Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 186 (2021) 40–47

Procedia Computer Science

www.elsevier.com/locate/procedia

14th International Symposium "Intelligent Systems"

Deployment of parallel computing in a hybrid high-performance cluster based on virtualization technologies

K.I. Volovich[a,*], S.A. Denisov[a], S.I. Malkovsky[b]

[a]*Federal research center "Computer Sciense and Control" of the Russian Academy of Sciences, 44 Vavilova st. korpus 2, Moscow, 119333, Russia*

[b]*Computing Center of the Far Eastern Branch of the Russian Academy of Sciences, 65Kim Yu Chen st., Khabarovsk, 680000, Russia*

Abstract

The article discusses the application of MPI technology when performing calculations on a hybrid high-performance computing cluster using virtualization on a container base. The features of the application of interprocess interaction in a virtualization environment associated with the construction of a single space of containers interacting over a high-performance computing network interconnect are considered. Approaches and algorithms for deploying and executing parallel processes in a computer cluster are proposed. The issue of the functioning of computing process control systems in the provision of PaaS services using virtualization technologies is considered.

*Keywords:*HPC cluster; hybrid architecture; graphics accelerator; workload management system; CPU; GPU; MPI

1. Introduction

The solution of modern applied and fundamental scientific problems is inextricably linked with the use of computing technologies. Almost any scientific field uses mathematical modeling with the use of computer and calculation technology [1].

To solve scientific problems requires an ever-increasing computing performance.

* Corresponding author. Tel.: +7-499-135-4320.
E-mail address: kvolovich@frccsc.ru

1877-0509
10.1016/j.procs.2021.04.122

The main approach to increasing productivity is the development of parallel computing technology, which allows solving scientific problems simultaneously on a group of computing nodes. Approaches to the organization of parallel computing can be different, but, in the general case, they are aimed at creating computing clusters that distribute the parts of the computing problem among themselves. Examples of such clusters are grid and hadoop [2, 3], which allow parallel execution of tasks on distributed computing nodes.

Another approach to the construction of computer systems intended for parallel computing is the creation of computer clusters consisting of many computing nodes connected by a high-performance data transmission network. Such systems include supercomputers and high-performance clusters built by combining many nodes into a single computing cluster through a high-performance network. When building supercomputers, we are talking about systems containing many identical computing modules[1]. High-performance computing clusters can also be built on the basis of various server components that allow performing scientific tasks in parallel mode [4].

The parallel execution of one task on a group of computing nodes requires the use of specialized technologies that provide a group launch of processes on one or more cluster nodes, synchronization of processes, data exchange between parallel processes.

The most common technology for solving this problem when performing calculations in the field of applied and fundamental science is Message Passing Interface (MPI) [5]. Today MPI is a generally accepted mechanism for organizing parallel computing in a multi-machine cluster.

There are a number of implementations of MPI technology based on open source code and proprietary solutions (OpenMPI, Intel MPI, IBM Spectrum MPI), and other manufacturers of high-performance equipment are also involved in the implementation and optimization of MPI solutions.

At present, hybrid supercomputers [7] and high-performance computing systems [6], built on the basis of classical central processors and specialized computing accelerators, are becoming more widespread.

The peculiarities of the use of such clusters is the distribution of computational workload between classical computers and accelerators, which allows you to get a significant increase in performance within a single computing node.

Combining the hybrid architecture of the computing clusters with MPI technology requires the development of new approaches to organizing the computing process in a multi-nodes computing clusters.

When developing approaches, the following points are the starting points:

- A hybrid high-performance computing complex is designed for the simultaneous parallel execution of several scientific tasks;
- Applied and system software of various scientific tasks may be incompatible with each other;
- The preparation of calculations on graphic accelerators and switching tasks to them require a lot of time, therefore it is advisable to allocate the accelerator exclusively for one scientific task;
- For the preparation of program code that uses the technology of programming on graphic accelerators (CUDA) and MPI technology, specialized settings of the programming code development environment are required.

Thus, for the full use of MPI technology in hybrid computing systems, the creation of individual software environments using virtualization is required. This will allow for each applied task to allocate an independent software environment that does not affect other scientific tasks.

The optimal virtualization technology for use in a hybrid high-performance computing complex is container technology[8]. In this case, each virtual environment from the point of view of the operating system of the computing node is a normal process and the cost of resources for scheduling and managing virtual environments is minimal.

For the effective use of MPI when running applications in individual containers, it is necessary to develop methods for applying MPI technology when using containers, develop algorithms for deploying individual runtime environments and include them in the general area of the MPI environment, develop approaches to the interaction of the computing task management system with container management systems and MPI processes, to manage computing in a hybrid high-performance cluster.

2. Container MPI Approaches

MPI technology allows you to organize various schemes for the parallel execution of computational processes aimed at solving one scientific problem. Process execution can be organized on one or several nodes of a high-performance computing cluster. In this case, the exchange of data between processes is organized through the MPI interface.

The use of container technology introduces an additional level of nesting when distributing application processes across computing components.

The following distribution variants for MPI processes are possible when using containers in a high-performance computing cluster.

Variant 1. Execution of several processes in one container by executing mpirun in the same container.

The advantage of this method is the absence of the need for additional configuration of the software environment of the computing cluster for MPI processes. To implement this option, it is enough to deploy MPI libraries in the container. The user has the ability to perform a group of parallel processes inside the container and get the resources of the computing node on which the container operates.

The disadvantages of this method is the inability to run this option on a group of nodes connected by an interconnect network.

Thus, this scheme can be implemented in order to debug application software or perform scientific calculations that do not require the resources of a whole cluster and which can be performed on a single computing node.

Variant 2. Running one process in one container by running mpirun in an external environment.

With this method of execution, containers are created on the computing nodes by the number of processes being executed, and then the mpirun utility creates one process in each container that is designed to solve an applied scientific problem. Parallel processes can exchange messages and data via the MPI interface using shared memory of a computing node or a high-performance interconnect network. The mpirun utility is executed outside the containers with application processes; it can be performed either on one of the computing nodes of the cluster or in a specialized container.

The advantage of this method is the ability to perform parallel processes on a group of nodes of a high-performance computing cluster using standard MPI mechanisms for interaction between parallel processes.

The disadvantage of this method is the need to dynamically form a pool of containers (based on their IP addresses) for transferring to mpirun utility as source data for creating parallel processes.

These shortcomings are not critical, the problem of creating a dynamic pool of container addresses is solved by developing special scripts to create a container environment and execute parallel processes. The problem of container multiplicity is not critical, because, unlike virtual machines, the overhead of container operation is small compared to operating system processes.

Variant 3. The combination of variants 1 and 2.

With a combination of variants 1 and 2, it is possible to execute processes on a group of nodes of a high-performance cluster. At the same time, one container is created on each cluster node to solve a specific application, in which a group of parallel processes is performed. Utility execution should also be performed outside containers intended for the application.

The advantages of this option include the reduction in the number of containers operating on one node of a high-performance cluster.

The disadvantages are similar to variant 2 and are associated with the need to create a functioning environment using special software.

Note that for the application of MPI technologies and containers in a hybrid high-performance cluster, it is necessary to ensure their integration with the workload management system. It is required to create a mechanism that allows the control system to create a group of containers intended for the application, to create a list of ip-addresses of the nodes to be transmitted to the mpirun utility and to execute the utility on one of the cluster nodes or in a separate container.

When performing such actions, the system should keep track of the resources allocated to each parallel process, allow managing the execution of a group of containers, provide prioritization, queuing, and monitoring of resource use. Also, using the control system or additional software integrated with it, “garbage collection” should be carried

out in case of program failures and errors. In the event of a malfunction in any parallel process running in the container, leading to the need to complete the program, all containers related to this program must be unloaded.

Management of jobs using GPU resources imposes additional requirements on the operation of the workload management system. Based on the fact that switching the accelerator between computational tasks is a lengthy process, the control system should allocate GPU resources exclusively for the computational process. In the presence of parallel processes, it is optimal to allocate to each of them one or more graphics accelerators for the entire duration of the calculations.

Below are the algorithms for the formation of an individual software environment using container technology, focused on the use of MPI and allowing the use of computing resources of graphic accelerators of calculations.

3. MPI-based computing job deployment algorithms using container technology

When deploying containers according to option 1, you must perform the following steps:

- Using the docker virtualization environment, create a container from the base image;
- Configure the individual execution environment to solve an applied scientific problem [9];
- Install MPI libraries into the container (if they are not in the base image);
- Execute mpirun in the container indicating the number of parallel processes on the local node involved in solving the scientific problem.

Figure 1 shows the MPI task execution algorithm in one container.

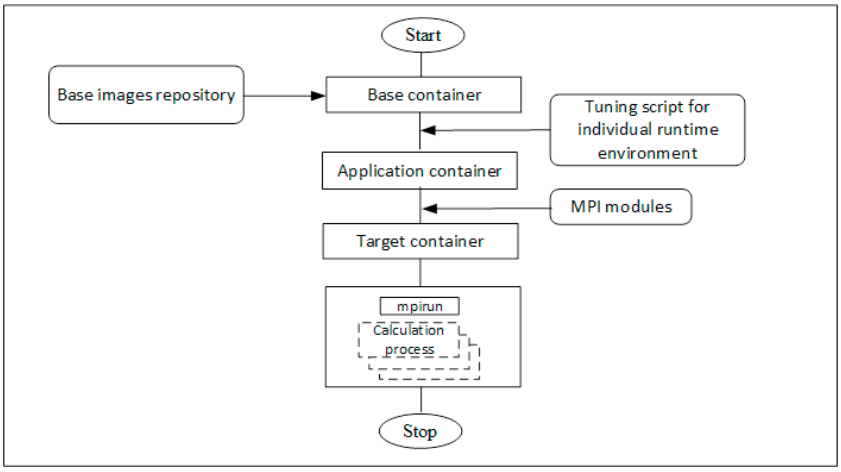

Fig. 1. Algorithm for executing a task in a single container

Note that when using parallel graphics accelerator resources, it is preferable to proceed from the rule of exclusive allocation of the accelerator to the process in order to avoid task switching and performance degradation.

When deploying containers according to variant 2, you must perform the following steps:

- Using the docker virtualization environment, create a group of containers from the base image according to the number of planned parallel processes on one or more nodes of the computing cluster;
- Algorithm for executing a task in a single container;
- In each container, configure the individual execution environment. Note that at high costs for the dynamic creation of an individual environment, it is preferable to create and save it in the base container in advance. In this case, you can skip the data step;
- Install MPI libraries in containers (if not in the base image);

- Create a list of ip addresses of deployed containers for transfer to mpirun utility as source data;
- Execute mpirun on the cluster node, specifying as a source data a list of ip addresses of deployed containers. In this case, the number of processes in the container should be limited to one.

Figure 2 shows the MPI task execution algorithm in a group of containers on different nodes of the cluster.

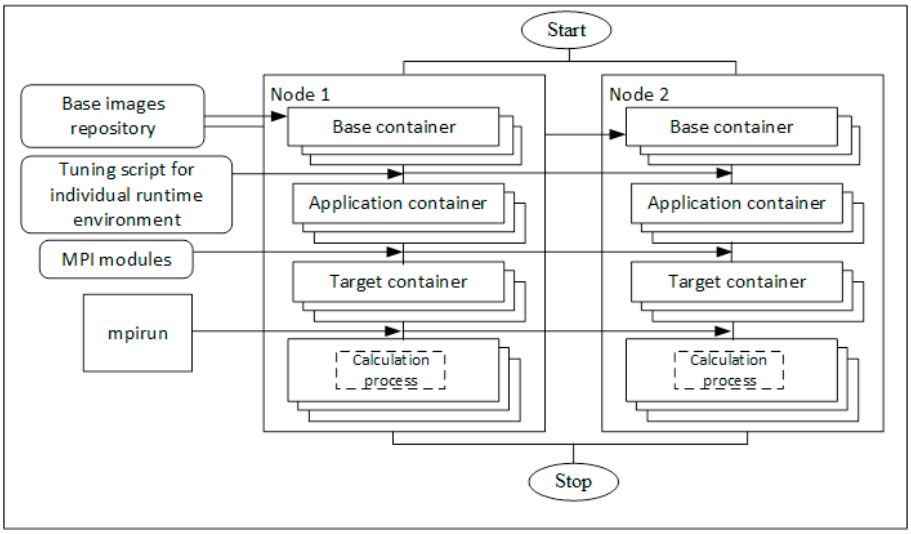

Fig. 2. Algorithm for executing a task in a container group

In this variant, you should pay attention to the configuration of docker virtualization tools and the configuration of parameters and drivers for the interconnect network. It is necessary to ensure interaction at the IP addresses of containers running on different nodes of the computing cluster. The visibility of the interconnect network devices of their containers at all nodes of the cluster must also be ensured.

When deploying containers according to variant 3, you must perform the following steps:

- Using the docker virtualization environment, create one container from the base image on each of the cluster nodes involved in solving the scientific problem. We will call such a container a “pseudo-node” in which all parallel processes of one scientific task are carried out.
- In each container, configure the individual execution environment similarly to option 2;
- Install MPI libraries in containers (if not in the base image);
- Create a list of ip-addresses of deployed pseudo-nodes for transmission to mpirun utility as source data;
- Execute mpirun on the cluster node, specifying as a source data a list of pseudo-node ip-addresses. Moreover, the number of processes in the container can be different and meet the requirements of a scientific task.

Figure 3 shows the MPI task execution algorithm using pseudo nodes.

When organizing the computational process according to this option, several “pseudo-nodes” can be executed on the physical node of the cluster — containers in which parallel processes of different applied tasks are performed. For each application on a group of physical nodes, one pseudo-host is executed — a container with a number of parallel processes.

Note that for the hybrid high-performance computing complex to work correctly in parallel with several scientific tasks, the following software must be deployed and configured on the nodes of the cluster:

- Modules and libraries of the MPI interface;
- Drivers, utilities, software modules of the interconnect network;

- A system for managing computational tasks;
- In containers of an individual software environment, the following should be deployed;
- Application software, integrated environments, utilities designed to solve the applied problem;
- MPI libraries that provide interprocess communication;
- Interconnect libraries and network modules for MPI functioning on the basis of a high-performance network.

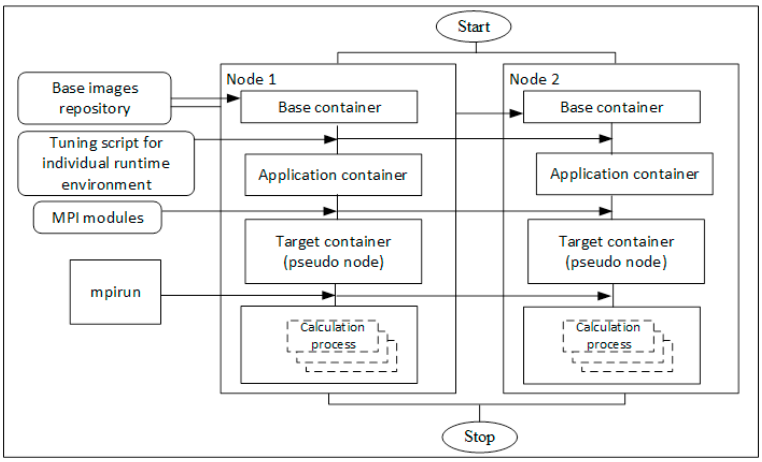

Fig. 3. MPI task execution algorithm using pseudo nodes

Providing the resources of a hybrid high-performance cluster can be considered as a cloud service of a digital platform [10].

The formation of a repository of basic images intended for applied research is one of the main tasks of the competence center, which is part of the unit that operates the computing cluster and provides the provision of cloud services.

The hybrid high-performance computing cluster of the CKP «Computer science» of FRCCSC RAS [11] in the repository of basic images contains software environments designed to solve applied problems in the areas of artificial intelligence, mathematical modeling, materials science, and biomedical chemistry. The presence of such images allows us to provide PaaS-type cloud services to clients to solve applied problems in these fields of science [12-15].

4. The functioning of the workload management system in the organization of parallel computing

In the general case, the problems of managing computing jobs in a hybrid high-performance computing cluster boils down to distributing computing resources between tasks, managing job queues, priorities, and accounting for resources used. Open source and proprietary workload management systems, such as, for example, SLURM, IBM Spectrum LSF, successfully cope with these tasks.

Support for MPI technology is provided in these computing process control systems by default. However, when using this technology in combination with container virtualization technology, it creates problems related to the creation and management of virtual environments and the execution of applied computing tasks in them.

Integration through the development of middleware (scripts), which acts as an aggregator of information about containers and nodes, allows you to perform the following actions necessary to execute custom applications:

- Allocate the necessary computing resources to a group of physical nodes of a hybrid high-performance computing cluster;
- Create the necessary number of containers or pseudo-hosts;
- Configure the individual execution environment;

- Perform automatic (dynamic) and predefined (static, by template) allocation of the processes of one applied task on the hosts and pseudo-hosts of the hybrid cluster;
- To manage the execution of the computational task on all nodes and pseudo nodes of the cluster in accordance with the policies for servicing users of the computing cluster.

When allocating computing resources to increase the efficiency of using the cluster, the workload management system should proceed from the rule that GPU resources should be allocated exclusively to each parallel process within the MPI framework to avoid loss of time for switching GPUs. CPU resources should be allocated by cores to ensure parallel execution of processes that do not require GPU resources. Features of reserving CPU and GPU resources when performing one or more tasks using MPI. Such a resource management policy makes it possible to increase the efficiency of servicing user tasks, but does not affect the efficiency of executing the applications themselves, which depends on the quality of the program code. Approaches to evaluating the effectiveness of program code using CPUs and GPUs are associated with profiling user applications when working with central processes and computing accelerators [7].

5. Conclusion

The parallel execution of MPI in the cloud infrastructure using container virtualization requires the development of algorithms for deploying containers of individual runtime environments and their inclusion in the general field of the MPI task.

To provide a cloud service, the creation of specialized containers is required, including software libraries for working with MPI and intrconnect, as well as integrated packages that are configured to interact between parallel processes.

For the high-quality provision of a PaaS-type cloud service, the competence center must create and keep up to date a repository of container images focused on solving applied problems in various fields of science and technology.

The development of middleware for managing tasks oriented to parallel execution under MPI control will allow controlling the computing process in a hybrid cluster using standard control systems, taking into account resources, ensuring cluster loading and priority task execution.

Acknowledgments

The research is partially supported by the Russian Foundation for Basic Research (project 18-29-03100). The research was carried out using infrastructure of sharing research facilities CKP «Computer science» of FRCCSC RAS [11].

References

[1] Abramov, S.M. and Lilitko, E.P. (2012). The status and development prospects of computing systems of super-high performance. *Information Technologies and Computing Systems*, 2, pp.6-22.

[2] Bryukhov, D.O., Vovchenko, A.E., Zakharov, V.N., Zhelenkova, O.P., Kalinichenko, L.A., Martynov, D.O., Skvortsov, N.A. and Stupnikov, S.A. (2008). The middle ware architecture of the subjectmediators for problemsolving over a set of integrated heterogeneous distributed information resources in the hybrid grid-infrastructure of virtual observatories. *Informatics and applications,* 2(1), pp.2-34.

[3] Budzko, V.I., Bryukhov, D.O., Devyatkin, D.A., Skvortsov, N.A., Smetanin, N.N., Stupnikov, S.A. and Shelmanov, A.O. (2017). Scalable architecture of a system for extracting information from data on the Arctic zone. In: *Collection of Information Technologies and Mathematical Modeling of Systems 2017. Proceedings of the international scientific and technical conference*. pp.90-94.

[4] Zatsarinny, A.A., Gorshenin, A.K., Kondrashev, V.A., Volovich, K.I. and Denisov, S.A. (2019). Toward high performance solutions as services of research digital platform. In: *Procedia Computer Science*. vol.150, pp.622-627.

[5] Gorchakov, A.Y. (2019). K-frontal method of nonuniform coverings. *International Journal of Open Information Technologies*, 7(8), pp.65-69.

[6] Abramov, S.M. (2018). Analysis of supercomputer cyber infrastructures of the leading countries of the world. In: *Supercomputer technologies. SKT-2018. Materials of the 5th All-Russian Scientific and Technical Conference.* Rostov-on-Don, pp.11-18.

[7] Volovich, K.I., Denisov, S.A., Shabanov, A.P. and Malkovsky, S.I. (2019). Aspects of the assessment of the quality of loading hybrid high-performance computing cluster. In: *CEUR Workshop Proceedings*. vol.2426, pp.7-11.
[8] Volovich, K.I. and Denisov, S.A. (2019). The main scientific and technical problems of using hybrid HPC clusters in materials science. In: *Materials of the I international conference "Mathematical modeling in materials science of electronic components. MMMEC-2019.* Moscow, pp.15-18.
[9] Volovich, K.I., Denisov, S.A. and Malkovsky, S.I. (2019). The formation of an individual modeling environment in a hybrid high-performance computing cluster. *News of higher educational institutions. Materials of electronic equipment*, 22(3), pp.197-201.
[10] Zatsarinny, A.A., Kondrashev, V.A. and Suchkov, A.P. (2019). The system of scientific services as a relevant component of scientific research. *Systems and Means of Informatics*, 29(1), pp.25-40.
[11] FRC CSC (2020). *The regulation CKP «Informatics»*. [online] Available at: http://www.frccsc.ru/ckp [Accessed 23 Jun. 2020].
[12] Mikurova, A.V., Skvortsov, V.S. and Raevsky, O.A. (2018). Computer assessment of the selectivity of inhibition of muscarinic receptors M1-M4. *Biomedical Chemistry: Research and Methods,* 1(3), pp.1-9.
[13] Abgaryan, K.K., Zhuravlev, A.A. and Reviznikov, D.L. (2017). Parallel data processing in computer modeling problems of high-speed interaction of solids. In: *Materials of the XX International Conference on Computational Mechanics and Modern Applied Software Systems. VMSPSPS 2017.* Alushta, pp.27-28.
[14] Yadrintsev, V.V., Klyubina, K.V., Tikhomirov, I.A. and Gershelman, A.F. (2018). Choosing a server solution for a digital text search and analysis platform. *Systems and Means of Informatics,*28(3), pp.26-38.
[15] Goloviznin, V.M., Gorchakov, A.Yu., Zalesny, V.B, Mayorov, P.A., Mayorov, P.A., Semenov, E.V. and Soloviev, A.V. (2019). A numerical model for solving equations of geophysical hydrodynamics using a new class of conservative difference schemes that preserve angular momentum on computational grids. In: *Seas of Russia: Fundamental and Applied Research Abstracts of the All-Russian Scientific Conference*.
[16] Zatsarinny, A.A., Gorshenin, A.K., Volovich, K.I., Kolin, K.K., Kondrashev, V.A. and Stepanov, P.V. (2017). Management of scientific services as the basis of the national digital platform "Science and Education". *Strategic priorities*, 2(14), pp.103-113.
[17] Kondrashev, V.A., Volovich, K.I. Service management of a digital platform on the example of high-performance computing services. In: *Materials of the International scientific conference.* Voronezh, September 3-6. pp.217-223.