engineering proceedings

MDPI

Proceeding Paper

Dynamic Job Queue Management for Interactive and Batch Computation on HPC System †

Sergey Denisov *, Vadim Kondrashev and Alexander Zatsarinny

Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44, Build.2, Vavilova Str., Moscow 119333, Russia; vkondrashev@frccsc.ru (V.K.); azatsarinny@ipiran.ru (A.Z.)

* Correspondence: sdenisov@frccsc.ru; Tel.: +7-910-434-08-93

† Presented at the 15th International Conference "Intelligent Systems" (INTELS'22), Moscow, Russia, 14–16 December 2022.

Abstract: The article discusses HPC system computing resources distribution management during execution of interactive and batch jobs. A set of queues for interactive and batch jobs is proposed, and an algorithm for the dynamic resources allocation between the proposed job queues is described.

Keywords: computing resources; hybrid high-performance computing system; hybrid architecture; graphic coprocessor; individual execution environment; interactive mode; computational task management system; job queue

1. Introduction

Modern high-performance systems provide solutions to a wide range of tasks and interdisciplinary research, including machine learning, artificial intelligence, big data, digital twins, behavior modeling, and more. Such an extensive set of tasks determines the specifics of the organization of the computational process on such systems. The computing environment is required to be flexible and adaptable to the requirements from both the application and user task sides.

In [1], in addition to the classical batch mode of operation of the computing environment, an interactive mode of operation in an individual execution environment in the computing environment of a high-performance system was substantiated.

The interactive mode of operation is implemented in many software products (graphics editors, spreadsheets, tutorials, computer games, etc.). In other cases, it is used in systems operating in the mode of a computational experiment, including the development and training of artificial intelligence models, when the ongoing processes must remain under the supervision and control of an expert. The interactive mode provides developers, experts, and users with the opportunity to test the functionality of models online (especially during the initial stages of their development), test various hypotheses, and analyze and visualize the obtained data.

The current level of development of microelectronics makes it possible to introduce high-performance computing systems in large corporations as well as in small scientific and scientific-production teams, with the only difference being the "size" of the implemented systems. As a rule, high-performance systems of scientific and scientific-production teams are significantly inferior to the computing complexes of corporations in terms of the amount of computing resources, which means that the issue of efficient allocation of computing resources while taking into account interactive and batch approaches to computing is an urgent task.

Keeping in mind that there are many different families of tools for developing applications for modeling, machine learning, and artificial intelligence, the simultaneous presence of several tools in a single underlying computing environment is inappropriate, as it creates conditions for the occurrence of compatibility conflicts and limits the ability to

Citation: Denisov, S.; Kondrashev, V.; Zatsarinny, A. Dynamic Job Queue Management for Interactive and Batch Computation on HPC System. *Eng. Proc.* **2023**, *33*, 55. https://doi.org/10.3390/engproc2023033055

Academic Editors: Askhat Diveev, Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 21 July 2023

customize these tools for use by different groups of users. The limitations on the settings are primarily due to the fact that their implementation requires administrator rights to install additional system software and software libraries, taking into account the compatibility requirements of the software components of the entire complex. The performance of such operations by the administrator, which require a change in the computing environment, creates discreteness in the work of users.

In [1], the authors proposed an approach to the placement of computational jobs belonging to the class of batch or interactive jobs while taking into account the organization of an individual execution environment. In this paper, we consider the distribution of computing resources and the allowable time intervals for executing computational jobs for different job queues. Using the example of the "High Performance Computing and Big Data" (CKP "Informatics") Shared Research Facilities of FRC CSC RAS (Moscow) [2], an approach is proposed to dynamically allocate computing resources between job queues serving flows of interactive and batch jobs.

2. Organization of the Computing Process in a High-Performance System

The functions of managing the computing environment of a high-performance system are assigned to the computing task management system (which may be referred to as a "task manager", "workload management system", "job scheduler", etcetera).

As a rule, a computing task management system operates within the "client–server" paradigm and manages the process of executing computational jobs based on the configured service policies in the form of a set of rules (algorithms) for distributing computational resources between computational jobs and managing the progress of their competitive execution.

The process of managing the execution of computational jobs is reduced to the implementation of three key functions of the computing task management system:

1. Providing users with exclusive and/or non-exclusive access to computing resources for a certain period of time;
2. Providing an environment for launching, executing, and monitoring the operation of a parallel task, for example, MPI (Message Passing Interface);
3. Performing resource contention arbitration by managing a queue of pending jobs.

As a rule, computing task management systems have a built-in set of algorithms that implement the following service policies:

- A policy for processing the input stream of computational jobs;
- A policy for managing the behavior of the computational task being performed.

The classical approach to the organization of the computational process in a high-performance system is the batch execution of computational jobs, which ensures the most efficient use of computational resources [3].

However, in a number of cases the established practice of using computing tools dictates the need to implement an interactive mode of operation for users in their individual execution environment together with a batch service mode. An individual execution environment is created through the use of virtual containerization technology based on the Docker system [4]. From the point of view of the computing task management system, there is no difference between the management of a task containing an applied task and that of a container with an applied task. This feature is implemented in the CKP "Informatics", where an individual runtime environment is integrated into the computing environment of a high-performance system, allowing different groups of users to solve their tasks in parallel in different computing environments. In [5], approaches to the formation and maintenance of a functional individual execution environment in the interests of a particular user are substantiated.

Currently, the following types of individual execution environments are implemented in the CKP "Informatics":

- The GROMACS environment [6] for modeling physical and chemical processes in molecular dynamics, with the ability to run several copies and organize MPI interaction between them;
- The Ansys environment [7] for automated engineering calculations;
- A development environment for machine learning and artificial intelligence models based on the Jupyter Notebook solution [8].

3. Queues of the Computing Task Management System

In this paper, we propose the creation of the following queues for simultaneous operation of batch and interactive modes.

The *interactive queue* is designed to perform development and debugging functions, and permits almost-online execution of computational jobs. Service policies for this queue represent a limitation on the execution time of a given job.

The *main queue* is intended for execution of computational jobs in batch mode, and allows computational jobs to be placed for execution, for which the user determines the operation time by different values.

An *additional queue* is intended for executing computational jobs in batch mode, and allows computational jobs to be placed for execution when it is not possible to estimate the time of their execution. This queue has the lowest priority; a limited pool of resources is provided for the queue, and jobs can be suspended.

The listed queues are defined by the following parameters:

- *Interactive queue*: priority is high, availability of resources is based on the "best possible" principle, and the maximum task execution time is very limited;
- *Main queue*: priority is medium, availability of resources is within a limited pool, and the maximum task execution time is limited;
- *Additional queue*: priority is low, resource availability is within a limited pool, and the maximum task execution time is unlimited.

Next, we consider the approach of dynamic distribution of computing resources between job queues serving the flows of interactive and batch computational jobs. The computing task management system (for example, the system based on the Slurm software product [9]) provides users with flexible options for solving their scientific and practical problems by launching computational jobs either

- On one server from the computing system; or
- On a group of servers belonging to the same computing system.

4. Dynamic Distribution of Computing Resources

The parameters that characterize the queue, as indicated in the previous section, are as follows:

- Queue priority;
- Resource pool, i.e., the maximum amount of computing resources available to the queue for distribution between computational jobs;
- Resource portions, i.e., the maximum amount of computing resources allocated to the calculation task in the queue;
- Operation mode;
- Reservation time, i.e., the maximum time for the execution of the calculation task in the queue.

Below, we explain several of these parameters.

4.1. Resource Pool

Usually, in heterogeneous computing systems, all computing resources, which are understood as physical computing servers (nodes), are divided into several classes depending on the type of equipment. For the CKP "Informatics" these are of two classes: hybrid

computing nodes (computing servers equipped with graphics co-processor GPUs) and computing nodes (computing servers without graphics co-processors).

4.2. Resource Portions

A resource portion R is described by the set

$$R = \left\{ nvCPU, nGPU, nRAM \right\}, \quad (1)$$

where

- $nvCPU$—the requested number of virtual processors (vCPU);
- $nGPU$—the requested number of graphics coprocessors (GPU);
- $nRAM$—the requested amount of RAM.

The $nvCPU$ is the sum of the products

$$nvCPU = \sum_{i=1}^{N} (nCPU * nCore * nThread), \quad (2)$$

where

- N—the number of computing nodes;
- $nCPU$—the number of CPU of the computing node;
- $nCore$—the number of CPU cores of the computing node;
- $nThread$—the number of threads per CPU core.

Typically, resource portions are available to users in discrete sets or user-defined sets. For example, the following types of resource portions are available for users of the CKP "Informatics":

(a) numa node—a resource portion containing n CPUs and n GPUs belonging to the same numa domain [10].
The configuration of numa nodes depends directly on the physical architecture of the hybrid computing nodes.
Possible configuration options are
   - numa.small, containing one GPU and 12 vCPU;
   - numa.medium, containing two GPUs and 24 vCPUs;
   - numa.big, containing four GPUs and 48 vCPUs;
   - numa.large, containing eight GPUs and 96 vCPUs.

(b) user-defined, in which a resource portion is described by the set R and is not more than the size of the "resource pool".

4.3. Operation Mode

The "open" mode means that the computational jobs in the queue have a claim on the allocation of resources, and can begin to be executed. The "closed" mode means that the computational jobs have accumulated in the queue and are waiting for the queue opening time before starting their execution.

The specified parameters are dynamically assigned to the previously designated queues depending on the current mode of operation of the computing task management system. There are two modes of operation for the computing task management system of the CKP "Informatics": "day mode" (daytime operation from 8.31 to 21.30) and "night mode" (night-time operation from 21.31 to 8.30).

It can be seen that there is a time interval of one minute between the "day–night" and "night–day" intervals. During this period, switching between operating modes occurs. The queue parameters are reassigned, while calculation jobs are not executed and remain pending.

The proposed approach for the dynamic distribution of computing resources between job queues serving the flows of interactive and batch jobs is performed twice a day when

changing operating modes to day and night modes. The change of modes is carried out automatically by reassigning the parameters of the queues in accordance with the following conditions.

In "day mode", the following parameters are assigned to the *interactive queue*:

1. Priority—high;
2. Resource pool—all hybrid computing nodes;
3. Resource portion—one numa node from the list of available options;
4. Operation mode—open;
5. Reservation time—user-defined, with the following conditions:
    - No more than one hour;
    - If the reservation is made less than an hour before the change of operation mode, then until the change of operation mode.

In "day mode", the following parameters are assigned to the *common queue*:

1. Priority—medium;
2. Resource pool:
    - All computing nodes for computational jobs with reservation times that do not exceed the time limit for the end of "day mode";
    - Three computing nodes for computational jobs with reservation times that exceeds the time limit for the end of "day mode";
3. Resource portion—user-defined;
4. Operation mode—open;
5. Reservation time—user-defined.

In "day mode", only the "operation mode—closed" parameter is assigned to the *additional queue*. The priority, resource pool, resource portion, and reservation time parameters are not assigned due to the mode of operation.

Thus, in the "day mode" the users have access to both interactive and batch computing modes in a high-performance system, while interactive jobs have priority advantage over batch jobs.

In "night mode", only the "operation mode—closed" parameter is assigned to the *interactive queue*. Thepriority, resource pool, resource portion, and reservation time parameters are not assigned due to the mode of operation.

In "night mode", the following parameters are assigned to the *common queue*:

1. Priority—high;
2. Resource pool:
    - All nodes for computational jobs with reservation times that do not exceed the time limit for the end of "night mode";
    - One hybrid computing node for computational jobs with reservation times that exceed the time limit for the end of "night mode";
    - Three computing nodes for computational jobs with reservation times that exceed the time limit for the end of "night mode";
3. Resource portion—user-defined;
4. Operation mode—open;
5. Reservation time—user-defined.

In "night mode", the following parameters are assigned to the *additional queue*:

1. Priority—medium;
2. Resource pool:
    - One hybrid computing node;
    - One computing node;
3. Resource portion—user-defined;
4. Operation mode—open.

Thus, in the "night mode" only the batch mode of calculations is available to users. At the same time, computational jobs for which the resource reservation time is determined have a priority advantage over computational jobs for which it is not possible to estimate the time of execution. When placing a computational job in a computing environment, the user specifies values for the resource portion and reservation time while taking into account the limitation on the maximum execution time of a computational task in the queue.

After that, the computational task management system operates according to the basic algorithm embedded in it:

1. the system places the computational job in the queue in accordance with the policy for processing the input stream of computational job.
   In "day mode":
   If all resource portions are reserved at the current moment, the computing task management system performs the following actions when queuing a computational job:
   - In the interactive queue, the system offers the user either the nearest available time for reservation and available options for numa nodes, or offers the user the option to select a convenient time and available options for numa nodes, including on subsequent calendar days;
   - In the general queue, the system leaves the computational job in the queue in "resources pending" status.

   If the user has specified an invalid reservation time, the computational job is not queued.
   For an interactive queue, it is possible to prolongation the amount of time required by the user 10 min before the expiration of the reservation time; it may be the case that there are no available resource portions, in which case the system offers the user the same options as during the initial reservation of resources.
   When changing the current operating mode to "night mode",
   - For computational jobs pending in queues, the reservation is preserved, while jobs from the general queue are allowed to execute at night;
   - Computational jobs running from the interactive queue are forcibly terminated;
   - Computational jobs running from the general queue continue their execution.

   In "night mode":
   If all resource portions are reserved at the current moment, the computing task management system performs the following actions when queuing a computational job:
   - In the general and additional queues, the system leaves the computational jobs in the queue in "resources pending" status.

   If the user has specified an invalid reservation time, the computational job is not queued.
   When changing the current operating mode to "day mode":
   - For computational jobs pending in queues, the reservation is preserved, while jobs from the general queue are allowed to be executed during the daytime;
   - Computational jobs running from the general and additional queues continue their execution.
2. The system analyzes the parameters of jobs in the queue in a cycle. The system determines which jobs that requires a control action and performs this action in accordance with the policy for managing the behavior of the executed computational jobs;
3. The system removes completed computational jobs from the queue. The job can complete its execution on its own or under the influence of the system in accordance with the policy for managing the behavior of the executed computational jobs;
4. Upon completion of a job, the system checks the correctness of the release of computing resources and returns the resources to the resource pool.

5. Conclusions

This article proposes an approach to the dynamic distribution of computing resources between job queues serving flows of interactive and batch computational jobs. This approach improves efficiency when loading the resources of a high-performance computing system under conditions of joint application of interactive and batch approaches to computing.

The presented mode of operation of the computing task management system with dynamic distribution of computing resources is currently being tested in the computing environment of the CKP "Informatics". The dynamic distribution of computing resources of a high-performance system successfully provides for heterogeneous tasks of interdisciplinary research (including machine learning, artificial intelligence, big data analysis, creation of digital twins, etc.) with the necessary flexibility and adaptability for the requirements of applied applications, including interactive modes.

Taking into account that an interactive mode is used during the initial stages of model creation, while the classic batch mode approach is well suited for operation (i.e., training models in deep learning tasks), the approach to organizing calculations proposed in this article can greatly expand access to computing resources. As a result, the solution proposed in this paper is able to effectively solve the problem of allocating computing resources for interactive and batch computing modes in high-performance computing systems belonging to small scientific and scientific-production teams.

Author Contributions: Conceptualization, S.D., V.K. and A.Z.; methodology, S.D., V.K. and A.Z.; validation, S.D.; formal analysis, V.K.; investigation, S.D., V.K. and A.Z.; writing—original draft preparation, S.D., V.K.; writing—review and editing, S.D., V.K.; supervision, A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The research was carried out using the infrastructure of the Shared Research Facilities "High Performance Computing and Big Data" (CKP "Informatics") of FRC CSC RAS (Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Volovich, K.; Kondrashev, V.; Posypkin, M.; Denisov S. Some Approaches to Managing Computing Resources of a Hybrid High-Performance Cluster in a Cloud Environment. In Proceedings of the VI International Conference Information Technologies and High-Performance Computing (ITHPC-2021), Khabarovsk, Russia, 14–16 September 2021; Volume 2930, pp. 47–53.
2. CKP "Informatics". Available online: http://www.frccsc.ru/ckp (accessed on 19 July 2022).
3. Volovich, K.I.; Denisov, S.A.; Shabanov, A.P.; Malkovsky S.I. Aspects of the assessment of the quality of loading hybrid high-performance computing cluster. In Proceedings of the V International Conference Information Technologies and High-Performance Computing (ITHPC-2019), Khabarovsk, Russia, 16–19 September 2019; Volume 2426, pp. 7–11.
4. Docker Docs. Available online: https://docs.docker.com/get-started/overview/ (accessed on 19 July 2022).
5. Volovich, K.I.; Denisov, S.A.; Malkovsky, S.I. Formation of an Individual Modeling Environment in a Hybrid High-Performance Computing System, *Russ. Microelectron.* **2020**, *49*, 580–583. [CrossRef]
6. Gromacs. Available online: https://www.gromacs.org/ (accessed on 19 July 2022).
7. Ansys. Available online: https://www.ansys.com/ (accessed on 19 July 2022).
8. Jupyter Notebook. Available online: https://jupyter.org/ (accessed on 19 July 2022).
9. Slurm Documentation. Available online: https://slurm.schedmd.com/documentation.html (accessed on 19 July 2022).
10. The Linux Kernel. Numa. Available online: https://www.kernel.org/doc/html/v4.18/vm/numa.html (accessed on 19 July 2022).