

Integration of NetSolve with Globus-based Grids

Maciej Brzezniak, Tomasz Makiela, Norbert Meyer
Poznan Supercomputing and Networking Centre
Noskowskiego 10, 61-704 Poznan, Poland
{maciekb,tomem,meyer}@man.poznan.pl

Abstract

The paper describes the techniques for integrating NetSolve, a client-server system for solving complex scientific problems, with the Globus-based Grid environments. The integration includes applying the Globus identification and authorisation techniques to the NetSolve modules and adapting the communication and computation control mechanism in NetSolve to operate in the Globus-based Grids. The control communication in NetSolve is modified to use the Globus-IO. Large input and output data are transmitted through the Grid FTP protocol. Computations on the remote resources are run through Globus GRAM services.

The paper presents the idea and technical details of the integration. It discusses advantages and disadvantages of the modified system in comparison to the original NetSolve and provides preliminary experimental results.

1. Introduction

Many distributed applications, portals and systems as well as many experimental and production Grids base on the Globus Toolkit [1]. The different applications exploit the Globus-provided services and Globus modules as the building blocks. This leads to the unification of the design and management in Grids.

In theory, the different applications that base on Globus can be integrated easily. The unification of processing schema should broaden the range of the resources that can potentially be used by a single application. The newly deployed application can get access to all the resources of a given environment that are controlled by Globus.

In practise, the integration of different applications through the usage of Globus entails many challenges. Firstly, the applications adjusted to the processing schema of Globus-based Grid can lose their interesting features. The integration can thwart the lightweight, simplicity and

effectiveness of the original applications as the Globus Services introduce some overhead on authenticating and securing the communication, running and controlling the computations etc. Secondly, in many cases, adapting the applications to the policies of a given Grid, concerning the resource usage (method of accessing, limits etc.), communication, authentication etc., requires significant modifications in the processing schema of the applications. The modification requires much work and can further demote the valuable characteristics of the original application.

The paper presents the idea and the issues of integrating NetSolve with Globus. NetSolve is a client-server system for solving complex scientific problems. A NetSolve-like service for remote execution of mathematical functions on top of the Globus-based Grid is provided as a result of the integration.

The work includes encapsulation of the NetSolve control communication in the Globus-IO protocol and applying the Grid FTP protocol for transmitting large input and output sets of data. The remote execution mechanism is also modified. It uses the Globus GRAM services instead of the fork-exec mechanism exploited in the original NetSolve in order to activate remote computations. As the result of work, the modified version of NetSolve system is developed.

The paper presents the features of NetSolve that are important from the point of view the integration as well as their relation to the requirements of Globus Grids. The technical details of the integration and the discussion of both their positive and negative impact on the NetSolve characteristics are also provided and supported by some experimental data.

The described work is lead by PSNC, Poznan, Poland in the collaboration with the Tricity Academic Computing Center in Gdansk, Poland within the confines of the government project: SGIgrid: "High Performance Computing and Visualisation with the SGI Grid for Virtual Laboratory Applications" (no. 6 T11 0052 2002 C/05836). In the reference Grid system, Globus services (version 2.4) manage the SGI computers as well as Intel- and AMD-based PCs dis-

tributed across Poland. These machines work under Irix and Linux operating systems. SGI computers are grouped into LSF clusters. Part of the presented works is connected with the project "Framework for Virtual Laboratory system" (no. 4 T11F 010 24).

The features of the original NetSolve are discussed in the second chapter. The third chapter provides the description of the modifications introduced into NetSolve in order to integrate it with Globus services. The fourth part of the paper contains the discussion of the features of the modified NetSolve and their comparison to the original package. The fifth and the sixth sections provide the information concerning to the related work and plans for future researches. The considerations are summed up in the seventh part of the paper.

2. The original NetSolve

NetSolve [2] [3] has been developed in US by Innovative Computing Labs at the University of Tennessee. It allows the end-users to launch functions of mathematical libraries installed on the remote machines from the inside of their applications. The processing schema is similar to RPC model.

The programmer calls the NetSolve API function (provided by the NetSolve client library) specifying the name of the mathematical function to be executed and its input and output arguments (see Figure 1). The NetSolve Agent selects the resource providing the shortest predicted response time to a given mathematical function call. The Agent maintains the list of machines that can execute the mathematical function and monitors their state (i.e. node workload, bandwidth and latency of the network between the Agent and the Server). The function calls are executed on a selected computing resource by the NetSolve Server. It listens for the requests incoming from the Clients and uses the fork-exec mechanism to activate the binary (the NetSolve Service) that implements the mathematical functionality. Each control request is transferred by a separate connection. The communication between the NetSolve modules is performed by using the NetSolve protocol (NetSolve-IO). Input/output data are transmitted through a single stream between the NetSolve Client module and the server node. The processing schema is presented in Figure 1. In reality, The Client-Agent and Client-Server communication is performed through the NetSolve Proxy. It is not marked in the figure to keep it readable. For more information concerning NetSolve, please refer to [2][3].

The NetSolve features are very attractive to some applications. It can be used e.g. as the front-end to the dedicated cluster of machines with installed mathematical libraries. NetSolve does not bring a significant overhead on processing the remote computations thanks to the lightweight of its mechanisms. Therefore, the end-user can get access to such

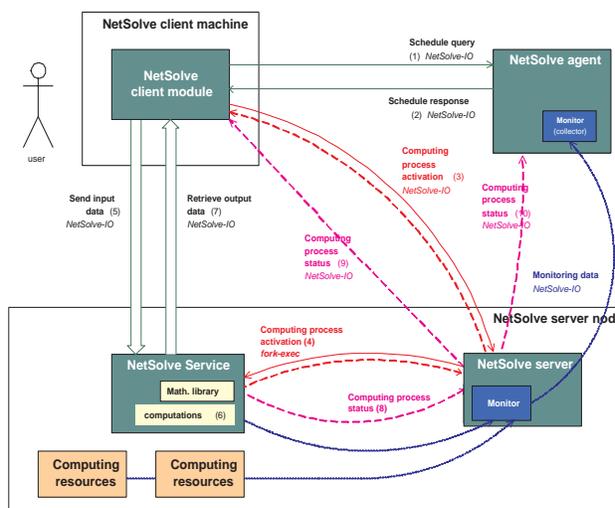


Figure 1. Processing schema in NetSolve

"mathematical cluster" by using a simple NetSolve Client API.

Unfortunately, using NetSolve in shared, geographically and organisationally distributed Grids faces some issues. The following subsections of the paper discuss them and provide details of the solution worked out.

3. Description of work

We have developed a modified version of NetSolve. The first step of the work was the encapsulation of the NetSolve control communication into the Globus-IO protocol. Next, we have modified the way of transmitting the input and output data between the Client and Server modules (Grid FTP instead of single TCP/IP stream). This work is connected with the activities in the third step, i.e. the modification of the mechanism for activating the remote computations.

We decided to introduce an intermediate stage in processing the user requests. The input data are stored by the Client in the NetSolve Repository, from where they are retrieved by the NetSolve Service. The Service is activated through Globus GRAM services (instead of the fork-exec mechanism in the original NetSolve). The Service stores the results in the Repository after the computations. They are retrieved by the Client module from that place.

The schema of processing the user requests in the modified NetSolve is presented in Figure 2. In reality, the Client-Agent and Client-GRAM Server communication is performed through the NetSolve Proxy. This fact is not marked in the figure to keep it readable. The Proxy module uses the same Globus certificate as the Client.

The following subsections of the paper discuss details of the modifications and the new processing schema.

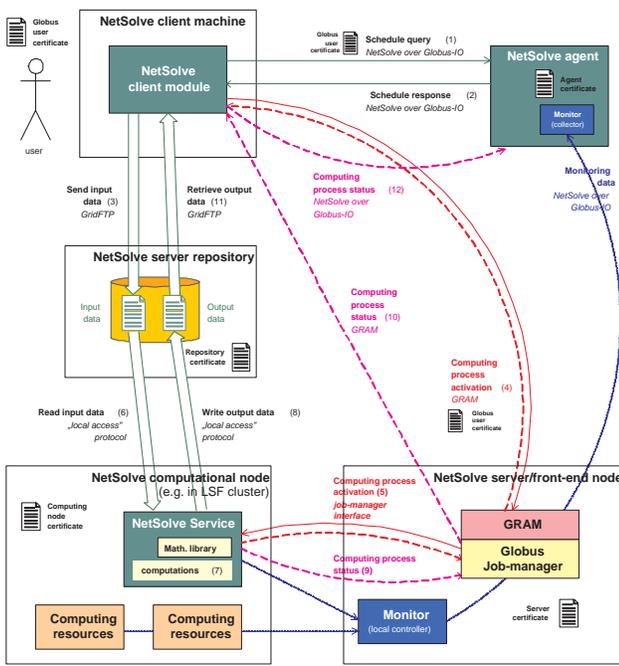


Figure 2. Processing schema in the modified NetSolve

3.1. Authentication and communication

The distributed ownership of the Grid resources forces the need for secure (authenticated, integral, confidential and safe) communication. The Globus Toolkit provides the Globus-IO library that addresses these issues.

Our idea is to exploit the Globus-IO in NetSolve instead of the ordinary TCP streams used so far. The Globus-IO makes securing the internal NetSolve communication possible as it assures the integrity and confidentiality of the data channels. It also provides the mutual authentication of parties by the use of X.509 certificates. Each NetSolve Client is authenticated by the ordinary Globus end-user certificate in the proposed approach (see Figure 2). The Agents and the Servers have special Globus user certificates signed for them.

The NetSolve-over-Globus-IO protocol is used to transmit the control requests between the Client and the Agent module as well as the notifications which are sent from the Server to the Client and from the Client to the Agent. Hence, the communication between all the NetSolve modules is secured. The mutual authentication is assured, since each request inside NetSolve is "signed" by the certificate of the appropriate user or the system module. Therefore, the modified NetSolve can operate in the physically and organisationally distributed installations

However, using Globus-IO to carry the requests inside

NetSolve has some drawbacks. NetSolve creates a considerable number of connections in order to process a single Client request. It also periodically exchanges some messages that are important for the NetSolve operation. Globus-IO assures the security of these transfers but it also introduces a noticeable overhead on the communication. This mainly applies to the creation of the connections, which are authenticated by the exchange of the certificates of communication parties. The overhead on ciphering the content of the requests is not significant, as they are short.

The overhead increases the communication costs in NetSolve. The delays affect the performance of processing the remote calls, while the exchange of certificates and the ciphering of transmitted data consumes some additional resources (network and processors).

The experiment shows that the overhead brought in by Globus-IO to the communication in NetSolve is significant. It can even degrade the performance of computations in NetSolve, if the big number of short tasks is submitted to the system. However, in case of large, computationally complex tasks run in the geographically distributed environment, the additional communication overhead is acceptable.

Summing up, the usage of the NetSolve with Globus-IO makes sense only in the environments where the authentication and the communication safety is essential and not provided by other means. The increased communication cost threatens the profitability of the execution of mathematical functions on the remote Grid resources. We provide the Globus-IO-secured communication as the option in the modified NetSolve. In the SGIgrid the secured version of the communication is performed.

3.2. GridFTP for input/output data transmission

The communication cost is the significant part of the overhead brought in by NetSolve to the remote function execution. The cost of exchanging the requests between the NetSolve Client and the Agent relates mainly to the creation of connections. However, the characteristics of the network traffic generated while transmitting the large computation input and output data between the Client and Server is different. The cost of these transmissions is connected with the cost of moving data through a single stream mainly. The cost of creating the connections is not significant in that case.

The Globus Toolkit provides the Grid FTP service that allows transferring the data in parallel. The transmission of data over multiple data connections overcomes the limitation of bandwidth that is observed on a single connection. Grid FTP API gives also the possibility of assuring high reliability of transmission. Additionally, the authentication and authorisation in Grid FTP is coherent with the techniques used in Globus-IO (X.509 certificates).

Our idea is to perform the input and output data transfers via the Grid FTP protocol instead of using a single TCP connection (as in the original NetSolve). In the modified NetSolve the Client transmits the input data to the Repository that is appropriate for a selected computing node by exploiting the Grid FTP service (see Figure 2). Then the Client initiates the computations on a given computing node (details of the activation process are described in the next subsection). The activated computing process (the Service) acquires the input data from the Repository by using the local access method (read from file, NFS etc.). After the computations, the Service stores the results in the Repository. Then the Client downloads these data through Grid FTP.

Profit of the speedup achieved by the usage of the Grid FTP instead of a single stream for transferring the input and output data depends on the features of the environment and the characteristic of user computations. In widely distributed installations where the user applications call the remote functions with huge input and output arguments, it is very beneficial

Our experiments prove that the parallel transfer with Grid FTP can shorten the time needed for the input/output data movement when comparing with the single TCP stream. The experiment was performed between Poznan and Gdansk, two academic centres in Poland.

In this setup, the maximal bandwidth was achieved for 32 parallel streams after tuning the TCP settings of operating systems for a wide area configuration.

For each setup, there is the optimal number of parallel streams that guarantees the maximum overall bandwidth. In the extreme case, e.g. in the local network, using the GridFTP parallel mode is not profitable, since the cost of maintaining the parallel streams is bigger than the profit of the multiple streams. The default number of streams should be set by the NetSolve administrator in an experimental way. We provide the GridFTP feature for transferring the computations input and output data as the option for the modified NetSolve package. It can be used if it is profitable. Otherwise, the single Globus-IO stream can be used.

Note also that in the presented approach, the processing of the remote call requires four accesses to the data in the Repository. Therefore, the efficiency of the remote calls depends on the performance of the Repository disk system.

3.3. GRAM services for running the remote functions

The original NetSolve uses the fork-exec mechanism for running the computations. The Server runs as the daemon process on the server machine with the privileges of the selected user. Each mathematical task is solved by a separate

child process, that executes the NetSolve Service which implements the mathematical function (see Figure 1).

Usually Grid nodes have their own policies of making the resources available for Grid users. These policies include the users and domain restrictions, limitations of resource utilisation, accounting, determination of the access method etc. Some sites do not allow running a separate daemon processes for each employed application. The GRAM service is the only method for running the computations on these nodes. The nodes of the mentioned systems do not accept the fork-exec mechanism for activating the computing processes. Moreover, the interaction of the computing process with the environment is often limited to the access to the input and output files only. This applies e.g. to the elements of the SGIgrid installation and to the components of many other Grids.

The original NetSolve cannot exploit such resources, since normally the NetSolve Server uses fork-exec mechanism in order to activate the Service and the Service communicates with the Server and the Client modules.

We have developed the modification of NetSolve that uses the Globus GRAM services for running the mathematical functions on the computing nodes. In this approach (see Figure 2), the NetSolve Client uses GRAM and its plug-ins to activate the NetSolve Service. The input data are retrieved by the Service from the Repository assigned with a given computational node. The computation output is also stored in the Repository and retrieved by the Client (details in the previous subsection). The functionality of the NetSolve Server is partially taken over by the GRAM server, which is responsible for activating the Service process. The mathematical functions that are available on the computing node are registered in the Agent by a separate tool. This tool also launches the Monitor process, which reports the state of the node to the Agent periodically (see Figure 2). Note that there is no longer the daemon process (the Service) that forks the computing processes for serving the Client requests.

We also have introduced some modifications into the NetSolve Service. The Service gets all the data needed to solve a mathematical task from the input files and puts the results and the information on the status of computations in the output files. Input and output files are placed in the Repository.

The additional advantage of the proposed mechanism, is the fact that it complies with the rules of batch processing. In our opinion it is the important feature. Some resources, e.g. nodes of the clusters or the queue systems (e.g. LSF [4], PBS [5]) incorporated to Globus-based Grids are accessible only through the special tools or services on the front-end nodes. Normally, the GRAM server uses the plug-ins in order to run the computing processes in such embedded systems. The original NetSolve cannot exploit them. The

proposed method for activating the Service process, supported by the technique for transmitting the input/output data, makes it possible to convert the NetSolve API calls (seen from the end-user perspective) to the batch jobs (executed on the NetSolve Server side). Using GRAM and its plug-ins (job-managers), e.g. for LSF [4], PBS [5] and Condor [6], the modified NetSolve can exploit the computing systems, which were inaccessible for the original version of NetSolve.

Note that the original NetSolve Service can behave similarly to the NetSolve service provided by us. The original Service can retrieve the output and input data from the file and store them into a file. However, it communicates with the Server and Client modules through the TCP/IP connections. Therefore, it does not comply with the rules of batch processing.

In the proposed solution, the access method is unique (GRAM protocol) from the NetSolve Client perspective, no matter which job-manager is used by the GRAM server. There is no need to develop separate interfaces to various systems for the Client. Additionally, NetSolve can benefit from the fact that the GRAM plug-ins for new systems or their new versions are continuously developed by the Globus community.

Moreover, exploiting the GRAM service, NetSolve automatically respects the resource management policies (concerning resources usage limits, user or organisation constraints, access method etc.) that are introduced to the configuration of that service.

The developed remote execution technique requires additional activities on the computing node side and on the Repository side, when comparing to the original NetSolve. Therefore, the simplicity and effectiveness of the resources access method of the original NetSolve is lost. However, the cost of the additional processing overhead pays off in many situations. The broader set of the Grid resources the NetSolve can exploit provides potentially more computational power to the NetSolve users.

The modified NetSolve Proxy and NetSolve Service as well as the tools replacing the original NetSolve Server are provided as an option of the modified NetSolve package.

4. Discussion

In our opinion the integration of NetSolve with Globus Grids brings many benefits. The modified NetSolve can utilise the authorisation and security mechanisms provided by Globus-IO. It is very important in many Grid environments, where computing nodes belong to independent organisations and the mutual trust of the processing parties is crucial. In that way, the modified NetSolve can be used in the Grids composed of independent sites, domains and virtual organisations. The effective and reliable Grid FTP

protocol used to transfer the large input and output data (instead of a single stream) shortens the time needed to serve the remote calls. Finally, the GRAM service and its plug-ins for the other systems (LSF, PBS etc.) give the possibility of running the computations on the remote systems that were inaccessible for the original NetSolve.

However, the modified NetSolve has some disadvantages. It has lost the lightweight and simplicity. The overhead introduced by the modified NetSolve into the processing of the remote function calls has increased in comparison to the original system. This is caused mainly by the cost of authenticating and ciphering the communication channels and switching from the fork-exec mechanism to the GRAM services for activating the computing process on the remote machine. The intermediate step in the processing of the remote function execution (storing the input and output data in the Repository) entails also some additional overhead.

We have shown that profitability of exploiting the three techniques (Globus-IO for internal communication, Grid-FTP for the large data movement and GRAM for activating the Service process) depends on the characteristics of a given Grid environment and the end-user application. Therefore, we provide all the techniques as an option for the modified NetSolve package. They can be used independently and activated if it is beneficial. In the extreme case, e.g. in the local setup or in the dedicated cluster of machines, usage of any of the presented modifications is not beneficial. But in a widely distributed Grid, where the end-users need much computational power for solving very complex mathematical problems, exploitation of the modified version of NetSolve is profitable. The mode of operating should comply with the local Grid management policy. The decision to enable or disable particular features should be supported by the analysis and experiments concerning the environment and the application.

The solution can be incorporated to the consecutive versions of NetSolve. Fragments of the code that implement the additional functionality are separated from the original NetSolve code. The modifications of the existing fragments are clearly marked.

An important feature of the three techniques applied in the modified NetSolve is their coherency. Globus-IO, Grid FTP and GRAM use the same authorisation and identification mechanisms (X.509 certificates).

Another interesting feature of the developed solution is the fact that the NetSolve and the other applications and systems can authenticate the end-user by using the same Globus user certificate. Moreover, the credentials of the end-user can be delegated by any of the NetSolve modules to the other NetSolve modules or the external systems. For instance, the GRAM server can start the computations in the LSF or PBS clusters on behalf of the end-user. In some Grids, e.g. in SGIgrid, this becomes an important feature.

The modified NetSolve can be integrated with Globus-based Grids and respect the processing schema and management policies (concerning resources usage limits, user or organisation constraints etc.) in these Grids. It is possible under the condition that these policies are reflected in the configuration of the Globus services.

5. Related work

NetSolve developers have worked out the Globus version of the NetSolve Proxy (globus-proxy) as well as the Globus version of NetSolve Service (globus-service). In this solution the NetSolve Service is activated through GRAM [7] [8]. The existence of the Service on a given node is noticed in the MDS database, with a special tool which replaces the Server module. However, these are the only phases of the NetSolve operation, which are integrated with Globus services. The control communication as well as the manner of moving the input and output data inside NetSolve remains unchanged. The solution worked out by us goes further. It uses secure, authenticated Globus-IO channels to carry the control communication and the Grid FTP protocol in order to move the input and output data efficiently. The activation mechanism allows NetSolve to exploit even these resources which do not accept the fork-exec mechanism.

Moreover, the globus-proxy provided in the original NetSolve distribution is rather the test or the development version than the final product. It has hard-coded user credentials and can be used with a special NetSolve Service included in the package. The version of NetSolve-Globus proxy developed by us can be used with any end-user Globus certificate.

The NetSolve team has developed the version of the Service for Condor-G (condor-service). It allows direct submission of the NetSolve calls to the Condor-G pools [6]. Our solution can submit the jobs to Condor-G indirectly, through the GRAM services and its plug-ins for Condor-G pools.

6. Future work

The original NetSolve does not provide the accounting feature. The possibility of charging the end-users for the resources consumed by the remote computations is very important in the production Grids that are composed of the independent sites. Therefore, we plan to add this functionality to the modified NetSolve.

We are also working on the extensions of the scheduling technique for NetSolve. The developed algorithm should consider the new performance characteristics of the modified NetSolve. They include the additional costs of communication over the Globus-IO and the performance gains provided by the Grid FTP applied to large data sets transfers as

well as the additional costs of activating the NetSolve Service through GRAM. The modified scheduler should also consider the performance features of the Repository. Another issue is the estimation of the amount of time that is needed to execute the mathematical function on the resources that are not directly accessible by NetSolve (nodes of LSF and PBS clusters etc.). This problem is currently addressed by our researches. They are partially described in [9].

7. Summary

The paper discusses the problems related to the integration of the NetSolve system with Globus-based Grids. We have shown that the usage of tools and services provided by Globus is beneficial. The most important is the fact that NetSolve gains the possibility of using the resources of a given Globus-based Grid environment after the integration. Another attractive feature is the ability of interoperation with the external systems and applications that use the common Globus-provided building blocks to implement their functionality. The range of the resources that the application can use broadens thanks to the integration. This potentially increases the amount of the computational power that the end-user of the application can exploit.

However, the integration also entails the costs. The processing schema of the original application and system must be adapted to the rules of the Globus-based Grids. In the extreme case the modifications can even waste the attractive features of the original application, such as lightweight, simplicity and effectiveness. It is caused by the additional overhead introduced by Globus services. Therefore, the number of modification made in the original application should be minimised. To achieve this, we provide the modifications to the NetSolve package as the independent options, which can be used if it is profitable.

In our opinion, the mentioned costs of integration are worth to pay. The security issues disable the usage of the original NetSolve in many of real Grids, nevertheless the fact that NetSolve provides potentially broad range of the computing resources to the end-user application. Exploitation of the Globus services in the critical phases of the processing in NetSolve allows the NetSolve system to operate in shared, production Grids.

We work on putting the modified NetSolve into practice in the geographically and organisationally distributed Grid system, which is the reference system for the SGI-grid project. We are also considering the exploitation of the modified NetSolve in the meta-cluster built in the confines of the CLUSTERIX project "National Cluster of Linux Systems" [10]. The installation consists of local PC clusters with 64- and 32-bit architectures, located in 12 geographically distant and organisationally independent sites in

Poland. We believe that the modified NetSolve will be able to exploit the power of this environment very effectively. On the other hand, this installation is going to be a very interesting environment for testing the modified NetSolve package

References

- [1] I. Foster, C. Kesselman. "Globus: A Metacomputing Infrastructure Toolkit." *Intl J. Supercomputer Applications*, 11(2): 115-128, 1997.
- [2] H. Casanova and J. Dongarra. "NetSolve: A Network Server for Solving Computational Science Problems." *The International Journal of Supercomputer Applications and High Performance Computing*, Volume 11, Number 3, pp. 212-223, 1997.
- [3] D. Arnold, H. Casanova, J. Dongarra. "Innovation of the NetSolve Grid Computing System." *Concurrency: Practice and Experience*, Volume 14, numbers 13-15, pp. 1457-1479, 2002.
- [4] Load Sharing Facility. <http://www.platform.com/products/wm/LSF/index.asp>.
- [5] Portable Batch System. <http://www.openpbs.org>.
- [6] D. Thain, T. Tannenbaum, and M. Livny. "Condor and the Grid." in F. Berman, A.J.G. Hey, G. Fox, editors, *Grid Computing: Making The Global Infrastructure a Reality*, John Wiley, 2003. ISBN: 0-470-85319-0
- [7] M. Miller. "How to use NetSolve." Workshop on Grid Computing. January 23, 2002. Presentation available on <http://citr.cs.utk.edu/012302gridcomputing.html>.
- [8] D. Arnold, S. Agrawal, S. Blackford, J. Dongarra, C. Fabianek, T. Hiroyasu, E. Meek, M. Miller, K. Sagi, K. Seymour, Z. Shi, S. Vadhiyar. "Users' Guide to NetSolve V2.0". <http://icl.cs.utk.edu/netsolvedev/documents/ug/nug.ps>.
- [9] M. Brzezniak, N. Meyer. "Evaluation of execution time of mathematical library functions based on historical performance information." in *Parallel Processing and Applied Mathematics*. R. Wyrzykowski, J. Dongarra, M. Paprzycki, J. Wasniewski. Springer-Verlag, Berlin, 2004, pp. 161-168. (Lecture Notes in Computer Science, Vol. 3019).
- [10] R. Wyrzykowski, N. Meyer, M. Stroinski. "CLUSTERIX: National Cluster of Linux Systems." *Across Grids 2004*, Nicosia, Cyprus. <http://clusterix.pcz.pl/project/publications.html>.