How to Build the CNS Network

Nate McNamara, nate@cs

Alok Agrawal, aloka@cs

- Overview of CNS
- Performance Model
- Results
- Conclusions

CNS Overview

Connectionist Network Supercomputer

Optimized for neural net applications

- Highly parallel
- Fixed point vector processors

Benchmarks

- Benchmark in the CNS Architecture Specification
  - Sparse network
  - 10^6 neurons
  - 10^9 random connections
- ICSI speech group's neural net
  - Fully connected
  - Few inputs and outputs
  - Thousands of hidden units

Most real applications resemble the speech net

A Performance Model

Usual approach: Simulation

- Accurate

  Often simulate entire system down to the cycle

- Hard to integrate with real benchmarks

  Existing literature employs contrived communication patterns

Alternative approach: High level model

- Determine the time for primitive network operations

  Depends on topology, interface, special features

  Example: One-to-all broadcast on a mesh

$$T_{one-to-all} = \sum_{d=x,y}\left(\sum_{i=1}^{\log p_d} t_f + wt_w + \frac{p_d}{2^i}t_h\right)$$

- Express benchmark as a sequence of these primitives

Estimation of Model Parameters

Interface

- Launch in one cycle
- Poll in one cycle

Router

- Link width
  - Bidirectionality cuts in half
  - Added dimension cuts in half for mesh
- Hop cost
  - Assume input buffer without head-of-line blocking
  - Assume arbitration takes only one cycle, because there really isn't any

Computation

- Vector units
  - 8 multiply-adds per cycle
- SRAM
  - 4 words per cycle

The Benchmark Models

Both benchmarks have the same five steps:

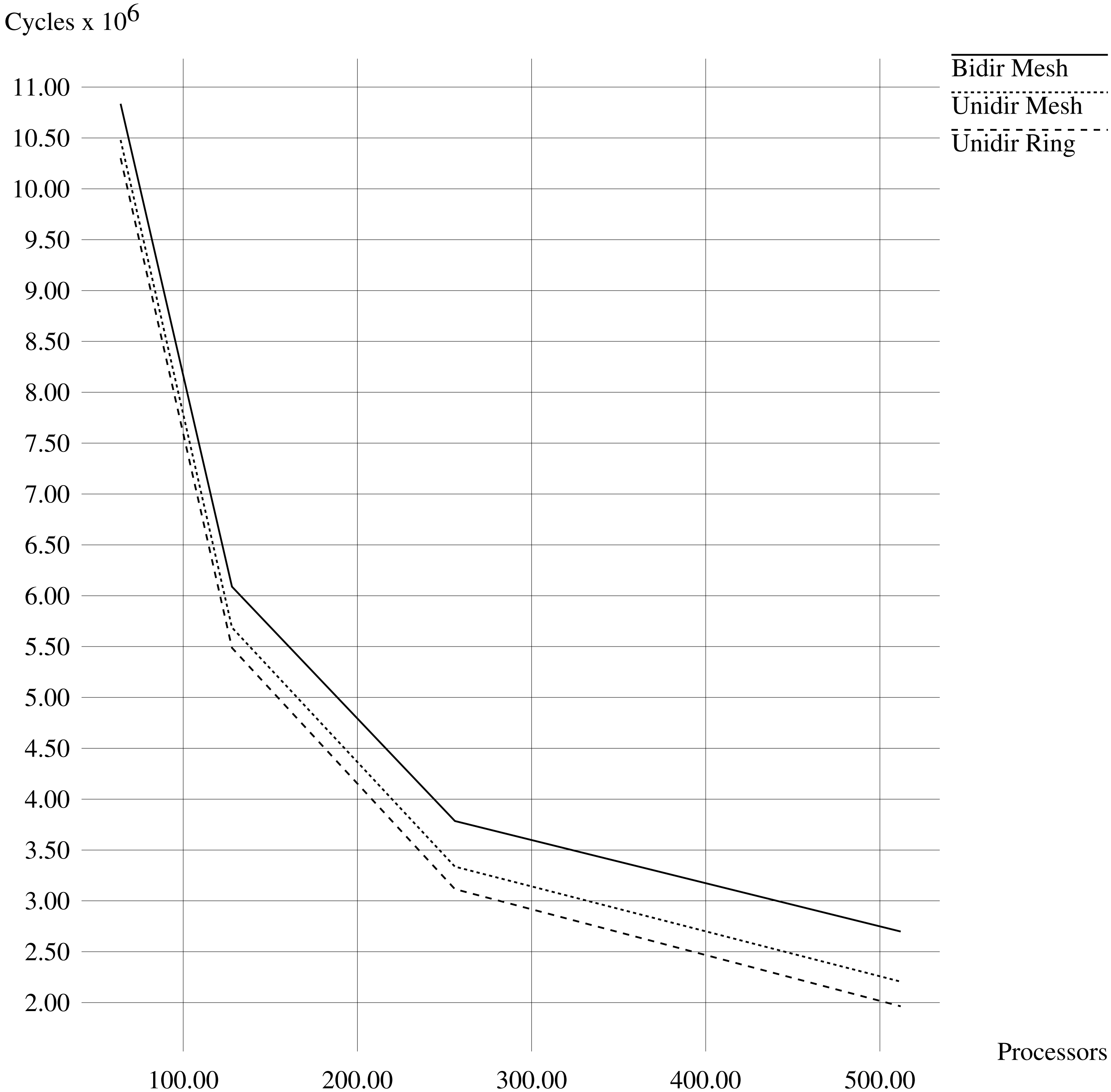
- Distribute the input vector from the host
- Compute the hidden layer activations
- Distribute the hidden layer activations
- Compute the output layer activations
- Collect the output at the host

One possible implementation:

- One-to-all broadcast
- Computation
- All-to-all broadcast
- Computation
- All-to-one collect

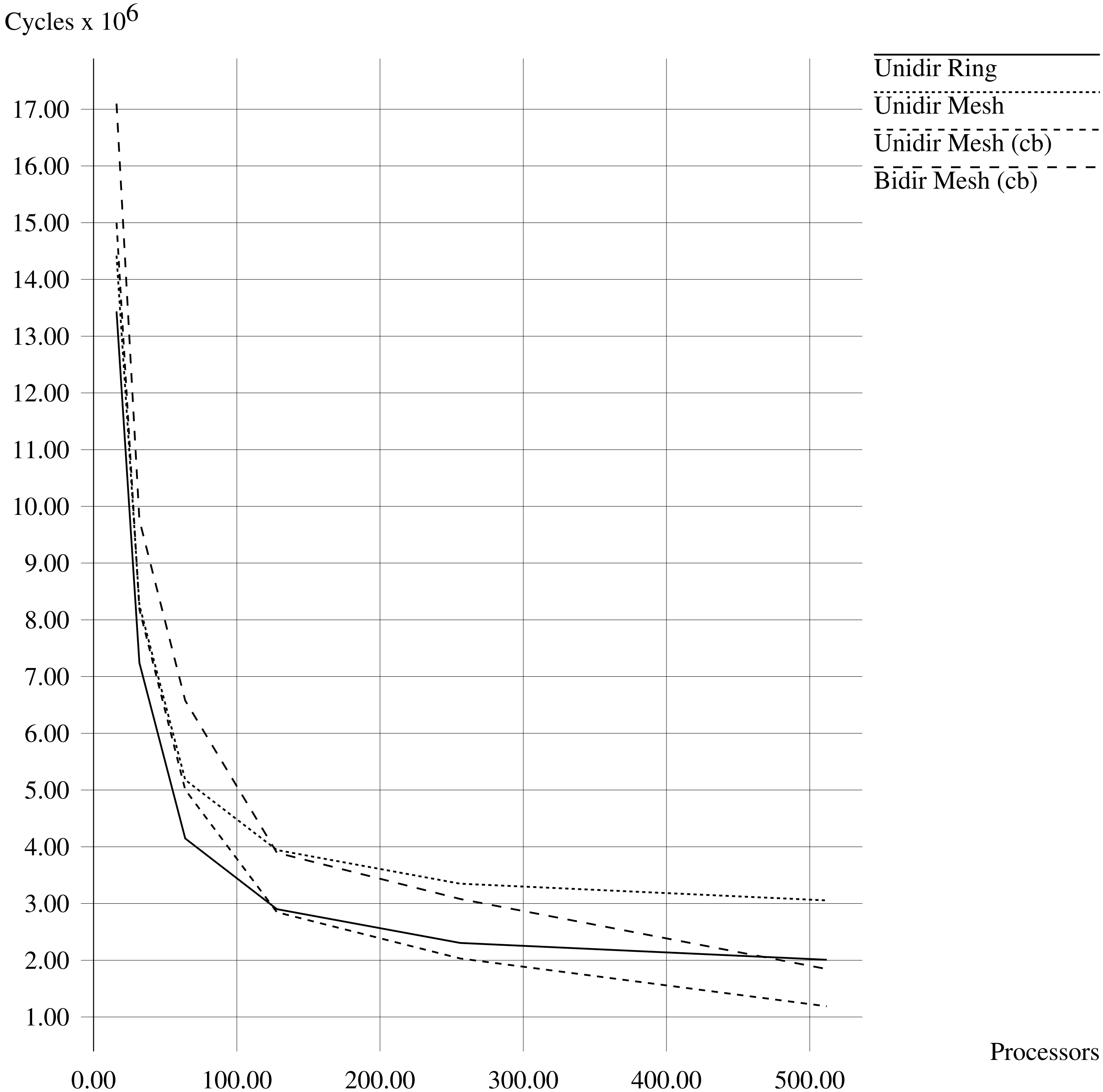
Results

Time v Machine Size for sparse case with 10^9 connections



Results continued

Time v Machine Size for speech net with 10^6 hidden neurons



Conclusions

Network design depends heavily on application

- For most problems, a unidirectional ring is best
- For large, dense networks on large machines, a unidirectional 2-D mesh is fastest, but:

    Price/performance is very poor above 128 processors

    Big problems => lots of SRAM

Caveats

- Need a model that accounts for finite packet size

    DMA could solve this, but not part of current CNS design

- Need a better handle on constants
- Other benchmarks to consider

    Backpropagation

    Clustered neural nets