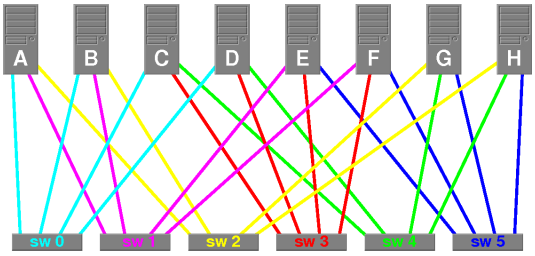
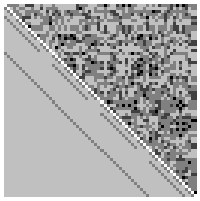
Flat Outperformance

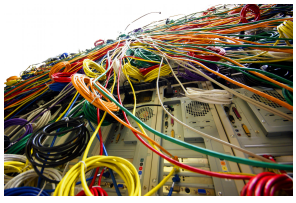
INTERCONNECTION NETWORKS play a critical role on the performance of parallel computers – be they clusters spanning many racks or massively parallel supercomputers with multiple cores. Communications within parallel programs tend to have specific properties that allow a well-engineered network such as a **Flat Neighborhood Network (FNN)** to dramatically outperform straightforward topologies such as Fat trees. Flat Neighborhood Networks come in three flavors: *Universal*, *Sparse,* and *Fractional* Flat Neighborhood Networks.

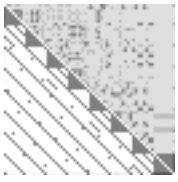


UNIVERSAL FLAT NEIGHBORHOOD NETWORKS (FNNs). *Bisection bandwidth* and *latency* are the critical metrics that affect performance on parallel systems. Using multiple network interfaces per node, Universal FNNs provide *single-switch* latency, full link bandwidth and better bisection bandwidth than a Fat tree with comparable hardware complexity. As shown above for 4-port switches connecting 8 nodes, an FNN connects nodes to switches such that each node-pair has at least one switch in common. The best wiring pattern usually is *asymmetric*; the design is *evolved* from random wiring patterns using a genetic algorithm.

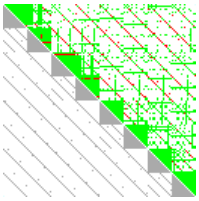


FNN design problems and solution quality are summarized by square maps in which the darkness of each point indicates how many single-switch paths exist between the corresponding pair of nodes; the lower left triangle is the minimum design requirement, the upper right is what the FNN design actually provides. The map above shows the world's first FNN, which in April 2000 connected the 66 nodes of KLAT2 (KENTUCKY LINUX ATHLON TESTBED 2) using 31-port 100Mb/s ETHERNET switches and standard IP to deliver close to 25Gb/s bisection bandwidth, and $30\mu s$ latency, at a network cost of about $8,100.





SPARSE FLAT NEIGHBORHOOD NETWORKS (SFNNs). Surprisingly, very few parallel programs depend on every node talking to every other; usually, each node talks to at most $O(log(N))$ other nodes. By ensuring single-switch latency *only for node pairs that are expected to communicate*, SFNNs can provide single-switch latency for all critical communications in a typical application suite using cheap, narrow, switches for systems having many thousands of nodes. The first SFNN was KASY0 (KENTUCKY ASYMMETRIC ZERO), built in 2003, covering 128 nodes using 24-port switches – at about half the network cost of KLAT2's FNN. Our latest demonstration of SFNN is a 64-node cluster **NAK** (**N**VIDIA **A**thlon cluster in **K**entucky), built in April, 2010, primarily intended for GPU research. The SFNN wiring diagram and design map for NAK are shown above.



FRACTIONAL FLAT NEIGHBORHOOD NETWORKS (FFNNs). Although SFNNs have great price/performance, the search is driven by the performance; FFNNs flip priorities, finding the *best coverage possible* for a *fixed network cost*. For example, using far less hardware than KASY0, the above map shows coverage of an FFNN in **green** – only the **red** spots deliver poorer latency. A 98-node cluster **HAK** (**H**alf-powered **A**thlon cluster in **K**entucky) built in June, 2010 is the primary testbed for FFNNs and performance results will be released soon.

Note that FNN, SFNN, and FFNN topologies all are fully compatible with ETHERNET, IP, and most other commonly available network technologies and protocols. FNN design tools and have been ported to the cluster provisioning toolkit, **Perceus** (http://perceus.org) ver. 1.5.0. The FNN driver has been ported to the latest Linux kernel version, 2.6.31.6-2.caos provided by Perceus. Design tools, including interactive WWW forms and the FNN driver are freely available online at **`Aggregate.Org`**.

This document should be cited as:

```
@techreport{sc10flat,
author={Henry Dietz and Krishna Prabhala},
title={Flat Outperformance},
institution={University of Kentucky},
address={http://aggregate.org/WHITE/sc10flat.pdf},
month={Nov}, year={2010}}
```

Aggregate.Org UNBRIDLED COMPUTING

SC10 New Orleans, LA