Cluster Design For The Lazy Or Paranoid

What is a cluster supercomputer? Traditionally, each supercomputer was the result of years of custom design and manufacturing effort; in contrast, a cluster supercomputer is built taking advantage of standardized interfaces and interchangeable parts. This allows clusters to be built quickly, using the latest technology at low cost, while simultaneously allowing an exceptionally wide range of design choices for customizing the system. Unfortunately, picking the best cluster design for a given set of applications and resource constraints requires searching a huge and complex design space that is often significantly altered by commodity pricing and other fluctuations.

Since February 1994, when we built the world's first LINUX PC cluster, we've been helping people design and build clusters as a free service to the community. However, good design is hard even for us experts, so we built computer-aided engineering software to partially automate the design space exploration. The CDR (CLUSTER DESIGN RULES) software tool, also known as the BDR (BEOWULF DESIGN RULES) project at SOURCEFORGE, combines years of experience with sophisticated performance models to perform an intelligent exhaustive search of the cluster design space. The search is completely vendor and technology neutral, literally considering *all* designs that could be constructed using components from a user-supplied parts database.

Cluster Design For The Lazy. A live version of the CDR is freely available via an HTML forms interface from `Aggregate.Org/CDR` (incidentally, all the information you enter is kept private... we don't even store a copy of the form data). There is a default part database that you can use directly or modify to create your own. Simply fill-in the forms and let the tool tell you about the K best designs it found to meet or exceed your requirements. The forms allow you to specify application-driven performance requirements like message latency, memory bandwidth needed per FLOP, etc.; alternatively, you can tell it to maximize performance on any of the applications for which it has built-in detailed performance models. *Want to maximize the* TOP500 *rank of the machine you can buy for $100K? – just enter your budget, floorspace, power, and cooling and select the detailed* HPL *performance model.* Of course, various approximations are made in the analysis so you shouldn't blindly do what the tool suggests, but the designs it recommends provide a great starting point for a human designer. They also are useful when writing an equipment budget for a proposal.

Cluster Design For The Paranoid. If you're paranoid, like we are, you really want to know that you're building the absolute best possible design. You can use multiple manual runs, or the new scripting support, to do things like examine what if scenarios such as, "how cheap do ATHLON64 X2 chips need to get before my upgrade budget would give me a factor of two speedup on my application?" In fact, not only can you get an approximate answer to such questions, but the BDR actually allows you to construct your own plug-in model for performance of your application.

How Smart Is The CDR/BDR? That depends on your definition of smart. For example, the tool is dumb in that it does not have the concept of a single switch in which different ports can have different, otherwise incompatible, physical interfaces (e.g., mixing copper GigE, fiber GigE, and ATM ports). On the other hand, it is very smart about knowing all possible ways to interconnect network components that it understands; topologies it generates and models include:

- Direct connections between nodes
- 1D, 2D, and 3D toroidal meshes
- Simple switched networks
- Two-level trees of switches, including "short circuit" routing enhancements to improve bisection bandwidth
- Two-level FAT TREES
- CHANNEL-BONDED versions of all the above
- FLAT NEIGHBORHOOD NETWORKS (FNNs), custom designed using a fast algorithm that derives designs from a large set precomputed using a genetic algorithm
- FNNs of two-level trees of switches

These classes of topologies are not just tested in representative cases, but in *all possible configurations using the parts in the database*. The bounds on the configurations tested are obtained by pruning partial designs as soon as they become worse than the K^{th}-best complete design found so far. Similarly thorough analysis is used for each aspect of system design.

This document should be cited as:

```
@techreport{sc07cdr,
author={William Dieter and Henry Dietz},
title={Cluster Design for the Lazy or Paranoid},
institution={University of Kentucky},
address={http://aggregate.org/WHITE/sc07cdr.pdf},
month={Nov}, year={2007}}
```

Aggregate.Org UNBRIDLED COMPUTING

SC07