Parallel Protein Information Analysis (PAPIA) System Running on a 64-Node PC Cluster

Yutaka Akiyama akiyama@rwcp.or.jp
Kentaro Onizuka onizuka@rwcp.or.jp
Tamotsu Noguchi noguchi@rwcp.or.jp
Makoto Ando ando@rwcp.or.jp

Parallel Application Laboratory, Real World Computing Partnership (RWCP)
Tsukuba Mitsui Bldg., 1-6-1 Takezono, Tsukuba 305-0032, Japan

Abstract

Protein information analysis is widely regarded as a key technology in drug design, macromolecular engineering, and understanding genome sequences. Because vast amount of calculations are required, further speed-up for protein information analysis is very much in demand. We have implemented the PAPIA (PArallel Protein Information Analysis) system on the RWC PC cluster IIa ("PAPIA cluster") which consists of 64 Pentium Pro 200MHz microprocessors. The PAPIA system performs fast parallel processing for typical calculations in protein analysis, such as structure similarity search, sequence homology search and multiple sequence alignment, nearly 60 times faster than a single processor. We have started a WWW service (`http://www.rwcp.or.jp/papia/`), allowing any biologist to easily submit jobs to the PAPIA system through a WWW browser. The user can experience the power of current parallel processing technology.

1 Introduction

Recently, analysis of protein molecules and genetic DNA sequences has depended heavily on the power of computational approaches. "Computational biology" has been rising as a new and active area of study which is necessary for understanding disease mechanisms, designing drugs and macromolecular materials, and improving agricultural resources. Molecular biology databases are rapidly growing in size (a 10 fold increase in 4 years, in the case of the GenBank DNA database) and the entire genomic sequencing of several organisms has or will be finished in a few years. Even the human genome sequence will be almost fully obtained by 2005. Vast amount of calculations are required for the analysis of huge databases, but fortunately, large-scale parallelism can be exploited in most cases.

From very early on, computational biology research has been carried out by utilizing the power of the Internet. Hundreds of WWW service sites are now available on the Internet. However only a few sites or programs are employing parallel calculation techniques (*e.g.* BLAST [2] is capable of small-scale multithread processing on SGI, the MPsrch [14] program (BLITZ/BLAZE server) for Smith-Waterman algorithm has been used on the MasPar parallel computers).

We are aiming at demonstrating the power of the latest parallel calculation technology in efficiently solving computational biology problems. For this purpose, we have developed the **PAPIA** (**PA**rallel **P**rotein **I**nformation **A**nalysis) system [1].

In this paper, we will describe the development of the PAPIA system. A newly assembled PC cluster dedicated to the PAPIA system is shown in section 2. In section 3, the implementation as well as the performance of the PAPIA system is described. In section 4, usage of the PAPIA system through the WWW is introduced.

Table 1: History of RWC Cluster Developments.

1995	Aug.	RWC WS Cluster I
		SPARCstation 20, 9 nodes
1996	Feb.	RWC WS Cluster II
		SPARCstation 20, 36 nodes
	Oct.	RWC PC Cluster I
		Pentium 166MHz, 32 nodes
1997	Oct.	RWC PC Cluster II
		Pentium Pro 200MHz, 64 nodes (now 128 nodes)
1998	Feb.	**RWC PC Cluster IIa (PAPIA)**
		Pentium Pro 200MHz, 64 nodes
	Aug.	RWC Alpha Cluster I
		DEC Alpha 21164 500MHz, 32 nodes

2 RWC PC Cluster

The RWCP Tsukuba Research Center has been building workstation clusters and PC clusters (see Table 1) as environments for research and development of parallel operating systems and parallel programming languages since 1995.

The RWC PC Cluster II was developed in October 1997 and is a highly efficient parallel computer with powerful network connections. The NPB benchmark results show its performance is comparable to the latest massively parallel processors (`http://www.rwcp.or.jp/lab/pdslab/`).

The PAPIA system prototype was implemented on the RWC PC Cluster II and was demonstrated at the SC97 conference exhibition. We then built a new application-dedicated cluster, the RWC PC Cluster IIa, "**PAPIA cluster**", in February 1998. The PAPIA system described in this paper is operated using the new PAPIA cluster.

The RWC PC Cluster series all have the following features:

- Connected by a high-speed network (Myrinet) with an originally-developed driver(PM) [15],
- Original parallel (global) operating system (SCore-D) [8] which enables parallel multi-user environment with gang-scheduling,
- Compact packing in originally designed chassis bodies, and
- Unix (NetBSD or Linux) based highly portable environments.
  The MPICH-PM (MPI compatible) communication library is available.

2.1 PAPIA Cluster (RWC PC Cluster IIa)

We have built an application-dedicated PC cluster, the RWC PC Cluster IIa ('a' stands for application) or the "PAPIA cluster". The system specifications are shown in Table 2 and a view of the PAPIA cluster is shown in Fig. 1. Each node of the PAPIA cluster has 256MB of memory and a 4.1GB hard disk. The disk capacity is expanded from 2.1GB (RWC PC cluster II) to 4.1GB in order to store a whole copy of the Protein Data Bank (PDB) [4] and other databases on each node.

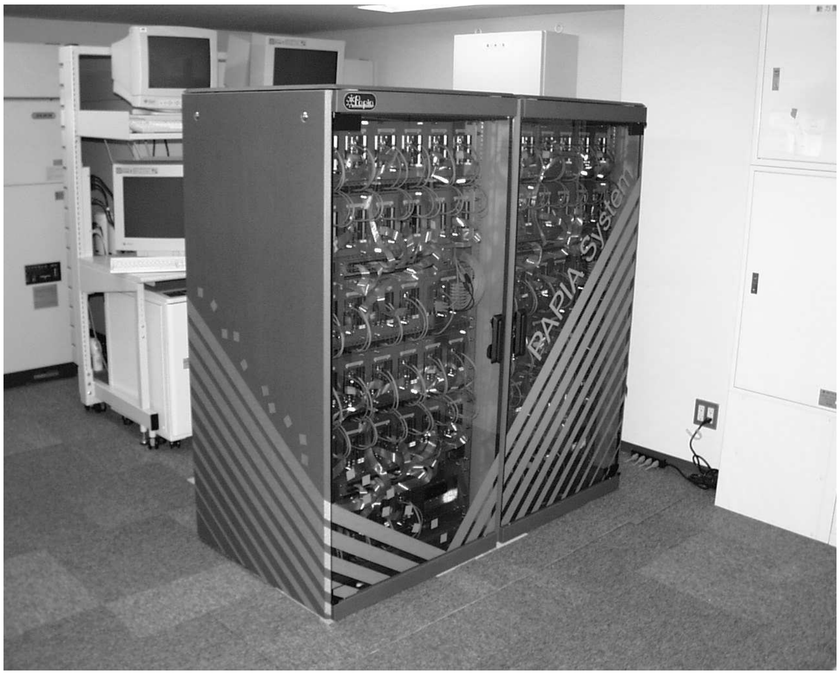

Figure 1: PAPIA Cluster (RWC PC Cluster IIa).

Table 2: PAPIA Cluster Specifications.

No. of Nodes	64 calculation nodes +2 monitor nodes
Node Processor Main Memory Hard Disk	Intel Pentium Pro, 200MHz (8KB L1, 512KB L2 cache) 256MB ECC DRAM /node 4.1GB EIDE HD /node
Network H/W Network Driver	Myrinet 1.28Gbit/sec/link, 100 Base-T Ethernet **PM** [15]
Basic OS Parallel OS Languages	NetBSD **SCore-D** [8] **MPC++** [9], C++, C, **MPICH-PM** comm. library
Physical Size	W80cm $\times$ D80cm $\times$ H160cm $\times$ 2 Chassis

Some minor differences between the PAPIA cluster and the original RWC PC cluster II are:

- Hard disk expanded from 2.1GB to 4.1GB,
- Automatic shutdown/reboot system with UPS,
- Node position rearrangement for simplifying Myrinet connection pattern, and
- Stronger cooling fans and additional floor fans for cool air intake.

3 PAPIA System

In order to rapidly and efficiently develop application software for protein analysis, making a library of commonly used program modules is important. However, there is currently no common library for protein research. One reason might be the highly-complex semantics and notoriously ill-defined format of the Protein Data Bank (PDB) [4], the principal database of protein tertiary structures.

Onizuka *et.al* [13] have developed a C++ class library, the "**PAPIA library**", for protein information analysis. In the PAPIA library, a protein structure is hierarchically described with clear object-oriented *class* definitions. Commonly required calculations, like PDB parsing, geometric rotation, structure matching, sequence alignment, multivariate analysis, and so on are defined as *methods* of the corresponding classes. We have been using the PAPIA library for several research themes in protein information analysis.

Using the PAPIA library, we have also been building an assorted collection of efficient parallel programs used for typical protein analysis. We have ported this collection of practical programs onto the PAPIA cluster. We call the total system the "**PAPIA system**" which includes the collection of programs and the hardware/software system itself.

Significant functions in the current PAPIA system are the following three parallel calculations.

1. Parallel Protein Structure Similarity Search
   (Kabsch method)
2. Parallel Protein Sequence Homology Search
   (Needleman-Wunsch algorithm)
3. Parallel Protein Multiple Sequence Alignment
   (an extended Berger-Munson algorithm)

3.1 Implementation

A single calculation server program for the PAPIA system was designed. Any calculation service can be performed by the same server program with a specific function call. The server program is launched simultaneously on all calculation nodes, at the booting of the PAPIA cluster. The first node works as a master node communicating with the client outside. The other nodes work as worker nodes (master-worker model).

The PAPIA calculation server program resides on each node as a daemon. Any calculation request from the outside will be received as a socket communication (specifying a service and input parameters). The master node dispatches sub-tasks to worker nodes and waits for answers from the workers. In general, dispatching will be repeated until a calculation task is finished. The result list is sorted and edited by the master before it is sent back to the client.

In order to speed-up a database search, each worker node reads the specified (dispatched) part of the database into the local memory at booting. Because of the limited memory capacity (256MB), each worker node is in charge of searching only a small fraction of the database. The master node decides an arrangement for data partitioning. Because each node has an entire copy of the databases on its local disk, the data partitioning and dispatching may be dynamically changed at the cost of loading the data from the local disk. In the current implementation, no pair of nodes works on the same database entry, that is, dispatching is done in a non-redundant way.

Communication between nodes has been implemented through two different methods. The first implementation uses the MPC++ programming language developed by Ishikawa [9]. MPC++ provides object-oriented remote function invocation.

The second implementation uses the MPICH-PM (MPI) communication library for message-passing between processes on different nodes. A drawback of this approach is that a programmer has to design and generate communication packets for transmitting complex protein objects.

The authors prefer the former approach, however, we provide the MPI implementation to guarantee easy porting to other parallel environments.

An important point is to avoid memory overflow in the gathering phase. This is rather difficult because the size of the result of database searching varies drastically and we cannot easily predict it. Smart memory management, or a less-centralized mechanism will be required in the future.

To improve load balancing, protein data is sorted by length and dispatched to the worker nodes in that order. However the effect of sorting is limited because computation time is not always proportional to the sequence length especially in the case of structure comparison. There is no *a priori* best dispatch, without knowing the characteristics of the incoming queries.

3.2 Parallel Efficiency

The parallel efficiency of the PAPIA system is excellent for protein structure similarity searches and protein sequence homology searches, because these parallelized database searches are performed by a simple data partitioning scheme. The PDB protein structure database has more than 7,500, and SWISS-PROT protein sequence database has more than 74,000 entries. We can assume large granularity in our data partitioning scheme (even if we use a non-redundant representative set of the PDB called the PDB-REPRDB [12], which consists of about 2,800 chains). Overhead for the dispatching and gathering phase is negligible.

Multiple alignment of protein sequences is not a such trivial case. All processor nodes must frequently exchange current alignment status. Our method [16] is an extension and parallelization of the Burger-Munson type iterative refinement algorithm [7]. It is a kind of parallel hill-climbing search and the best move found by a worker (not only the score but also the alignment) must be sent back to the master in each calculation step. Thus the ratio of communication versus calculation is much larger than the simple database scanning cases.

Even for this calculation, experiments showed good parallel efficiency on the PAPIA cluster. Fig. 2 shows the execution time of our precise multiple alignment algorithm for a middle-range example problem (64 *kinase* enzyme sequences, length 80 amino acids).

The horizontal axis shows the number of worker nodes (log scale), and the vertical axis shows execution time in seconds (log scale). The alignment takes 79 minutes with a single node, and takes only 89 seconds with 64 nodes (parallel efficiency is 0.83).

Thus we have succeeded in providing a very accurate multiple alignment algorithm [16], which used to take hours, in an interactive fashion. It will improve the way biologists discover evolutionarily conserved blocks of protein sequences.

4 Using PAPIA through WWW

We have implemented a job submission mechanism using a WWW browser as a user interface for entering query sequences and several other parameters. By means of the WWW, any user can easily submit jobs from a remote site on the Internet.

The mechanism consists of the following modules: a) HTML-based input forms as a user interface, b) CGI scripts for submitting and monitoring jobs on the PAPIA system, c) FIFO queues for each service, and d) JAVA and HTML-based graphic output screens as the user interface.

We have operated a WWW service for the PAPIA system since April 1998. The six services currently available on the **PAPIA WWW** page (Fig. 3) at `http://www.rwcp.or.jp/papia/` are described below.

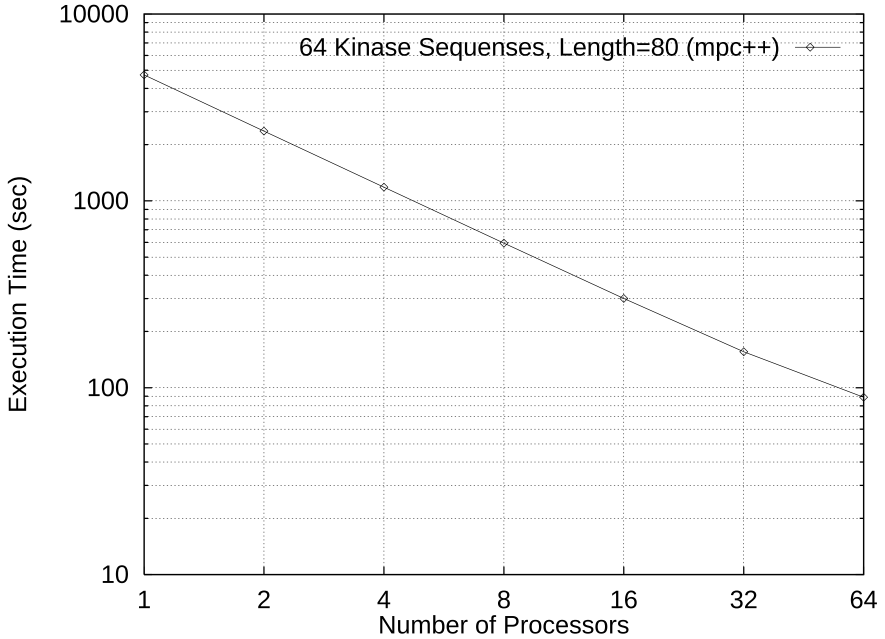

Figure 2: Execution Time for Protein Sequence Multiple Alignment on the PAPIA Cluster, (64 kinase sequences, 80 amino acids length, horizontal: No. of nodes, vertical: Time in seconds, log-log scale).

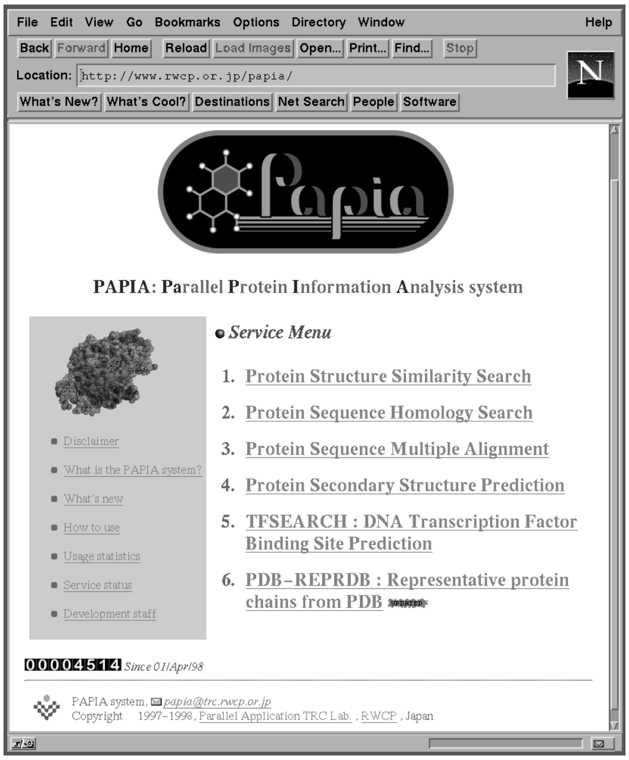

Figure 3: PAPIA WWW Top Page (`http://www.rwcp.or.jp/papia/`).

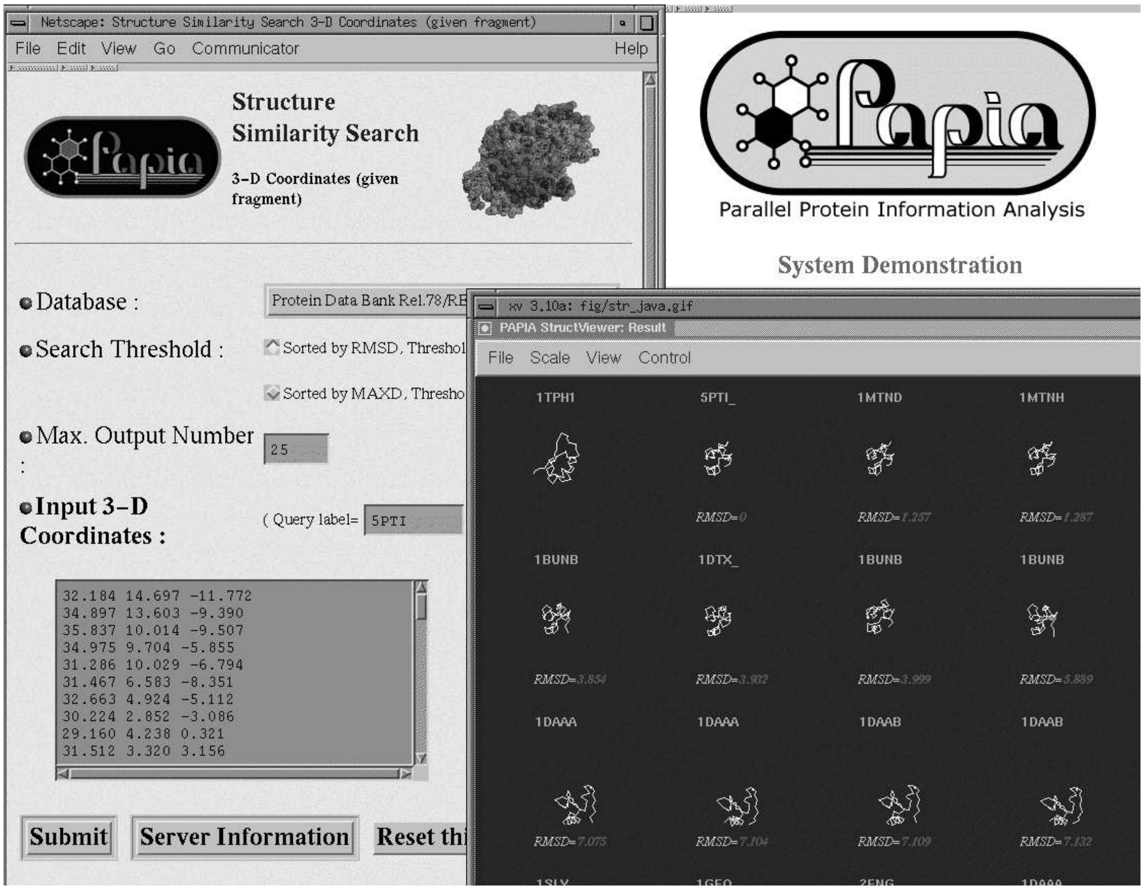

Figure 4: Protein Structure Similarity Search on PAPIA.

4.1 Protein Structure Similarity Search

This is a parallelized search on the PDB protein structure database for full or partial structures similar to a given query fragment. Main-chain skeleton structures composed of C_α carbons are the subject of comparison. Three dimensional best fit structures are searched through the PDB via precise residue by residue comparisons (Kabsch method). Discovered similar structures are reported through the browser sorted by matching scores. The result list is clickable and is linked to the corresponding PDB entries. Some of them are further linked to the DBGET service [5] on the GenomeNet. The graphical structure viewer using JAVA can also be launched from the result list (Fig. 4).

4.2 Protein Sequence Homology Search

This is a parallelized search of the SWISS-PROT protein sequence database (or the PDB database as a sequence database) for sequences which are evolutionarily related (homologous) to a given query sequence. The rigorous Needleman-Wunsch algorithm, based on dynamic programming, is employed for detecting even weak relationships. (A local alignment based on Smith-Waterman algorithm will also be made available soon.) The result list is also clickable. Related protein functions, chemical reactions, genetic diseases, and literature databases can be accessed by following these links. In addition, a user can easily obtain the multiple alignment of the query and the detected sequences by just selecting (by click) and sending them to the multiple alignment service described next.

Figure 5: Protein Multiple Sequence Alignment on PAPIA.

4.3 Protein Multiple Sequence Alignment

The aim of this calculation is aligning evolutionarily related sequences so that the related amino acid residue characters are lined up in the same row to show blocks conserved through evolution. This is a combinatorial optimization problem and previous programs like ClusterW are based on rough approximation methods using a tree-based tournament. On the other hand, our original algorithm [16] is an extension and parallelization of an accurate iterative method. We also employ an A^*-pruning technique in some cases as well as parallelization to realize a major speed up than our original sequential program. An accurate multiple alignment can now be obtained in a few minutes on the PAPIA system. The characters in the resulting alignment are displayed in six different colors to emphasize the physico-chemical tendencies of locally aligned blocks.

4.4 Protein Secondary Structure Prediction

The New Joint method developed by Nishikawa and Noguchi [11] was implemented and is fully parallelized. The SSThread program [10] based on 3D-1D method has recently been added in order to improve prediction accuracy. This calculation usually runs not on the PAPIA cluster but on a SGI machine at our site because the calculation uses only 10 processors. (For user's convenience, this calculation is provided as a part of the PAPIA WWW service.)

4.5 DNA Transcription Factor Binding Site Prediction: TFSEARCH

The TFSEARCH program developed by Akiyama is provided to predict transcription factor binding sites by calculating correlation scores with the TFMATRIX consensus matrix database [6]. This service has already been used by biologists world-wide with about one hundred accesses per day. (This and the following service also run on a different machine than the PAPIA cluster.)

4.6 Non-redundant Protein Structure Database: PDB-REPRDB

PDB-REPRDB is a non-redundant representative protein chain database developed by Noguchi [12]. 48 representative chain sets are provided, according to different sequence homology thresholds and structure similarity thresholds. In the future, we are planning to allow the 'on-line' dynamic construction of a representative set where a user can define the ranking policies and the similarity thresholds.

4.7 Usage statistics

The PAPIA WWW service has been accessed by many users since the public opening at April 1998. Table 3 shows the usage counts for each calculation service, while 'WWW total' counts include all accesses to the WWW content. The TFSEARCH service was already available before April 1998 and is being frequently accessed by foreign researchers. Massively parallel services have been used from at least 10 countries. Detailed statistics information is available on the PAPIA WWW page.

Table 3: PAPIA WWW service usage statistics.

	Structure similarity	Sequence homology	Multiple alignment	Secondary structure	TFSEARCH	PDB-REPRDB	WWW Total
Japan	2815	2068	1892	2037	6200	1159	148202
Foreign	130	105	111	94	20855	206	80730
Unknown	146	209	169	120	9602	15	47528
Total	3091	2382	2172	2251	36657	1380	276460

(Statistics period: January 1, 1998 through October 24, 1998)

5 Discussion

The PAPIA WWW service still has limitations in terms of practical usefulness.
1) query size: We have set a short calculation time-out for each service in order to avoid long occupation of the cluster by a single task. In some cases this time-out is too short and thus query size is strongly limited. For example, the homology search service will reject a query with more than 300 amino acids. We can easily loosen it technically, but we had been operating our service conservatively.
2) database: The accessible databases are currently limited to PDB and SWISS-PROT because they are enough for our research activity. As an open service, however, more databases will be required. At that time we will have to consider static partitioning of databases (because of 4GB limit of disks).
3) scoring matrix: The PAPIA WWW service had provided only two scoring matrices. Now we are providing at least five important matrices (BLOSUM 45, 62, 80, PAM 120, 250) .
4) functions: The PAPIA system only provides simple calculations. For example, no sequence filtering or taxonomic restriction is available with the homology search (BLAST [2] provides them). The PAPIA homology search should not be used as a simple replacement of previous sophisticated services. BLAST is sufficient for most homology search tasks but the rigorous DP matching provided in PAPIA is suitable for weak homology cases.

For structure similarity search, however, we do not know another sufficient service. Our parallel multiple alignment service is also practical given reasonable sequence lengths and numbers.

6 Conclusion

The PAPIA system running on a 64-node PC cluster enables very fast computational analysis of proteins. A calculation which used to take hours is now carried out in minutes, and calculations that took a week can be completed less than three hours. The wide memory space (256MB $\times$ 64 = 16GB) allows on-memory database searches, and 4GB local disks provide flexibility in parallel database searches. We will keep enhancing the PAPIA system and will continue to open services through the Internet as a way to demonstrate and evaluate the parallel applications we are developing.

References

[1] Akiyama, Y., Onizuka, K., Noguchi, T., Ando, M. and Saito M., Parallel Protein Information Analysis (PAPIA) system implemented on RWC PC cluster II, *IPSJ SIG Notes, 97-HPC-70-6*, 31–36, 1998.

[2] Altschul, S. F., Madden, T. L., Schffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res.*, 25:3389-3402, 1997. (BLAST service is available at `http://www.ncbi.nlm.nih.gov/BLAST/`)

[3] Bairoch, A. and Apweiler, R., The SWISS-PROT protein sequence data bank and its supplement TrEMBL, *Nucleic Acids Res.*, 25:31–36, 1997.

[4] Berstein, F., Koetzle, T., Williams, G., Meyer, E., Brice, M., Rodgers, J., Kennard, O., Shimanouchi, T. and Tasumi M., The Protein Data Bank: A Computer-based Archival File for Macromolecular Structures, *J. Mol. Biol.*, 112:535–542, 1977. (`http://www.pdb.bnl.gov/`)

[5] Fujibuchi, W., Goto, S., Migimatsu, H., Uchiyama, I., Ogiwara, A., Akiyama, Y. and Kanehisa, M., DBGET/LinkDB: an Integrated Database Retrieval System, *Proc. of PSB'98*, 683–694, 1998.

[6] Heinemeyer, T., Wingender, E., Reuter, I., Hermjakob, H., Kel, A. E., Kel, O. V., Ignatieva, E. V., Ananko, E. A., Podkolodnaya, O. A., Kolpakov, F. A. , Podkolodny, N. L. and Kolchanov, N. A., Databases on Transcriptional Regulation: TRANSFAC, TRRD, and COMPEL, *Nucleic Acids Res.*, 26:364–370, 1998.

[7] Hirosawa, M., Totoki, Y., Hoshida, M. and Ishikawa, M., Comprehensive Study on Iterative Algorithms of Multiple Sequence Alignment, *Comput. Applic. Biosci.*, 11:13–18, 1995.

[8] Hori, A., Tezuka, H., Ishikawa, Y., Soda, N., Konaka, H. and Maeda M., Implementation of Gang-Scheduling on Workstation Cluster, *Lecture Notes in Computer Science*, 1162:76–83,Springer-Verlag, 1996.

[9] Ishikawa, Y., *RWC Tech. Repo. TR-96012*, 1996. (See also `http://www.rwcp.or.jp/lab/pdslab/mpc++/`)

[10] Ito, M., Matsuo Y. and Nishikawa, K., Prediction of protein secondary structure using the 3D-1D compatibility algorithm, *Comput. Appl. Biosci.*, 13:415–424, 1997.

[11] Nishikawa, K. and Noguchi, T., Predicting protein secondary structure based on amino acid sequence, *Methods in Enzymology*, 202:31–44, 1991.

[12] Noguchi, T., Onizuka, K., Akiyama, Y. and Saito M., PDB-REPRDB: A Database of Representative Protein Chains in PDB (Protein Data Bank), *Proc. ISMB97*, 214–217, AAAI Press, 1997.

[13] Onizuka, K., Noguchi, T. and Akiyama Y., Parallel PDB Data Retriever 'PDB Diving Booster', *Lecture Notes in Computer Science*, 1336:389–396, Springer-Verlag, 1997.

[14] Sturrock, S. S. and Collins, J. F., MPsrch version 1.3. Biocomputing Research Unit, Univ. of Edinburgh, 1993. (MPsrch service is available at `http://www.dna.affrc.go.jp/htdocs/MPsrch/`, for example.)

[15] Tezuka, H., Hori, A., Ishikawa, Y. and Sato M., PM: An Operating System Coordinated High Performance Communication Library, *Lecture Notes in Computer Science*, 1225:708–717, Springer-Verlag 1997. (See `http://www.rwcp.or.jp/lab/pdslab/` for the RWC PC cluster on-line documents.)

[16] Totoki, Y., Akiyama, Y., Onizuka, K., Noguchi, T., Saito, M. and Ando, M., Employing A* Algorithm in Parallel Multiple Protein Sequence Alignment, *IPSJ SIG Notes, 97-MPS-16-4*, 19–24, 1997.