id. 27822 SP 19160 METAS 2006

"C" int

Online journal ISSN - 1676-5680
GMR Genetics and Molecular Research
EVOLUTION AND TECHNOLOGY
Home GMR Editorial Board Current Issue All Issues Search Indexers For Authors Contact Sponsors Links

A distributed computation of Interpro Pfam, PROSITE and ProDom for protein annotation

Edward de O. Ribeiro[1], Gustavo G. Zerlotini[1], Irving R.M. Lopes[1], Victor B.R. Ribeiro[1], Alba C.M. Melo[1], Maria Emilia M.T. Walter[1], Marcos Mota Costa[2] and BIOFOCO Network
[1]Departamento de Ciência da Computação, Universidade de Brasília, Brasília, DF, Brasil
[2]EMBRAPA Recursos Genéticos e Biotecnologia, Brasília, DF, Brasil
Corresponding author: E.O. Ribeiro
E-mail: edward@cic.unb.br

Genet. Mol. Res. 4 (3): 590-598 (2005)
Received May 20, 2005
Accepted July 8, 2005
Published September 30, 2005

ABSTRACT. Interpro is a widely used tool for protein annotation in genome sequencing projects, demanding a large amount of computation and representing a huge time-consuming step. We present a strategy to execute programs using databases Pfam, PROSITE and ProDom of Interpro in a distributed environment using a Java-based messaging system. We developed a two-layer scheduling architecture of the distributed infrastructure. Then, we made experiments and analyzed the results. Our distributed system gave much better results than Interpro Pfam, PROSITE and ProDom running in a centralized platform. This approach seems to be appropriate and promising for highly demanding computational tools used for biological applications.

Key words: Distributed systems, Interpro, Computational biology, Tools

INTRODUCTION

In recent years, the availability of many and powerful computers has allowed the creation of new ways of performing intensive computational tasks at a low cost. Many computationally demanding problems could be solved using idle resources in existing fast computer networks. In fact, the scientific community has already adopted various distributed strategies, such as Peer-to-Peer (P2P) networks (JXTA, 2004), CORBA (CORBA, 2004) and Web Services (Web Services, 2004) to perform their tasks. Nevertheless, many software tools do not take full advantage of this distributed infrastructure. They are usually restricted to a previously defined, highly controlled and static environment (Foster et al., 2001).

Among the distributed technologies, grid computing has received much attention from different sectors in academia and industry for its innovative approach to distributed computing as a widespread, secure, and economically viable solution for problems requiring huge amounts of memory and storage. Foster and Kesselman (1997) defined a grid as a "coordinated resource sharing and problem solving in a dynamic, multi-institutional virtual organization. … This sharing is necessarily highly controlled, with resources, providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs''. A grid is created based on the concept of a virtual organization, that is, a dynamic group of individuals and/or organizations that use the grid infrastructures (protocols, services and programming interfaces) to decide what kind of resources (instruments, CPU cycles, data storage, RAM, software, and data) will be shared in a highly controlled manner. A particular grid has to adapt itself to the local policies of the virtual organization's members, and it must be highly scalable.

Globus (Foster and Kesselman, 1997) developed by Globus Alliance (Globus/Alliance, 2004) is a well-known system that implements the basic services of a grid. It is composed of a huge set of protocols and services to support grid computing using the Web Services infrastructure. In Brazil, MyGrid (Costa et al., 2004) is another system implementing a grid. There is considerable research on grid computing and the concepts are still under development.

Recently, computationally intensive applications for molecular biology have been developed, motivated mainly by genome-sequencing projects all around the world (NCBI, 2004; TIGR, 2004). These projects demand huge amounts of computation and storage, and so they are very interesting challenges for researchers who study distributed and parallel computation (Pande and Shirts, 2000).

We propose a distributed architecture based on a Java-based messaging system (JMS) (JORAM, 2004) to connect remote sites. Our architecture does not implement all the required functionalities of a grid system, but its use in a real system can shape our understanding of this kind of application and drive future developments. This choice was driven by the easiness of development and testing of JMS, when compared to other systems, such as Globus (Foster and Kesselman, 1997) or MyGrid (Costa et al., 2004). We installed the application on heterogeneous machines located at different institutions, and our distributed infrastructure used the available network resources.

We chose a biological application demanding huge amounts of computation. Interpro (Zdobnov and Apweiler, 2001) is a widely used tool during the annotation phase of genome sequencing projects that runs different annotation programs under a common interface. Although achieving parallelization, it does not scale well when processing a large number of sequences. It can be executed in a parallel-processing machine, but it does not execute in a distributed environment. Interpro uses many databases, and among them we selected three to use in our distributed execution environment: Pfam, PROSITE and ProDom. Pfam (Bateman et al., 2000) is a database of protein domain families. It contains multiple sequence alignments for each family and profile Hidden Markov Models for finding these domains in new sequences. Pfam can automatic classify predicted proteins into protein domain families. PROSITE (Falquet et al., 2002) is used for determining the function of proteins translated from genomic sequences using signature patterns. Its collection of biologically significant sites is used to determine if a given sequence belongs to a known family of proteins. ProDom (Servant et al., 2002) is a database of protein domain families obtained by automated analysis of the SWISS-PROT and TrEMBL protein sequences. It is useful for analyzing the domain arrangements of complex protein families and the homology relationships in modular proteins.

We noticed that Pfam is the most time-consuming database when running Interpro. It severely limits the number of sequences executed by Interpro during a period of time. For example, when processing a 1000 base-pair sequence, Pfam runs in approximately 20 min, PROSITE runs in 1.5 min and ProDom runs in 14 s.

DESCRIBING THE DISTRIBUTED INFRASTRUCTURE

At first, we decided to propose a distributed infrastructure for a biological application requiring intensive computation. We wanted to connect remote sites with heterogeneous hardware and operating systems, using available machines at different institutions. So, we proposed a distributed infrastructure composed of a two-level scheduling system-BioGridDF (Figure 1). We did not examine scheduling policies in this project.

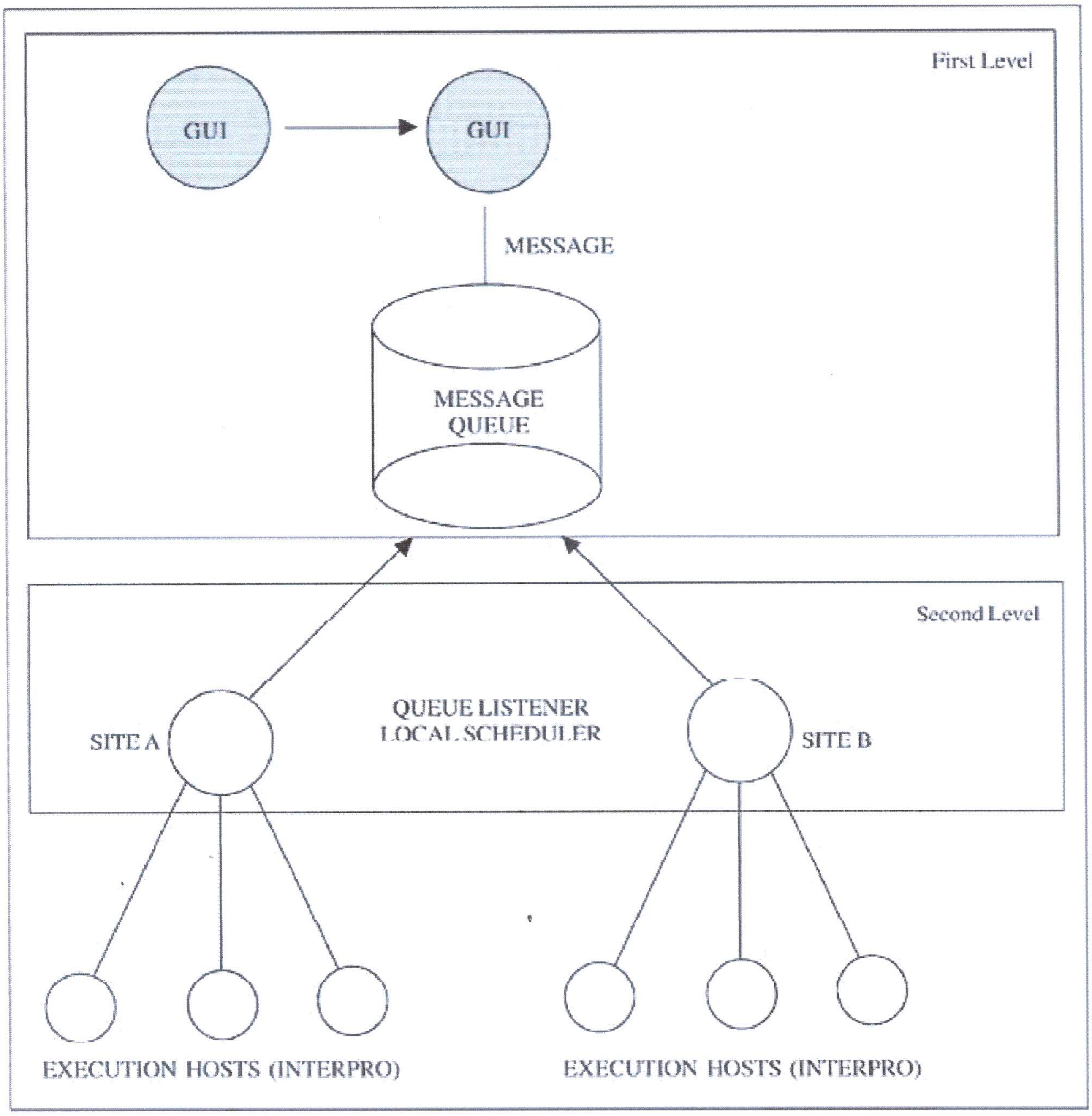
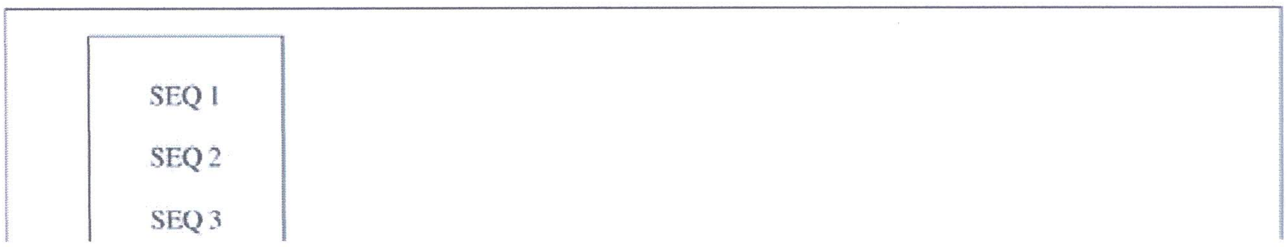

First Level
GUI
GUI
MESSAGE
MESSAGE QUEUE
Second Level
QUEUE LISTENER LOCAL SCHEDULER
SITE A
SITE B
EXECUTION HOSTS (INTERPRO)
EXECUTION HOSTS (INTERPRO)

Figure 1. The proposed distributed architecture system - BioGridDF.

The first level is composed of a Web-based Graphical User Interface, a parser to the biological sequences, a high level scheduler and a JMS server running a message queue (Figure 2). These components of the system were installed on a single machine, but they could be distributed among the participating machines.



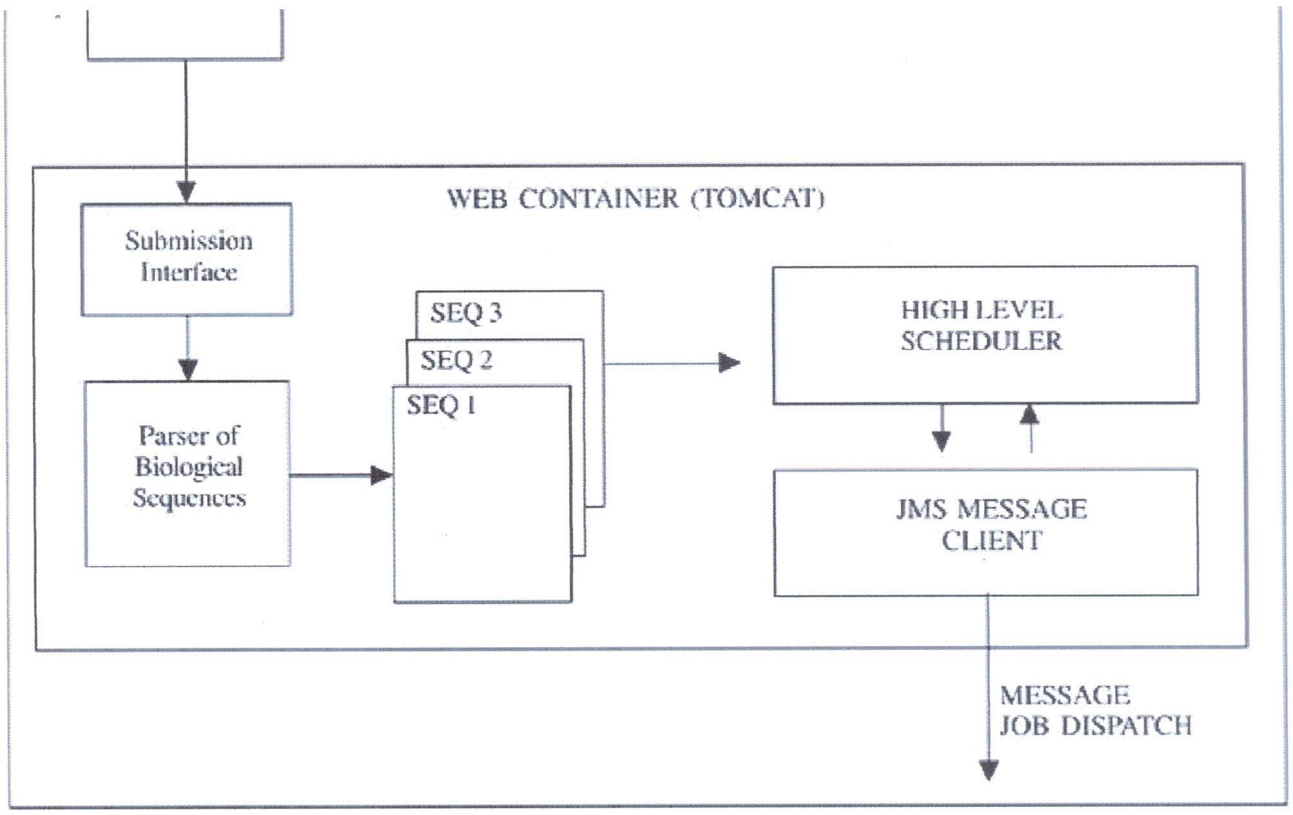
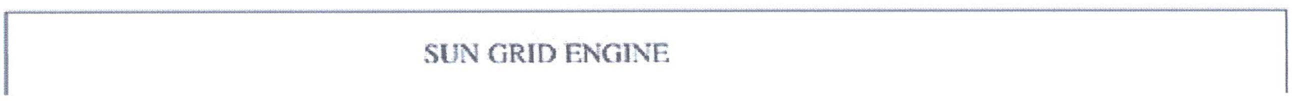


Figure 2. The first level of BioGridDF.

We adopted Java language (Horstmann and Cornell, 1997) to implement the first layer of our distributed architecture, because it offers portable, clear and efficient primitives to solve the proposed application. It offers a rich set of Application Programming Interfaces that are valuable to reduce development and debugging time. We also tried to make extensive use of XML (XML, 2004) whenever appropriate. Loosely coupled systems need a consistent way to represent data and query interfaces, and XML is a self-contained and flexible data format.

The system is centered on the JMS message queue that makes the scaling of the routing messages among the remote sites. Graphical User Interface offers Web interfaces to submit biological sequences, in which the user indicates the location of the input sequence file in FASTA format. Once the file is submitted, a parser does data pre-processing, splitting each sequence into a single file and storing all these sequence files into an NFS shared directory that is accessible from the different sites. Then, the system decides the remote site that will process each sequence and sends a message to the message queue in the JMS server. This message is composed of the input sequence data file name, the chosen database (Pfam, PROSITE or ProDom), the output directory, and the processing site name.

The second level (Figure 3) is composed of a master machine located at each remote site, which "listens" to the first level JMS message queue through TCP connections. This master machine is notified as soon as a message addressed to it is stored on the JMS message queue. Then, the request message is consumed, and this request is submitted as a job to the local scheduler. The local scheduler chooses a machine to execute the job. We used SunGrid Engine (SGE) as our local scheduler. SGE (Sun Grid Engine, 2004) is an open source program, widely used for distributed tasks, and it offers load balancing and monitoring facilities that help to control the processing jobs. Our system is not tightly bound to SGE, so it could be replaced by any other local scheduler, like Condor (Livny, 1995), for example.



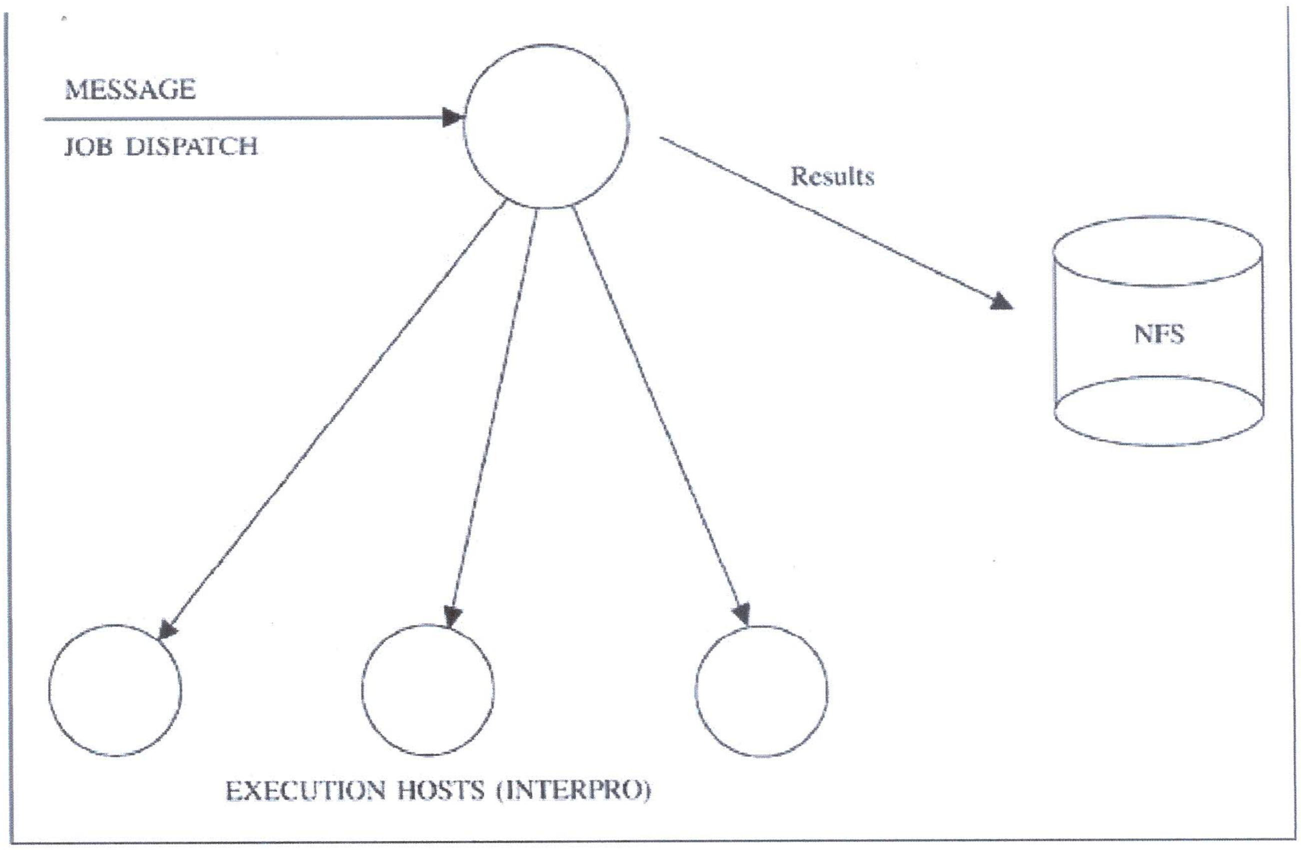


Figure 3. The second level of BioGridDF.

A biological sequence is processed using tools and resources provided by each site, so each host machine that executes the application must have the resources (programs and databases) installed. Consequently, for BioGridDF, the Interpro Pfam, PROSITE, and ProDom databases and programs were installed on each execution machine.

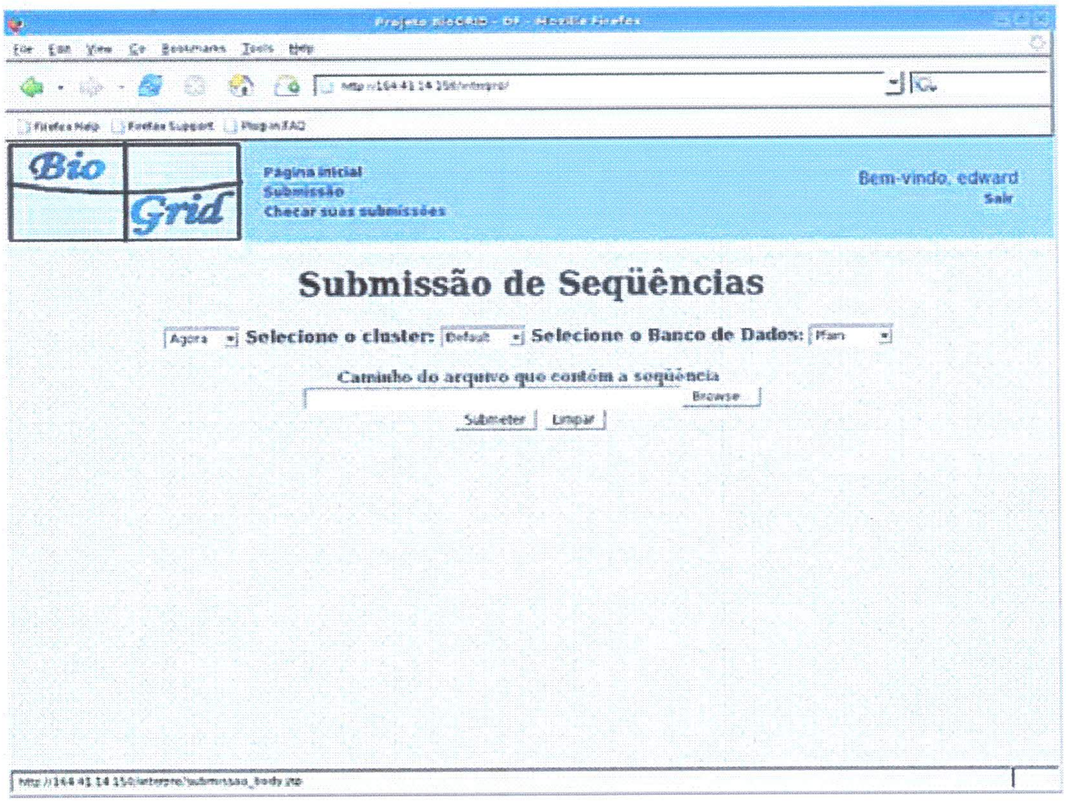
We built a prototype of the proposed distributed architecture on PCs, running different versions of the Linux operating system. These machines were installed at University of Brasília and EMBRAPA/Recursos Genéticos e Biotecnologia (Table 1).

Table 1. Description of the machines composing BioGridDF.

	EMBRAPA		University of Brasília	
	Machine 1	Machine 2	Machine 1	Machine 2
Processors	XEON 1.70 GHz	XEON 1.70 GHz	1.70 GHz	1.70 GHz
Number of processors	2	2	1	1
RAM	2 GB	2 GB	256 MB	256 MB
Cache	512 KB	512 KB	256 KB	256 KB
Swap Space	2 GB	2 GB	1 GB	1 GB
HD Space	40 GB	111 GB	28 GB	28 GB
Operating System	Fedora Core 1	Fedora Core 1	Red Hat 9	Red Hat 9
Java Virtual Machine	1.4.2_0	1.4.2_0	1.4.1_0	1.4.1_0
Sun Grid Engine	6.0	6.0	6.0	6.0
JORAM JMS Server	4.0.0	4.0.0	4.0.0	4.0.0

In Figure 4, we show some pages of our system.

A


Projeto BioGRID - DF - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://164.41.14.150/interpro/
Firefox Help
Firefox Support
Plug-in FAQ
Bio Grid
Página inicial
Submissão
Checar suas submissões
Bem-vindo, edward
Sair

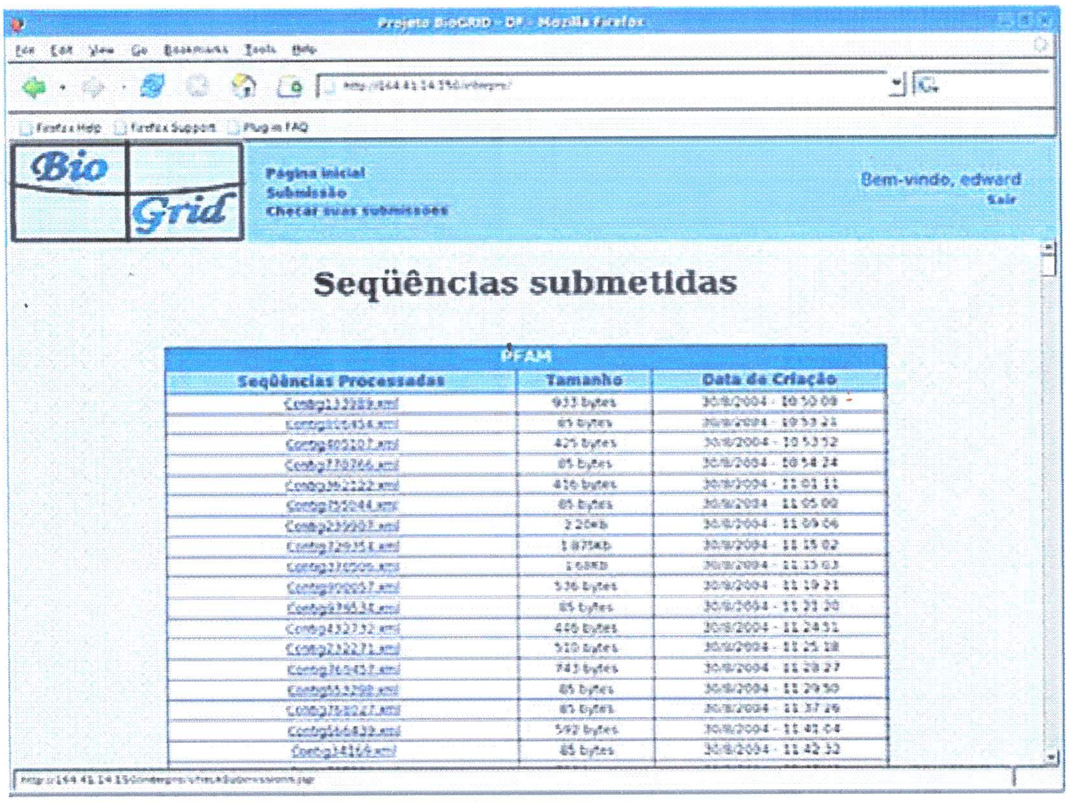
Submissão de Seqüências

Agora | Selecione o cluster: Default | Selecione o Banco de Dados: Pfam

Caminho do arquivo que contém a seqüência

Browse
Submeter
Limpar
http://164.41.14.150/interpro/submissao_body.jsp

B



Seqüências submetidas

PFAM		
Seqüências Processadas	Tamanho	Data de Criação
Contig133389.xml	933 bytes	30/8/2004 - 10:50:08
Contig016454.xml	85 bytes	30/8/2004 - 10:53:21
Contig405107.xml	425 bytes	30/8/2004 - 10:53:52
Contig770766.xml	85 bytes	30/8/2004 - 10:54:24
Contig362122.xml	416 bytes	30/8/2004 - 11:01:11
Contig755044.xml	85 bytes	30/8/2004 - 11:05:00
Contig239907.xml	2.20kb	30/8/2004 - 11:09:06
Contig729351.xml	1.875kb	30/8/2004 - 11:15:02
Contig376509.xml	1.68KB	30/8/2004 - 11:15:03
Contig900057.xml	536 bytes	30/8/2004 - 11:19:21
Contig876531.xml	85 bytes	30/8/2004 - 11:21:20
Contig432732.xml	446 bytes	30/8/2004 - 11:24:51
Contig232271.xml	510 bytes	30/8/2004 - 11:25:18
Contig769457.xml	743 bytes	30/8/2004 - 11:28:27
Contig653298.xml	85 bytes	30/8/2004 - 11:29:50
Contig758027.xml	85 bytes	30/8/2004 - 11:37:26
Contig566839.xml	592 bytes	30/8/2004 - 11:41:04
Contig34169.xml	85 bytes	30/8/2004 - 11:42:32

Figure 4. Screenshot of the system. **A.** User can inform his input data, and can choose the database and the processing site. **B.** A page of the output data.

EXPERIMENTS AND DISCUSSION

Experiments were developed to test the functionality of the proposed distributed infrastructure. We made experiments using sequences with lengths of 50, 100, 200, and 400 base pairs on a single PC with four processors, 2 GB RAM, 700 MHz, 2048 KB Cache on each processor, 100 GB HD Space, Red Hat 7.2 Operating System (Table 1).

We executed these sequences with BioGridDF. These execution times were computed during the interval from the time the input sequences were all stored in the NFS directory to the time the outputs of the three databases in XML formats were stored in the same NFS directory. This NFS directory is currently stored on a single machine that can be accessed by all machines composing BioGridDF.

The execution times on the distributed infrastructure with four machines (six processors) were approximately 50% of the execution time on a single four-processor machine (Table 1). The 700 MHz speed of the processors in the single PC could explain the higher execution times. This machine is used for the genome sequencing projects of the MidWest Genome Network. The execution time for running Interpro with 2,655 biological sequences in this four-processor PC was almost a month, which is unacceptable for genome projects. So we developed the distributed infrastructure in order to decrease this time, and simultaneously use available machines in the BIOFOCO Network. In a widely distributed environment, with a high communication overhead, the jobs can be distributed among the machines, achieving a high throughput on each machine. These results are motivating and indicate that the distribution of jobs is a good strategy to run massive time-consuming biological applications.

CONCLUSIONS AND FUTURE WORK

We devised a distributed architecture to execute a massive computation biological application. The two-layer scheduling distributed infrastructure relies on JMS (JORAM, 2004) connecting remote sites (first level) and a local scheduler on each site (second level). This choice was driven by the easiness of development and testing of JMS when compared to other systems, such as Globus (Foster and Kesselman, 1997) or MyGrid (Costa et al., 2004). The application must be installed in the execution machines. We built a running prototype at the University of Brasília and EMBRAPA, using four execution machines, with the databases Pfam, PROSITE and ProDom of Interpro, used for protein annotation in genome sequencing projects. We made some experiments and the results were much better when compared to the results obtained with a single four-processor PC. This approach proved to be efficient to distribute the execution time of these databases among heterogeneous machines on the network.

As future work, we suggest studies on communications security, data encryption, authorization, and authentication, not considered in this first prototype, but important due to the different environments involved in the distributed system. It would be interesting to try other middleware layers incorporating these features, such as Globus (Foster and Kesselman, 1997) or MyGrid (Costa et al., 2004), in the first level scheduling, following the same basic design principles of this first version. For the second level, we could also use a different local scheduling tool. We could investigate different ways to store input files and results, and make performance comparisons with the NFS-based strategy currently used at BioGridDF. A comparison of these different topologies could bring new information and somehow improve distributed processing. Other interesting work would be to distribute the components of the first layer among the participating institutions, to use the JMS messaging queue to send the output processing back to the user's machine, and to develop scheduling policies. The effectiveness of grid technology is also highly dependent on the kind of task to be executed as well as the policies of the institutions involved. This system should contribute to optimize a time-consuming biological tool that is widely used during the annotation phase of genome-sequencing projects.

REFERENCES

Bateman, A., Birney, E., Durbin, R., Eddy, S.R., Howe, K.L. and **Sonnhammer, E.L.L.** (2000). The Pfam protein families database. *Nucleic Acids Res. 28*: 263-266.

CORBA IIOP Specification (2004). http://www.omg.org/technology/documents/formal/corba_iiop.htm. Accessed

August, 2004.

Costa, L.B., Feitosa, L., Araújo, E., Mendes, G., Cirne, W. and **Fireman, D.** (2004). My-Grid: A complete solution for running bag-of-tasks applications. In: *22nd Brazilian Symposium on Computer Networks - III Special Tools Session*, Gramado, RS, Brazil, May 2004.

Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C.J., Hofmann, K. and **Bairoch, A.** (2002). The PROSITE database. *Nucleic Acids Res. 30*: 235-238.

Foster, I. and **Kesselman, C.** (1997). Globus: A metacomputing infrastructure toolkit. *Int. J. Supercomput. Appl. 11*: 115-128.

Foster, I., Kesselman, C. and **Tuecke, S.** (2001). The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. Supercomput. Appl. 15*: 200-222.

Globus/Alliance Project (2004). http://www.globus.org. Accessed April, 2004.

Horstmann, C.S. and **Cornell, G.** (1997). *Core Java 1.1: Fundamentals*. Vol. 1. Prentice Hall, Uppersaddle River, NJ, USA.

JORAM Project (2004). http://www.objectweb.org/joram. Accessed July, 2004.

JXTA (2004). http://www.jxta.org. Accessed September, 2004.

Livny, M. (1995). The Condor Distributed Processing System. *Dr. Dobbs J.*, pp. 40-48.

NCBI (2004). National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov. Accessed July, 2005.

Pande, V. and **Shirts, M.** (2000). Screensavers of the world, unite! *Science 290*: 1903-1904.

Servant, F., Bru, C., Carrere, S., Courcelle, E., Gouzy, J., Peyruc, D. and **Kahn, D.** (2002). ProDom: Automated clustering of homologous domains. *Brief. Bioinform. 3*: 246-251.

Sun Grid Engine (2004). http://gridengine.sunsource.net. Accessed July, 2004.

TIGR (2004). The Institute For Genome Research. http://www.tigr.org. Accessed January, 2004.

Web Services Specification (2004). http://www.w3.org/2002/ws/. Accessed August, 2004.

XML (2004). eXtensible Markup Language. http://www.w3.org/TR/2004/REC-xml11-20040204. Accessed June, 2004.

Zdobnov, E.M. and **Apweiler, R.** (2001). Interproscan - an integration platform for the signature-recognition methods in interpro. *Bioinformatics 17*: 847-848.