Desktop Grid for the Deployment and Execution of Highly Loaded Webservice Components of Distributed Information Systems

Mykola O. Aliksićiev, Oles V. Maiboroda, Andrii O. Onysko, Olexandr Stepurin

Scientific and Educational Institute of Telecommunication Systems
National Technical University of Ukraine "KPI"
alexeyev@its.kpi.ua, oles.maiboroda@gmail.com,
onysko.andrew@gmail.com, olexander.stepurin@gmail.com

Abstract. The article is devoted to the current issues of creating an environment for the placement and execution highly loaded webservice components of distributed information systems, which have a dynamically changing heterogeneous architecture.

1 Introduction

The use of high-performance distributed computing systems based on non-clustered inalienable resource is one of the strategic directions of development of computer technology.

On the one hand this is due to the development of scientific fields, such as climate modeling, estimates of genetic engineering, analysis of pollution using intensive computing tasks of pattern recognition, optimization problems with many parameters, multi-dimensional time-dependent problems, etc.

On the other hand, the use of high-performance distributed computing systems based on non-clustered inalienable resources, the so-called grid systems of personal computers (GSPC, Desktop Grids) is due to economic factors: the cost of such a system may differ from that of the same computing power of a supercomputer by several orders.

This situation is particularly critical for organizations, especially non-profit, that cannot afford to buy equipment for a computer or other type of cluster, while its fleet of personal computers and portable devices can have a sufficiently high computing power, significant amounts of memory for both short and long term storage data or files. It is obvious that not all distributed applications can run efficiently on such systems due to severe limitations on data transmission capabilities and the ability to switch off inalienable nodes involved in the calculations.

2 Building a grid system of personal computers

In this paper we study the example of Institute of Telecommunication Systems of NTUU "KPI" as a small scientific and educational institution possibility of constructing on the existing resource base environment for solving computational problems which can be formulated as a set of small interdependent tasks, and for placement and performance of highly loaded components of distributed information systems, developed under the research and production projects. Research is considered as a set of tasks based on the following architecture (see Figure 1).
Figure 1. The architecture of a heterogeneous environment for the deployment and execution of heavily webservice component of distributed information systems based on the GSPC

2.1 Software technology for the implementation of the Desktop Grids core

The first task was the selection of software technology for the implementation of the Desktop Grids core for system with relatively constant configuration. Based on data obtained in [1] for the analysis of Desktop Grid systems: Condor, BOINC, XtremWeb, OurGrid, SARD, X-COM, was decided to make further analysis of software solutions for high-performance distributed MPI-calculations MPICH2 and PelicanHPC which in contrast to the above have the best multiplatform and relatively easy to install and configure. Also, it was necessary to investigate the effect of network connection speed (taking into account the possibility of connecting network participants at a relatively low-speed radio channels) on the characteristics of these Desktop Grid based software solutions.

For constructing the desktop grid test model based on the MPICH2 and PelicanHPC has been used seven laboratory computers which have roughly same computing power. Four of them were able to connect using IEEE 802.11a/g (54Mbit/s). Testing was performed on parallel distributed calculating of number $\pi$ with a given accuracy. The results of calculations for up to 7, 8 and 9 decimal places are shown in Figure 2.a for an 8-node fixed structure; 2.b - for a structure in which 4 nodes are connected via wireless channels; 2.c - for a structure only with wireless nodes. Figure 2.d shows a comparison of Desktop Grids based on the MPICH2 and PelicanHPC. In this case to construct the theoretical curve was used a method of calculation for determining the effectiveness of the cluster, described in [2].
From the graphs it is obvious that a more powerful host computer system (in this case to calculate the task with less accuracy) require higher bandwidth data transfer. The obtained practical result on the one hand confirms the theory that a system of parallel computing is effective for as long as the data transfer time is less than the computation time of all tasks. On the other hand, these results demonstrate that the analyzed software implementations can be used to construct and use a GSPC core.

The comparison of two distributed computing systems proved that the system which is using a clustered inalienable resource in the case of PelicanHPC is more effective by deploying the same number of computing nodes than MPICH-2. But we must note the fact that the system PelicanHPC requires constant number of nodes in the structure, and does not allow simultaneous use of the site resources for other tasks of user.

Therefore, taking into account the small difference in the computation time of the same problems, as well as efficient use of network nodes, it is better to use MPICH-2 because it provides the ability to attract third-party sites to the calculations, carried out without affecting the basic tasks of the user and allows you to dynamically change the number of computing nodes in a distributed computing system (Desktop Grids).

Based on calculations made in the work and practical testing of the system, we can conclude that the presence of even morally obsolete computer park, and a data network with high bandwidth, you can organize a fairly powerful computer system that can meet the educational and scientific purposes.

Figure 2. The results of the parallel distributed computing of to 7, 8 and 9 decimal places.

a) for an 8-node fixed architecture;
b) for the architecture, with 4 wireless nodes;
c) for the wireless architecture;
d) comparison of results for Desktop Grids based on the MPICH2 and PelicanHPC.
2.2 Authentication of nonstationary resources for operation in GSPC

To attract the computing resources of nonstationary network subscribers by providing them access to paid information resources (electronic publishing, distance learning courses, etc.) and unlimited access to the Internet it was necessary to investigate users and computing resources authentication to connect to information corporate networks and further use in the environment with dynamic architecture. The mechanisms should be cross-platform, do not depend on the hardware implementation of the network, should be easily implemented on both the user- and network- sides and have a reliable level of security.

Conducted in the work analysis allows to identify several authentication methods that can be used for constructing an environment with non-clustered resources and dynamic architecture (Fig. 3).

![Classification of EAP methods](image)

Figure 3. Classification of EAP methods

The results of the analysis and comparative analysis of these authentication methods is given in Table 1. Methods that provide two-phase authentication, offer a higher level of protection level for the transmission of credentials by creating a secure tunnel transport.

Among all the methods of authentication, Windows 7 has built-in software only for work with EAP-PEAP and GTC. Analysis of Unix distributives has shown, that they can have variable built-in or independently installed supplicants. Operating system Mac OS X 10.7 Lion can use all methods, except for GTC.

As one of the requirements was to provide a cross-platform for chosen mechanism to ensure the ability to connect the largest number of nodes, so it is possible, only when using the EAP-PEAP, as shown in Table 1. Also, this method is relatively easy to implement on both the user and the network side and has a sufficient level of security.

Thus, for a heterogeneous environment on the basis of the existing resource base with a dynamic connection of working nodes construction appropriate to use EAP-PEAP-MSCHAPv2.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Methods</th>
<th>EAP-TLS</th>
<th>EAP-TTLS</th>
<th>EAP-PEAP</th>
<th>EAP-FAST</th>
<th>EAP-GTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client side certificate required</td>
<td></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+ (PAC)</td>
<td>-</td>
</tr>
<tr>
<td>Server side certificate required</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+ (PAC)</td>
<td>-</td>
</tr>
<tr>
<td>Provider</td>
<td></td>
<td>Microsoft</td>
<td>Funk</td>
<td>Microsoft</td>
<td>Cisco</td>
<td>Cisco</td>
</tr>
<tr>
<td>Two-phase authentication</td>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Windows XP SP3</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Windows 7</td>
<td></td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Mac OS X 10.7 Lion</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Ubuntu 11.10</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Deployment Difficulty</td>
<td></td>
<td>High</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Moderate</td>
</tr>
<tr>
<td>Wi-Fi Protection Level</td>
<td></td>
<td>Very High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>
2.3 Core and nonstationary resources interoperaibility for operation in Desktop Grid

To ensure interoperability between Desktop Grid nodes this paper proposes to use the technology of interaction at the program and at the system level, as described in [1]. Under the interoperability at the software level implies the existence of the universal interface for connecting the computing resource, which allows to combine heterogeneous resources into a single distributed system. SAGA, P-GRADE, MathCloud, BNDGrid systems are based on this principle. Interoperability at the system level presupposes the existence of transparent for the application mechanism that ensure “seamless” integration of computing resources. An example of this approach is the mechanism of bridges used for the of service grids and desktop grid integration.

2.4 Modeling of parallel software algorithms for Desktop Grids

Efficient use of Desktop grid resources depends on the algorithms used. The key to success of complex software for distributed information systems using parallel algorithms is their proper modeling. To create a complex software models of distributed information systems, there is a known approach to software development of systems based on models, known as the MDSD (Model-Driven Software Development) or the MDE (Model-Driven Engineering).

In previous works [3,4,5,6, etc.] were proposed a modification of this approach, MMDSD, which provides an integrated model, which has four levels: the abstract business logic, automated business processes, the level of a platform independent graphical user interface with the simulation by FaceXML and platform-independent modeling functionality based on the UML [6]. However, this approach does not take into account the multi-threaded execution of various algorithms. As one of the options in this paper, we propose to use UML for this purpose. In particular, to describe the multi-threading programs is proposed to use class diagrams reflecting the basic relations between processes and threads, as well as sequence diagrams, taking into account the order of the single thread in a system with multi-core processor. For more information this problem is considered in [8].

3 Conclusions and Future Work

High-performance distributed computing systems based on non-clustered inalienable resources, the so-called grid systems of personal computers (Desktop Grids) based on analyzed software implementations can be successfully used in organizations that own personal computers and portable devices with a sufficiently high computing power, significant amounts of memory - for both short and long term storage of data or files.

Obtained practical result confirms that such a system of parallel computing on Desktop Grids, involving computing resources of unsteady subscribers by providing them with access to paid information resources (electronic publishing, distance learning courses, internet, etc.) can be very effective for computing tasks, consisting of a few mutually dependent tasks, as well as to accommodate and performance of heavy duty components of distributed information systems.

In further work within the architecture considered in Section 2 will be explored the scenarios for Desktop Grids as cluster for load balancing of Web applications, Desktop Grid cluster as “runtime” for applications that use Java Remote Method Invocation, Desktop Grid as a cluster-based storage of distributed fault-tolerant replicated file system.

References