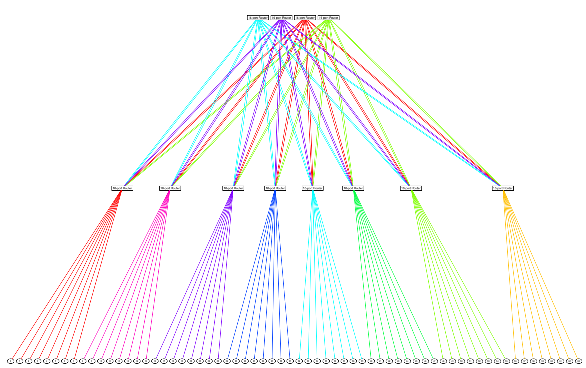
Adventures in KNITTing

Knitting is a process that converts one-dimensional material into a two-dimensional object. Normally, one thinks of knitting yarn or thread; however, in cluster supercomputers the disturbingly abundant one-dimensional material is network cable. **KNITT**, **Kentucky's Network Implementation Topology Tool**, uses GA (Genetic Algorithm) evolutionary search technology to create the physical placement structure that will most efficiently implement the wiring of a given logical interconnection pattern.

Logical network topologies are normally expressed in the form of an interconnection list. There are objects of various types, such as compute nodes and switches, that are listed as being connected to each other. The purpose of KNITT is to add a physical dimension to that structure: a specification of how the objects should be grouped on physical racks in order to minimize the wiring complexity across racks.

It might seem that an optimal physical object placement should be easy to create for a regular network design. However, differences in racking parameters and network features commonly yield non-obvious asymmetric results. Consider a simple 64-node cluster with Fat Tree logical topology using 16-port routers:

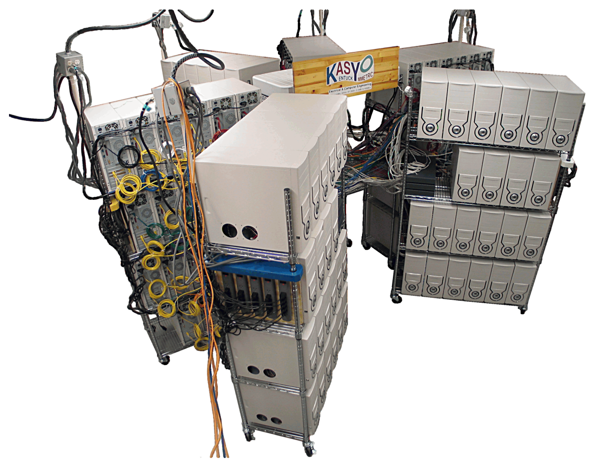


How should this very regular design be physically partitioned into three racks containing no more than 4 switches and 22 nodes each? Here's the implementation topology that KNITT found:

```
[s00 s01 s05 s08 n06 n08 n09 n10 n11 n12
 n13 n14 n15 n32 n33 n34 n35 n36 n37 n38
 n39 n50 n61 n63 n n]
[s02 s04 s07 s10 n00 n01 n02 n03 n04 n05
 n07 n24 n25 n26 n27 n28 n29 n30 n31 n48
 n49 n51 n52 n53 n54 n55]
[s03 s06 s09 s11 n16 n17 n18 n19 n20 n21
 n22 n23 n40 n41 n42 n43 n44 n45 n46 n47
 n56 n57 n58 n59 n60 n62]
```

Each bracketed set represents one physical rack. The first letter of an object name is a letter – here, `s` for a switch and `n` for a node. The first letter by itself represents an empty slot that could hold that type of object.. The most obvious of physical layouts for the above fat tree would have resulted in 40 of the system's 96 cables crossing from one rack to another. In contrast, just 26 cables need to cross between racks in this implementation topology.

It shouldn't be surprising that we've been using GA technology to optimize physical layout ever since we built **KASY0** in 2003:



However, we had not made the user interface simple enough for others to use until 2011. For more information, see `http://aggregate.org/KNITT/`

This document should be cited as:

```
@techreport{sc13knitt,
author={Henry Dietz},
title={Adventures in {KNITTing}},
institution={University of Kentucky},
address={http://aggregate.org/WHITE/sc13knitt.pdf},
month={Nov}, year={2013}}
```