Available online at www.sciencedirect.com
ScienceDirect
Future Generation Computer Systems 23 (2007) 580–586
FGCS FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

Experiences with GeneRecon on MiG

Thomas Mailund[a], Christian N.S. Pedersen[a,b], Jonas Bardino[c,*], Brian Vinter[c], Henrik H. Karlsen[c]

[a] *Bioinformatics Research Center, University of Aarhus, Denmark*
[b] *Department of Computer Science, University of Aarhus, Denmark*
[c] *Department of Computer Science, University of Copenhagen, Denmark*

Received 28 February 2006; received in revised form 14 July 2006; accepted 16 September 2006
Available online 14 November 2006

Abstract

We report on our experiences so far with running a bioinformatics simulation study on a newly developed Grid architecture. We briefly describe the bioinformatics application – an association mapping algorithm for both locating disease loci and separating cases into those diseased due to genetic factors and those diseased solely due to environment factors – and describe the Grid architecture and how the application is set up to run on the Grid.

Keywords: Minimum intrusion Grid; Association mapping; Simulation study

1. Introduction

Locating the genetic factors behind common inheritable diseases is a highly relevant problem in medicine and bioinformatics [3]. The complexity of the problem, however, implies that finding biologically sound solutions requires algorithms of high time complexity, which makes extensive experimentation difficult. This paper describes the work done, and results from, utilising Grid computing [1] for large scale experiments with an algorithm, GeneRecon, for detecting genetic mutation clusters.

GeneRecon is a tool for analysis of population genetic data from case/control studies. The analysis works on genetic data which is collected from patients with a specific disease and a control group without this disease, and tries to locate genes that are increasing the risk of disease. The analysis, which is based on a *Markov-chain Monte Carlo* (MCMC) method, is extremely CPU-demanding, since it searches through a very large parameter-space. The presented work seeks to evaluate a method that both significantly reduces the running time of the method and additionally can determine which patients have a given disease because of a genetic markup, rather than another cause, e.g. environmental effects. The work includes thousands of GeneRecon runs – an infeasible task without access to a large number of processors.

Minimum intrusion Grid (MiG) [10] is a new platform for Grid computing which is driven by a stand-alone approach to Grid, rather than integration with existing systems. The goal of the MiG project is to provide a Grid infrastructure where the requirements on users and resources alike, to join Grid, is as small as possible. While striving for minimum intrusion, MiG will still seek to provide a feature-rich and dependable Grid solution.

2. Association mapping and mutation clusters

For diseases with a genetic component, carriers of high risk alleles are often descendants of one or a few founders, in whose genome the allele appeared by mutation. Recent shared ancestry of the carriers in the disease position results in a higher homogeneity, among the carriers, in the genomic region around the disease affecting gene, than in the population at large. This, in turn, causes markers – points of genetic variation – around the disease gene to be more associated with each other — in a statistical sense, a phenomenon known as *linkage disequilibrium*, or LD. The use of LD for disease gene mapping has in recent years received much attention, and is believed by

* Corresponding author.
E-mail addresses: mailund@birc.au.dk (T. Mailund), cstorm@birc.au.dk (C.N.S. Pedersen), bardino@diku.dk (J. Bardino), vinter@diku.dk (B. Vinter), karlsen@diku.dk (H.H. Karlsen).

0167-739X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2006.09.003

many to hold great promise in the mapping of common complex diseases [3].

The *Common Disease, Common Variant* (CD/CV) hypothesis [9], fundamental to association mapping and based on the history of the human population, states that most common diseases with a genetic component are affected by a few alleles with a (relatively) high frequency in the population but with a (relatively) low penetrance. If it holds true, we would expect that, in a case/control study, several unaffected individuals would be carriers of the increased risk mutation – but not diseased, due to the low penetrance – and perhaps, at the same time, only a few of the diseased individuals would carry the increased risk mutation while the remaining affected individuals have the disease due to environmental causes. In such a study, identifying the disease carriers among the affected individuals would greatly benefit the search for the disease gene once a candidate genome region has been found.

2.1. The mapping method based on mutation and null clusters

To approach this problem, we have developed an MCMC algorithm[1] based on Morris et al.'s shattered coalescent method [8], but incorporating ideas from Liu et al. [4] and Molitor et al. [7], where we separated the affected individuals into *mutation clusters* – where affected individuals in the same cluster are assumed to be descendants of a common founder – and a *null cluster* — for individuals affected due to environmental factors and not genetic factors. By sampling, during the run of the MCMC, the distribution of affected individuals among the mutation and null clusters, we hope to be able to infer which individuals carry an increased risk of mutation and which are affected solely due to environmental factors, while at the same time locating the locus of the disease affecting gene. Introducing the mutation and null cluster can potentially also greatly reduce the running time of the algorithm, as we only explicitly model the genealogy of the cases in the mutation clusters, not of all cases, thus reducing the search space significantly.

While the methods we have based our new technique upon have proved themselves highly efficient in locating the disease locus on case/control data, they are unfortunately also very CPU-demanding and require several hours – or days – for a successful computation with a few hundreds of cases and controls, with a few tens of markers. This makes computer clusters or computer Grid architectures essential for validating new mapping methods such as ours, since such validation requires analysis of a large number of (simulated) data sets to get useful statistics about the performance of the method.

2.2. The experimental setup

We have simulated haplotype data sets using the CoaSim simulator [5] under varying recombination rate ($\rho = 40$ and $\rho = 400$, roughly corresponding to 0.1 cM and 1 cM), with varying marker densities (20 markers on the region, 40 markers on the region, and a twice as wide region with 40 markers, 20 of which are in the middle region – containing the disease marker – and 10 on each side), and varying disease models (m/w where m is the fraction of affected individuals among the mutants and w is the fraction of affected individuals among the wild types).

For each data set, four chains of GeneRecon was run for cluster sizes 100 (all affected individuals), 75, and 50. The analysis of the results was conducted using a set of *R* scripts. In calculating the error of disease locus inference, we have used a simple measure of the distance from the point with maximum posterior value to the true disease locus. This does not catch the confidence of the inference, but gives a simple measure of error that lets us compare the accuracy for various cluster sizes.

3. Minimum intrusion Grid

MiG (http://www.migrid.org) is a Grid middleware model and implementation designed with previous Grid middleware experiences in mind. In MiG central issues such as security, scalability, privacy, strong scheduling and fault tolerance are included by design. Other Grid middlewares tend to suffer from problems with one or more of those issues.

The MiG model seeks to be non-intrusive in the sense that both users and resources should be able to join the Grid with a minimal initial effort and with little or no maintenance required. One way to obtain these features is keeping the required software installation to a functional minimum, e.g. the software that is required to run MiG includes only 'need to have' features, while any 'nice to have' features are completely optional. This design philosophy has been used, and reiterated, so stringently that in fact neither users nor resources are required to install *any* software that is MiG specific.

Another area where MiG strives to be non-intrusive is the communication with users and resources. Users in general and resources in particular cannot be expected to have unrestricted network access in either direction. Therefore the MiG design enforces that all communication with resources and users should use only the most common protocols known to be allowed even with severely restricted networking configurations. Furthermore resources should neither be forced to provide any special privileges to the MiG job execution user(s) nor run any additional network-listening daemons.

Fig. 1 depicts the way MiG separates the users and resources with a physical Grid layer, which users and resources securely access through one of a number of MiG servers. The MiG model resembles a classic client–server model where clients are represented by either users or resources. The servers are represented by the Grid itself, which in the case of MiG is a set of actual computers, not simply a protocol for communicating between computers. All clients can contact the Grid in order to either upload or download files. User clients can additionally contact the server in order to do job management and file manipulation while resource clients can additionally request a job to execute. As much of the actual functionality as possible is located at the MiG servers, where it is maintained by the MiG developers. Thus, in addition to minimising the user

[1] The new MCMC algorithm has been implemented in the association mapping tool *GeneRecon* (http://www.daimi.au.dk/~mailund/GeneRecon) [6].