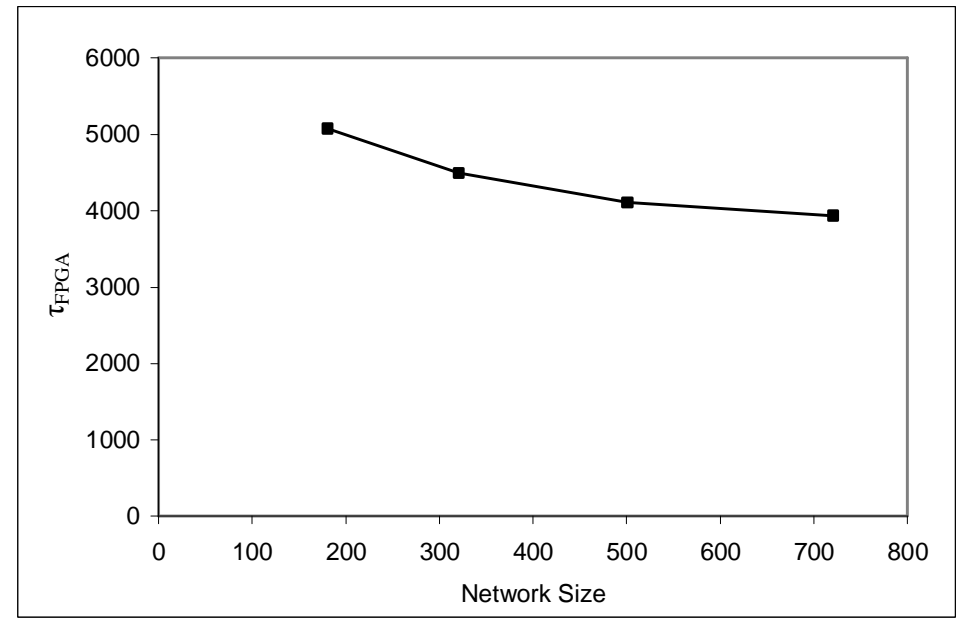
Throughput for a Single Network

Throughput per processing core (T_{CPU})



Nodes/(Second*Core)

Throughput per FPGA (T_{FPGA})



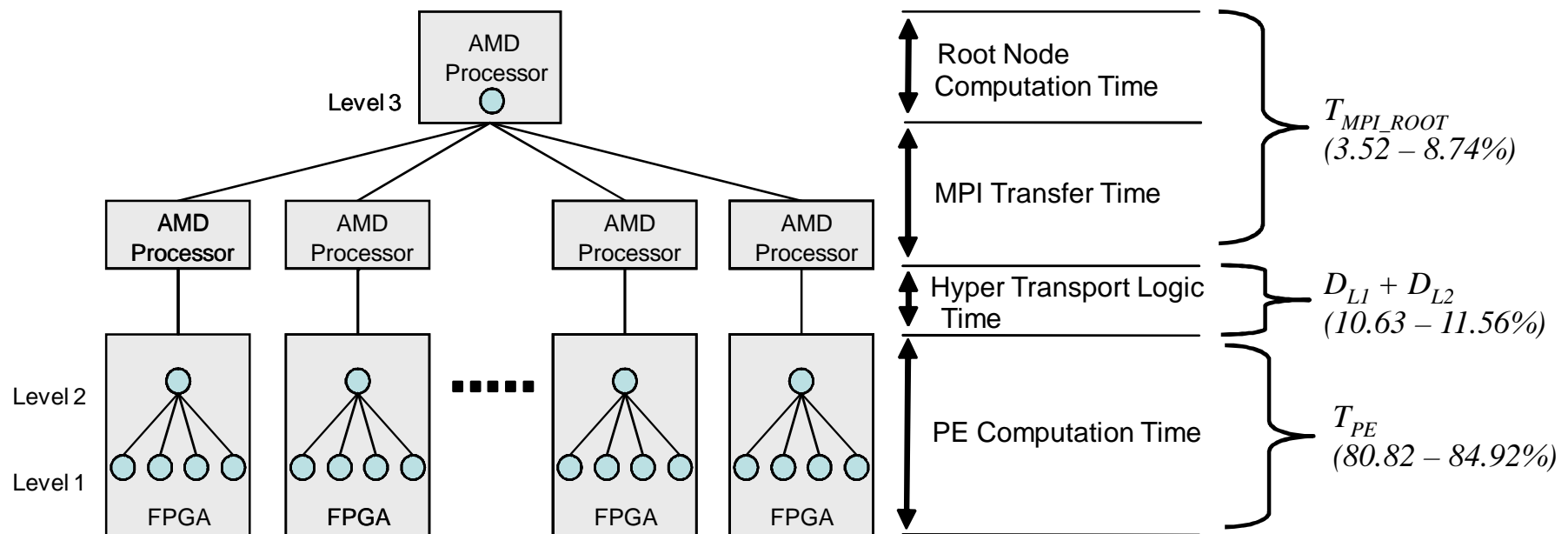
Nodes/(Second*FPGA)

Hardware offers an average throughput gain of 123

Timing Breakdown

T_{SP} – The time required to complete one pass through a network

- ❑ T_{PE} – Time to compute level 1 and level 2 node computations
- ❑ D_{L1} - Time to send λ_{in} to level 1 nodes
- ❑ D_{L2} - time to transfer level 2 λ_{out} and π_{in} between level 2 nodes and processors
- ❑ T_{MPL_ROOT} – Time for root node computation and MPI communication



FPGA Resource Utilization

All Hardware-Accelerated Network performed at 138 MHz

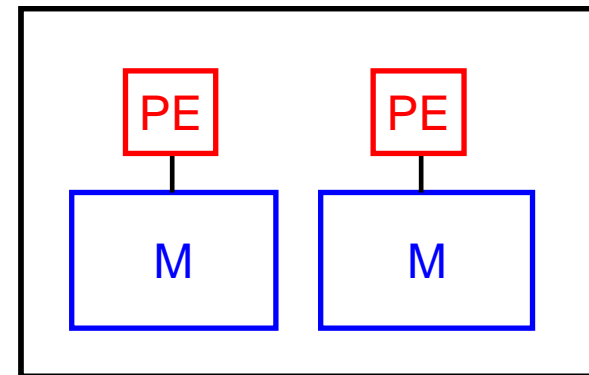
Resource Utilization	Network Size ($nodes_{NET}$)			
	181	321	501	721
Logic	21%	22%	22%	22%
Memory	77%	85%	91%	91%

Scaling Analysis

Full Cray Evaluation

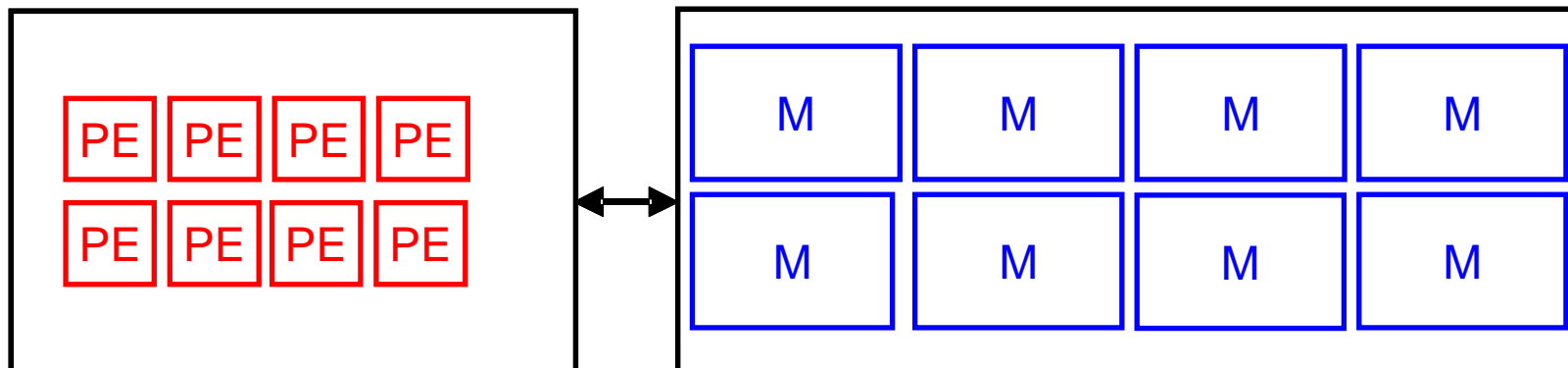
- Software
- Hardware Accelerated:
 - Using On-Chip Memory
 - Using Off-Chip Memory

Using On-chip Memory



FPGA

Using Off-chip Memory



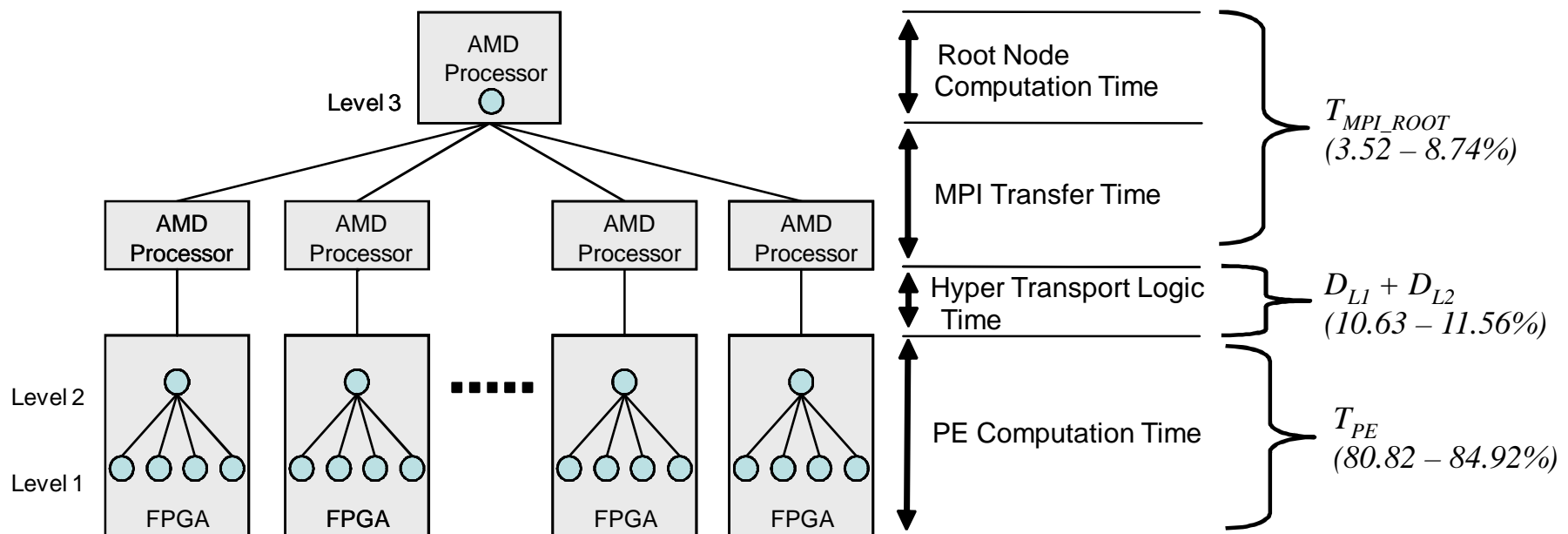
FPGA

SRAM

Hardware Scaling

Time for 1 pass through network (T_{SP}) changes based on number of PEs on FPGA

$$T_{SP} = (D_{L1} \times PE_{FPGA}) + \left(\left\lceil \frac{PE_{FPGA}}{L2_par_io} \right\rceil \times D_{L2} \right) + T_{PE} + T_{MPI_ROOT}$$



Hardware Scaling (Continued)

- Throughput for using the entire Cray XD1 ($T_{System,HW}$) is estimated as

$$PE_{System} = FPGAs_available \times PE_{FPGA}$$

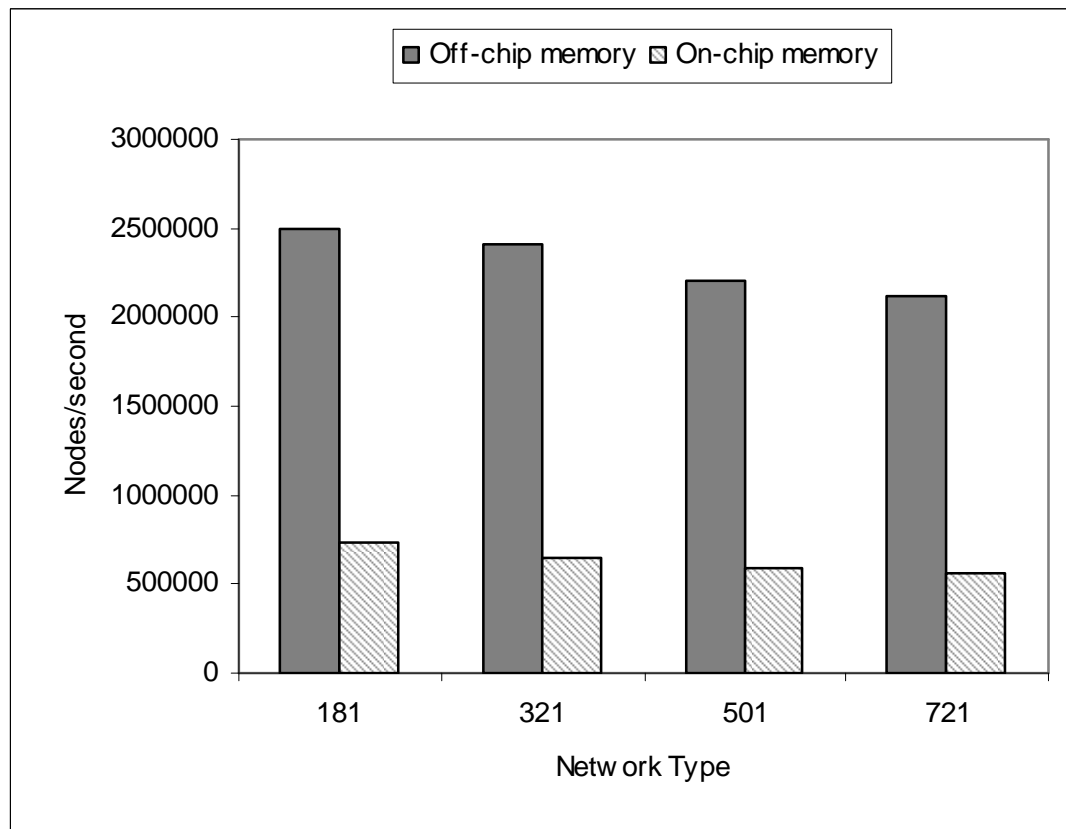
$$networks_{System} = \frac{PE_{System}}{NPE}$$

$$\tau_{System,HW} = \frac{networks_{System} \times nodes_{NET}}{Time\ for\ 1\ pass} = \frac{PE_{System} \times nodes_{NET}}{NPE \times T_{SP}}$$

Hardware Scaling (Continued)

- Throughput for using the entire Cray XD1 ($T_{System,HW}$) is estimated as

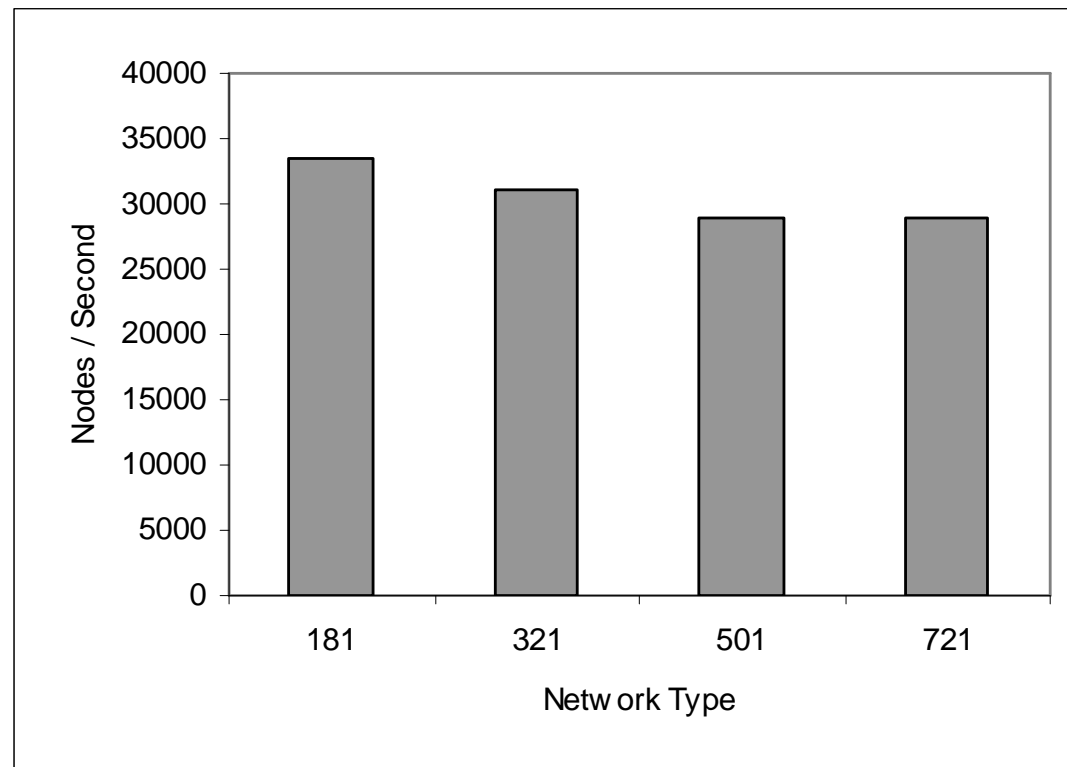
$$\tau_{System,HW} = \frac{networks_{System} \times nodes_{NET}}{Time\ for\ 1\ pass} = \frac{PE_{System} \times nodes_{NET}}{NPE \times T_{SP}}$$



Software Scaling

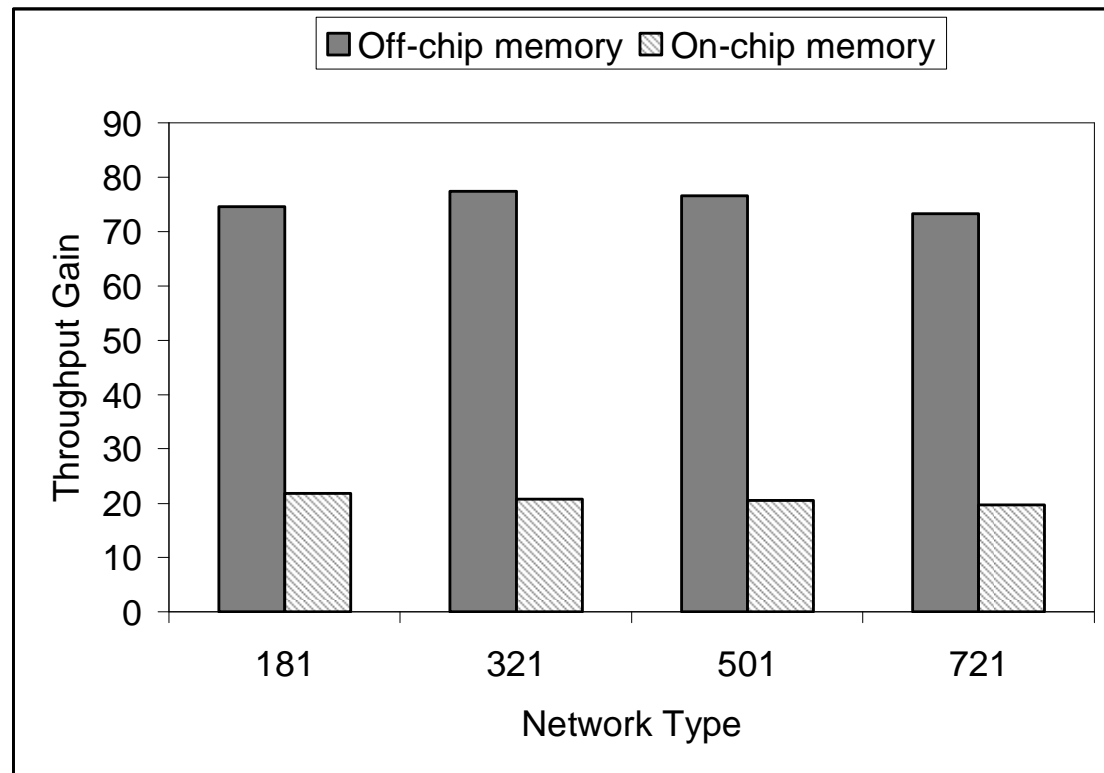
- Based on performance of software system
($T_{System,SW}$)

$$\tau_{System,SW} = \tau_{CPU} \times cores_available$$



Full Cray Performance Comparison

Throughput Gain of Hardware-Accelerated designs over a software based design utilizing full Cray resources.



On average, using off-chip memory shows a 75 X throughput gain and using on-chip memory shows a 20 X throughput gain

Conclusions

Conclusions

- Cognitive Model

- Implemented various sized model networks on XD1
- Estimated performance of a model using full Cray
- FPGA based system can provide an average throughput gain of 75 X over software implementation on full Cray

Extensions

- Cognitive Model
 - Implement Hardware-Accelerated versions of other cognitive models
 - Compare performance of other FPGA cognitive models with George and Hawkins model

Acknowledgements

- Air Force Research Laboratory (AFRL)
- National Science Foundation (NSF)
- Center for Computational Science at Navy Research Laboratory (NRL)

For more on this work see:

Rice, K, Vutsinas, C., and Taha, T. M., “A Scaling Analysis of a Neocortex Model Implementation on the Cray XD1,” *Journal of Supercomputing*, (accepted February 2008).
