



THE WHITE ROSE GRID

e-Science Centre

High Throughput Computing

Parallel File Serving...Made Easy



“RepliCator from eXludus uses patented technology in the network stack to improve the achievable network bandwidth, thus reducing the network contention and greatly improving the ability of the file servers in a cluster to serve data to the cluster nodes.”

Introduction

Here we describe results of experiments on a method to improve the efficiency of high throughput cluster computing. This method utilises the RepliCator software from eXludus Technologies to efficiently transfer data sets from a file server to the cluster nodes. The results demonstrate that this can result in a dramatic increase in overall utilisation efficiency of the cluster.

As the data dependency of compute jobs becomes ever greater then contention for file serving and/or network resources often occurs which cause the jobs to wait on the required data. This contention is exacerbated in cases where arrays of jobs are started requiring access to the same data, for example when a parameter search is conducted using different initial job parameters but the same data set. This contention can seriously affect the overall throughput possible in a cluster.

RepliCator

RepliCator from eXludus uses patented technology in the network stack to improve the achievable network bandwidth, thus reducing the network contention and greatly improving the ability of the file servers in a cluster to serve data to the cluster nodes. By reducing this network contention and allowing data serving efficiency to be improved data-dependent jobs can access the required data faster and the real time job execution times are reduced. This either results in users getting results in a more timely matter or the ability to schedule more jobs on the cluster sooner, resulting in higher overall cluster efficiency. RepliCator also allows for data pre-staging to take place to the cluster thus completely removing data latency to jobs seen in traditional on-demand data access schemes.

Experimental Set-up

An experiment was run to benchmark the performance with and without the RepliCator software to determine the real world performance of RepliCator.

The system chosen for the experiment was a Linux Beowulf system which is part of the White Rose Grid. This machine is hosted at York.

The system comprises 40 single processor nodes, each comprising a Pentium III processor, 768MB local memory and disk, and a 100Mb/s network bandwidth to each node. This is a relatively old system, but was chosen partly as it reflects a system similar to what could be expected on a campus grid. Due to existing user load on the system 10 cluster nodes were used for the experiment.

The file server used was the head node for the Nevada system, a dual 700Mhz processor Linux box with SCSI disks and gigabit bandwidth to the Beowulf's network switch.

The job chosen for the test was a BLAST. This is a bioinformatics program of a type widely used at York and elsewhere. The data set used was HomoSap sets 00-13, of 568MB total size. The baseline for comparison was the identical jobs started simultaneously on the 10 selected cluster nodes, accessing the data via NFS from the file server. This was compared to the same jobs utilising RepliCator, which in turn accessed the NFS-mounted directory containing the data set.

Results

Two metrics were measured: the aggregate network bandwidth and the execution time of the compute jobs.

Aggregate Network Bandwidth

Caveat: There was no direct access to the file server, so RepliCator tests were achieved by using an NFS mounted directory to broadcast data. Under normal circumstances higher bandwidth would have been obtained.



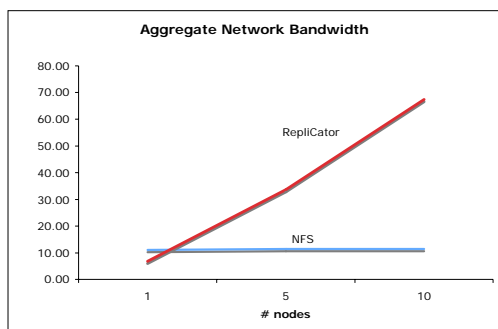


figure 1

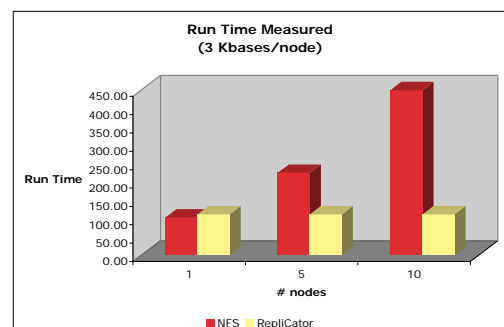


figure 2

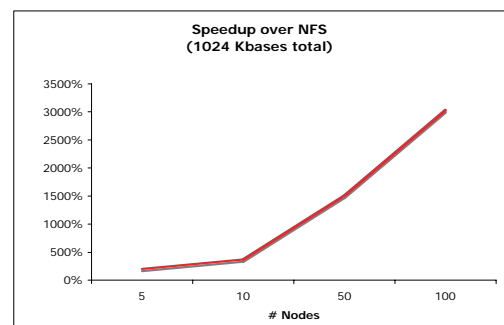


figure 3

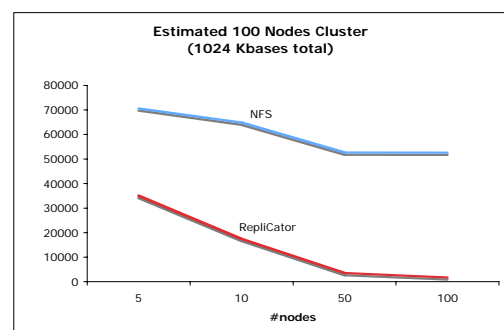


figure 4

In spite of the use of NFS to broadcast data the aggregate network bandwidth was increased by 589% using 10 nodes (figure 1).

BLAST Test

Caveat: Due to limited local memory on the cluster nodes only a subset of the full HomoSap data set could be used.

A speed-up of 4.4 times over NFS was achieved with only 10 nodes (figure 2).

Scaled results

The node count used in the tests was rather modest, and here we present estimates indicating how RepliCator performance would scale to a larger number of nodes.

The methodology for this estimation is as follows:

- Divide results in the first search per node per job and all other searches (the first search takes longer to load the initial database into memory).
- Measure the average 1st and other searches on increasing numbers of nodes.
- Determine equations for the 1st and other searches per number of nodes.
- Correlate equations with existing results.
- Determine total time for the target number of searches.

Based on the benchmark results it is anticipated that a BLAST search of average length (48 hours run time on a single CPU with a 500MB data set) on 100 nodes of similar performance would result in speed ups of 30 times over NFS (fig. 3). Scaled over the full available system speed up of 9 times would be expected.

Economics

Cluster computing offers a good price:performance ratio, especially when coupled with relatively modest file serving. However, the graphs above reveal that for data-dependent jobs the effective utilisation of a cluster can be reduced dramatically, and this must be accounted for when examining the price:performance ratio. For a cluster similar to the one described above running BLAST jobs the peak efficiency would be reduced

by a factor of 4 if all jobs running on the system were similar BLAST jobs, which may significantly affect the achievable price:performance ratio.

There are strategies other than the use of RepliCator that might be used to increase data transfer efficiency for data-dependent jobs by increasing the file serving bandwidth. This requires a combination of higher bandwidth disk arrays, file servers, and network infrastructure. This will generally incur additional system costs, again adversely affecting the price:performance ratio of the cluster.

To the contrary RepliCator is a software only solution, hence less expensive. And it was designed specifically with the intent to yield a positive ROI (Return-on-Investment). That is, for every pound invested in RepliCator users can expect a cost reduction in infrastructure and increased productivity worth at least three pounds. Economic reality is a cornerstone of RepliCator's design philosophy.

Given the degree of data dependency of the job profile run on a cluster and the details of the file serving and network infrastructure various different strategies may be appropriate on a price:performance argument. RepliCator is one of these possible strategies and may be worthwhile in price:performance terms compared to new file serving or network infrastructure.

Conclusion

Cluster computing using commodity components offers a good price:performance ratio when jobs have low data dependency, but we have shown that there is a potential data transfer bottleneck for jobs which are data-dependent, but that this bottleneck can be reduced in an economic way by the use of RepliCator software from eXludus.

Further Information

Contact:
Aaron Turner, University of York
(email: aaron@cs.york.ac.uk)
or
info@exludus.com

The Project Web site:
<http://www.exludus.com>
info@exludus.com

