

Apply Bioinformatics Applications on Parallel and Grid Computing Environment

應用生物資訊軟體於平行及網格計算環境

Yu-Lun Kuo
郭育倫

High-Performance Computing Laboratory
高效能計算實驗室
Department of Computer Science and
Information Engineering
Tunghai University
東海大學資訊工程與科學系
Taichung, 407, Taiwan, R.O.C.
E-mail: g912814@student.thu.edu.tw

Chao-Tung Yang
楊朝棟

High-Performance Computing Laboratory
高效能計算實驗室
Department of Computer Science and
Information Engineering
Tunghai University
東海大學資訊工程與科學系
Taichung, 407, Taiwan, R.O.C.
E-mail: ctyang@mail.thu.edu.tw

Abstract

In addition to the traditional massively parallel computers, distributed workstation clusters now play an important role in scientific computing perhaps due to the advent of commodity high performance processors, low-latency/high-band width networks and powerful development tools. As we know, bioinformatics tools can speed up the analysis of large-scale sequence data, especially about sequence alignment. To fully utilize the relatively inexpensive CPU cycles available to today's scientists, a PC cluster consists of one master node and seven slave nodes (16 processors totally), is proposed and built for bioinformatics applications. In this paper, we also experiment on Sun Fire 6800 Server and Grid System and compare with the bioinformatics tools performance. We use the mpiBLAST, FASTA and HMMer on parallel computer and Grid System to speed up the process for sequence alignment. The mpiBLAST and FASTA software use a message-passing library called MPI (Message Passing Interface) and the HMMer software uses a software package called PVM (Parallel Virtual Machine), respectively. The system architecture and performances of these three platforms are also presented in this paper.

Keywords: Parallel computing, Grid Computing, Sun Fire 6800 SMP, Bioinformatics, Speedup

摘要

除了傳統的大型平型電腦之外，由於許多高效能處理器的誕生以及擁有高速傳輸頻寬的網路和許多有用發展工具的出現，使得分散式電腦叢集在現今的科學計算領域中扮演了非常重要的角色。如我們所知，生物資訊領域的相關軟體可以加速巨量序列資料的分析，尤其是針對序列的分析比對。因此我們利用了目前處理器價格低廉的特點，利用八台雙處理器的電腦建構一組主從式架構的個人電腦叢集以作為生物資訊之計算平台。在此篇論文中，我們還利用了Sun Fire 6800伺服器與網格系統去分析平行生物資訊的效能。實驗中利用mpiBLAST、FASTA與HMMer三種平行版本的生物資訊應用軟體，並且紀錄與比較其序列比對所需的時間與效能。而論文中也會提及關於此系統之架構以及叢集電腦的效能分析等。

關鍵詞: 平行計算, 網格計算, Sun Fire 6800 SMP伺服器, 生物資訊, 加速

1 Introduction

Cluster computing is not new, but in company with other technical capabilities, particularly in the area of networking, this class of machines is becoming a

high-performance platform for parallel and distributed applications.

Scalable computing clusters, ranging from a cluster of (homogeneous or heterogeneous) PCs or workstations to SMP (Symmetric MultiProcessors), are rapidly becoming the standard platforms for high-performance and large-scale computing. A cluster is a group of independent computer systems and thus forms a loosely coupled multiprocessor system as show in Figure 1.

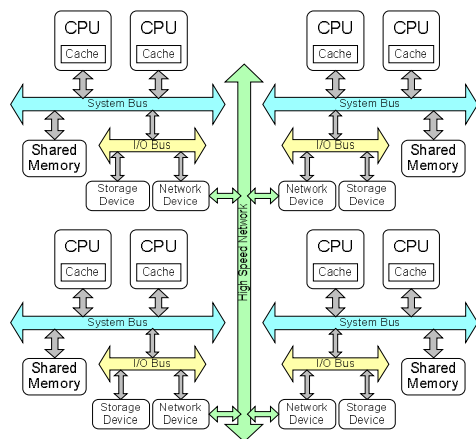


Figure 1: A cluster system by connecting four SMPs.

Inexpensive systems such as Beowulf clusters have become increasingly popular in both the commercial and academic sectors of the bioinformatics community. In this paper; we also use Sun Fire 6800 Server and Grid system for our TestBed. The supercomputer is a SMP system. It construct by 24 CPUs and 24 GB main memory. We use 8 CPUs and 8 GB main memory to applying the bioinformatics software and analysis the performance on the Sun Fire 6800 Server. The grid system is constructing by many computers or clusters and it can provide large computational resource to process sequence analysis.

There exist some parallel version bioinformatics application [1] which can be installed and conducted on a PC cluster, for example, HMMer, FASTA, mpiBLAST, PARACEL BLAST [2], ClustalW-MPI [3], Wrapping up BLAST [4], and TREE-PUZZLE [5] et al. Using these parallel programs on sequence alignment can always save much time and cost.

2 Background Review

2.1 Bioinformatics

Bioinformatics is the marriage of biology and information technology. The discipline encompasses

any computational tools and methods used to manage, analyze and manipulate large sets of biological data. Essentially, bioinformatics has tree components:

- The creation of database allowing the storage and management of large biological data set.
- The development of algorithms and statistics to determine relationships among members of large data set.
- The use of these tools for the analysis and interpretation of various types of biological data, including DNA, RNA, and protein sequences, protein structures, gene expression profiles and biochemical pathways.

2.2 Cluster Computing

A Beowulf cluster is a parallel computer system, it suits applications that can be partitioned into tasks, which can then be executed concurrently by a number of processors.

The previous study lists four benefits that can be achieved with clustering. These can also be thought of as objectives or design requirements:

- **Absolute scalability:** It is possible to create large clusters that far surpass the power of even the largest standalone machines. A cluster can have dozens of machines, each of which is a multiprocessor.
- **Incremental Scalability:** A cluster is configured in such a way that it is possible to add new systems to the cluster in small increments. Thus, a user can start out with a modest system and expand it as needs grow, without having to go through a major upgrade in which an existing small system is replaced with a larger system.
- **High availability:** Because each node in a cluster is a standalone computer, the failure of one node does not mean loss of service. In many products, fault tolerance is handled automatically in software.
- **Superior price/performance:** By using commodity building blocks, it is possible to put together a cluster with equal or greater computing power than a single large machine, at much lower cost.

2.3 Grid Computing

Grid computing (or the use of a computational grid) is applying the resources of many computers in a network to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data.

Grid computing requires the use of software that can divide and farm out pieces of a program to as

many as several thousand computers. Grid computing can be thought of as distributed and large-scale cluster computing and as a form of network-distributed parallel processing. It can be confined to the network of computer workstations within a corporation or it can be a public collaboration.

Grid computing appears to be a promising trend for three reasons: (1) its ability to make more cost-effective use of a given amount of computer resources, (2) as a way to solve problems that can't be approached without an enormous amount of computing power, and (3) because it suggests that the resources of many computers can be cooperatively and perhaps synergistically harnessed and managed as a collaboration toward a common objective. In some grid computing systems, the computers may collaborate rather than being directed by one managing computer.

2.4 BioGrid

Grid computing has a potential for expansion in computing performance by connecting a large number of parallel computers or PC clusters with high performance networks. Therefore, Grid system can help us speedup the experimentation time. The hybrid system sometimes called BioGrid [12]. BioGrid is a large-scale distributed computing environment, including couple of computers, storage systems, and other devices. BioGrid system can improve performance more than parallel computing on PC clusters. In this paper, the FASTA bioinformatics application software is used and ported on the grid system, and we must re-compile it by using MPICH-G2. The FASTA can be executed on grid system. There are some MPI based biology software applications, for example, mpiBLAST [7] and ClustalW [3] etc. Therefore construct the BioGrid system is necessary for research to accelerate the sequence alignment time.

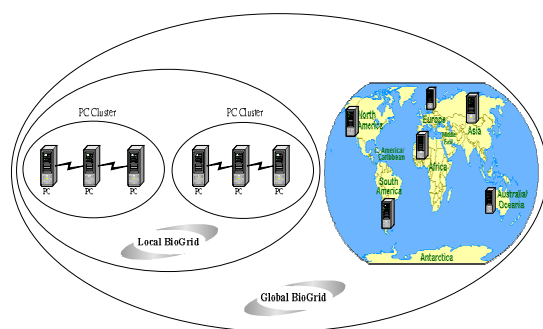


Fig 2: The BioGrid relationship diagram

2.4.1 Related Works

Recent advances in computer technology, especially grid tools makes them good candidate for

development of user interfaces to computing programs and resources. Computational grids enable sharing a wide variety of geographically distributed resources and allow selection and aggregation of distributed resources across multiple organizations for solving large scale computational and data intensive problems in science. There are some BioGrid projects and works proceed on the world. Such as EuroGrid BioGrid [13], Asia Pacific BioGrid, UK BioGrid [14, 15], North Carolina (NC) BioGrid [16, 17], Osaka University BioGrid, Indiana University BioGrid, Minnesota University [18] and Singapore BioGrid [3] etc. The EuroGrid aim of Grid opens new perspectives for bioinformatics. The integration of a platform dedicated to biology into GRID opens up new perspectives in terms of computing resources and data storage. Many genomes have been sequences and their annotation requires larger and larger databases. The storage and the exploitation of these genomes and of the huge flux of data coming from post-genomics put a quickly growing pressure on the computing tools and resources in the laboratories. The UK BioGrid is called MyGrid. MyGrid is an e-Science Grid project that aims to help biologists and bioinformaticians to perform workflow-based in silico experiments, and help to automate the management of such workflows through personalization, notification of change and publication of experiments. The NC Grid is providing the computing data storage, and networking capabilities to support the genomics revolution; members of the North Carolina Genomics and Bioinformatics Consortium are working with computer and networking companies to create the North Carolina Bioinformatics Grid. An information network will be built at Indiana University for large data and computationally intensive applications in several sciences, using advanced data grid technologies. With national and international collaborations in physics, bioinformatics, geology, and computer science, this will provide scientists access to local and globally distributed computing resources. The Singapore BioGrid is applying Clustal-G on Grid system to implement sequence alignment. These projects of the world are all use grid technology to implement bioinformatics. And the greater part wants applying the grid system to save much time to solve the bioinformatics problem. Therefore the BioGrid technology is the most popular and effective about solving biology problems.

3 Parallel Bioinformatics Applications

3.1 BLAST and mpiBLAST

3.1.1 BLAST

The most popular tool for searching sequence databases is a program called BLAST (Basic Local Alignment Search Tool). BLAST compares two sequences by trying to align them, and is also used to search sequences in a database. The algorithm starts by looking for exact matches, and then expands the aligned regions by allowing for mismatches. It performs pairwise comparisons of sequences, seeking regions of local similarity, rather than optimal global alignments between whole sequences.

Here are the four main executable programs in the BLAST distribution [8, 9]:

- [*blastall*]
Performs BLAST searches using one of five BLAST programs: *blastn*, *blastp*, *blastx*, *tblastn*, or *tblastx*

The following table summarizes the query, database sequence, and alignment types for the various BLAST commands.

Program	Query sequence type	Database sequence type	Alignment sequence type
<i>blastn</i>	nucleotide	nucleotide	nucleotide
<i>blastp</i>	protein	protein	protein
<i>blastx</i>	nucleotide	protein	protein
<i>tblastn</i>	protein	nucleotide	protein
<i>tblastx</i>	nucleotide	nucleotide	protein

- [*blastpgp*]
Performs searches in PSI-BLAST or PHI-BLAST mode. *blastpgp* performs gapped *blastp* searches and can be used to perform iterative searches in psi-blast and phi-blast mode.
- [*bl2seq*]
Performs a local alignment of two sequences. *bl2seq* allows the comparison of two known sequences using *blastp* or *blastn* programs. Most of the command-line options for *bl2seq* are similar to those for *blastall*.
- [*formatdb*]
formatdb is used to format protein or nucleotide source database. It converts a FASTA-format flat file sequence database into a BLAST database

3.1.2 mpiBLAST

Using the ubiquitous parallel-programming library called MPI: Message Passing Interface, mpiBLAST [7] segments a database into several fragments such that each node in a computational cluster searches a unique portion of the database. Database segmentation offers two primary advantages over existing parallel BLAST algorithms. First, the current size of sequence databases is larger than core memory on most computers, forcing BLAST searches to use disk I/O. Segmenting the database permits each node to search a smaller portion of the database, eliminating disk I/O and vastly improving BLAST performance. Second, because database segmentation does not create heavy interprocessor communication demands, it allows BLAST users to take advantage of power-efficient, space-efficient, low-cost clusters. The mpiBLAST is a freely available open source parallelization of NCBI BLAST. The mpiBLAST segments the BLAST database and distributes it across cluster nodes, permitting BLAST queries to be processed on many nodes simultaneously. mpiBLAST is based on MPI.

3.2 FASTA

The popular tool for searching sequence databases is a program called FASTA. FASTA compares two sequences by trying to align them, and is also used to lookup sequences in a database. FASTA provide very fast searches of sequence databases.

FASTA distribution contains search programs that are analogous to the main BLAST modes, with the exception of PHI-BLAST and PSI-BLAST, as well as programs for global and local pairwise alignment and other useful functions. The FASTA programs listed here all compile easily on a Linux system:

- [*fasta*]
Compares a protein sequence against a protein database or a DNA sequence against a DNA database using the FASTA algorithm.
- [*ssearch*]
Compares a protein sequence against a protein database or DNA sequence against a DNA database using the Smith-Waterman algorithm.
- [*fastx/fasty*]
Compares a DNA sequence against a protein database, performing translations on the DNA sequence.
- [*tfastx/tfasty*]
Compares a protein sequence against a DNA database, performing translations on the DNA sequence database.
- [*align*]
Computes the global alignment between two DNA or protein sequences.

- [*lalign*] Computes the local alignment between two DNA or protein sequences.

The FASTA package contains many programs, and they are inconveniently named after both the version number of the package and the parallel programming library that was used to build them. Nicknames are provided for most programs in the following table.

Nickname(s)	Binary
fasta	mp34compfa
ssearch	mp34compsw
fastx	mp34compfx
fasty	mp34compfy
tfastx	mp34comptfx
tfasty	mp34comptfy

3.3 HMMer

Profile hidden Markov models (profile HMMs) can be used to do sensitive database searching using statistical descriptions of a sequence family's consensus. HMMer uses profile HMMs for several types of homology searches. HMMer is a software package which is an implementation of profile hidden Markov model (HMM) methods for sensitive database searches using multiple sequence alignment as queries. About HMMer's sequence file format, it attempts to read most common biological sequence file formats. The programs automatically detect what format the file is in and whether the sequences are DNA, RNA, or protein. List some HMMer tools here:

- [*hmmpfam*] Searches a profile HMM database with a query sequence, trying to annotate an unknown sequence.
- [*hmmindex*] Create a binary SSI index for HMM database.
- [*hmmsearch*] Searches a sequence database with a profile HMM, looking for more instances of a pattern in a sequence database.
- [*hmmalign*] Align multiple sequences to a profile HMM.
- [*hmmbuild*] Builds a profile HMM from a multiple sequence alignment.
- [*hmmcalibrate*] Reads an HMM and calibrates its search statistics.
- [*hmmconvert*] Converts an HMM into other profile formats.
- [*hmmemit*] Generates sequences probabilistically based on a profile HMM. It can also generate a consensus sequence.
- [*hmmfetch*] Retrieves a profile HMM from HMM database.

3.3.1 Hmm Database-Pfam Database

Pfam [10] is a database of alignments of protein domain families and a database of profile Hidden Markov Models. Pfam includes two sub-dataset: Pfam-A and Pfam-B. Pfam-A contains over 2700 gapped profiles, and most of them cover whole protein domains; Pfam-B entries are generated automatically by applying a clustering method to the sequences left over from the creation of Pfam-A. Pfam-A entries being with a "seed alignment", it is a biologically meaningful multiple sequence alignment and sometimes may involve some manual editing.

4 Our System Environment

4.1 Linux PC Cluster

Our cluster is a low cost Beowulf-type class supercomputer and connected by one 24-port 100Mbps Ethernet switches with Fast Ethernet interface. There are one server node and seven computing nodes. The server node has two AMD ATHLON MP 2000+ processors and 1GBytes of shared local memory. Each AMD ATHLON processor has 128K on-chip instruction and data caches (L1 cache), a 256K on-chip four-way second-level cache with full speed of CPU. Each computing node has dual AMD ATHLON MP 1800+ with 512MB shared-memory.

4.2 Sun Fire 6800 Server

The following experiment is base on Sun Fire 6800 SMP [11] system with 8 CPUs and 8GB main memory. In our system, we configure on the Sun Fire[tm] 6800 as a domain which with 8 CPUs, 8GB main memory and setup by solaris 8 (5.8) operation system. This NUMA machine built from 4-processor building blocks ("quads") interconnected with a fast switch that delivers 9.6GB/sec. In each quad, it is a UMA SMP.

4.3 The Grid System

The test environment, described in the next table, we build 2 clusters to form a multiple cluster environment. Each cluster has two slave nodes and one master node. Each nodes are interconnected through 3COM 3C9051 10/100 Fast Ethernet Card to Accton CheetahSwitch AC-EX3016B Switch HUB; Each master node is running SGE QMaster daemon and SGE execute daemon to running, manage and monitor incoming job and Globus Toolkit v2.4.

Cluster 1		
Grid	Grid1*	Grid2
grid.hpc.csie.thu.edu.tw	grid1.hpc.csie.thu.edu.tw	grid2.hpc.csie.thu.edu.tw
140.128.101.172	140.128.101.188	140.128.101.189
Intel Pentium 3 - 1Ghz x2	Intel Celeron 1.7GHz	Intel Celeron 300MHz
512MB SDRAM	768MB DDR RAM	256MB SDRAM
Cluster2		
Grid3*	Grid4	Grid5
grid3.hpc.csie.thu.edu.tw	grid4.hpc.csie.thu.edu.tw	grid5.hpc.csie.thu.edu.tw
140.128.102.187	140.128.102.188	140.128.102.189
Intel Celeron 1.7GHz	Intel Pentium 3 - 866Mhz x2	Intel Pentium 3 - 366MHz
256MB DDR RAM	512MB DDR RAM	256MB SDRAM

Table 1. Hardware Configuration

5 Experimental Results

5.1 The Experimental Results on PC Cluster

5.1.1 The Performance of mpiBLAST

The software is execution by 2, 4, 6, and 8 nodes to compare the time. According to the Figure 3, we can easily to discover the speedup is near two times as we used the four nodes to executing the mpiBLAST.

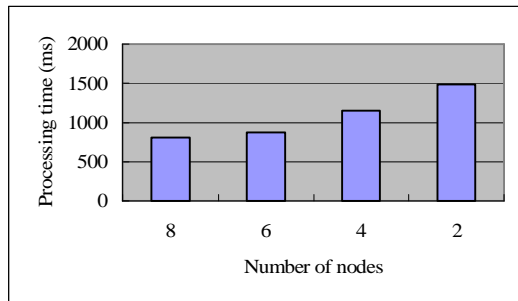


Figure 3: The average execution time of mpiBLAST using nodes from 2 to 8.

5.1.2 The Performance of HMMer

We used 2, 4, 6, 8, 10, 12, 14, 16-processor's to calculate the execution time. It is get more speed up

and save more time to do multiple sequences alignments.

When we use 4 processors to executing the software, it saved about a half time. Therefore, the speedup is near two degree which compare with 2 processors. And it produced near two degree speedup as we used 8 processors which compares with 4 processors. The Graph is draw below (Figure 4).

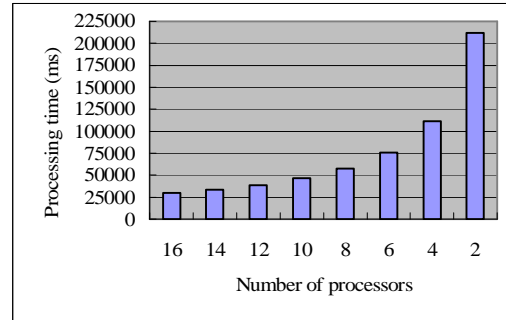


Figure 4: The average execution time of HMMer using processors from 2 to 16.

5.2 The Experimental Results on Sun Fire 6800

5.2.1 The Performance of FASTA

It is execution by 2, 3, 4, 5, 6, 7 and 8 processors to compare the time. According to Figure 5, we can easily to discover the speedup is near two times as we used the four nodes to executing the FASTA.

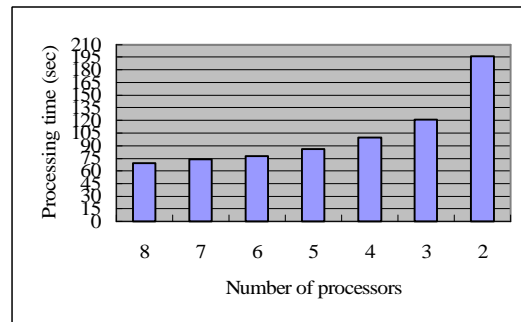


Figure 5: The average execution time of FASTA using processors from 2 to 8.

5.2.2 The Performance of HMMer

We use 2 processors to executing the software, it saved about a half time. Therefore, the speedup is near two degree which compare with 1 processor. It produced near two degree speedup as we used 4 processors which compares with 2 processors. The Graph is draw below (Figure 6.).

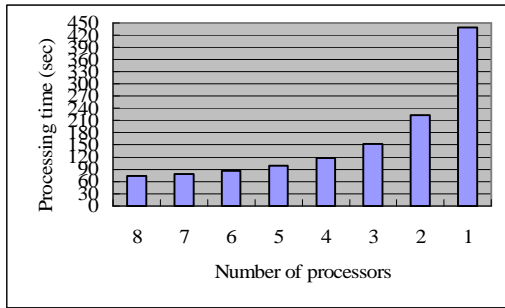


Figure 6: The average execution time of HMMer using processors from 1 to 8.

5.3 The Experimental Results on Grid System

The experiment on our grid system is used FASTA software. Using MPICH-G2 re-compile FASTA and let it run on grid. About the experimental result statement such like the Table 6. The result data is used np3 (2 CPU work) and np5 (4 CPU work) respectively. According to the statistics, we can know that the performance has obvious improvement and it can save about one-third time.

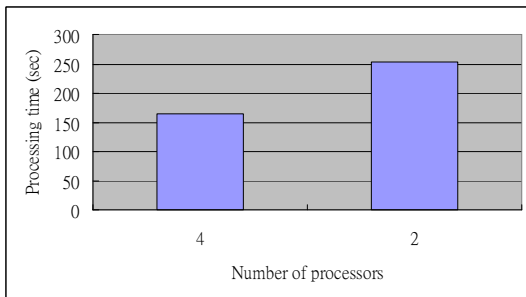


Table 6. The FASTA experimental result about np3 (2 CPUs) and np5 (4 CPUs) on grid system

6 Conclusions

In this paper, we experiment some parallel bioinformatics tools on PC Clusters, Sun Fire 6800 SMP and Grid System. And we compare the parallel alignment software performance and record the execution time. According to the experimental result, we know that the parallel computer and grid system can save more time for sequence analysis. Therefore, the parallel version bioinformatics tools can help us reduce the waiting time of alignment and improve performance about sequence alignment.

References

- [1] Oswaldo Trelles. On the Parallelization of Bioinformatic Applications.
- [2] PARACEL BLAST-Accelerated BLAST software optimized for Linux clusters.
- [3] Kuo-Bin Li, ClustalW-MPI: ClustalW Analysis Using Distributed and Parallel Computing
- [4] Karsten Hokamp, Denis C. Shields, Kenneth H. Wolfe and Daniel R. Caffrey. Wrapping up BLAST and other applications for use on Unix clusters.
- [5] Heiko A. Schmidt, Korbinian Strimmer, Martin Vingron and Arndt von Haeseler. TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing.
- [6] <http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html>, *NCBI GenBank main page*.
- [7] <http://mpiblast.lanl.gov/index.html>, *mpiBLAST main page*
- [8] <http://www.ncbi.nlm.nih.gov/BLAST/>, *NCBI BLAST main page*
- [9] <http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html>, *NCBI BLAST information guide*
- [10] <http://pfam.wustl.edu/>, *Pfam Database Home*
- [11] <http://www.sun.com/servers/midrange/sunfire6800/>, *Sun Fire 6800 Server main page*
- [12] Michael Karo, Christopher Dwan, John Freeman, Jon Weissman, Miron Livny, Ernest Retzel. Applying Grid Technologies to Bioinformatics.
- [13] BioGRID – An European grid for molecular biology
- [14] On the Use of Agents in a Bioinformatics Grid, University of Southampton, University of Manchester, University of Nottingham, University of Newcastle, University of Sheffield and EMBL Outstation.
- [15] <http://www.mygrid.org.uk>, *the UK MyGrid project site*.
- [16] <http://www.ncbiogrid.org>, *North Carolina Bioinformatics Grid (BioGrid) web site*.
- [17] <http://www.ncgbc.org>, *North Carolina Genomics and Bioinformatics Consortium web site*.
- [18] Applying Grid Technologies to Bioinformatics. Michael Karo, Christopher Dwan, John Freeman, Jon Weissman, Miron Livny, Ernest Retzel.