http://www.moderntechno.de/index.php/meit/article/view/meit22-01-040
DOI: 10.30890/2567-5273.2022-22-01-040

UDC 004.272.75

ORGANIZATION OF DISTRIBUTED HIGH-PERFOMANCE COMPUTING ON THE UNICORE-PLATFORM

Skakalina E. V.
c.t.s., as.prof.
ORCID: 0000-0002-6441-3467
National University "Yuri Kondratyuk Poltava Politechnics"
Pershotravnevyi ave. 24, Poltava, UA-36011, Ukraine

Abstract. *The paper presents the result of building a grid system for implementing distributed computing technology. To build the system, the architecture of the UNICORE platform was used, which is focused on providing transparent secure access to the resources of a distributed computing environment. The general concept and advantages of using grid systems are considered. To solve the problem, a hybrid algorithm is chosen. This choice is substantiated, its performance is estimated on a test example.*

Keywords: *grid, UNICORE, distributed computing, grid computing system, computing node.*

Introduction.

The subject of the study is the process of building computing grids and optimizing the distribution of allocated resources to fulfill the tasks. The ideological basis of Grid technology is the pooling of resources by creating a new type of computer infrastructure that provides global integration of information and computing resources based on network technologies and special middle-level software (between basic and application software), as well as a set of standardized services to ensure reliable shared access to geographically distributed information and computing resources: individual computers, clusters, information stores and networks. The relevance of the chosen topic is due to the constant need to improve the process of processing large volumes of data with relatively simple programs and to achieve the maximum speed of calculations due to the global distribution of these calculations between computers. Grid technology is used to create a geographically distributed computing infrastructure that combines resources of various types with collective access to these resources within virtual organizations.

GRID – concept, architecture, development

The last fifteen years are the years of the birth and development of a new direction in information technologies, the name of which (as is traditionally believed) was given in 1998 by Y. Foster and K. Kesselman - "Grid". Grid as a means of compatible use of computing power and data storage allows you to go beyond the simple exchange of data between computers and, ultimately, to turn their global network into a kind of giant virtual computer that can be accessed remotely from anywhere, regardless of from the user's location. The idea of using a network of supercomputers to solve problems probably originated much earlier (attempts were made since the 60s of the 20th century), but now it has acquired the completed form of the "Grid concept". Traditionally, nuclear physicists are considered to be involved in the development of Grid computing, and until now their need to process colossal volumes of research data is the driving force for the implementation of Grid implementation programs, let's mention at least the activities of the European Center

for Nuclear Research (CERN). However, Grid has a potentially large number of other areas of application, as it offers a universal approach to solving the problem of lack of computing resources - because it is obvious that, in general, a network of supercomputers is capable of solving more complex problems than each of its component nodes individually [1].

That is, the Grid is a type of parallel distributed system that allows the separation, selection and accumulation of geographically distributed "autonomous" resources in real time depending on their suitability, capability, performance, price and quality-of-service requirements of users.

In the past few years, Grid technology has evolved from a carefully configured infrastructure that supported the execution of a limited number of Grant Challenge applications on high-performance hardware to a dynamic environment, the development of which is guided by the international community. As Grid technology becomes a reality, industry is increasingly involved in its development. The participation of commercial organizations accelerates the development of robust software that supports Grid environments outside of academic laboratories. In turn, this affects both the Grid architecture and related protocols and standards. The adaptation of Web services technology to the grid architecture has brought significant benefits and led to the emergence of a somewhat fragmented development environment. Developers of software and Grid services strive to comply with agreements and standards widely spread in their community. However, for various political and technical reasons, there are several competing viewpoints about how the architecture should be implemented and what standards should be relied upon. This rivalry is holding back the developers of Grid software because they are not sure that future standards will include those used today [2].

There are two main criteria that distinguish Grid systems from other systems that provide distributed access to shared resources [3]:

1. The Grid system coordinates disparate resources. Resources do not have a common management center, and the Grid system coordinates their use, for example, load balancing. Therefore, a simple cluster resource management system is not a Grid system, as it provides centralized management of all nodes of a given cluster, having full access to them. Grid systems have only limited access to resources, which depends on the policy of the administrative domain (owner organization) in which this resource is located.

2. The Grid system is built on the basis of standard and open protocols, services and interfaces. Without standard protocols, it is impossible to easily and quickly connect new resources to the Grid system, to develop new types of services.

The main properties of grids:

- ✓ *joint use of distributed resources.* It opens up opportunities for cooperation that would be difficult to achieve by other means. At the same time, issues of "fairness" of resource distribution, creation and management of virtual organizations (VO) arise.
- ✓ *unification of capacities.* In this way, a system is built with a total potential computing power that exceeds the power of its components, and at the same time, a more efficient use of hardware is achieved (downtime is reduced).

There are strict requirements for communication channels.

- ✓ *virtualization.* It includes such concepts as: hiding (masking) from the user the complexity of the hardware and software implementation of the system and its components, geographical distances between nodes, the belonging of nodes to different organizations, creating the illusion of working with a "virtual supercomputer".
- ✓ *inhomogeneity (heterogeneity).* A typical Grid consists of (and must function successfully in) a variety of heterogeneous hardware and software. Decentralized management. There is no single "owner" of the entire system, which requires the use of distributed management mechanisms.
- ✓ *interoperability.* The functional compatibility of different Grid components and even different Grid infrastructures is based on the standardization of interfaces. Approaches that do not take into account standards are not very promising.
- ✓ *transparency of access.* Grid should provide access to system resources to users, regardless of the specific network topology or local implementation of access mechanisms to certain nodes and their resources.
- ✓ *scalability.* The Grid should provide mechanisms to include new resource sources, users, data stores, etc. without affecting existing participants: The Grid should have the ability to dynamically reconfigure. Security. One of the main requirements for the Grid system is security of access to resources, which determines a limited set of permitted operations for authorized users and programs.

Although the Grid technology itself is not tied to certain resources, most often implementations of Grid systems provide work with the following types of resources:

- computing resources
- separate computers, clusters;
- data storage resources
- disks and disk arrays, mass data storage systems;
- network resources;
- software
- any specialized software.

Let's note the difference between Grid technology and implementations of Grid systems. Grid technology includes only the most general and universal aspects that are the same for any system (architecture, protocols, interfaces, services). Using this technology and filling it with specific content, it is possible to implement one or another Grid-system designed to solve one or another class of applied tasks.

Grid technology should not be confused with parallel computing technology. Within the framework of a specific Grid system, of course, it is possible to organize parallel calculations using existing technologies (PVM, MPI), since the Grid system can be considered as a kind of meta-computer that has many computing nodes. However, the Grid technology is not a technology of parallel computing, its task includes only the coordination of the use of resources.

The Grid architecture [4] defines the system components, the goals and functions of these components and reflects the ways in which the components interact with each other. The Grid architecture is an architecture of interacting protocols, services and interfaces that define the basic mechanisms by which users connect to the Grid system and share computing resources to solve various tasks. The architecture of Grid protocols is divided into levels (Fig. 1), the components of each of them can use the capabilities of the components of any of the levels below.

In general, this architecture sets requirements for the main components of technology (protocols, services, application interfaces and software development tools), without providing a strict set of specifications, leaving the possibility of their development within the framework of the adopted concept.

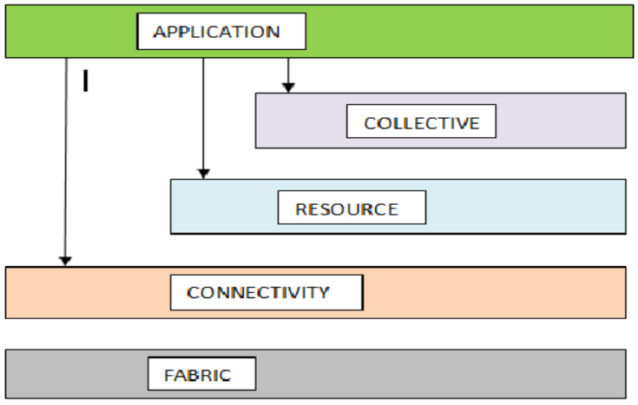

Figure 1 - Grid protocol architecture levels and their correspondence to Internet protocol architecture levels

Since 2007, the development of the European Grid infrastructure (EGI) began, the operation of which is planned on a permanent basis in the form of a coordinated network of national Grid systems, or NGI. CERN reserves the overall coordination and responsibility for the modernization of the PGO and the general security system. The Grid system becomes national (NGI - National Grid Initiative) if it meets the following criteria: it has state support, for example, by including the project to create a Grid infrastructure in the State Program with guaranteed funding; represents the interests of all layers of society (scientists, university employees, industrialists, merchants, etc.); has an extensive structure of coordinating, regional and resource centers that ensure the functioning of basic Grid services, monitoring and response to emergency situations, accounting of resources and completed works (accounting), management and support of virtual organizations (B), Grid software certification; is based on compliance with international standards and rules (for example, using the SOAP protocol to access the resource, the WSDL language for describing Web services, UDDI directories as registers of descriptions, etc.); supports infrastructure security, has the right to generate CA user certificates with the knowledge of EUGridPMA (European Grid Authentication system); connected to GEANT, the European Scientific and Educational Computer Network; has Grid infrastructure governing bodies in the form of the National Grid Infrastructure Development Council in NA, SA and JRA directions [5].

Application of the UNICORE platform for building a grid system

A UNIX-like system is required to install UNICORE services (except for the graphical client). It can be used both with and without a connected graphical interface. As part of the implementation of the task, the client and the first components of the system grid will be installed on CentOS 7 with a graphical interface [6].

Since all components of the UNICORE grid system are essentially Java processes, the Java package installed on all systems (without exception) is required to start their operation. The version of the UNICORE installation package directly depends on the Java version and only requires a specific version of the Java JDK. The latest version of the UNICORE 8 installation package requires Java version 1.8.0 or higher.

In the GUI (Graphical User Interface) version of CentOS 7, such a package comes by default with the installation

UNICORE is not only a Java application, but also bash and python scripts for configuration and installation of services. And if the bash interpreter comes by default in all linux-distributions, then python must be installed separately. It is the default in the GUI version of CentOS 7.

The TSI component can be installed in two ways - by specifying its parameters in configure. properties, or using the Install.sh script. During installation, you must select the task scheduler. UNICORE TSI supports the integration of Torque, Slurm, LSF and Load Leveler schedulers. The use of third-party schedulers is beyond the scope of the task, so the default nobatch scheduler option is selected during installation. It is necessary to take into account the fact that after installation, access to TSI configuration files and its scripts is possible only with root user rights. The installation process is presented in fig. 2.

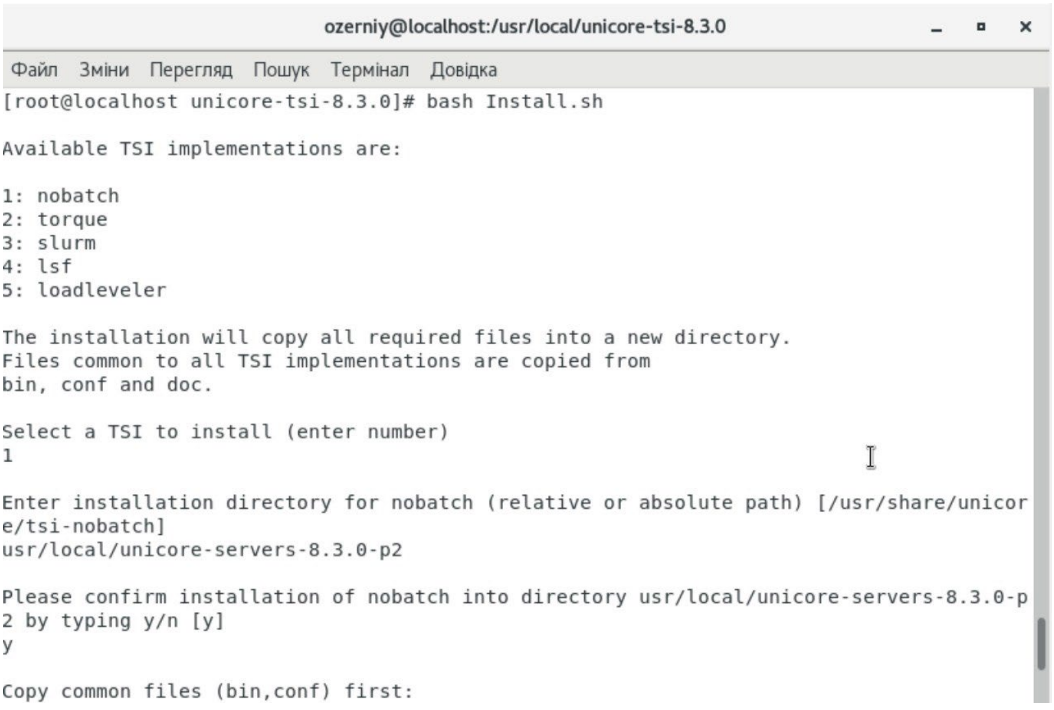

Figure 2 - Installation TSI

The main control script will be used for launch. The process was carried out from the command line configuration of the Linux system using the command "./start.sh" from the directories with the configuration files of the UNICORE system. The final result is presented in Figure 3.

After confirming the ability to connect to the server, it was necessary to create a job that can be called from the console. UCC can perform tasks specified in the JSON task description format used by the UNICORE REST API, as well as several extensions related to local file handling, presentation options, and more.

To test the work sent by the client to the server, an example of calculating the Fibonacci number up to 100 was chosen. bash was chosen as the compiler.

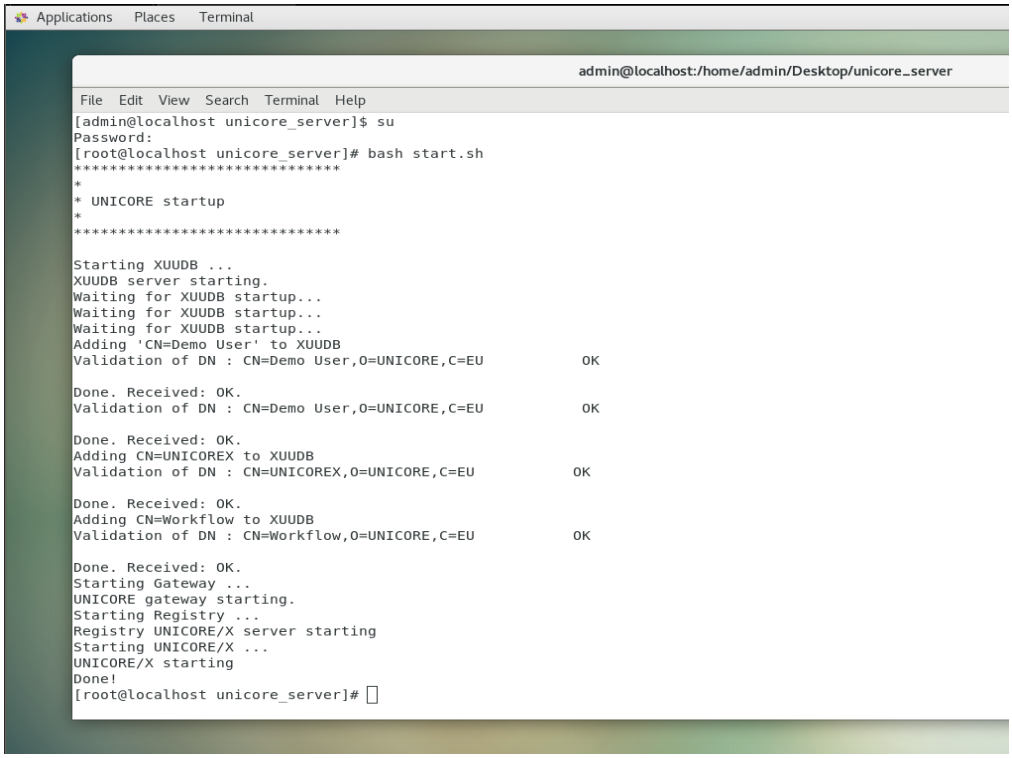

Figure 3 - UNICORE Platform launch

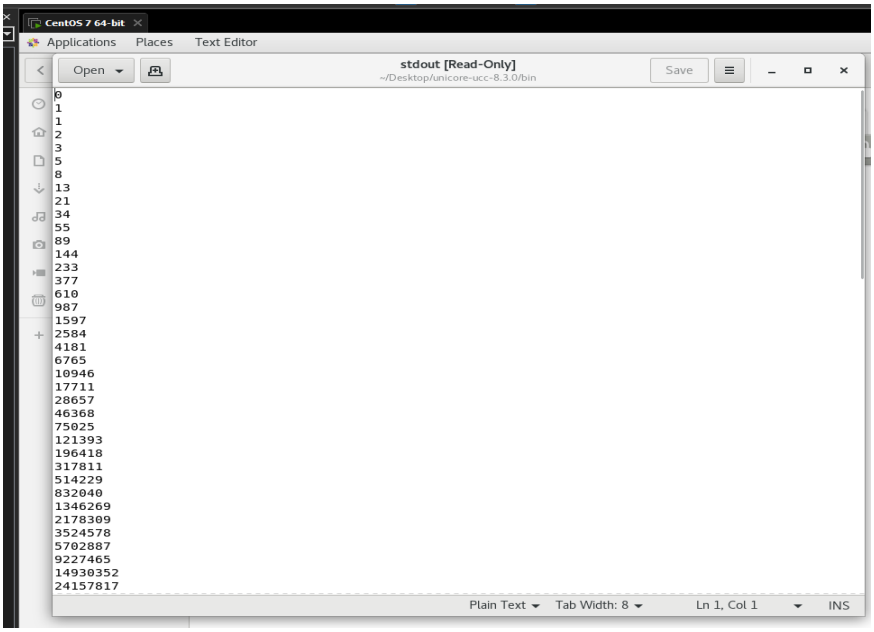

Figure 4 - The result of calculating the Fibonacci number up to 100

After successfully creating a new task, we send it for execution through the UCC console using the run command with arguments in the form of the job name and -v, which provides detailed information about the progress of sending and executing the job [7].

After starting, the job is sent to the selected computing node (or determined by the site automatically, if a specific system was not specified) and started for processing. After the calculations are finished, the work ends with a positive or negative result. Regardless of the success of the calculations, the work saves all output data for further analysis and possible editing of the work being performed in the place specified in the work.

The results of the work can be seen in the created stdout - file in Figure 4.

Summary and conclusions.

A computing grid is a software and hardware infrastructure that provides reliable and transparent access to high-performance computing resources. The Grid provides a common environment for deploying a service-oriented infrastructure that supports the creation and sharing of resources across distributed organizations. Resources are understood as hardware, tools, software and data, as well as services connected through an intermediate software layer and providing security, monitoring, resource management, etc. The grid network was implemented taking into account all the necessary standards for distributed computing, installing and configuring all its components in practice made it possible to demonstrate the basic and necessary elements for distributed computing, such as a heterogeneous and easily scalable environment, combining decentralized resources, the ability to use a graphical interface, creating works on computing tasks in various programming languages.

References :

1. Ivashko E.E. (2016). Modern technologies of high-performance computing. Petrozavodsk: KPIYF, 61 p.

2. Shpakovsky G.I. (2011). Implementation of parallel computing. M.:BGU, 76 p.

3. Grid Computing: Making the Global Infrastructure a Reality (2005). / edited by Fran Berman, Geoffrey Fox, Tony Hey. – John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, England, 1020 p.

4. GRID architecture - http //jre.cplire.ru/alt/dec03/4/text.html#_Toc63762391 Access date: 07/03/2022.

5. GRID software testing. - http://mybiblioteka.su/tom2/10-45461.html - Access date: 07/18/2022.

6. Installation Guide for UNICORE Server Components. - https://www.unicore.eu/wp-content/uploads/2016/02/manual_installation.html#_installation_of_the_perl_tsi_and_tsi_related_configuration_of_the_unicore_x_server – Access date: 07/25/2022.

7. UNICORE XUUDB . Manual. - https://www.unicore.eu/docstore/xuudb-1.3.2/manual.pdf - Access date: 07/28/2022

Article sent: 30.08.2022 г.