Sparse Flat Neighborhood Networks (SFNNs): Scalable Guaranteed Pairwise Bandwidth & Unit Latency

Timothy I. Mattox, Henry G. Dietz, & *William R. Dieter*

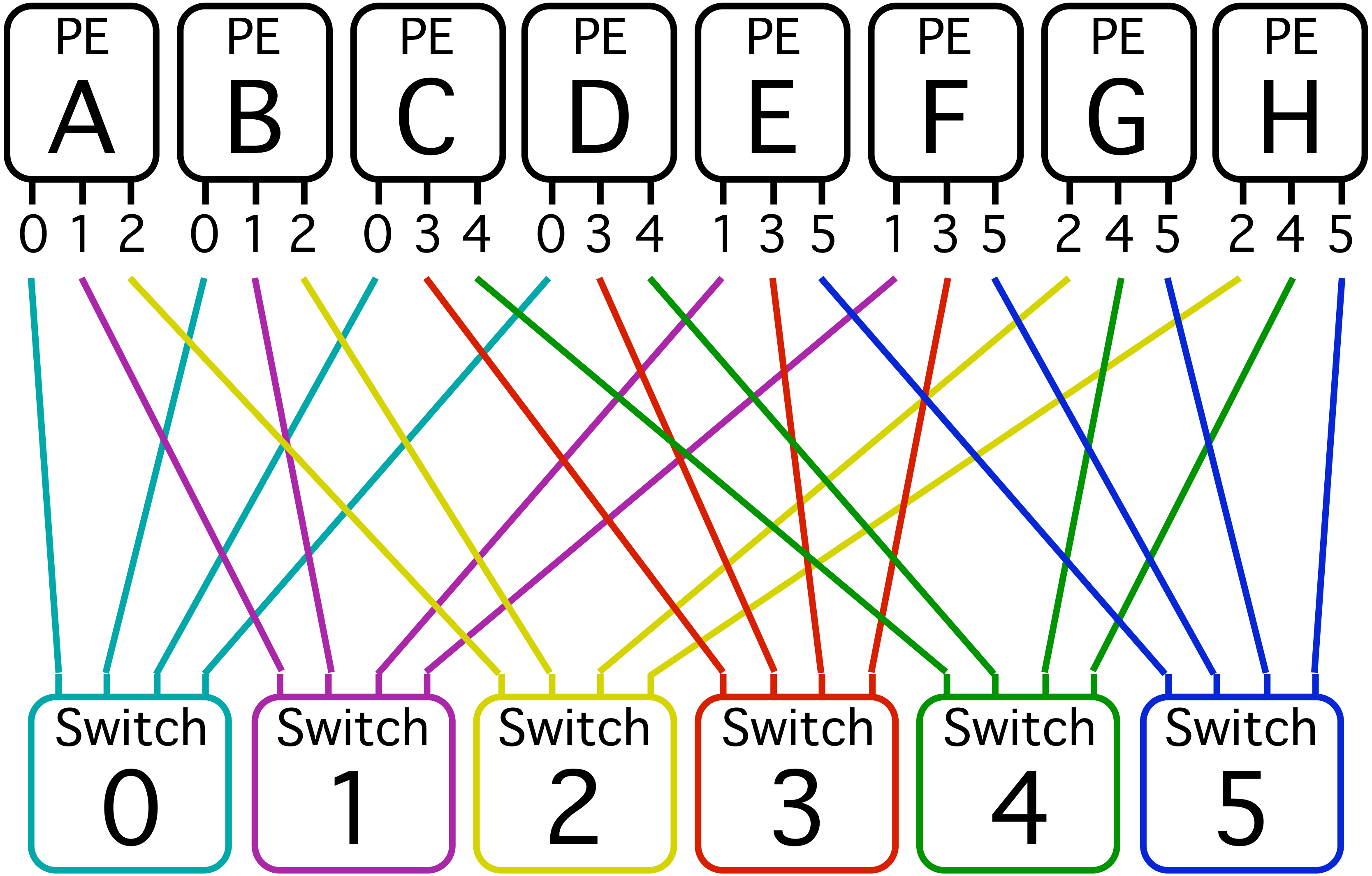
Electrical and Computer Engineering Department
University of Kentucky
Lexington, KY 40506-0046

tmattox@engr.uky.edu,
hankd@engr.uky.edu,
dieter@engr.uky.edu

Flat Neighborhood Networks

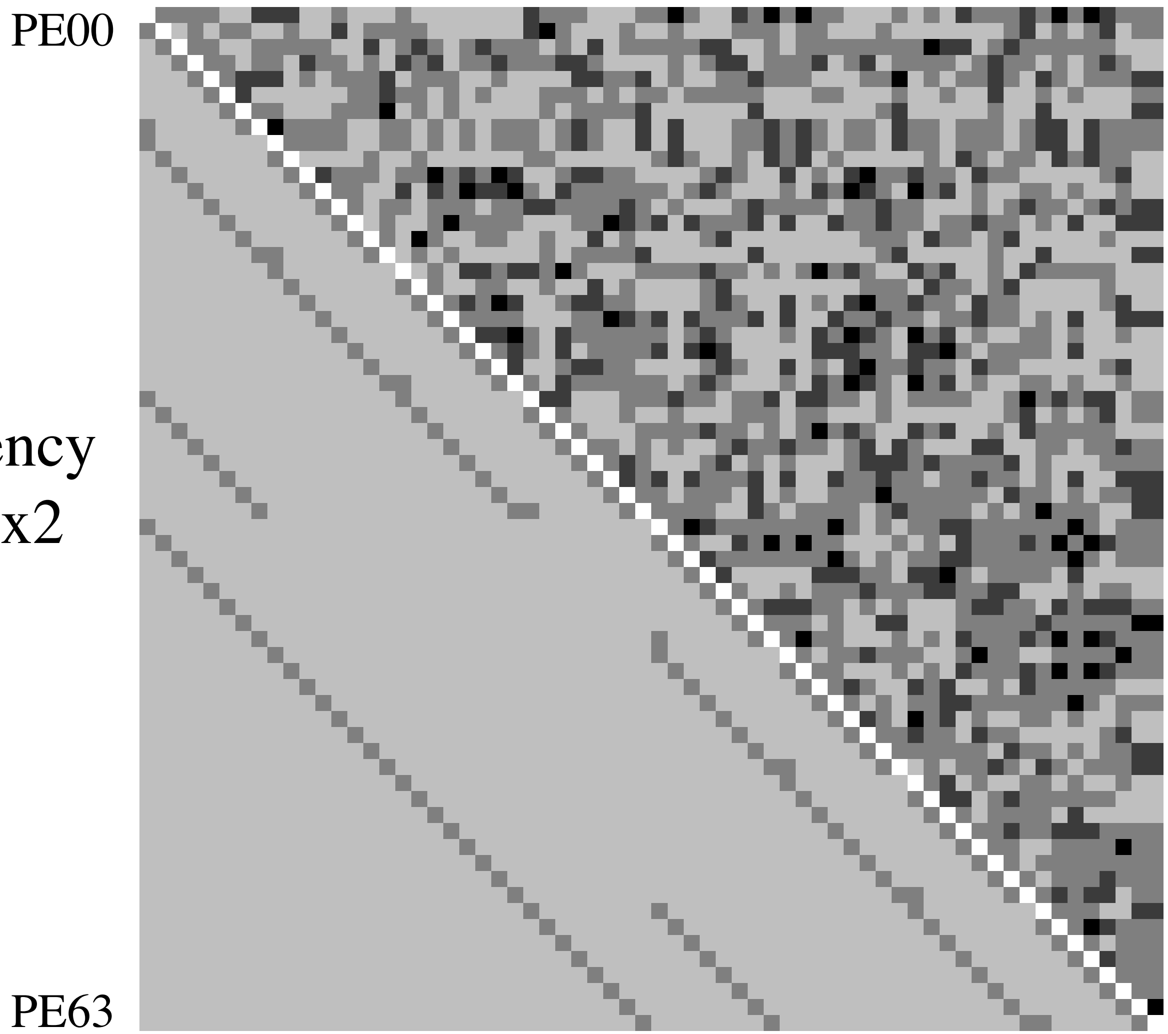
- Single switch-hop = low latency
- No shared links = guaranteed bandwidth
- Multiple Network Interfaces (NIs) per node
- Can be lower cost and faster than a fat-tree
- Designed by GA (genetic algorithm)
    - Incorporate program requirements
    - Search includes asymmetric designs

Example Universal FNN



KLAT2's Universal FNN

Specification:
- All PE pairs want low latency
- 3D torus 8x4x2 ±1 offsets want extra bandwidth



Solution:
- All PE pairs have unit latency
- 3D torus 8x4x2 ±1 offsets 153 of 160 pairs have at least two units of bandwidth
- 4 NIs per PE
- 9 switches (32-ports each)

Note: KLAT2 was first supercomputer under $1000/GFLOPS, 2000

Universal FNN?

- All node pairs are equidistant
- All permutations pass in one hop
- Many non-permutations also are single-hop
- A PE's list of neighbors contains every other PE
- Scalability is only ~2-5x of a single switch

Relax these to get better scaling...

Sparse FNN Idea

- Select a target suite of parallel programs
- Find the set of communication patterns used
- Take the union of the important patterns
- Construct a desired neighbor list for each PE
- Satisfy the FNN property for these neighbors

Communication Patterns

- O(1): shuffle, bit-reversal, mesh/tori neighbor

  ± 1 offsets in 2D and 3D

- O(log(N)): hypercube, reductions

  ± power of 2 offsets in any dimension

- $O(N^{1/D})$: scatter, gather, all-to-all

  i.e., not permutations

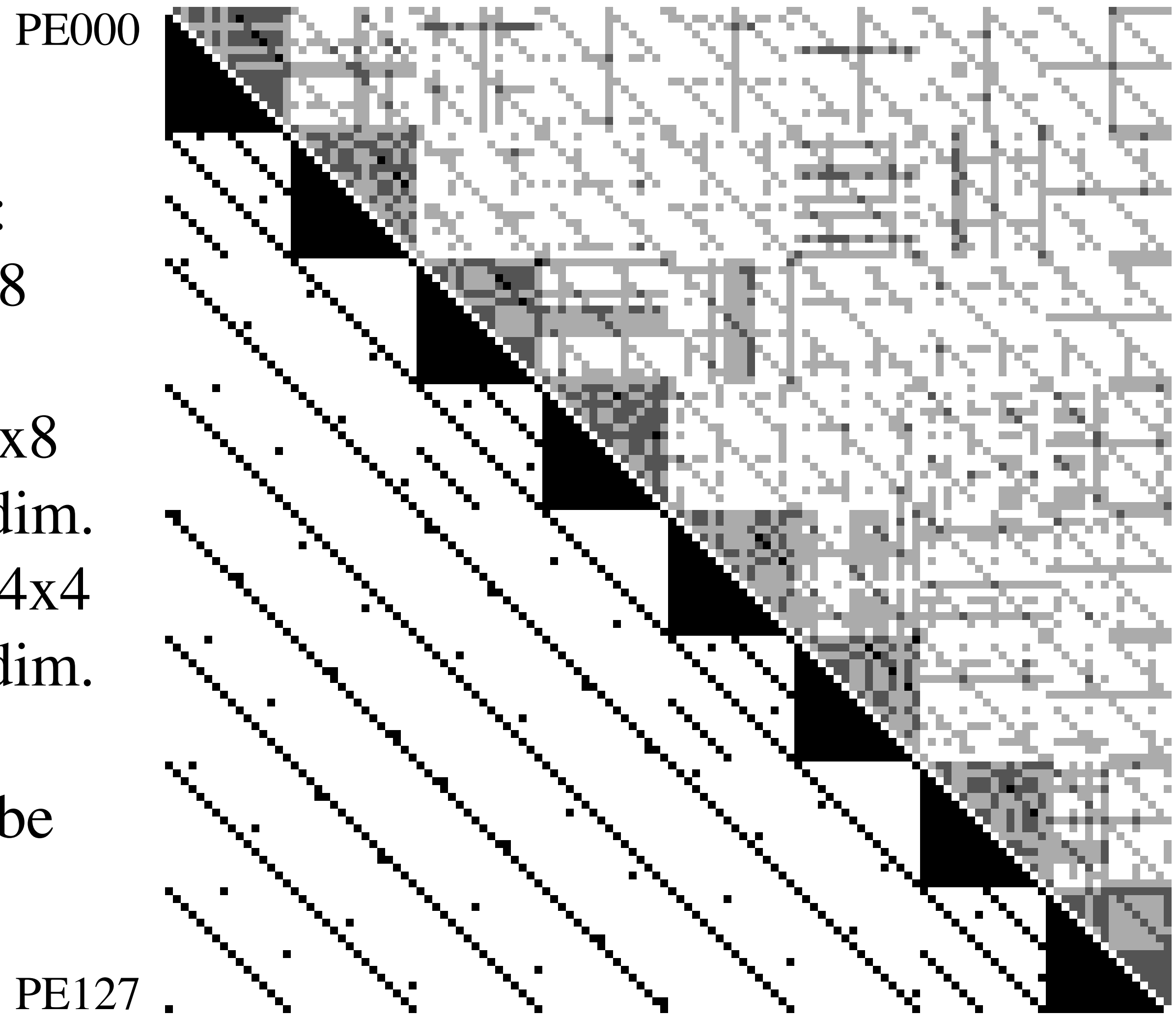
- Overlap between patterns: pair synergy

Sparse FNN Properties:

- Single switch latency for chosen patterns
- Full bisection bandwidth for chosen patterns
- Scales better than Universal FNNs
  - Neighbor lists scale as ~O(log(N)) vs. O(N)
  - Lower cost (uses narrower switches)
  - Design solutions found for over 10K PEs

KASY0's Sparse FNN

Specification:
- 1D torus 128 ±1 offsets
- 2D torus 16x8 all in same dim.
- 3D torus 8x4x4 all in same dim.
- bit-reversal
- 7D hypercube



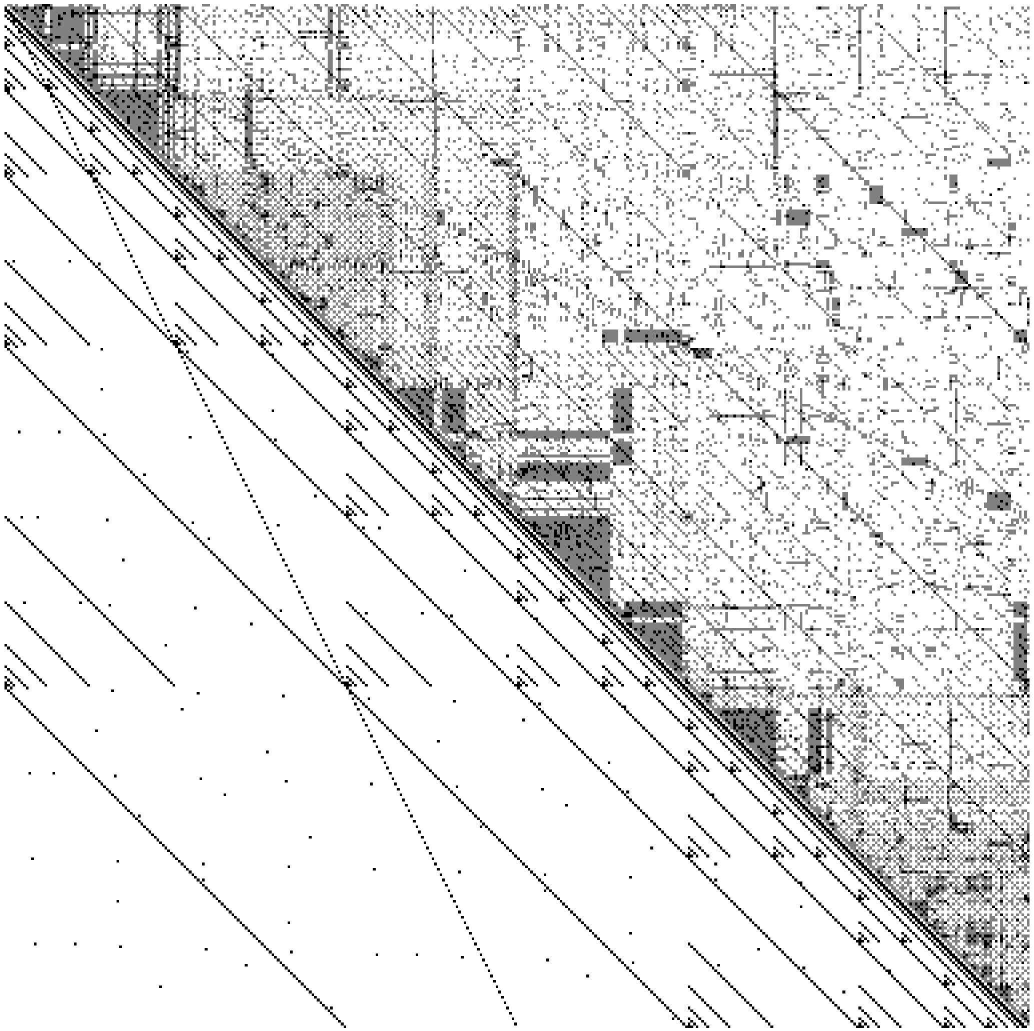
Solution:
- All requested PE pairs have unit latency
- All requested PE pairs have at least 1 unit of bandwidth
- 3 NIs per PE
- 17 switches (24-ports each)

Note: KASY0 was first supercomputer under $100/GFLOPS, 2003

1024-PE Sparse FNN Example

Specification:
- 1D torus
  ±2^k offsets
- 2D tori
  ±2^k offsets
- 3D tori
  ±2^k offsets
- shuffle
- bit-reversal
- 10D hypercube
- 2D transpose

PE000



PE383

Solution:
- All requested PE pairs have unit latency
- All requested PE pairs have at least 1 unit of bandwidth
- 2-6 NIs per PE
- 101 switches (48-ports each)

523,766 possible PE Pairs:
- 2.78% requested
- 19.6% covered in this solution

•

•

•

FNN Runtime Support

- Support coming to the Warewulf cluster toolset
- Modified Linux 2.4 & 2.6 Bonding Driver
  - Run any IP layer software, unmodified
  - Compressed routing table (4KB for 1024 PEs)
  - MAC addresses locally administered by driver

Conclusion: Sparse FNNs

- Give more control over cost/performance trade-offs
- Take account of what the parallel programs actually need
- Can achieve single-switch latency for very large systems
- Can guarantee pairwise bandwidth for very large systems