European Journal of Human Genetics (2007) 15, 694–702
© 2007 Nature Publishing Group All rights reserved 1018-4813/07 $30.00
www.nature.com/ejhg

ARTICLE

The use of grid computing to drive data-intensive genetic research

Jorge Andrade[1], Malin Andersen[1,2], Anna Sillén[3], Caroline Graff[3] and Jacob Odeberg*[,1,2,4]

[1]*Department of Biotechnology, AlbaNova University Center, Royal Institute of Technology (KTH), Stockholm, Sweden;* [2]*Department of Medicine, Atherosclerosis Research Unit, Gustaf V Research Institute, Karolinska Institutet, Karolinska University Hospital, Stockholm, Sweden;* [3]*Karolinska Institutet Dainippon Sumitomo Pharma Alzheimer Center (KASPAC) Department of Neurobiology, Care Science and Society, Karolinska Institutet, Novum, Huddinge, Sweden;* [4]*Karolinska Biomics Centre, Karolinska University Hospital, Stockholm, Sweden*

In genetics, with increasing data sizes and more advanced algorithms for mining complex data, a point is reached where increased computational capacity or alternative solutions becomes unavoidable. Most contemporary methods for linkage analysis are based on the Lander-Green hidden Markov model (HMM), which scales exponentially with the number of pedigree members. In whole genome linkage analysis, genotype simulations become prohibitively time consuming to perform on single computers. We have developed 'Grid-Allegro', a Grid aware implementation of the Allegro software, by which several thousands of genotype simulations can be performed in parallel in short time. With temporary installations of the Allegro executable and datasets on remote nodes at submission, the need of predefined Grid run-time environments is circumvented. We evaluated the performance, efficiency and scalability of this implementation in a genome scan on Swedish multiplex Alzheimer's disease families. We demonstrate that 'Grid-Allegro' allows for the full exploitation of the features available in Allegro for genome-wide linkage. The implementation of existing bioinformatics applications on Grids (Distributed Computing) represent a cost-effective alternative for addressing highly resource-demanding and data-intensive bioinformatics task, compared to acquiring and setting up clusters of computational hardware in house (Parallel Computing), a resource not available to most geneticists today.
European Journal of Human Genetics (2007) **15,** 694–702. doi:10.1038/sj.ejhg.5201815; published online 21 March 2007

Keywords: grid; bioinformatics; genome-wide; linkage analysis; genotype simulation

Introduction

With an increasing amount and complexity of data in genomics and genetics that is generated by today's high-throughput technologies, the demand for computational power has become an issue that sometimes defines the practical limit for the analysis rather than the size of the study. Several statistical and analytical algorithms have been developed that become prohibitively time consuming when applied on a genome-wide scale. One example is the analysis of computer-based genotype simulations given phenotypes with and without a defined disease model, a commonly used approach in linkage analysis of complex traits.[1,2]

Allegro v1.1[3] is one computer program for linkage analysis that is free for non-commercial use. It uses a hidden Markov model,[4] in which time and computer memory cost grow exponentially with the pedigree size and linearly with the number of markers. Allegro represents an improvement of the computational algorithms used in Genehunter,[5] and has much of the functionality of

*Correspondence: Dr J Odeberg, Department of Biotechnology, Royal Institute of Technology, Alba Nova University Center, SE – 10691 Stockholm, Sweden. Tel: +46 85 537 8332; Fax: +46 85 537 8481;
E-mail: Jacob.odeberg@biotech.kth.se
Received 4 October 2006; revised 10 January 2007; accepted 14 February 2007; published online 21 March 2007

Genehunter. It specifically calculates single- and multipoint parametric LOD scores, non-parametric linkage (NPL) scores and allele-sharing LOD scores. Haplotype reconstruction and genotype simulations in the absence of linkage or assuming linkage are also part of Allegro's features. Allegro can be used to estimate the power to detect linkage in a sample set, to estimate global *P*-values associated with given LOD scores, to explore linear or exponential methods of calculating *P*-values, to compare parametric and non-parametric methods or, in general, to give approximate answers to many statistical problems.

Computer simulations of genotypes using Allegro to evaluate the significance of genome-wide linkage results have been used previously;[6–9] however, execution time increases rapidly with the number of genotypes and pedigrees. Despite the fact that Allegro is considerably faster than Genehunter, sequential genotype simulations of large pedigree sizes could consume weeks or even months in run-time in a state-of-the art single computer. One alternative is to distribute such time-consuming tasks subdivided into a set of smaller jobs executed on several computers in parallel. Since Allegro runs are independent of each other (not data-dependent), Allegro is an ideal application for a distributed execution.

The Grid paradigm[10] offers CPU and data-handling capabilities that far exceeds what can be attained within most research institutions and budgets at the same time as it allows for parallel execution of existing algorithms and software without re-codification. Grid computing is often confused with cluster computing; however, a key difference is that a cluster is a single set of nodes sitting in one location, whereas a Grid is composed of many clusters and other kinds of resources including computers, supercomputers, storage systems, data sources, and specialized devices that are geographically distributed and owned by different virtual organizations,[11] to which users other than the owners can be granted access.

On the basis of Grid technology,[12] we have developed an application of Allegro by which several thousands of simulations can be performed in parallel in a distributed dynamic Grid environment, providing theoretically unlimited simulation power; however still with the current Allegro restriction of limited family size. The grid implementation enables, among other things, accurate evaluation of the significance of the results from analyses of genome-wide human genetic linkage data. To the geneticist without direct access to expensive resources in-house such as dedicated clusters or computer farms, this represents a hitherto unexploited resource.

Materials and methods

Architecture overview

To develop a Grid aware implementation of the Allegro software, we joined the Swegrid virtual organization. The Swedish Swegrid (http://www.swegrid.se/) is a Globus (http://www.globus.org/) based Swedish national computational resource, consisting of 600 computers in six clusters at six different sites across Sweden. Swegrid is a member of the NorduGrid[13] virtual organization, initiated in January 2001 by several Nordic universities and research centres and at present it puts together more than 2500 processors. For the current implementation, we were granted access to about 600 nodes (CPU) through the different clusters.

The first step to get access to Grid facilities is to download and install the client package from http://ftp.nordugrid.org/download/. Binary distributions are available for several GNU/Linux flavours. A full client installation (<10 MB) was performed in our local Grid proxy server (Master node). The Grid user has to hold an electronic signed certificate expended by an appropriate Certificate Authority, which ensures unique authentication. After the NorduGrid client installation in our local Grid proxy server, the task of creating and renewing a Grid proxy session was automatically scheduled in the local master node, using Linux shell scripts. A schematic system specification of our environment setup is described in Figure 1.

Implementation

To implement the Grid-Allegro, two programs written in Perl were developed. *Gridallegrosteep1.pl* runs locally in the master node and its task is to prepare the input files that will be submitted to the Grid environment given the specified input parameters. In the case of genotype simulations, for instance, it will require the specification of family structure, a phenotype, a disease model (in parametric linkage analysis), a choice of the type of analysis to be performed (multipoint or single point), and optional choices as to include *P*-values to be listed in the output files (LOD, NPL or Z1r scores). *Gridallegrosteep1.pl* also creates a specific number of Grid jobs using the Globus RSL (Resource Specification Language). After the single atomistic jobs are defined and created, the Grid-broker *gridallegrosteep2.pl* handles the distribution of the jobs to the remote workers, constantly evaluates the status of each job, manages re-submission in case of failure or excessive delay in the Grid queue system, and finally collects the output results of the calculations.

A detailed description of the 'Grid-Allegro' workflow, implementation, environment setup and configuration is available at http://kthgridproxy.biotech.kth.se/grid-allegro/index.html.

Results

Simulation of genotypes in a study of Alzheimers' disease

As a 'proof of concept', this Grid implementation was used in the simulation analyses of an expanded study on Swedish families with Alzheimer's disease (AD).[14] The aim of the study was to identify novel genes involved in the AD

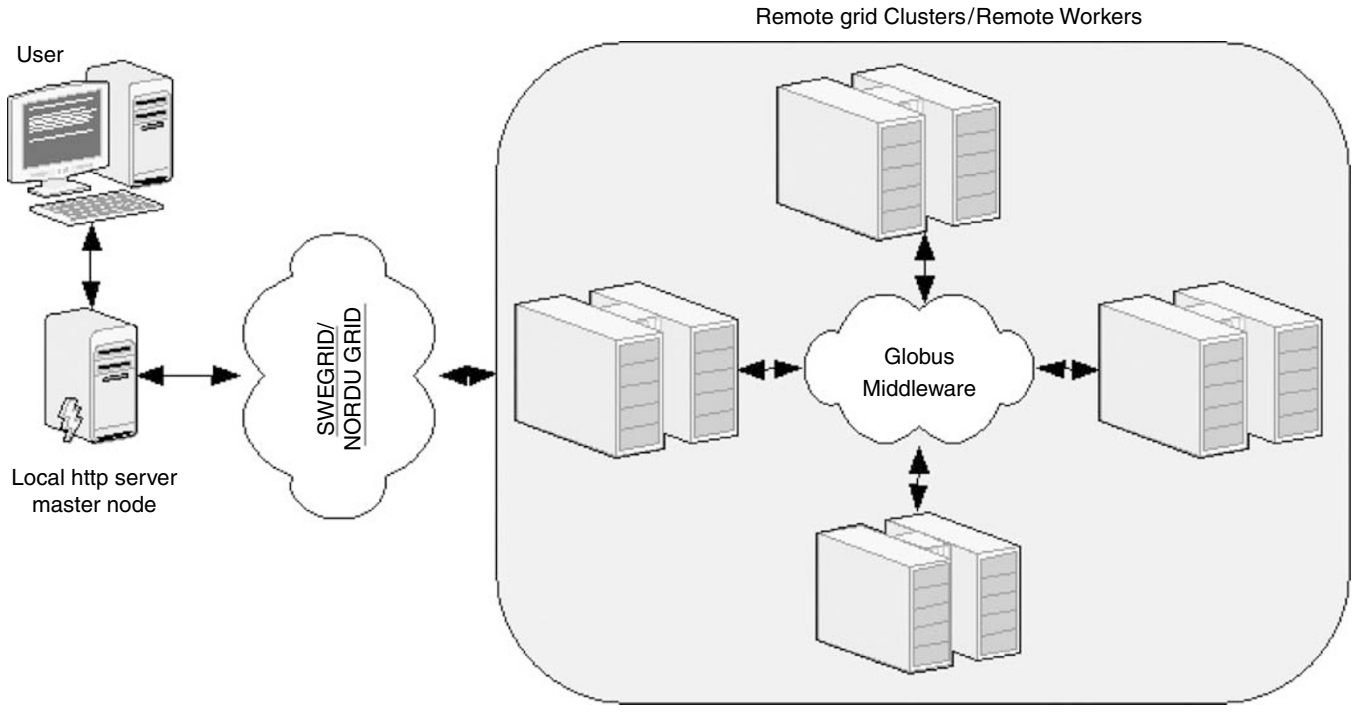

Figure 1 Grid-Allegro environment setup. The figure shows a schematic system specification of the environment configuration. A full installation of the Grid stand-alone client package in the local Master node enables communication with the remote Grid workers through the Swegrid/Nordugrid middleware.

Table 1 Expected run time for 1000 simulations with seven different input data sizes using Allegro v1.1 in a serial execution

Sample set identifier	*Set size no. of families*	*Largest pedigree size in bits*	*Real run time in minutes for 1 simulation in a single processor chromosome 1*	*Projected run time in days for 1000 simulation in a single processor in one chromosome*
1	10	20	18	12.5
2	14	16	20	13.8
3	18	20	21	14.5
4	24	20	28	19.4
5	45	20	31	21.5
6	63	23	42	29.1
7	109	23	86	59.7

The expected run time needed to perform 1000 genotype simulations using Allegro v1.1 with seven different sets of input pedigree data sizes and six different models for each simulation (mpt exp pairs power, 0.5; spt exp pairs power, 0.5; mpt exp all power, 0.5; spt exp all power, 0.5; mpt par het and spt par het). The argument 'mpt' specifies that multipoint analysis is to be performed. Its alternative, 'spt', specifies single point analysis. Run times are projected based on the real run-time measure needed for a single computer (Linux-based, CPU of 2 GHz speed/512 Mb RAM) to perform one single simulation of one chromosome (Chromosome 1). This time is then escalated to project an approximate run time for 1000 simulations.

pathogenesis by performing a genome-wide non-parametric linkage analysis on AD families from the relatively genetically homogeneous Swedish population. The study was performed on seven different pedigree sets identified with numbers 1–7 (Table 1). Set 7 constitutes the total family material and contains 109 families, made up of 470 family members genotyped for 1289 microsatellite markers. Set 7 was further subdivided into six substrata based on phenotypic similarity and/or genotypes of the known Alzheimer's disease susceptibility gene *APOE*. Sets 1, 2, 3, 4, 5 and 6 correspond to 10, 14, 18, 24, 45 and 63 families of the total 109 families respectively.

Simulated genotypes were created using the 'SIMULATE' option of the Allegro program, using the same marker map, allele frequencies and pedigree structures as for the authentic linkage analyses. A thousand serial simulations under the null hypothesis of no linkage across the whole genome were performed for each chromosome, except for

the sex chromosomes. Yield was set to 93%, which corresponds to the actual genotyping success rate obtained for the genotyping in the genome scan.

Grid-Allegro was used to evaluate the statistical significance of the linkage data (Global *P*-value of 0.001 is highly significant[15]), under the null hypothesis of no linkage. The three highest obtained LOD scores for each simulation data sets were used as thresholds to estimate their global significance, that is the number of times that the threshold value was obtained in the simulations by chance in the absence of linkage.

In the sets of analyzed families there were large differences in pedigree sizes, and to compensate for this and not making the linkage calculations biased due to some large family's individual linkage scores, we used the variable 'power: 0.5' as the family weighting scheme, as recommended in the Allegro manual. The scoring function that is the best to use can be discussed, depending on the disease model.[16] We performed the linkage analysis using both of the scoring functions 'pairs' and 'all', since the families under investigation show both apparent dominant disease model, unclear disease model and a mixture of models.

The major part of the 109 analyzed AD families are of Swedish origin, and vary in size from small pedigrees with only two affected siblings to large family trees with at least six affected individuals and many genotyped siblings with unknown disease status. The largest pedigree in our study comprises 20 genotyped persons, resulting in a bit size of 36. Existing multipoint linkage analysis programs based on Lander-Green HMM like Genehunter,[5] Allegro[3] and Merlin[17] can handle arbitrarily many markers, but are currently limited to ~25-bit pedigrees, irrespective of computational power available. The bit size of a pedigree is $2n\text{-}f\text{-}g$, where n is the number of non-founders, f the number of founders and g the number of ungenotyped founders. Therefore, in the present study the largest family and two other families were cut in size in the calculations.

Serial projected run time

The excessive run time expected for a single computer to perform a complete genome-wide linkage analysis was illustrated using a standard serial execution of Allegro v1.1 in a single processor (CPU of 2 GHz speed/512 Mb RAM). The real time needed to perform a single simulation for seven different sets of input pedigree data sizes and six different models for each simulation was measured (Table 1), and then used to project the expected run time for 1000 simulations. This measure shows that even for pedigrees of small to moderate sizes, the projected time needed to perform thousands of simulations constitute a technological bottleneck using ordinary computers. This is a result of the exponential increase in run time of Allegro for a linear increase in the number of members of a pedigree. As shown in Table 2, on this scale the number of markers has no significant impact on the total runtime.

Table 2 Projected run time for a complete genome wide linkage analysis, based on real run-time measures using Allegro v1.1

Chromosome	*No. of markers*	*Real run time in minutes for 1 simulation in a single processor*	*Projected run time in days for 1000 simulations in a single processor*
Chr 1:	88	86	59.7
Chr 2:	86	85	59.0
Chr 3:	76	80	55.5
Chr 4:	75	80	55.5
Chr 5:	74	79	54.8
Chr 6:	64	70	48.6
Chr 7:	78	81	56.2
Chr 8:	59	78	54.1
Chr 9:	57	78	54.1
Chr 10:	56	78	54.1
Chr 11:	58	79	54.8
Chr12:	59	80	55.5
Chr 13:	38	76	52.7
Chr 14:	51	80	55.5
Chr 15:	50	79	54.8
Chr 16:	39	76	52.7
Chr 17:	50	79	54.8
Chr 18:	40	76	52.7
Chr 19:	47	77	53.4
Chr 20:	38	76	52.7
Chr 21:	21	75	52.0
Chr 22:	24	74	51.3
		Total projected time in days/years:	1195.8/3.2

Projected Allegro v1.1 run time needed to perform 1000 simulations for each chromosome in an up-to-date single processor (Linux-based, CPU of 2 GHz speed/512 Mb RAM), using the largest input data size corresponding to 109 families with largest pedigree size of 23 bits. Real timings for one single simulation were performed and then multiplied by 1000 to obtain the expected run -time for a complete genome wide linkage analysis. The exact run-time can vary in different computers, depending on hardware configurations.

In the application described here (Alzheimer's disease), a minimum of 22 000 simulations in total (1000 for each chromosome, the X chromosome was not included in the simulation analysis) were required to achieve an estimation for global *P*-value of 0.001. The projected execution time for the accumulated total material of this study using Allegro v 1.1 becomes approximately 3.2 years on a single up-to-date computer (Table 2).

Grid-allegro performance

To define a model of performance, theoretical speed-up on P nodes was calculated to evaluate the expected improvement in run time achieved by the 'Gridification' of Allegro v 1.1. The theoretical speed-up can be calculated as

$$S_p = \frac{T_1^S}{T_p}$$

where T_1^S is the expected calculated sequential run time to perform the simulations. For the 22 000 simulations using

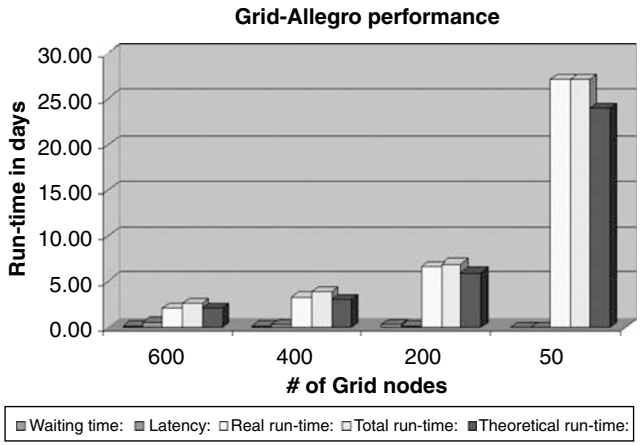

Figure 2 Grid-Allegro performance in Swegrid and predicted run-time. The figure shows the real Grid-Allegro run-time in days needed to perform 22 000 simulations on data from a complete genome-wide linkage analysis, using the largest available input data set (sample set 7 with 109 families and with the largest pedigree size of 23 bits). Nearly linear increase in speed-up is achieved as the number of Grid nodes is increased. The exact real Grid run time can vary depending of hardware configuration in different Grid nodes. Waiting time can vary depending on Grid work load conditions. Theoretical run-time was calculated by dividing the total projected serial run-time (Table 2) by the number of Grid workers.

the largest available input data set (109 families with the largest pedigree size of 23 bits), T_1^S is 3.2 years, (see Table 2). T_p is the Grid-Allegro execution time for the same data, clustered in $P = 50$, 200, 400 and 600 Grid processors in Swegrid. Because we are submitting jobs of equal sizes to the Grid workers, one may expect nearly linear increase in speed-up as the number of computation nodes increases. Figure 2 shows the theoretical and real execution times for identically clustering the same data set on remote Grid nodes of different sizes.

The total Grid execution time for a particular task is defined by the total Grid latency (the time needed to submit a complete set of jobs (Input), and collect a complete set of results (Output)), the accumulated waiting time in case a job or a set of jobs are delayed in Grid queue systems and the real execution time on the nodes. Grid latency increases linearly with the number of Grid jobs, but the total Grid run time for a particular task decrease proportionally with the amount of jobs that can be processed in parallel in the Grid workers. Since the executions on the nodes are overlapping (processes are run in parallel but started sequentially), and hardware configurations may differ between nodes, the actual Grid execution time is difficult to calculate theoretically beforehand. Run-time measures were performed using external Linux timers. For the largest data set corresponding to sample set 7 with 109 families and the largest pedigree size of 23 bits, the calculated speed-up was 455-fold in 600 Grid nodes, 313 in 400 nodes, 172 in 200 nodes, and 44 in 50 nodes. We find that the total run time at different submission time-points remain stable. The average of the real execution time to perform 22 000 simulations on 600 remote Grid nodes (2.6 days, Table 3) represents a significant improvement, compared with the projected execution time to perform the same number of simulations in an up-to-date single processor (1193.5 days, Table 1). The theoretical 600-fold speed-up using 600 nodes is not achieved mainly due to Grid latency.

Discussion

Multipoint linkage analysis applications that are based on hidden Markov models scales exponentially with the pedigree size, and while algorithm improvements in the Allegro program has resulted in significantly shorter run times than Genehunter, the computational requirements still creates a bottle neck in genome-wide linkage analysis with many markers and large pedigrees. Performing a sufficient number of simulations is important to evaluate if a positive linkage signal obtained in a specific chromosomal region reaches the threshold of global significance. Some, but far from all institutes, possess or have access to the computational resources required, and in the absence of sufficient computational resources a reduction of the number of simulations performed is forced, which can lead to the estimation of insufficient global significance levels and false positive linkage claims.

In the Alzheimer's disease gene mapping project described in this work, even a modest number of simulations (1000 simulations) for the larger pedigree sizes creates a computational load that is incompatible with standalone CPU computers, prompting us to consider an alternative solution that distribute the time-consuming tasks subdivided into a set of smaller jobs executed in parallel with existing algorithms and software. Our developed Grid-Allegro implementation makes it possible to evaluate the level of significance of variation in different simulation parameters, as several thousands of simulations with different parameter calibrations can be done in the Grid, irrespective of computational demand. The results can be compared afterwards to test the robustness of the different statistical models that are available in Allegro software. To our knowledge, there are currently no publications describing parallel or distributed execution software approaches for genome-wide linkage analysis with large pedigree sizes.

We have here demonstrated how it is possible to exploit the advantages of the computational Grid to facilitate scaled-up analyses using existing algorithms and software for high computationally demanding task. Grids are cost-efficient resources that could have much use in genetics research, particularly for larger projects. Grids enable the sharing, selection, and aggregation of a wide variety of resources including computers, supercomputers, storage

Table 3 Real Grid-Allegro run times in minutes for a complete genome-wide linkage analysis, using 600 nodes in Swegrid

Chromosome	*No. of markers*	*Waiting time 1*	*Latency 1*	*Real run time 1*	*Total-run-time 1*	*Waiting time 2*	*Latency 2*	*Real run time 2*	*Total-run-time 2*
Chr 1:	88	0.00	140.00	143	562.00	48.00	140.00	145.00	613
Chr 2:	86			142				143.00	
Chr 3:	76			137				137.00	
Chr 4:	75	2.00	140.00	137	680.00	0.00	140.00	138.00	676
Chr 5:	74			136				136.00	
Chr 6:	64			127				126.00	
Chr 18:	40			138				136.00	
Chr 7:	78	8.00	140.00	135	689.00	76.00	140.00	135.00	754
Chr 8:	59			135				134.00	
Chr 9:	57			135				133.00	
Chr 10:	56			136				136.00	
Chr 11:	58	30.00	140.00	137	846.00	30.00	140.00	137.00	845
Chr 12:	59			133				132.00	
Chr 13:	38			137				137.00	
Chr 14:	51			136				135.00	
Chr 16:	39			133				134.00	
Chr 15:	50	0.00	140.00	136	939.00	2.00	140.00	136.00	938
Chr 17:	50			133				134.00	
Chr 19:	47			134				134.00	
Chr 20:	38			133				131.00	
Chr 21:	21			132				132.00	
Chr 22:	24			131				129.00	
	Total	40.00	700.00	*2976.00*	3716.00	156.00	700.00	*2970.00*	3826
				Real run-time in days	2.58				2.66

Table 3 shows representative real Grid-Allegro run-time measures for data on a complete genome-wide linkage analysis (1000 simulations for each chromosome), using the largest input data size corresponding to 109 families with largest pedigree size of 23 bits. In the different run-time measures performed, the total run time was found to be stable but the waiting time in the Grid queue system can vary depending on work load conditions, as shown here for two runs. As Grid latency time is directly proportional to the number of Grid jobs, chromosomes were conveniently grouped into five data sets to be submitted in parallel to the Grid.

systems, data sources, and specialized devices that are geographically distributed and owned by several different organizations. It offers CPU and data-handling capabilities that far exceed what can be attained within ordinary or even well-supported research facilities.[11,12]

However, concerning usability, there has been a clear need to ease the interface between the Grid and the users. Especially to the biologically oriented researcher a current obstacle is the middleware that is still raw and hardly accessible to the non-computer scientist. The job submission process is relatively complex and non-automated. A Grid user has had to deal with the middleware command line interface to submit jobs manually, periodically check the resource broker for the status of the job, and finally retrieve the raw data file. For an application in biological sciences where available computer expertise may not always run as deep, more user-friendly solutions are needed. With our programs and procedural descriptions, these tasks are automated and simplified.

Grid-Allegro involves temporary installations of the Allegro executables and datasets on remote nodes at submission, followed by uninstallation after the return of results. For the Allegro executable, the time for distribution and installation time to the nodes is negligible; however, for executable files of very large size, this could theoretically introduce an increase in overall latency and run time. As our strategy avoids the use of predefined run-time environments (preinstalled software and databases at specific Grid-nodes), it greatly improves the usability as no knowledge of the Grid structure is required of the users. This solution is also attractive from a Grid administrator's point of view as no resources are occupied on the nodes between submissions. Furthermore, from a user perspective, it also limits the interaction with Grid administrators for setup, installation and maintenance of run-time environments. Given the dynamic status of the Grid environment where clusters at some locations may have been added or removed between submission times, a predefined run-time environment would be more impractical and likely restrict the number of possibly allocated Grid nodes/workers.

There are several factors that could affect Grid-Allegro performance. Most importantly, the latency time in a Grid environment will increase directly proportional to the number of jobs that has to be submitted to the workers. In our work, the latency is increased by the requirement for serial submission of jobs that is part of the Grid

administration principles, while downloads can be done in parallel (forked download), which of course speeds up the process. However, in the case of the largest dataset, the latency of less than 12 h for a Grid job that takes 63 h in total but otherwise would consume over 3 years on a single computer (Table 3) could be considered negligible.

Table 4 Selected Major Grid application and deployment projects in Science and Engineering

Name	*Domain*	*Support*	*Focus*
Crossgrid	crossgrid.org	European Union	Develop, implement, and exploit new Grid components for interactive compute and data-intensive applications
DATATAG	datatag.org	European Union	Research and technology development for a transAtlantic Grid
European Union (EU) Datagrid	eu-datagrid.org	European Union	Create and apply an operational grid for applications in high-energy physics, environmental science, bioinformatics
Damien	www.hlrs.de/organization/pds/projects/damien	European Union	Develop building blocks for a middleware environment for distributed simulation and visualization
European Grid of Solar Observatories (EGSO)	www.mssl.ucl.ac.uk/grid/egso	European Union	Create a virtual archive by federating solar data centers scattered across Europe into a Data Grid
Eurogrid and Grid Interoperability (GRIP)	eurogrid.org	European Union	Create technologies for remote access to supercomputer resources and simulation codes; in GRIP, integrate with Globus
Grid Resources for Industrial Applications (GRIA)		European union	Application-driven Grid testbed focused on outsourcing of computational services for structural analysis and digital film post production
DOE Science Grid	sciencegrid.org	Department of Energy, Office of Science USA	Create an operational Grid providing access to resources and applications at US DOE science laboratories and partner universities
Earth System Grid (ESG)	earthsystemgrid.org	Department of Energy, Office of Science USA	Delivery and analysis of large climate model datasets for the climate research community
Fusion Collaboratory	fusiongrid.org	DOE Office of Science	Create a national computational collaboratory for fusion research
Globus	globus.org	DARPA, DOE, NASA, National Science Foundation	Research on Grid infrastructure and tools; development of community-based, open source, open architecture Globus Toolkit
(GRIDS) Center	grids-center.org	National Science Foundation	Integration, deployment, support of the NSF Middleware Infrastructure for research & education
Grid Application Development Software	hipersoft.rice.edu/grads	National science foundation	Research into program development methodologies for Grid applications
Grid Physics Network (GriPhyN)	griphyn.org	National science foundation	Technology R&D for data analysis in physics experiments: ATLAS, CMS, LIGO, SDSS
Information Power Grid	ipg.nasa.gov	NASA	Create and apply a production Grid for aerosciences and other NASA missions
Network for Earthquake Engineering Simulation Grid	neesgrid.org	National Science Foundation	Create and apply a production Grid for earthquake engineering
National Virtual Observatory (NVO)	www.srl.caltech.edu/nvo	National Science Foundation	Create and apply production Grids for data analysis in astronomy
Particle Physics Data Grid (PPDG)	ppdg.net	DOE Office of Science USA	Create and apply production Grids for data analysis in high-energy and nuclear physics experiments
Southern California Earthquake Center 2	scec.org	National Science Foundation	Full geophysics modeling using Grids and knowledge-based systems
Teragrid	teragrid.org	National Science Foundation	US science infrastructure linking four major resource sites at 40 Gb/s
GridPP	gridpp.ac.uk	UK eScience Program	Create and apply an operational grid within the UK for particle physics research
Mygrid	mygrid.org.uk	UK eScience Program	Develop and apply eScience workbench for bioinformatics applications
Reality Grid		UK eScience Program	Enable the realistic modelling of complex condensed matter systems at the molecular and mesocale levels
UK Grid Center	grid-support.ac.uk	UK eScience Program	Support center for Grid projects within the UK

Geographical distributions and local area networks conditions could also create a small increase in the latency time. Waiting time in the Grid queue system may also vary depending of the amount of previously submitted jobs to the Grid environment. A way of overcoming excessive waiting times in the Grid queue systems has been implemented in Grid-Allegro by defining a cut-off of maximum allowed waiting time. If this time (by default set to 1 h but is modifiable) is exceeded for a specific job, Grid-Allegro will kill this specific job in the queue system and resubmit it to another available Grid node in a different Grid cluster. For the largest dataset available in this study, several run-time experiments were performed to test that the total run time for a given Grid node size remained stable at different submission times. This shows that at least in SweGrid, waiting times are currently not an issue. Also when a relatively small number of Grid nodes is accessed, the improvements in run time (27 days for 50 nodes) compared with a serial execution in a single computer (3.2 years) are significant, however with a run time still 24 days more than in the 600 node example although latency would be reduced by a factor of 12 when using only 50 nodes. This suggests that Grid-based implementations are cost- and time-efficient alternatives compared with acquiring and setting-up local small computer clusters. Most genetic research departments do not own or have direct access to larger clusters of computers. Moreover, cluster computing requires the development of implementations based on the message passing interface (MPI) library specification. This task is not trivial even for computer experts in comparison with the alternative of implementing existing algorithms and compilations on external Grids. No MPI-based software implementation for linkage analysis is available according to our knowledge.

A large number of Grids have been established to address the rapidly growing needs of the high-performance scientific computing community. Globus software toolkit is the most popular Grid middleware used for building Grid systems and applications around the world. A list of selected major Grid applications and deployment projects in science and engineering is shown in Table 4. It is possible to migrate our application to any Globus- based Grid environment with no or very minor modifications. Many of these Grid initiatives are initiated and funded by governments and non-profit research organizations with the aim of providing large- scale computational resources to scientific and academic research. Access is given upon joining a virtual organization and allocations are granted to the members on a project basis according to their scientific contribution and importance as viewed by peer researchers. Administrative policies can define an upper limit for computational resources and run-time quota that can be allocated to a certain user. Current policies are based on a queuing principle of equal opportunity at a specific submission time for Grid users granted access to the same nodes, irrespective of the total run time. Recently, market-based automatic resource allocation solutions have been demonstrated that could enable a more efficient allocation and use of the available resources at any specific run time if implemented on a broader scale.[18] Grid security implementations are build on public key infrastructure,[19] in which each Grid user is authenticated by processing a set of credential comprised of a cryptographic key and a certificate, the authentication process results in the generation of a unique session key, which is used to protect further communication. However, recent review of security issues in large distributed systems[20] indicated that there are many issues still to be considered. Earnest and conscientious efforts are made in the different Grid organizations, and new mechanisms are being proposed to increase the Grid security.[21–23]

The one-time task to establish access to resources on a Grid (joining a Grid virtual organization, setup a local proxy server, installation of the standalone Grid client, installation of the Grid certificate, and so on) according to the detailed procedures on our website should comprise a few days for someone with basic knowledge in Linux system administration. The availability of our scripts and procedural descriptions could enable a broader use of Grid technology by other research groups in genetics. Importantly, this Perl-based Grid-Allegro implementation is generic, where the Allegro executable can easily be replaced by other software tools to facilitate analyses suitable for parallelization, examples of such programs are among others: ANALYZE,[24] LINKAGE package,[25] MERLIN[17] and ARLEQUIN.[26]

The scripts and procedural documentation are freely available from the authors at: http://kthgridproxy.biotech.kth.se/grid-allegro/index.html. However, a license for Allegro v1.1[3] (which is available free for non-commercial use) needs to be obtained independently (e-mail: allegro@decode.is).

Conclusions

In this study, a high-performance execution for simulation of genetic linkage data has been presented. On a genome scale, these operations take a prohibitively long time. By 'griddifying' the existing executables and software package, computation times are reduced from years to days with basically no investment in hardware. This allows the features of Allegro[3] to be fully exploited in, for example, genome-wide scans. As we show, the use of Grid computing is a low-cost high-performance alternative when computing needs in bioinformatics go beyond institution hardware capabilities.

Bioinformatics analysis of the massive quantities of molecular data produced by complete genome sequencing projects is one of the major challenges of the following years. Facing this challenge, the use of distributed Grid

environments is and efficient solution to distribute and integrate up-to-date databanks, algorithms and storage resources for genomics.

Acknowledgements

The study was supported by grants from Magn. Bergvalls foundation, Vinnova, Strategic Research Foundation (SSF), National Research Counsel (VR), and the Knut and Alice Wallenberg foundation. The genotyping and the genome-scan was generosity supported by Dainippon Sumitomo Pharma Co., Ltd.

References

1 Van Steen K, McQueen MB, Herbert A *et al*: Genomic screening and replication using the same data set in family-based association testing. *Nat Genet* 2005; **37**: 683–691.
2 Falchi M, Forabosco P, Mocci E *et al*: A genomewide search using an original pairwise sampling approach for large genealogies identifies a new locus for total and low-density lipoprotein cholesterol in two genetically differentiated isolates of Sardinia. *Am J Hum Genet* 2004; **75**: 1015–1031.
3 Gudbjartsson DF, Jonasson K, Frigge ML, Kong A: Allegro, a new computer program for multipoint linkage analysis. *Nat Genet* 2000; **25**: 12–13.
4 Idury RM, Elston RC: A faster and more general hidden Markov model algorithm for multipoint likelihood calculations. *Hum Hered* 1997; **47**: 197–202.
5 Kruglyak L, Daly MJ, Reeve-Daly MP, Lander ES: Parametric and nonparametric linkage analysis: a unified multipoint approach. *Am J Hum Genet* 1996; **58**: 1347–1363.
6 Frisen L, Soderhall C, Tapper-Persson M, Luthman H, Kockum I, Nordenskjold A: Genome-wide linkage analysis for hypospadias susceptibility genes. *J Urol* 2004; **172**: 1460–1463.
7 Holmans P, Zubenko GS, Crowe RR *et al*: Genomewide significant linkage to recurrent, early-onset major depressive disorder on chromosome 15q. *Am J Hum Genet* 2004; **74**: 1154–1167.
8 Gretarsdottir S, Sveinbjornsdottir S, Jonsson HH *et al*: Localization of a susceptibility gene for common forms of stroke to 5q12. *Am J Hum Genet* 2002; **70**: 593–603.
9 Sanna-Cherchi S, Reese A, Hensle T *et al*: Familial vesicoureteral reflux: testing replication of linkage in seven new multigenerational kindreds. *J Am Soc Nephrol* 2005; **16**: 1781–1787.
10 Foster I, Kesselman C, Tuecke S: The anatomy of the grid: enabling scalable virtual organizations. *Int J High Perform Comput Applic* 2001; **15**: 200–222.
11 Baker M, Buyya R, Laforenza D: Grids and Grid technologies for wide-area distributed computing. *Software-Practice Exp* 2002; **32**: 1437–1466.
12 Foster I, Kesselman C: Computational grids – Invited talk (Reprinted from The Grid: Blueprint for a new computing infrastructure, 1998). *Vector and Parallel Processing – Vecpar 2000* 2001; **1981**: 3–37.
13 Smirnova O, Eerola P, Ekelof T *et al*: The NorduGrid architecture and middleware for scientific applications. *Computational Science – ICCS 2003, Pt I, Proceedings* 2003; **2657**: 264–273.
14 Sillen A, Forsell C, Lilius L *et al*: Genome scan on Swedish Alzheimer's disease families. *Mol Psychiatry* 2006; **11**: 182–186.
15 Lander E, Kruglyak L: Genetic dissection of complex traits: guidelines for interpreting and reporting linkage results. *Nat Genet* 1995; **11**: 241–247.
16 McPeek MS: Optimal allele-sharing statistics for genetic mapping using affected relatives. *Genet Epidemiol* 1999; **16**: 225–249.
17 Abecasis GR, Cherny SS, Cookson WO, Cardon LR: Merlin – rapid analysis of dense genetic maps using sparse gene flow trees. *Nat Genet* 2002; **30**: 97–101.
18 Sandholm T, Lai K, Andrade J, Odeberg J: Market-Based Resource Allocation using Price Prediction in a High Performance Computing Grid for Scientific Applications. *IEEE Computer Society Press* 2006, HPDC '06: Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing.
19 Huang LC, Wu ZH: A PKI-based scalable security infrastructure for scalable grid. *Grid Cooperative Comput, Pt 2* 2004; **3033**: 1051–1054.
20 Ju SG, Yang Z, Wang CD, Guo DC: The analysis of authorization mechanisms in the Grid. *Grid Cooperative Comput, Pt 1* 2004; 1047–1050.
21 Beckles B, Welch V, Basney J: Mechanisms for increasing the usability of grid security. *Int J Human-Computer Stud* 2005; **63**: 74–101.
22 McNab A: The GridSite Web/Grid security system. *Software-Practice Exp* 2005; **35**: 827–834.
23 Gui XL, Xie B, Li YN, Qian DP: A grid security infrastructure based on behaviors and trusts. *Grid and Cooperative Computing GCC 2004 Workshops, Proceedings* 2004; **3252**: 482–489.
24 Terwilliger JD: A powerful likelihood method for the analysis of linkage disequilibrium between trait loci and one or more polymorphic marker loci. *Am J Hum Genet* 1995; **56**: 777–787.
25 Lathrop GM, Lalouel JM, Julier C, Ott J: Multilocus linkage analysis in humans: detection of linkage and estimation of recombination. *Am J Hum Genet* 1985; **37**: 482–498.
26 Laurent E, Guillaume L, Stefan S: Arlequin ver. 3.0: an integrated software package for population genetics data analysis. *Evol Bioinform Online* 2005; **1**: 47–50.