

# Experiments with a small supercomputer

With fail-over and load balancing software, these clusters could become reliable enough for many services in the future

BY THOMAS B. HICKEY, Ph.D., Chief Scientist, OCLC Research

Over the last year OCLC Research has been experimenting with the application of parallel processing to searching and processing large files of bibliographic records, such as those in WorldCat. For this we acquired a 24-node computer configured in what is known as a Beowulf cluster. This type of configuration is becoming very popular for scientific computing because it's cheaper to connect many small machines together than to build a single machine with the same computational power. We have found that this type of machine works just as well for bibliographic processing. In this, we are following the lead of Google and other large-scale Web search and gaming engines that employ similar configurations.

A small supercomputer may sound like an oxymoron; how can a 'supercomputer' be 'small'? The main

searching, but, more generally, we have found it to be useful for virtually any work with large numbers of bibliographic records. WorldCat now contains well over 55 million records, even accounting for records that have been deleted and merged over the years. Since our cluster has 24 separate nodes with a total of 48 processors, we typically get 30-fold speedups in processing, and occasionally much more than that because the entire database can be cached in main memory.

A 30-fold speedup means that processes that previously would take a month can be done in 24 hours. Tasks that took a day now take less than an hour. One of the more dramatic speedups is to extract records based on a sequential scan of the whole database. That can be done in three seconds, rather than 20 minutes.

The organization of the cluster consists of one *head* node that controls all the others and manages all outside

## Breaking WorldCat up into 69 pieces results in less than a million

difference between our computer and some of the largest computers built for scientific computation is that our machine has 24 nodes in it rather than the thousands of nodes the largest commercial machines have. To put this in perspective, however, our machine has approximately the speed of the fastest machine in the world ten years ago (which probably cost about one thousand times as much as ours!).

We obtained the machine to investigate parallel text searching. At OCLC we have always searched our databases in parallel, but in as few pieces as we could. In this project we took the opposite approach—to break our database into as many pieces as we could, search each at the same time, and then deal with the coordination needed to return a single result to a searcher. We are finding this works very well for

communication; 23 *compute* nodes, each with 4 Gigabytes of memory and two Xeon CPUs; and a Cisco gigabit switch that enables the nodes to communicate.

Beyond text searching, where we've been able to do well over 100 searches per second using open-source tools, we have been doing much of our FRBR work on the cluster.

FRBR stands for "Functional Requirements for Bibliographic Records," an IFLA report that, among other things, describes an approach for grouping records into 'works.' For example, searching a work-based database for Shakespeare's *Hamlet* might retrieve a whole set of records, each of which is a different edition (or expression in FRBR terms). Our work has been primarily in identifying those works quickly and reliably and understanding the relationship that



records that each processor needs to cope with, greatly easing many tasks.

library authority files have with FRBR sets.

The processors in the cluster are 'hyperthreaded,' a technique used by Intel on its Xeon chips, so that logically each of the physical chips appears to be two logical processors, giving us a total of 92 logical CPUs on the 23 compute nodes. Typically we use three of those logical processors, reserving one for communication with other nodes. Breaking WorldCat up into 69 pieces (three per node) results in less than a million records that each processor needs to cope with, greatly easing many tasks. Our 'FRBRization' of WorldCat can now be done in less than an hour, and searches can be completed in milliseconds.

So, why isn't all bibliographic processing done on such machines? There are at least two main reasons. The first is that this sort of configuration is relatively

new, and relatively few sites deal with the tens of millions of records that make this level of parallelism important. Another issue is that Beowulf clusters are designed more for speed than reliability. Google has pioneered technology in this area with redundant file systems and data centers, but since they do not use standard cluster software, this is not available in the open-source Linux cluster distribution we have been using (called ROCKS). Reliability is less of an issue in a research context, but our experience (once we got past some 'teething' problems) is that the cluster has been very stable. As part of our research we are looking at ways that we can duplicate our data across nodes. With fairly simple fail-over and load balancing software, these clusters could become reliable enough for many services in the future. ■