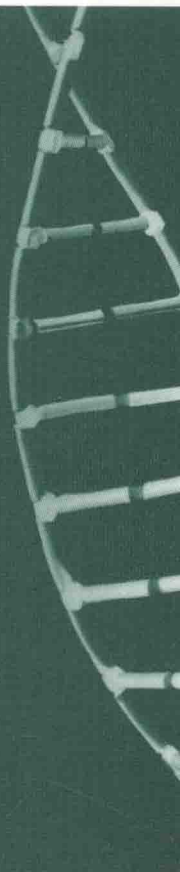
Werner Dubitzky Assaf Schuster
Peter M.A. Sloot Michael Schroeder
Mathilde Romberg (Eds.)

Distributed, High-Performance and Grid Computing in Computational Biology

LNBI 4360

International Workshop, GCCB 2006
Eilat, Israel, January 2007
Proceedings



Springer

Werner Dubitzky Assaf Schuster
Peter M.A. Sloot Michael Schroeder
Mathilde Romberg (Eds.)

Distributed, High-Performance and Grid Computing in Computational Biology

International Workshop, GCCB 2006
Eilat, Israel, January 21, 2007
Proceedings

Springer

Volume Editors

Werner Dubitzky
Mathilde Romberg

University of Ulster, School of Biomedial Sciences, Coleraine, UK
E-mail: {w.dubitzky, me.romberg}@ulster.ac.uk

Assaf Schuster
Technion – Israel Institute of Technology, Haifa, Israel
E-mail: assaf@cs.technion.ac.il

Peter M.A. Sloot
University of Amsterdam, Section of Computational Science, The Netherlands
E-mail: sloot@science.uva.nl

Michael Schroeder
Biotec, Dresden University of Technology, Germany
E-mail: Michael.Schroeder@biotec.tu-dresden.de

Library of Congress Control Number: 2006939917

CR Subject Classification (1998): F.1, F.2, J.3, D.1, D.2, D.4, C.2, C.4, H.4

LNCS Sublibrary: SL 8 – Bioinformatics

ISSN 0302-9743
ISBN-10 3-540-69841-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-69841-8 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11976158 06/3142 5 4 3 2 1 0

Lecture Notes in Bioinformatics 4360

Edited by S. Istrail, P. Pevzner, and M. Waterman

Editorial Board: A. Apostolico S. Brunak M. Gelfand
T. Lengauer S. Miyano G. Myers M.-F. Sagot D. Sankoff
R. Shamir T. Speed M. Vingron W. Wong

Subseries of Lecture Notes in Computer Science

Preface

This volume of the Springer *Lecture Notes in Computer Science* series contains the contributions presented at the International Workshop on Distributed, High-Performance and Grid Computing in Computational Biology 2006 (GCCB 2006) held in Eilat, January 21, 2007 in conjunction with the fifth European Conference on Computational Biology (ECCB 2006).

Modern computational biology and bioinformatics are characterized by large and complex-structured data and by applications requiring considerable computing resources, such as processing units, storage elements and software programs. In addition, these disciplines are intrinsically geographically distributed in terms of their instruments, communities and computing resources. Tackling the computational challenges in computational biology and bioinformatics increasingly requires high-end and distributed computing infrastructures, systems and tools. The main objective of this workshop is to bring together researchers and practitioners from these areas to discuss ideas and experiences in developing and applying distributed, high-performance and grid computing technology to problems in computational biology and bioinformatics.

The challenges in distributed, high-performance and grid computing in biology and biotechnology are inherently more complicated than those in such domains as physics, engineering and conventional business areas. Some of the added complexities arise from the:

- Conceptual complexity of biological knowledge and the methodologies used in biology and biotechnology
- Need to understand biological systems and processes at a detailed mechanistic, systemic and quantitative level across several levels of organization (ranging from molecules to cells, populations, and the environment)
- Growing availability of high-throughput data from genomics, transcriptomics, proteomics, metabolomics and other high-throughput methods
- Widespread use of image data in biological research and development (microscopy, NMR, MRI, PET, X-ray, CT, etc.)
- Increasing number of investigations studying the properties and dynamic behavior of biological systems and processes using computational techniques (molecular dynamics, QSAR/QSPR, simulation of gene-regulatory, signaling and metabolic networks, protein folding/unfolding, etc)
- Requirement to combine data, information and compute services (e.g., sequence alignments) residing on systems that are distributed around the world
- Variety of different technologies, instruments, infrastructures and systems used in life science R&D
- Huge variety of information formats and frequent addition of new formats arising from new experimental protocols, instruments and phenomena to be studied

- Large and growing number of investigated biological and biomedical phenomena
- Fact that life science R&D is based heavily on the use of distributed and globally accessible computing resources (databases, knowledge bases, model bases, instruments, text repositories, compute-intensive services)

The GCCB workshop brought together computational biologists, bioinformaticians and life scientists who have researched and applied distributed, high-performance and grid computing technologies in the context of computational biology and bioinformatics. The workshop discussed innovative work in progress and important new directions. By sharing the insights, discussing ongoing work and the results that have been achieved, we hope the workshop participants conveyed a comprehensive view of the state of the art in this area and identified emerging and future research issues. We believe that the GCCB workshop made a valuable contribution in encouraging and shaping future work in the field of distributed, high-performance and grid computing in computational biology.

Acknowledgements

GCCB 2006 was sponsored and organized by the University of Ulster, Coleraine, UK; the Technion - Israel Institute of Technology, Haifa, Israel; the University of Amsterdam, Amsterdam, The Netherlands; the Dresden University of Technology, Dresden, Germany; and the DataMiningGrid Consortium (EC grant IST-2004-004475, www.DataMiningGrid.org). We are indebted to the ECCB 2006 Conference organizers Hershel Safer, Weizmann Institute of Science, and Haim Wolfson, Tel Aviv University. Finally, we would like to extend our gratitude to the members of the GCCB 2006 International Program Committee and to those who helped in the review process. And last but not least we want to thank all authors and workshop participants for their contributions and the valuable discussions.

January 2007

Werner Dubitzky
Assaf Schuster
Peter Sloot
Michael Schroeder
Mathilde Romberg

Organization

Program Chairs

Werner Dubitzky, University of Ulster, Coleraine, UK
Assaf Schuster, Technion - Israel Institute of Technology, Haifa, Israel
Peter M.A. Sloot, University of Amsterdam, Faculty of Sciences, Amsterdam, The Netherlands
Michael Schroeder, Dresden University of Technology, Biotechnological Centre, Dresden, Germany
Mathilde Romberg, University of Ulster, Coleraine, UK

Program Committee

David A. Bader, Georgia Tech, College of Computing, Atlanta, Georgia, USA
Eric Bremer, Children's Memorial Hospital, Northwestern University, Chicago, USA
Rui M. Brito, Universidade de Coimbra, Coimbra, Portugal
Marian Bubak, AGH - University of Science and Technology, Krakow, Poland
Kevin Burrage, The University of Queensland, Australia
Gustavo Deco, Universitat Pompeu Fabra, Barcelona, Spain
Frank Dehne, University of Ottawa, Ottawa, Canada
Guiseppe di Fatta, University of Konstanz, Konstanz, Germany
Werner Dubitzky, University of Ulster, Coleraine, UK
Jordi Vill i Freixa, Universitat Pompeu Fabra, Barcelona, Spain
David Gilbert, University of Glasgow, Glasgow, UK
Carol Goble, University of Manchester, Manchester, UK
Danilo Gonzalez, Universidad de Talca, Talca, Chile
Ulrich Hansmann, Forschungszentrum Juelich, NIC, Juelich, Germany
Des Higgins, University College Dublin, Conway Institute, Dublin, Ireland
Alfons Hoekstra, University of Amsterdam, Amsterdam, The Netherlands
Martin Hoffmann, Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Sankt Augustin, Germany
Rod Hose, University of Sheffield, Sheffield, UK
Chun-Hsi (Vincent) Huang, University of Connecticut, Storrs, USA
Akihiko Konagaya, Riken Genomic Sciences Center, Yokohama City, Japan
Miron Livny, University of Wisconsin at Madison, Wisconsin, USA
Uko Maran, University of Tartu, Tartu, Estonia
Hartmut Mix, Dresden University of Technology, Dresden, Germany
Ron Perrot, Queens University, Belfast, UK
Mark Ragan, The University of Queensland, Australia

Stephen Robinson, University of Ulster, Coleraine, UK
Mathilde Romberg, University of Ulster, Coleraine, UK
Michael Schroeder, Dresden University of Technology, Biotechnological Centre, Dresden, Germany
Assaf Schuster, Technion - Israel Institute of Technology, Haifa, Israel
Gadi Schuster, Technion - Israel Institute of Technology, Haifa, Israel
Richard Sinnot, University of Glasgow, UK
Peter Sloot, University of Amsterdam, Faculty of Sciences, Amsterdam, The Netherlands
Craig A. Stewart, Indiana University, Indiana, USA
Domenico Talia, Università della Calabria, DEIS, Rende, Italy
Albert Zomaya University of Sydney, Sydney, Australia

External Reviewers

Pinar Alper, University of Manchester, Manchester, UK
Michael Johnston, Universitat Pompeu Fabra, Barcelona, Spain
David Jones, University of Manchester, Manchester, UK
Peter Kral, Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Sankt Augustin, Germany
Dean Kuo, University of Manchester, Manchester, UK
Jan Meinke, Forschungszentrum Juelich, NIC, Juelich, Germany
Javier Tamames, Universitat Pompeu Fabra, Barcelona, Spain

Lecture Notes in Bioinformatics

Vol. 4360: W. Dubitzky, A. Schuster, P.M.A. Sloot, M. Schroeder, M. Romberg (Eds.), Distributed, High-Performance and Grid Computing in Computational Biology. X, 192 pages. 2007.

Vol. 4345: N. Maglaveras, I. Chouvarda, V. Koutkias, R. Brause (Eds.), Biological and Medical Data Analysis. XIII, 496 pages. 2006.

Vol. 4316: M.M. Dalkilic, S. Kim, J. Yang (Eds.), Data Mining and Bioinformatics. VIII, 197 pages. 2006.

Vol. 4230: C. Priami, A. Ingólfsdóttir, B. Mishra, H.R. Nielson (Eds.), Transactions on Computational Systems Biology VII. VII, 185 pages. 2006.

Vol. 4220: C. Priami, G. Plotkin (Eds.), Transactions on Computational Systems Biology VI. VII, 247 pages. 2006.

Vol. 4216: M.R. Berthold, R.C. Glen, I. Fischer (Eds.), Computational Life Sciences II. XIII, 269 pages. 2006.

Vol. 4210: C. Priami (Ed.), Computational Methods in Systems Biology. X, 323 pages. 2006.

Vol. 4205: G. Bourque, N. El-Mabrouk (Eds.), Comparative Genomics. X, 231 pages. 2006.

Vol. 4175: P. Bücher, B.M.E. Moret (Eds.), Algorithms in Bioinformatics. XII, 402 pages. 2006.

Vol. 4146: J.C. Rajapakse, L. Wong, R. Acharya (Eds.), Pattern Recognition in Bioinformatics. XIV, 186 pages. 2006.

Vol. 4115: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence and Bioinformatics, Part III. XXI, 803 pages. 2006.

Vol. 4075: U. Leser, F. Naumann, B. Eckman (Eds.), Data Integration in the Life Sciences. XI, 298 pages. 2006.

Vol. 4070: C. Priami, X. Hu, Y. Pan, T.Y. Lin (Eds.), Transactions on Computational Systems Biology V. IX, 129 pages. 2006.

Vol. 4023: E. Eskin, T. Ideker, B. Raphael, C. Workman (Eds.), Systems Biology and Regulatory Genomics. X, 259 pages. 2007.

Vol. 3939: C. Priami, L. Cardelli, S. Emmott (Eds.), Transactions on Computational Systems Biology IV. VII, 141 pages. 2006.

Vol. 3916: J. Li, Q. Yang, A.-H. Tan (Eds.), Data Mining for Biomedical Applications. VIII, 155 pages. 2006.

Vol. 3909: A. Apostolico, C. Guerra, S. Istrail, P. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 612 pages. 2006.

Vol. 3886: E.G. Bremer, J. Hakenberg, E.-H.(S.) Han, D. Berrar, W. Dubitzky (Eds.), Knowledge Discovery in Life Science Literature. XIV, 147 pages. 2006.

Vol. 3745: J.L. Oliveira, V. Maojo, F. Martín-Sánchez, A.S. Pereira (Eds.), Biological and Medical Data Analysis. XII, 422 pages. 2005.

Vol. 3737: C. Priami, E. Merelli, P. Gonzalez, A. Omicini (Eds.), Transactions on Computational Systems Biology III. VII, 169 pages. 2005.

Vol. 3695: M.R. Berthold, R.C. Glen, K. Diederichs, O. Kohlbacher, I. Fischer (Eds.), Computational Life Sciences. XI, 277 pages. 2005.

Vol. 3692: R. Casadio, G. Myers (Eds.), Algorithms in Bioinformatics. X, 436 pages. 2005.

Vol. 3680: C. Priami, A. Zelikovsky (Eds.), Transactions on Computational Systems Biology II. IX, 153 pages. 2005.

Vol. 3678: A. McLysaght, D.H. Huson (Eds.), Comparative Genomics. VIII, 167 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005.

Vol. 3500: S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 632 pages. 2005.

Vol. 3388: J. Lagergren (Ed.), Comparative Genomics. VII, 133 pages. 2005.

Vol. 3380: C. Priami (Ed.), Transactions on Computational Systems Biology I. IX, 111 pages. 2005.

Vol. 3370: A. Konagaya, K. Satou (Eds.), Grid Computing in Life Science. X, 188 pages. 2005.

Vol. 3318: E. Eskin, C. Workman (Eds.), Regulatory Genomics. VII, 115 pages. 2005.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004.

Vol. 3082: V. Danos, V. Schachter (Eds.), Computational Methods in Systems Biology. IX, 280 pages. 2005.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004.

Vol. 2812: G. Benson, R.D.M. Page (Eds.), Algorithms in Bioinformatics. X, 528 pages. 2003.

Vol. 2666: C. Guerra, S. Istrail (Eds.), Mathematical Methods for Protein Structure Analysis and Design. XI, 157 pages. 2003.

Table of Contents

Session 1a. "Sequence Analysis"

Session 1b. "Grids for Screening and Property Prediction"

Session 2a. "Data Management"

Session 2b. "Collaborative Environments"

Combining a High-Throughput Bioinformatics Grid and Bioinformatics Web Services

Chunyan Wang[1], Paul M.K. Gordon[1], Andrei L. Turinsky[1], Jason Burgess[2], Terry Dalton[2], and Christoph W. Sensen[1]

[1] Sun Center of Excellence for Visual Genomics, University of Calgary, HS 1150, 3330 Hospital Dr. NW, Calgary, Alberta, T2N 4N1, Canada
[2] National Research Council Institute for Marine Biosciences, 1411 Oxford Street, Halifax, Nova Scotia, B3H 3Z1, Canada
{cwan, gordonp, aturinsk}@ucalgary.ca, {jason.burgess, terry.dalton}@nrc-cnrc.gc.ca, csensen@ucalgary.ca

Abstract. We have created a high-throughput grid for biological sequence analysis, which is freely accessible via bioinformatics Web services. The system allows the execution of computationally intensive sequence alignment algorithms, such as Smith-Waterman or hidden Markov model searches, with speedups up to three orders of magnitude over single-CPU installations. Users around the world can now process highly sensitive sequence alignments with a turnaround time similar to that of BLAST tools. The grid combines high-throughput accelerators at two bioinformatics facilities in different geographical locations. The tools include TimeLogic DeCypher boards, a Paracel GeneMatcher2 accelerator, and Paracel BlastMachines. The Sun N1 Grid Engine software performs distributed resource management. Clients communicate with the grid through existing open BioMOBY Web services infrastructure. We also illustrate bioinformatics grid strategies for distributed load balancing, and report several nontrivial technical solutions that may serve as templates for adaptation by other bioinformatics groups.

Keywords: Bioinformatics, sequence alignment, grid, Web services, BioMOBY, Sun N1 Grid Engine, Smith-Waterman, hidden Markov model.

1 Introduction

Analysis and functional annotation of genomic and proteomic sequences require fast and sensitive methods that are able to capture complex patterns within massive sequence collections in a reasonable time. The algorithms used to search for and rank sequence alignment regions, fall mostly into two classes: those based on dynamic programming, such as the Smith-Waterman algorithm [1] or the various BLAST methods [2]; and those based on probabilistic networks, such as hidden Markov model (HMM) searches [3]. The canonical Smith-Waterman or HMM alignments typically require massive computational power. For example, by incorporating hardware accelerators described in this paper into the MAGPIE genome annotation tool [4], more than 50 000 expressed sequence tags (EST), an

W. Dubitzky et al. (Eds.): GCCB 2006, LNBI 4360, pp. 1 – 10, 2007.

equivalent of six to eight full microbial genomes, can be annotated in one day. Without the accelerators, the annotation of one genome would require two to three weeks (unpublished data).

Most research teams cannot afford the local implementation of specialized high-throughput hardware, hence far less sensitive BLAST searches remain the most used type of sequence similarity searches today. Further hindrance to the research process is the disparity between data input specifications of many generic bioinformatics software tools [5, 6]. A typical bioinformatics analysis session likely involves switching between multiple tools, transferring data manually, and performing data format conversion to ensure compatibility.

To help overcome these challenges, we have created a high-throughput system for the alignment of biological sequences that combines the grid paradigm [7, 8] with the Web services paradigm [9]. There is a growing interest in distributed bioinformatics solutions, especially those that are Web services-oriented or grid-oriented [10, 11, 12]. Outside of bioinformatics, the grid and Web services-based computation methods are rapidly undergoing a standardization process [13]. Our system unites several hardware and software accelerators into a grid and provides universal access to it through an existing bioinformatics Web services interface, called BioMOBY [14]. The BioMOBY project endeavors to establish a *de facto* bioinformatics data interchange standard for the web, using open XML-based communication standards and protocols for service allocation and queries.

We provide access to several highly specialized sequence analysis hardware accelerators and software tools, while allowing dynamic load-balancing across geographically distributed computational nodes at the same time. Furthermore, the inner structure of our grid is entirely transparent to any BioMOBY user, and can be incorporated seamlessly into other BioMOBY data analysis pipelines via http://biomoby.org.

2 Materials and Methods

2.1 Architecture

Our architecture comprises three layers: the high-throughput accelerators, the grid middleware for job allocation, and the BioMOBY Web services interface (Fig. 1). At the bottom level, highly sensitive sequence alignments are performed using the resources of two bioinformatics facilities, which are three time zones apart: the Sun Center of Excellence for Visual Genomics at the University of Calgary; and the Institute for Marine Biosciences of the National Research Council Canada, in Halifax, Canada. The resources at the Calgary site include:

- Four FPGA-based (field programmable gate array) TimeLogic™ DeCypher® bioinformatics accelerator boards, mounted in a SunFire V880 server;
- A Paracel BlastMachine based on a Linux cluster with 12 dual-processor nodes (hereafter referred to as the 24-processor BlastMachine);

- An ASIC-based (application-specific integrated circuits) Paracel GeneMatcher2 bioinformatics accelerator with 27 648 specialized CPUs on nine boards; by design, jobs for the GeneMatcher2 are submitted through the Linux cluster;
- A Sun Fire E6900 UNIX server used for coordination tasks.

The resources at the Halifax side include:

- A Paracel BlastMachine based on a 12-node Linux cluster (hereafter referred to as the 12-processor BlastMachine);
- A dedicated Sun Enterprise 220R UNIX server used for the handling of command-line grid job submissions.

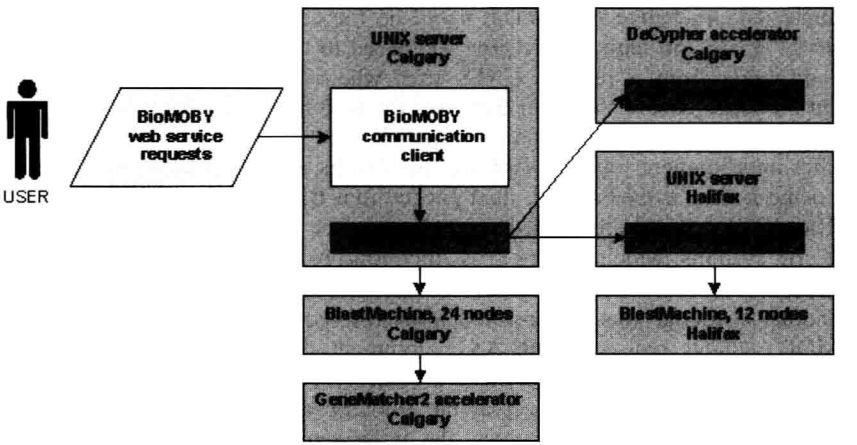


Fig. 1. The dataflow of the job submission to the BioMOBY-enabled grid test bed. The layered architecture comprises the computational nodes (*grey*), the grid middleware (*dark grey*) and the BioMOBY Web services components (*light grey*).

At the middle layer, the resources are united into a common wide area grid using Sun N1 Grid Engine 6 (SGE) middleware solution [15]. The SGE is installed on all machines as either a master host (main UNIX server in Calgary), which can submit jobs to other hosts, or an execution host (the remaining machines), which can only execute jobs locally. The BioMOBY-enabled SGE supports job execution over wide area networks, has a customizable policy module for formulating a flexible job submission policy, and allows easy out-of-the-box installation. The SGE defines internal queues for job scheduling, maintained at the master hosts, which may represent algorithm types, such as HMM searches, or individual execution hosts, or narrower job types, such as HMM Frame EST-to-HMM searches on the DeCypher system.

At the top layer, the BioMOBY Web services system provides a mechanism to discover the data processing resources and submit requests. Because all job submissions to the grid are handled by the main UNIX server in Calgary, the BioMOBY communication libraries only need to be installed on that host to create the

Web services endpoint. Using this endpoint, we registered the new BioMOBY web services, which are named for three types of sequence alignment analyses: runBlastVsPublicDBs, runSmithWatermanVsPublicDBs, and search_Pfam (for HMM searches). The registration process is described in [16]. Further information is available on the project website (http://moby.ucalgary.ca).

2.2 Data Flow Implementation

We had to overcome several unusual technical challenges in the context of implementing a geographically distributed grid test bed. First, traditional grids require users to log into their grid accounts to ensure remote authentication and secure data transfer. In contrast, BioMOBY analyses require no user accounts or authentication. Our solution was to create a specialized UNIX account on every SGE host and set up a remote data exchange mechanism based on the common gateway interface (CGI) scripts [17]. All job requests are now submitted to the SGE from this account, as if they originate from a regular UNIX user who is logged into the system. SGE execution hosts retrieve the input data over the web from the master host's CGI using the free Wget utility [18] and a unique input ID.

After an alignment job is completed, the results are retrieved by the SGE master host using a similar mechanism, then packed in a BioMOBY format and returned to the BioMOBY client. This CGI-based remote data exchange is in stark contrast to most grid systems, which require cross-mounting of disk drives between all hosts of the grid.

Another challenge was to translate BioMOBY requests into SGE submissions. If a BioMOBY client requests a BLAST alignment of a single sequence against a nucleotide sequence database, a custom-made Perl program parses the BioMOBY request at the main UNIX server at the Calgary site, which hosts the BioMOBY service endpoint. The program posts the input sequence as a separate file on the master host's CGI with a unique input identifier, and initializes a sequence-type parameter (nucleotide or amino acid). Subsequently it submits a pre-made SGE script `all_blast.sh` to the grid using the SGE submission command `qsub`, as follows:

```
qsub -hard -i numjobs=4 all_blast.sh $input_id $type
```

In this example, a simple threshold-based job scheduling is implemented. A SGE load sensor script goes through the list of hosts sorted in the order of their optimal BLAST performance. If it finds that the host has no more than four jobs in its SGE queue (`numjobs=4`), the new job of executing `all_blast.sh` is added to the host's queue. Otherwise the next best host from the list is examined.

The third challenge was to enable a generic SGE script to invoke job submissions with different syntax for different hosts. Continuing the BLAST example, the DeCypher and BlastMachine expect drastically different BLAST commands, whereas `qsub` invokes the same generic script `all_blast.sh` on either host. Our solution was to place on each host a local script `blast.sh` with BLAST commands specific to the host. A single `all_blast.sh` script can now trigger any of the local versions of `blast.sh` using the same command line, and pass on the required parameters.

3 Results and Discussion

Our objective was to explore possible grid load balancing policies that take into account the relative performances of the individual hosts. Even for single-sequence submissions, the grid considerably improves the performance of the alignment searches. For example, using the grid scheduling policy described in the previous section, we tested three types of alignment searches against public databases for a 599-amino-acid-long *Arabidopsis thaliana* calcium-dependent protein kinase AT3g19100/MVI11_1 (TrEMBL accession number Q9LJL9). The speedup was especially large for the computationally intensive Smith-Waterman searches, which took on average 5 100 seconds on a single CPU of the Sun Fire E6900 server from the Calgary site, but only 86 seconds on the grid – about 60 times faster (Fig. 2).

The benefits of parallel processing hardware become even more apparent when multiple input sequences are analyzed. This powerful option is available, for example, through the jMOBY Java-based toolkit for BioMOBY job submissions [19], plug-in modules for the Taverna workbench [20] as a front end to the BioMOBY system, or through other sequence analysis tools that support the BioMOBY Web services interface, such as [21]. At the time of this writing, the development of BioMOBY clients that access BioMOBY services or have an embedded BioMOBY functionality, is growing rapidly (for an up-to-date list see [22]). The features for the automated web-based and/or GUI-based submission of multiple sequences through BioMOBY clients, although a relatively new functionality, are also being constantly expanded.

For testing, we searched the NCBI Protein repository [23] for the keyword "alcohol" and retrieved the first N=100 sequences. This set was replicated to create the derivative sets of N=300, 1 000, 3 000, 10 000, 30 000, and 100 000 sequences, so that the values of log(N) cover the logarithmic scale almost uniformly between 2 and 5 with a step of approximately 0.5.

Alignments using a single CPU of the Sun Fire E6900 server in Calgary were tested on N=100 sequences, with the average execution time being 10 574 seconds for a BLAST, 3 961 seconds for an HMM search, and over 4 days(!) for a Smith-Waterman search. As Figure 2 shows, this constitutes a dramatic difference of two to three orders of magnitude compared to the grid performance on the same set of 100 sequences (listed in Table 1 for N=100). Tests for larger N values on a single CPU were not attempted due to the clear superiority of the grid and the specialized tools.

For each search type, the computational nodes exhibited near-linear performance with respect to the input size. However, we did not attempt linear regression because the data for smaller N are clearly outliers due to initial overhead and would cause linear least-squares fitting to underestimate consistently the "true" throughputs.

For BLAST searches, the throughputs were 4.25 seq/sec for the DeCypher, 1.13 seq/sec for the 24-processor BlastMachine, and 0.90 seq/sec for the 12-processor BlastMachine, measured at N=30 000, where deviations due to initial overhead were deemed minimal. We therefore split the input data among the three accelerators in proportion to their observed computational power: 68% for the DeCypher, 18% for the Calgary BlastMachine, and 14% for the Halifax BlastMachine. Generally, the load balancing should use the best available dynamic estimate for the throughput of each

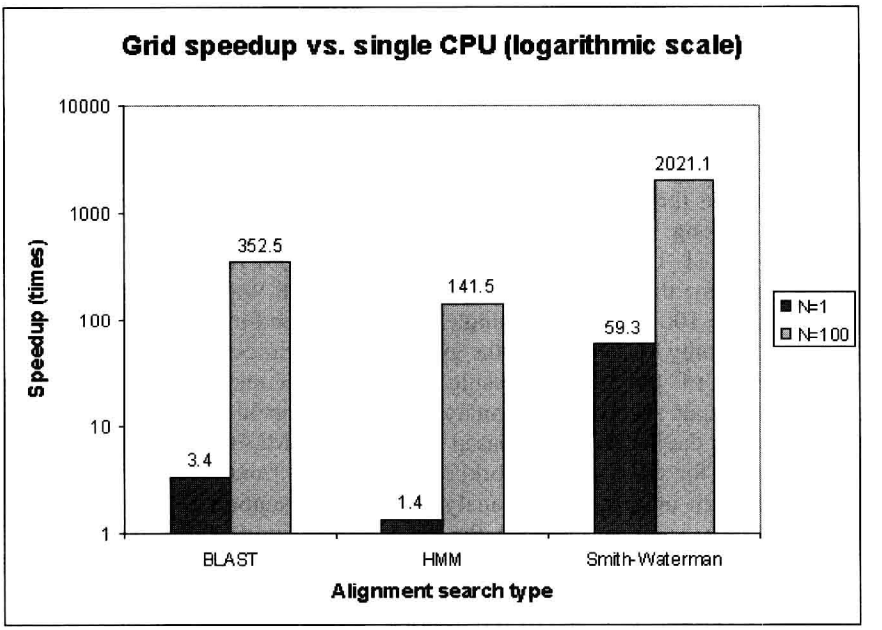


Fig. 2. The grid speedup compared to a single CPU on a Sun Fire E6900 server. Results reported in logarithmic scale for different numbers of input sequences and different alignment types (average of 6 trials).

computational node, taking into account such factors as the specific job type and search parameters. A proportional grid load balancing policy was defined:

Split large BLAST jobs among the DeCypher, Calgary BlastMachine, and Halifax BlastMachine in proportion to the (estimated) throughput of each accelerator.

For HMM searches, the DeCypher execution time ranged from 23 seconds for $N=10^2$ to 2 hours 43 minutes for $N=10^5$. In contrast, the execution time on the GeneMatcher2, which is a massively parallel supercomputer, stayed within the range of between 6 and 10 minutes. Using the approximate crossover point at N=5 000 between the two performance graphs, the grid load balancing policy was:

Submit the entire HMM job to the DeCypher if the input is less than 5 000 sequences, otherwise submit the entire job to the GeneMatcher2.

For Smith-Waterman searches, the GeneMatcher2 throughput was approximately 1.09 seq/sec for N=30 000. In contrast, the DeCypher throughput was only 0.07 seq/sec for N=3 000, with the execution time of nearly 12 hours. Due to the clear GeneMatcher2 superiority, DeCypher tests for larger N were not attempted. A trivial grid submission policy was defined:

For Smith-Waterman search requests, use the GeneMatcher2. Use the DeCypher only if the GeneMatcher2 is unavailable.