Approaches to the Organization of the Computing Process of a Hybrid High-Performance Computing Cluster in the Digital Platform Environment

Alexander A. Zatsarinny[1], Vadim A. Kondrashev[1], Aleksei A. Sorokin[2]

[1] Federal research center 'Computer Science and Control' of the Russian Academy of Sciences, Moscow, Russia, AZatsarinny@frccsc.ru, vd@ipi.ac.ru
[2]Computing Center of Far Eastern Branch Russian Academy of Sciences, Khabarovsk, Russia, alsor@febras.net

Abstract

The article discusses approaches to managing the distribution of resources of high-performance computing clusters in order to perform a wide range of calculations. The interdisciplinary scientific studies of artificial intelligence, "big data", modeling "digital twins" are the areas to apply the method that has been developed. A hybrid HPC jobs definition as a scientific service is proposed by declaring its parameters in a service-oriented computing cluster of a digital platform. Algorithms for managing services of a hybrid calculator are considered.

1 Introduction

Over the past few years, the need for high-performance computing has increased significantly [1-3]. This is due to interdisciplinary research and development of artificial intelligence technologies, processing of big data, modeling the behavior of digital twins.

At the same time, the progress of microelectronics and the widespread use of hybrid computing architectures made the operation and application of hybrid high-performance computing clusters available in medium research teams. Further, we will call them HHPCC (hybrid high-performance computing clusters). This expands the number of providers of computing services on the Internet and creates the prerequisites for the emergence of digital platforms with HHPCC services [4-6] in the form of scientific services.

Note that HHPCC is focused on solving problems of a wide range of interdisciplinary research, including tasks of artificial intelligence, "big data", digital twins, and determines the specifics of the organization of the computational process on such clusters.

The computing environment requires greater flexibility and adaptability to changing requirements from applications and user tasks.

At the same time, it is possible to identify the main problems that arise when using cloud technologies:

- the problem of deploying the science frameworks in a hybrid HHPCC;
- the problem of adapting the user code to the hybrid computer;
- the problem of creating an individual environment for the execution of tasks.

Usually, supercomputers use calculation technology, which is based on batch processing of tasks in a single computing environment. This is the most effective way to use the computing resources of a supercomputer.

However, in the HHPCC it is necessary to ensure the following modes for research teams:

- an individual computing environment for performing applications;
- the interactive work of research teams with the HHPCC in their individual environments;
- batch processing service.

The first. There are several tools to develop artificial intelligence applications, that are used by the different research teams. The simultaneous presence of several tools on one compute node is not expedient. This limits the ability to configure and use them for different teams. Working with tools involves a dynamic change in their settings. This includes performing operations that require administrative rights, for example, installing additional system software and software libraries.

In: Sergey I. Smagin, Alexander A. Zatsarinnyy (eds.): V International Conference Information Technologies and High-Performance Computing (ITHPC-2019), Khabarovsk, Russia, September 16-19, 2019, published at http://ceur-ws.org

Operations that require changes to the computing environment are performed only by the administrator by prior agreement (in accordance with the adopted information security policy). This creates discreteness in the research work of the research team.

The virtual containerization technology becomes the optimal solution for creating fully functional individual execution environments. Creating an individual execution environment for a research team is an urgent task. The individual execution environment should function in the HHPCC computing environment and solve the problem of constraints for the research team.

There are several containerization systems. A containerization system in HHPCC is used to create individual environments for a small number of users, and not to create microservice information systems. The well-known classical docker system is a good approach for these purposes. Like a virtual machine, the docker runs its processes in its own pre-configured operating environment. But at the same time, all docker processes run on a physical host, sharing all processors and all available memory with all other processes running on a host. The approach used by the docker is in the middle between running everything on a physical server and full virtualization offered by virtual machines. In this case there is no difference in the management of a task containing an application, or containing a container with an application. Therefore, further in the article we are not focused on the contents of the task.

The second. Development tools and access to HHPCC graphics accelerators for debugging are required to create efficient applications for hybrid computers. It is a complicated problem to create such a development environment outside the computing cluster. In practice, application development environments are deployed on HHPCC.

In interactive mode, the base container is loaded into the HHPCC computing environment and remains active. The user gets online access to the base container via the remote access protocol. Inside the container, the user has the ability to consistently perform actions on the installation, compilation, debugging of any software necessary to perform an application task. It is possible to fix the sequence of user actions in the form of a script. The script will further allow to deploy individual virtual environment automatically.

Thus, it is possible to create a whole series of dynamically formed containers with an individual computing environment based on a single image with a different set of software. Such a set of containers will allow solving similar application tasks of one research area that require fine tuning of the computing environment.

In this case, the workload management system should allocate resources for the development environment, which is provided to developers in the time-sharing mode.

The third. Batch processing is a classic way to efficiently load the computing infrastructure [7], which has been used on supercomputers. Obviously, the HHPCC workload management system should use this mode for long-term application testing or for performing calculations.

Therefore, the actual task is to describe applications in the form of scientific services in order to organize calculations at the HHPCC as part of a digital platform. In the rest of the article we discuss solutions to describe jobs and manage the computing process for the HHPCC as part of a digital platform.

2 Description of the Computing Application as a Scientific Service of the Digital Platform

The digital platform should provide a job representation as a scientific service in the cloud computing environment [8-10]. The scientific service should contain information that is necessary for the workload management system in order to perform the computing task. When ordering a scientific service workload management system must allocate the required computing resources in accordance with the specified maintenance policy.

Resources must be allocated to ensure the functioning of an interactive developer environment or to process computational models. The types of computing resources that are assigned to a scientific service can be as follows:

1) CPU resources (CPU cores or CPU core threads);
2) GPU resources;
3) RAM;
4) Disk space.

The maintenance policy of a scientific service sets a set of rules determining the processing of a task by the workload management system. It is proposed to select the maintenance policy of the computing task depending on the value of the following basic parameters of the scientific service:

1) consumed resources - the nomenclature of computational resources required to perform the task;
2) task priority - the value that the workload management system uses to apply the required service policy to the job;
3) task execution mode - interactive or batch;
4) execution time - the time limit for the task, after which the workload management system can perform the actions defined by the service policy (for example, stop the task).

Let us explain the listed parameters of the computing task of a scientific service, which determine the policy for its maintenance.

Consumed resources. Consumed resources are usually declared in order to reserve resources. In this case, until there are not enough available resources, the workload management system will not start the job. When performing a task, the workload management system will control the exceeding of the threshold for the volume of resources and execute control actions in case of crossing this boundary. Along with the reservation of resources for HHPCC it is advisable to be able to describe the resources consumed without reservation. This is necessary to classify the job and

determine its service policy. In this case, the job will begin to run without waiting for resources and without monitoring their use, but with the inclusion of a service policy consistent with this declaration.

A priority. Workload management system of HHPCC uses an initial priority, which is determined either when describing a job or by the service policy. A dynamic priority is also used, which is set by the workload management system depending on the job's behavior.

Execution mode. This parameter must be explicitly specified by the user when describing the job. This will enable the workload management system to perform the job either in time-sharing mode to provide interactive mode or in batch processing mode.

Execution time. During the job execution, the workload management system HHPCC determines a period of the job execution. If the period exceeds the deadline specified in the description of the scientific service, then the workload management system will perform the action specified by the service policy.

Maintenance policies are defined by the HHPCC administrators and include the values of the above parameters for the default setting. Therefore, when describing the computing job of a scientific service, it is possible to indicate the need for servicing an application in accordance with one of the maintenance policies.

3 Main Workload Management Loop for Computing Jobs Management of Scientific Services

The following basic workload management algorithm is proposed for handling the application of a scientific service for HHPCC:

Step 1: The workload management system places the job in the queue.

Step 2: The workload management system in a cycle analyzes the parameters of jobs in the queue. In accordance with the queue service policy, it defines a job that requires a control action and performs it.

Actions for the distribution of computing resources are determined by the policies of the HHPCC. This is a set of rules for distributing computing resources between tasks and managing the course of their competitive performance.

Step 3. The completed job is removed from the queue. A job can complete its execution on its own or under the influence of the workload management system of the HHPCC in accordance with the maintenance policy. For example, in the case of long-term execution or exceeding the limit of use of computing resources, the workload management system can stop the execution of the scientific service job. Note that the service policies may include other actions. For example, changing the priority of a job, pausing a job, movement a job to another queue.

After completing the job, the workload management system should check the correctness of the release of computing resources, and return the resources to the pool.

Resource management at the job level occurs within the same queue. The main criterion in the allocation of resources is the priority of the computing task. Tasks performed in interactive mode are given the highest priority and are limited in time to complete them.

4 Levels and Algorithms for Managing the Computing Tasks of Scientific Services.

Policies for managing the computing resources of the HHPCC digital platform are formed at the following levels:

1) job level;

2) queue level.

Resource management at the job level occurs within the same queue. The main criterion in the allocation of resources is the priority of the computing task. Tasks performed in interactive mode receive the highest priority, but their execution time is limited.

The job priority policy with the highest priority is as follows:

- the task receives the requested resources;
- in case of lack of resources, the workload management system for releasing the requested resources suspends the tasks with the lowest priority and the highest execution time;
- over time, the priority of the task decreases;
- among tasks with same priority, the task with the longest execution time is the priority task.

The job priority policy with the low priority is as follows:

- tasks are allocated resources on the Best Effort principle;
- in case of shortage of resources, the tasks are waiting in the queue until the appearance of free resources;
- while the job waits in the queue, its task priority increases.

Resource management at the queue level is based on the queue pool. For HHPCC, it is proposed to create at least three queues: two queues for jobs performed in batch mode (general and additional queues) and a queue for jobs performed in interactive mode (interactive queue).

The interactive queue allows you to put the jobs for execution in online mode. This queue is used to perform debugging and development functions. Queue service policy is limiting the execution time of the job. If necessary, the amount of resources may be limited, for example, no more than one GPU per node, no more than two nodes.

Such restrictions allow you to perform jobs using several nodes and the GPUs of the cluster, but do not allow to obtain high priority to bypass the existing queue service policies.

The main queue allows you to put the jobs for execution but the time of functioning is determined by one or another value. In general, there may be several queues of this type. In case of exceeding the task execution time in comparison with the declared workload management system can:

- continue the job;
- pause the job;
- pause the job and move it to a queue with a lower priority.

A limited resource pool is provided for the queue, and jobs may be suspended.

An additional queue allows to place jobs for execution for which it is not possible to estimate the time of their execution. The queue has the lowest priority. A limited resource pool is provided for the queue, and jobs may be suspended.

Queues are defined by the following parameters:

- interactive queue: priority - high, availability of resources - according to the Best Effort principle, the maximum execution time of the task - is strictly limited;
- general queue: priority - medium, availability of resources - within a limited pool, maximum time for task execution - is unlimited;
- additional queue: priority - low, availability of resources - within a limited pool, the maximum execution time of the task - is unlimited.

General recommendations for queuing are the following characteristics:

- there should be no queues with the same parameters;
- the higher the priority of the queue, the more resources available to it, but the maximum execution time of the task is lower compared to queues with lower priorities.

Workload management system analyzes the parameters that describe the job, and puts in the queue that most satisfies the parameters of the job. If all the job parameters are specified by default, the task is placed in an additional queue.

Jobs are pending or executed within the same queue to which they were assigned. If the task exceeds the maximum execution time, then it is either forcibly terminated (in the case of an interactive queue), or moved to the next queue, which has a lower priority.

Resource management can occur simultaneously at the queue and job level. Then control algorithms are applied at the job level and at the queue level.

5 Conclusion

The article proposes an approach to managing the computational process in a high-performance cluster as part of a digital research platform, which makes it possible to efficiently provide the necessary computational resources in the form of a modern scientific service [11, 12].

The use of hybrid architectures of computing systems for the provision of scientific services of a digital platform is a prospective direction for the development of high-performance computing. The proposed workload management solutions can be used to build the HHPCC for scientific calculations. Creating hybrid clusters is based on modern virtualization technology. This makes it possible to include in the cluster a wide range of computing platforms, architectures and technologies [13].

A feature of the use of virtualization based on docker is the possibility of simultaneous execution of runtime environments intended for research in various fields of science. Such a computational organization enhances the efficiency of the HHPCC and reduces the research cost.

Implementing workload management system to support research in the field of artificial intelligence, “big data”, modeling “digital twins” is possible either with the use of open technologies and the OpenSource software code, or proprietary solutions of commercial vendors.

The modes workload management system presented in the article are tested at the HHPCC of FRCCSC1 and are planned for deployment on its digital platform. The HHPCC of FRCCSC includes:

- IBM Power System AC922: 2 CPU Power9 2.87 GHz, 20 cores, 1 Tb RAM, 4 GPU NVidia Tesla V-100 16 Gb RAM – 2 nodes
- Huawei G560 V5: 2 CPU Intel Xeon Platinum 2.1 GHz, 24 cores и 1.5 Tb RAM, 4 and 8 GPU NVidia Tesla V-100 SXM2 32 Gb RAM - 2 nodes;

HHPCC uses InfiniBand EDR 100G computer networks, Ethernet 10G and the Internet digital platform FRCCSC, as well as a corporate 1 Pb data storage system.

Currently, the work is underway to create a service of personal accounts and an integration bus for the digital platform FRCCSC. Upon completion of the work, the high-performance computing service and services of various calculations based on the HHPCC will be available to platform users in the PaaS (Platform as a Service), SaaS (Software as a Service), RaaS (Research as a Service).

[1] Federal Research Center ‘Computer Science and Control’ of Russian Academy of Sciences. Available at: http://hhpcc.frccsc.ru

Acknowledgements

The research is partially supported by the Russian Foundation for Basic Research (project 18-29-03091).

References

1. Abramov, S.M.: Analysis of supercomputer cyber infrastructure of the leading countries of the world // Supercomputer technologies (CKT-2018). Materials of the 5th All-Russian Scientific and Technical Conference. Rostov-on-Don. p. 11-18. (2018)

2. Sobolev, S., Antonov, A, Shvets, P., Nikitenko, D., Stefanov, K., Voevodin, V., Voevodin, Vl., Zhumatiy, S.: Evaluation of the octotron system on the Lomonosov-2 supercomputer // Conference materials parallel computing technologies. Rostov-on-Don (2018)

3. Afanasyev I., Voevodin V. The comparison of large-scale graph processing algorithms implementation methods for Intel KNL and Nvidia GPU // Communications in Computer and Information Science. 2017. T. 793. C. 80-94.

4. Zatsarinny, A.A., Gorshenin, A.K., Kondrashev, V.A., Volovich, K.I., Denisov, S.A.: Toward high performance solutions as services of research digital platform. // Procedia Computer Science. Volume 150. p. 622-627. (2019)

5. Kondrashev, V.A., Volovich, K.I.: Service Management Digital Platform on the example of high performance computing services // Proceedings of the International Scientific Conference. Voronezh, September 3–6, (2018)

6. Kondrashev, V.A. Architecture of the service delivery system for the research services digital platform // Systems and Means of Informatics, 2018, vol. 28, no. 3, pp. 131–140

7. Nikitenko, D.A., Voevodin, Vl.V., Teplov, A.M., Zhumatiy, S.A., Voevodin, V.V., Stefanov, K.S., Shvets, P.A.: Supercomputer application integral characteristics analysis for the whole queued job collection of large-scale hpc systems // Parallel computing technologies (PaVT'2016): proceedings of the international scientific conference. Chelyabinsk. p. 20-30. (2016)

8. Wu W., Zhang H., Li Zh., Mao Ya. Creating a cloud-based life science gateway // e-Science and the Archaeological Frontier: 7th Conference (International) on eScience. |Piscataway, NJ, USA: IEEE, 2011. P. 55-61.

9. Ding F., Mey D., Wienke S., Zhang R, Li L. A study on today's cloud environments for HPC applications // 3rd Conference (International) on Cloud Computing and Services Science Proceedings. | Berlin, Germany: Springer, 2014. P. 114-127.

10. Zatsarinny, A.A., K. I. Volovich, and V.A. Kondrashev. 2017. Methodological problems of management of Russian Federation scientific and educational organizations science services. // Radiolocation, Navigation, Communication: Collection of Materials of the 23rd Scientific and Technical Conference (International). Voronezh: Velborn. 1:7-14.

11. Kondrashev V.A. Architecture of the Digital Services Platform for Scientific Research Services Presentation - Moscow: TORUS PRESS, Informatics Systems and Means, Vol. 28, No. 3. p. 131-140. (2018)

12. Zatsarinny, A.A., A.K. Gorshenin, V.A. Kondrashev, and K. I. Volovich. 2017. Scientific services management system as a basic service of a digital platform for scientific research. // 3rd Scientific and Practical Conference Proceedings. Moscow: IPU RAS, NITS “Zhukovsky Institute.” p. 53-64.

13. Volovich K.I. Organization of calculations in a hybrid high-performance computing cluster for parallel execution of heterogeneous tasks Presentation - Moscow: TORUS PRESS, Informatics Systems and Means, Vol. 28, No. 4. p. 98-109. (2018)