Free Software for the Educational Cluster of Parallel and Distributed Computing

Andriy Prodyvus
High-performance Computing Systems Laboratory
Faculty of Electronics and Computer Technologies
Ivan Franko National University of Lviv
Lviv, Ukraine
andriy.produvys@lnu.edu.ua

Roman Shuvar
The Department of System Design
Faculty of Electronics and Computer Technologies
Ivan Franko National University of Lviv
Lviv, Ukraine
shuwar@electronics.lnu.edu.ua

Vitalius Parubochyi
The Department of System Design
Faculty of Electronics and Computer Technologies
Ivan Franko National University
Lviv, Ukraine
vitalius.parubochyi@lnu.edu.ua

Roksolana Kovtko
The Department of System Design
Faculty of Electronics and Computer Technologies
Ivan Franko National University of Lviv
Lviv, Ukraine
rsulymko@ukr.net

Abstract—**The use of free and open source software in an educational cluster of parallel and distributed computing considered. The needs for the improvement of its functioning, software updating, increasing of work safety are determined. An overview of changes and improvements in the work of the cluster was made.**

Index Terms—**parallel computing, distributed computing, computing cluster, free software, GPGPU, CUDA.**

I. HIGH-PERFORMANCE COMPUTING SYSTEMS

Modern science and technology require significant computational resources for solving their tasks. These resources can give the high-performance computing systems that widely used in many countries. At the present stage, several approaches are using by creating such systems:

- technologies of parallel computing – computing clusters;
- technologies of distributed computing – Grid technologies;
- computing technology for general purpose GPU (GPGPU) – CUDA, OpenCL;
- hybrid technologies using various combinations of these approaches.

The following components are key for clusters that need to perform complex computing tasks: computing processors, RAM, volume and placing of data storage devices, communication between computers, and software. Important points are also homogeneity, scalability and ease of administration. The most important indicators of computing clusters are the high performance of processors in floating point operations (flops) and the low latency of the network that brings them together. The speed of I/O operations is not such significant for the performance of clusters.

Computational clusters reduce the calculation time compared to single computers by separating the executable task into parallel threads that exchange data over a network. Specially distinguish the HPC Cluster – High-Performance Computing Cluster. The list of the most powerful high-performance computer systems can be found in the Top500 world ranking [1]. This rating detects and tracks trends in high-performance computing. As of today, China is a leader in the number of HPC clusters with 206 systems. There are 124 systems in the US. In Western Europe are 97 clusters, in Asia (without China) – 19, in America (without the USA) – 7 supercomputers from the list, in other regions – 6. The total computing performance leader of the Top500 clusters is the United States.

Serious achievements in the field of high-performance computing are also in Ukraine. Currently, there are several clusters in academic institutions and universities in Kyiv, Kharkiv, Lviv, Dnipro, Sumy and other cities.

All large computing systems owned by government agencies or large corporations. Educational institutions and research institutes generally do not have the means to purchase powerful computers such as nCube, Cray or similar. However, with the progress of computer technology, software and the widening of the freely distributed Linux operating system, it is at the moment possible to create computing clusters with efficient performance comparable to that of supercomputers, but with a much smaller cost.

Computational clusters combine into national and international networks, which allow solving even more resource-intensive tasks by integrating and distributing computing power.

One of the most common configurations of computing clusters is the collection of computers that compiled from public components, with installed Linux operating system, and linked by Ethernet, Infiniband, or another relatively inexpensive network. In many classes of tasks and a sufficient number of nodes, they provide performance comparable to that which can be obtained using expensive supercomputers.

Distributed computations implemented on the basis of networks, which use for high-performance computing the free power of computers connected to these networks. Such technologies are called grid technologies.

Grid or Grid Infrastructure is a distributed hardware and software computer network designed to combine the computing power of individual workstations or organizations. The grid-based technology integrates regional and national computational infrastructures in order to create a united international resource for solving scientific and technical problems that require very large computing resources.

CUDA Technology is NVIDIA's software and hardware computing architecture based on the C language extension, which allows organizing access to the set of a graphic accelerator instruction and manages its memory while organizing parallel computing. CUDA helps implement algorithms that can be implemented on NVIDIA video accelerators.

II. Software for high-performance computer systems

An analysis of online information about clusters installed in various academic institutions (universities, research institutes, etc.) has shown that the most common operating systems in these clusters are different versions of Linux. For example, all supercomputer systems from the Top500 list use Linux. Such a choice is due to its free and open source status; stability and rapid correction of problems identified during the operation, as well as the availability of an extensive system of program repositories that allows quickly update the system.

The feature of the Linux operating system is a large number of distributions that differ in detail only (configuration files allocation, work with user accounts, installation package format, etc.). The most common distribution groups are Red Hat Enterprise and its derivatives (CentOS, Scientific Linux [2]), Ubuntu / Debian, SUSE Enterprise / OpenSUSE. Most often computing clusters are built based on these Linux distributions (see [1]).

The programs needed for clusters work divided into several categories: administrative, monitoring, computing. Software overview has shown that there is a large selection of both paid and free programs in each of these categories. When searching for and selecting programs for the cluster, our emphasis was on free software.

General administration of the system (OS, network, services, work with user accounts, etc.) can be carried out by the OS itself. However, in this case, the control of cluster nodes is very complicated. Often, administrative tasks are identical for all nodes, so searching for tools that would be able to manage groups of computers. Free software with such functionality can be called Spacewalk [3] and Webmin [4].

There are several software packages available to manage cluster tasks and resources distribution. The most known of them are the Torque - Maui / Moab Cluster Suite [5, 6], Open Grid Scheduler / Grid Engine [7] and SLURM [8]. The main task of these packages is the distribution of free cluster resources for user tasks, as well as creating queues of tasks in accordance with the priority of their execution.

Monitoring the status of the cluster is an important part of its administration, and therefore there is a large number of means available to monitor the status of the cluster in real time. Among the free software that can monitor the state of the cluster, the most common is Ganglia [9] and Nagios [10].

One of the main parts of the grid infrastructure software is the so-called "middleware" (intermediate software). Its task is to organize and consolidate resources in the network, to provide automatic machine-to-machine interaction (M2M) for integration into a single grid computing network. Obviously, the choice of middleware to a large extent depends on the grid network structure and the requirements for adapting the programs that will be implemented. In this regard, taking into account its capabilities and the requirements and tasks set for it, the task of selecting middleware arose.

Among the well-known middleware are both paid and free platforms. Among them, the most common are NorduGrid ARC [11], HTCondor [12] (formerly Condor), BOINC [13], and others.

General-Purpose computing on Graphics Processing Units (GPGPU) is a relatively new but promising trend in the construction of high-performance computing systems. The calculations performed by graphics processors, optimized for a large number of transactions with computer graphics arrays that can be easily paralleled.

Application packages that installed on a cluster are often available as source code and therefore require further compilation. This requires the cluster software to have a set of compilers for different programming languages. As a rule, in such cases, there is enough free set of GCC compilers, which is standard for Linux, but other compilers, including Open64, g95, LLVM, FreePascal, etc. can be installed if necessary.

Applications that use parallel computing require specialized libraries that implement the MPI standard. For today, there are two free implementations of this standard – OpenMPI [15] and MPICH [16]. Of the other computing libraries that often used in computing clusters, can be called FFTW [17] (fast Fourier transform) and BLAS [18] (linear algebra).

III. Updating the cluster software of the high-performance computing systems laboratory

A cluster of parallel and distributed computing in the Laboratory of high-performance computing systems of the Faculty of electronics and computer technologies functions since 2011. The work of the cluster has four main areas: parallel computing based on MPI technology, distributed computing using Grid technology, computing on powerful general-purpose graphics processors (CUDA technology), providing education, in particular, studying the basics of parallel and distributed computing.

The cluster can physically consist of nodes of one of two types: a control node and a computing node. In addition, the cluster may include specialized nodes: a data store, a database server, a firewall, and so on. If the cluster scales are small, then the role of specialized nodes can be placed on the control node.

The laboratory implemented an option with one control node and computing nodes.

The cluster has two separate segments of the computer network, one of which serves the messaging between nodes of the cluster, and the other provides the management of the cluster. In addition, on the computing nodes NVIDIA graphics cards installed. This enables computing using CUDA technology and the technology of hybrid computing.

The Scientific Linux (SL) [2] distribution suite, which was developed on the basis of Red Hat Enterprise Linux (RHEL) and supported by experts from one of the world's most renowned scientific institutions, Fermilab (USA), was chosen to update the cluster software. This distribution aimed at using for scientific and educational purposes. Its main advantages are the basing on one of the best Linux variants today, an operative update of the system, the availability or the ability to quickly install the necessary cluster software, an extensive system of program repositories. Current SL version is 7.5.

The cluster software update took place in three stages. At first, the operating system on the cluster control node installed. The next step was to install the operating system on the computing nodes. To automate the installation, the built-in for all RHEL-like distributions, and therefore, for the Scientific Linux, “kickstart” mechanism used. Using the *system-config-kickstart* utility, which has a graphical interface and practically repeats the interface of the local Scientific Linux installation program, a file containing all the necessary information that may be needed during the installation process was created. In addition, the program-generated file can be extended with the scripts that will be executed after the installation is complete. File access during installation was through the network.

When installing Scientific Linux on computing nodes, there was a question about the amount of installed software. Generally, it is solved by installing them the minimum number of required programs, since any extra package, especially launched as a service, takes system resources and reduces the productivity of the site. On this basis, the following software on the nodes installed:

- minimum required operating system (SL);
- programs required for cluster operation (MPI, NFS clients, SSH service, etc.);
- a nodal part of the queues and cluster resources distribution software (Grid Engine);
- the nodal part of the cluster monitoring and management software (Ganglia, Webmin);

In the process of cluster operation, it often becomes necessary to install new software or update an existing one. As already mentioned, for Scientific Linux on the Internet, an extensive repository system deployed, so in most cases, the installation procedure reduced to the use of the standard for Red Hat-like distributions *yum* utility, which automatically connects to them and installs the software and solves the necessary dependencies between packages. In the absence of the necessary software in the repositories, it is also possible to manually install using RPM-packages or compilation from the source code. The procedure for software installing on the cluster nodes greatly facilitated, because all of them (except the server) have identical hardware and software.

Webmin (version 1.881) was installed in the cluster to administer the system. This software package is actively developing and has most of the necessary tools for managing cluster nodes. It works through the Web interface and allows editing the system configuration, start, stop and restart services, manage the system remotely from other computers. A special “Cluster” interface section allows administering cluster nodes, executing commands on all nodes at once, creating and synchronizing cluster user accounts.

To work with user tasks that are performed on a cluster, the Grid Engine package (GE version 2011.11p1) is installed. It allows distributing free cluster resources to perform user tasks and create queues of tasks in accordance with the priority of their execution.

It was decided to monitor the cluster's work using the Ganglia package (version 3.7.2), which provides a real-time statistics and computing history (CPU, network) for each computing node through the Web interface.

To implement a distributed computing cluster, the HTCondor system (version 8.7.8) deployed. This system includes task queuing mechanisms, planning policies and priority task schemes, as well as a monitoring and resource management system.

To provide a cluster connection to the Ukrainian Grid system, the middleware NorduGrid ARC (version 5.4.2) has been deployed and configured.

For parallel computing using MPI technology, appropriate software libraries need to be installed. As already mentioned in section II above, free implementations of such libraries are two: OpenMPI (current version 3.1.1) and MPICH (current version 3.2.1). It was decided to install both, leaving the choice to use for users. In addition, commonly used application libraries FFTW (version 3.3.8) and BLAS (version 3.8.0) installed.

Since NVIDIA GeForce GTS 450 graphics processors installed on the cluster nodes, the freely distributed NVIDIA CUDA Toolkit package [14] is installed to provide the possibility of calculations with their use. The cluster nodes use the CUDA Toolkit 5.0 version, which corresponds to the hardware architecture of the cluster graphics accelerators and supports CUDA API 2.1. The CUDA Toolkit contains all the necessary tools for developing programs for graphics processors, including the nvcc compiler created on the basis of the Open64 compiler, application libraries: cuFFT, cuBLAS, cuRAND, cuSPARSE and NPP, shader, gdb debugger for the GPU, CUDA runtime driver included standard NVIDIA drivers, as well as background information on programming techniques on CUDA and the CUDA Developer SDK, which contains source codes and examples of programs, utilities and documentation necessary for the full development of applications on CUDA. Additionally, depending on the tasks, specialized packages and utilities can be used, including CUDA-MEMCHECK, cuDNN, TensorRT, OpenCV, Geometry Performance Primitives (GPP) and others.

To date, there are a large number of application software packages that allow for parallel computing. Installing such packages was carried out on user's request. In particular, it was installed ABINIT (version 8.8.4) – a software kit for calculating optical, mechanical, vibration and other properties of materials.

IV. Conclusions

There is a big need for high-performance computing systems in the world. As a result, their number is constantly increasing. Building such systems requires significant resources. However, due to the availability of free and open source software, it is possible to create high-performance computing systems at a fairly modest cost.

In the laboratory of high-performance computing systems of the Faculty of Electronics and Computer Technologies of the Ivan Franko National University of Lviv, the software of the parallel and distributed computing cluster updated.

In choosing the operating system, programs that provide cluster operation, application programs deliberately set the goal to rely on free and open source software. This is due to the fact that today there is wide selection of free software products that allow building a fully functional cluster system. A review of operating systems, system administration, monitoring, computing was performed and choice of corresponding software was made in each of the categories.

References

[1] TOP500 Supercomputer Sites [Electronical resource]. – Access mode: https://www.top500.org/.

[2] Scientific Linux [Electronical resource]. – Access mode: http://www.scientificlinux.org/.

[3] Spacewalk. Free & Open Source Systems Management. [Electronical resource]. – Access mode: https://spacewalkproject.github.io/.

[4] Webmin [Electronical resource]. – Access mode: http://www.webmin.com/.

[5] TORQUE Electronical resource]. – Access mode: https://en.wikipedia.org/wiki/TORQUE.

[6] Maui Cluster Scheduler [Electronical resource]. – Access mode: https://en.wikipedia.org/wiki/Maui.

[7] Open Grid Scheduler/Grid Engine [Electronical resource]. – Access mode: http://gridscheduler.sourceforge.net/.

[8] SLURM Workload Manager [Electronical resource]. – Access mode: https://slurm.schedmd.com/.

[9] Ganglia Monitoring System [Electronical resource]. – Access mode: http://ganglia.sourceforge.net/.

[10] Nagios [Electronical resource]. – Access mode: https://www.nagios.org/.

[11] NorduGrid [Electronical resource]. – Access mode: http://www.nordugrid.org/.

[12] HTCondor – Home – Computer Sciences Dept. – UW-Madison [Electronical resource]. – Access mode: https://research.cs.wisc.edu/htcondor/.

[13] BOINC – Open-source software for volunteer computing [Electronical resource]. – Access mode: https://boinc.berkeley.edu/.

[14] CUDA Toolkit [Electronical resource]. – Access mode: https://developer.nvidia.com/cuda-toolkit.

[15] Open MPI: Open Source High Performance Computing [Electronical resource]. – Access mode: https://www.open-mpi.org/.

[16] MPICH – High-Performance Portable MPI [Electronical resource]. – Access mode: https://www.mpich.org/.

[17] FFTW Parallel FFTW [Electronical resource]. – Access mode: http://www.fftw.org/fftw2_doc/fftw_4.html.

[18] BLAS (Basic Linear Algebra Subprograms) [Electronical resource]. – Access mode: http://www.netlib.org/blas/.