DSEARCH: sensitive database searching using distributed computing

Keane T.M.[1] and Naughton T.J.[1*]

[1]Department of Computer Science, National University of Ireland, Maynooth, Ireland
Email: tom.naughton@may.ie

Keywords: database search, sequence analysis, sensitive, distributed computing, Java

[*] To whom correspondence should be addressed

Abstract

Summary: We present a distributed and fully cross-platform database search program that allows the user to utilise the idle clock cycles of machines to perform large searches using the most sensitive algorithms. For those in an academic or corporate environment with hundreds of idle desktop machines, DSEARCH can deliver a 'free' database search supercomputer.
Availability: The software is publicly available under the GNU general public licence from http://www.cs.may.ie/distributed
Contact: tom.naughton@may.ie
Supplementary Information: Full documentation and a user manual is available from http://www.cs.may.ie/distributed

1 Introduction

Database searching for similar sequences is one of the fundamental tasks in bioinformatics. The two most rigorous search algorithms are the Needleman-Wunsch (Needleman and Wunsch, 1970) and Smith-Waterman (Smith and Waterman, 1981) algorithms. However for large databases it is not feasible to perform searches using these algorithms. Many heuristic search algorithms have been developed but these reduce the sensitivity of a search and can fail to detect certain matches. When given the choice, most biologists would prefer to use the more rigorous algorithms for their searches.

One way to significantly reduce the runtime of long searches is to parallelise the search process across multiple processors. Many approaches to parallelising database searching have been investigated because database searching is both computationally intensive and easily parallelised. However the overriding problem with many of these programs is that specialised parallel hardware and software is often required, making these programs either prohibitively expensive or simply too complicated to set up. DSEARCH is a fully cross-platform parallel database search program that does not require any specialised parallel hardware or software.

2 Description

DSEARCH operates in a master-slave environment and the search is parallelised by splitting the database into fixed sized units that are subsequently searched on the donor machines. DSEARCH is one of a number of distributed applications that runs on our general purpose distributed computing platform which provides the user with a remote interface to monitor the progress of the application as their search executes (Keane, 2004). The parallel granularity is dynamically controlled during each search to match the processing abilities of the current set of donor machines (Keane, 2004). The user edits a straightforward configuration file to tailor their computation and chooses one of the built-in search algorithms (Smith and Waterman, 1981; Needleman and Wunsch, 1970; Crochemore *et al.*, 2003). The inputs to the program are a FASTA database file, a FASTA query sequences file, a scoring scheme, and a configuration file.

The generality of DSEARCH is demonstrated by the fact that DSEARCH, written in Java, can run on virtually any architecture and operating system simultaneously while only using the spare clock cycles of donor machines. No specialised computer hardware or software is required, and no expense is incurred if idle computing resources are harnessed. This would not be as straightforward for a parallel application written in a native language because the application would have to be compiled for each particular architecture and operating system. We have demonstrated the ease of use and platform heterogeneity of

DSEARCH with experiments that utilise the spare computing resources of several architectures and operating systems simultaneously. DSEARCH's main features are:

- Full cross-platform compatibility
- Ability to run several database searches simultaneously
- Remote real-time progress updates via the remote interface
- Full support for arbitrary donor machine failure

3 Deployment and Performance

We have deployed DSEARCH in our university by running it as a low priority background service in a number of computing laboratories, consisting of approximately 200 desktop PC's of various modest specifications (Pentium II's up to Pentium IV's running assorted versions of Windows and Linux OSs) and on every node of an IBM Linux cluster (32 Dual PIII 1 GHz nodes) with all machines connecting via a 10 Mbit/s network to a single server (Pentium III 500 MHz). We tested the performance of DSEARCH with SSEARCH (Pearson and Lipman, 1988) by running DSEARCH on a single processor and searching a 13 Mbyte EST database with a 173 Kbyte set of query sequences. The comparative runtimes were 1207 minutes and 493 minutes, respectively. DSEARCH's performance reduction is overcome by its greater cross-platform compatibility. Figure 1 shows how DSEARCH scales with increasing numbers of processors. In our tests, DSEARCH reached a speed of 511 million cells per second on a network of 83 semi-idle workstations (Pentium III 900 MHz).

Acknowledgements

This research has been funded by the Embark Initiative from the Irish Research Council for Science, Engineering and Technology: funded by the National Development Plan. We thank Chris Creevey and Andrew Page for their assistance. DSEARCH uses the NeoBio Java library (http://neobio.sourceforge.net) to perform all comparisons.

References

Crochemore, M., Landau, G., and Ziv-Ukelson, M. (2003) A Subquadratic Sequence Alignment Algorithm for Unrestricted Scoring Matrices, *SIAM Journal of Computing* 32 (6), 1654-1673

Keane, T.M. (2004) A Programmable Heterogeneous Distributed Computing Platform with Bioinformatics Applications, *M.Sc. Thesis*, Department of Computer Science, National University of Ireland, Maynooth

Needleman, S.B and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequences of two proteins, *Journal of Molecular Biology*, 48, 443-453

Pearson, W.R. and D.J. Lipman (1988) Improved Tools for Biological Sequence Analysis, *Proceedings of the National Academy of Sciences*, 85, 2444-2448

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences, *Journal of Molecular Biology*, 147, 195-197

Figure Captions

Figure 1. Speedup achieved by DSEARCH over a network of 83 semi-idle machines

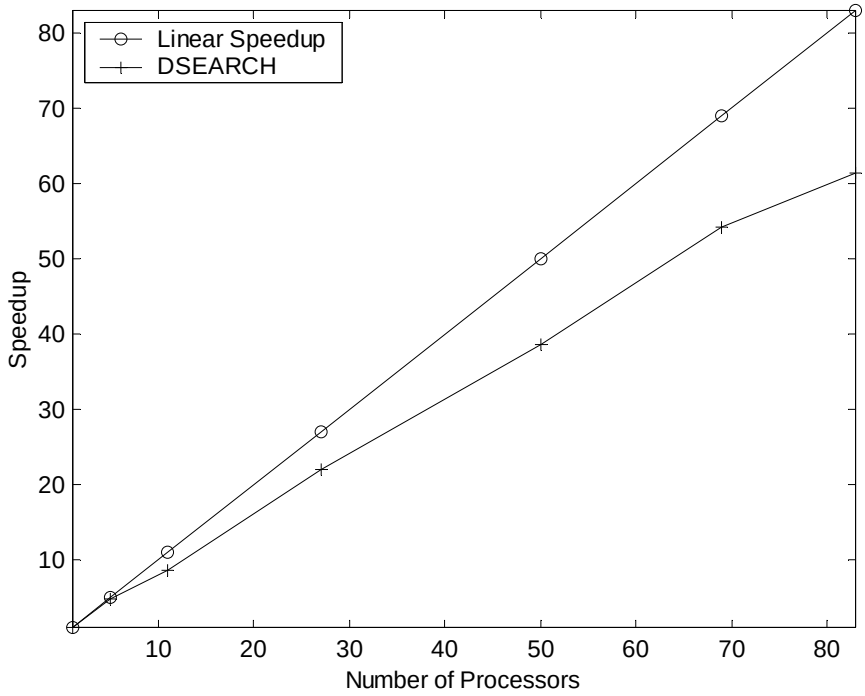
Linear Speedup
DSEARCH
Speedup
Number of Processors
10
20
30
40
50
60
70
80

Fig. 1