Aspects of the Assessment of the Quality of Loading Hybrid High-Performance Computing Cluster

Konstantin I. Volovich[1], Sergey A. Denisov[1],
Alexander P. Shabanov[1], Sergey I. Malkovsky[2]

[1]Federal research center 'Computer Science and Control' of the Russian Academy of Sciences, Moscow, Russia,
KVolovich@frccsc.ru, SDenisov@frccsc.ru, APShabanov@mail.ru
[2]Computing Center of Far Eastern Branch Russian Academy of Sciences, Khabarovsk, Russia,
sergey.malkovsky@gmail.com

Abstract

The article proposes a method for estimating workload, based on the calculation of peak performance, which is required to perform computational tasks. The system of dynamic priorities of computing tasks is considered, based on the resource efficiency indicators of the high-performance cluster. Keywords: high-performance computing cluster; hybrid architecture; graphics accelerator; performance efficiency; profiling; dynamic priority.

1 Introduction

The most important issue in the operation of a high-performance computing cluster is to provide the complete utilization of its resources. This is necessary for solving scientific problems and ensuring the return of investments (ROI).

We can distinguish two main areas in this problem [1, 2]:

- ensure execution of the maximum possible number of applications for a certain period of time;
- the most efficient use of cluster resources by user applications.

An important issue of operations is to determine the grade of loading of the cluster, because it allows to plan the provision of resources, to assess the necessity for modernization, to determine the quality of the services.

As a rule, the workload is defined as the ratio of the metric (parameter) of the workload to the maximum possible value of this parameter. The metric is determined by measurement or calculation.

The article proposes a new method for calculating the value of the workload using the peak performance of the cluster.

A high workload of the HPC cluster does not mean efficient use of its resources. It is possible that the resources requested by the application are not used and are idle. In this case, the workload factor of the cluster can be high, but the quality of the tasks is low.

To provide an advantage to applications that efficiently use the resources of the cluster, the article discusses a system of dynamic priorities. The system is based on determining the coefficient of profiling and using it to change the priorities of applications.

2 Technique for Estimating the Workload of the Hybrid HPC Cluster

For traditional supercomputers, the workload parameter may be the number of core-hours that were provided to the application for performing calculations [3]. The ratio of allocated core-hours to the maximum possible is an indicator of the cluster workload. These parameters are calculated for a certain period of time and are an integral indicator of the workload over a given period.

For hybrid architectures, this approach is less significant, since there are various types of cores in the hybrid HPC cluster, and applications reserve the calculator's resources not by the cores, but by entire graphics accelerators.

The proposed method allows to take into account this feature. The workload estimate is determined by comparing the requested by the applications and the maximum possible number of floating-point operations per period of time.

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

Note that there is a difference between the theoretically possible performance of cluster (peak performance) and practically achievable results. Results are determined by different tests and vary greatly depending on the type of tasks and configuration of the cluster [4].

To estimate the workload of a hybrid high-performance computing cluster, we use peak performance. It is defined as the sum of the peak productivities of its components — nodes (1).

$$P_{peak} = \sum_{i=1}^{K} P_{host\ i}, \quad (1)$$

where P_{peak} is peak performance of the computing cluster,

$P_{host\ i}$ – peak performance (P_{host}) of the i-th node of the computing cluster

Note that the summation does not take into account the performance losses that occur when the nodes interact over the computer network connecting them (interconnect) [4].

The peak performance of the P_{host} node is defined as the sum of the performance of the central processors of the node (P_{cpu}) and its graphic accelerators - P_{gpu}. It is assumed that they are fully loaded with floating point operations, do not perform any other operations, and there are no data transfer losses between the central processors and graphics accelerators (2).

$$P_{host} = N_{cpu}P_{cpu} + N_{gpu}P_{gpu}, \quad (2)$$

where N_{cpu} is number of CPUs in the compute node,

N_{gpu} – number of graphics accelerators in the compute node,

P_{cpu} – peak CPU performance,

P_{gpu} – graphics accelerator peak performance.

To calculate the peak performance of the CPU (3), we assume that the operations are performed by the cores in parallel, each core can process a group of threads, and the flow allows several operations to be performed in parallel if there are several operational blocks for this. Such a core-streaming architecture is characteristic of modern classical processors of various manufacturers.

$$P_{cpu} = n_{core}n_{stream}n_{unit}F_{cpu}, \quad (3)$$

where n_{core} is number of CPU cores,

n_{stream} – the number of threads processed by the CPU core,

n_{unit} – the number of operating units per flow corresponds to the number of operations performed in one flow per cycle,

F_{cpu} – CPU frequency.

To assess the performance of graphics accelerators, we use the the modern accelerator architecture of the NVidia company. Consider the family of accelerators Tesla Volta, as the most popular. Accelerators contain cuda- and tensor-cores, which allow performing parallel operations on floating-point numbers and matrices. The performance of a graphics accelerator is defined as the sum of the productivity of all cores without taking into account performance losses on scheduling and interaction (4) [5].

$$\mathrm{P_{gpu}} = \mathrm{P_{cuda}} + \mathrm{P_{tensor}}, \quad (4)$$

where P_{cuda} is total performance of the graphics accelerator cuda-cores,

P_{tensor} – total performance of tensor-cores of the graphics accelerator.

Let us determine the performance value of the cuda-cores of the graphics accelerator using formula (5), assuming that the floating-point operation is performed in one clock cycle.

Tensor-cores perform a multiplication of square matrices in one clock cycle. When calculating the number of operations performed in this case, we will take into account that the calculation of each element of the resulting matrix requires the execution of multiplication operations equal to the order of the matrix, as well as addition operations one less. Thus, the total performance of tensor-kernels is calculated as (6).

Note that the accuracy of performing floating point operations for different cores may differ. So, in the graphics accelerator NVidia Tesla V 100 cuda-cores work with double precision numbers, and tensor-cores with single precision numbers. In this method of performance evaluation, this feature is not taken into account.

The total performance of cuda- and tensor-cores are determined by formulas (5) and (6).

$$\mathrm{P_{cuda}} = \mathrm{n_{cuda}F_{gpu}} \quad (5)$$

$$\mathrm{P_{tensor}} = \mathrm{n_{tensor}r^2(2r-1)F_{gpu}} \quad (6)$$

where n_{cuda} is number of graphics accelerator cuda-cores,

n_{tensor} – number of tensor-cores of the graphics accelerator,

r – square matrix order,

F_{gpu} – graphics accelerator frequency.

Thus, the peak performance of the graphics accelerator is calculated by the formula (7).

$$P_{gpu} = (n_{cuda} + n_{tensor}r^2(2r-1))F_{gpu} \quad (7)$$

The peak performance of the computing node of a hybrid high-performance computing cluster is calculated by the formula (8).

$$P_{host} = N_{cpu}n_{core}n_{stream}n_{unit}F_{cpu} + (n_{cuda} + n_{tensor}r^2(2r-1))F_{gpu} \quad (8)$$

The total peak performance of a hybrid high-performance computing cluster is calculated as (1).

As shown above, the performance of the HPC cluster is calculated as the sum of the performances of its components and is expressed by the number of floating-point operations performed per second.

The resource of the hybrid high-performance computing cluster in the time interval will be the peak number of floating point operations available to users during this interval.

The total number of operations of the hybrid high-performance computing cluster Op(T) on the time interval T is defined as:

$$Op\ (T) = P_{peak}T, \tag{9}$$

where T is time interval.

The peak estimate differs from the actual, which is determined on the basis of various tests. However, as noted above, in this method we will use the peak values.

To estimate the requirements of applications to the resources of a hybrid high-performance computing cluster, we calculate the number of operations required for the execution of the application (10).

For each application, a number of CPU cores, graphics accelerators, and runtime are reserved. We take into account that the resources of graphic accelerators are reserved entirely, and the resources of central processors - by cores. Therefore, the total number of hybrid high-performance computing cluster operations performed by the task - $Op_{app}(t)$ - for a given time *t* is determined by the number of cores of the central processors (R_{core}) and graphics accelerators (R_{gpu}) reserved by the application.

$$Op_{app}\ (t) = (\frac{(R_{core}P_{cpu})}{n} + R_{gpu}P_{gpu})t, \tag{10}$$

where R_{core} is the number of cores reserved by the application,

R_{gpu} – the number of graphics accelerators reserved by the application,

n – total number of cores in CPU.

After calculations for all applications *i=1…N*, the execution of which accounted for the period T, we obtain the total number of operations required for the execution of applications on the period T (11):

$$Op_{app}(T) = \sum_{i=1}^{N}(\frac{(R_{cpu}P_{cpu})}{n} + R_{gpu\,i}P_{gpu})t_i \qquad \text{for} \qquad t_i \in T \tag{11}$$

Figure 1 shows a diagram of tasks performed on period T. Note that it is possible that only part of the execution time of the application falls on this period. In this case, when estimating the resources used, only the time interval t_i belonging to T is taken into account.

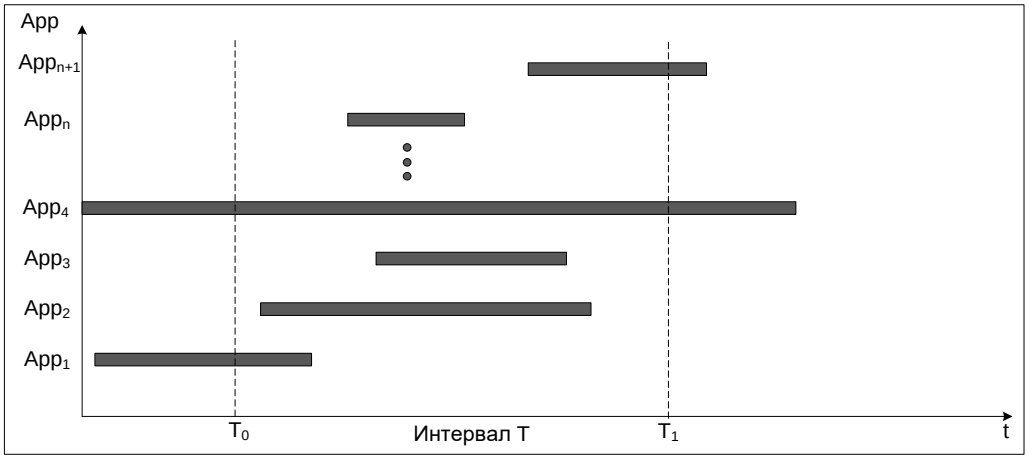

Figure 1: Execution of tasks on the time interval T

Based on the peak value of the available floating-point operations on the interval T, the workload quality ratio of the hybrid high-performance computing cluster is calculated (12).

$$Q(T) = \frac{Op_{app}(T)}{Op\ (T)} * 100\ \%. \tag{12}$$

The *Q(T)* indicator can be used to set and evaluate the performance indicator of high-performance hybrid computing systems, plan the modernization of the cluster and draw up plans for the calculations of users of the hybrid cluster.

Note that the proposed assessment of the quality of loading of processor resources *Q(T)* is purely declarative and does not take into account the degree of use of allocated resources, the efficiency of algorithms and the quality of program code.

3 System of Dynamic Priorities

To provide an advantage to applications that use the resources of the cluster qualitatively, a system of dynamic priorities should be used. The system is based on the cluster utilization indicator.

To obtain an indicator of the use of computing resources, it is proposed to introduce the task profiling coefficient as an integral indicator of the quality of resource use (13).

$$Op_{app-prof}\ (t) = K_{prof}\ Op_{app}\ (t) \tag{13}$$

where K_{prof} is profiling ratio.

Using such an assessment allows you to avoid a situation when the application does not use or irrationally uses the resources requested from the computing cluster. The coefficient of profiling is obtained by running a custom application under the control of a special debugging tool - a profiler that allows you to determine the degree of resource utilization, the execution time of individual code sections, bottlenecks and problems of memory usage. As part of development packages, there are profilers for both code executed on central processors and graphic accelerators.

Information on program profiling should be available both to the developer of a scientific application and to the division operating the computing cluster. This is necessary to take measures to improve the efficiency of the program code and increase the efficiency of the functioning of the hybrid HPC cluster as a whole.

Obviously, applications with a high profiling coefficient improve the quality indicator of a high-performance cluster workload. Therefore, a competitive advantage should be given to such applications. This encourages users to improve the calculation algorithms and taking into account the capabilities of the computing cluster. A classic way of encouraging tasks with a high profiling rate is to introduce a system of dynamic priorities based on the profiling coefficient.

The introduction of dynamic priorities allows within certain limits to change the priority of an application depending on its quality. This service policy is especially useful in conditions of heavy workload of the computing cluster. It allows to improve the quality of resource use and reduce the workload, as well as provide an advantage in the implementation of the applications that make the most use of the cluster's resources.

The decision to change the priority should be made on the basis of a comparison of the measured profiling coefficient with the recommended one, which is determined by expert. It is possible to set several threshold values of the profiling coefficient, for each of which there is a different priority rule. For example, for two quality thresholds (profiling coefficients K_1, K_2) that divide a multitude of applications into three subsets of quality "low", "medium", "high", the dynamic priority can be calculated based on a piecewise linear function (14).

$$Pr_{dyn} = \begin{cases} Pr_{base}\ (C_0 K_{prof} - C_0 K_1 + 1), & K_{prof} < K_0 \\ Pr_{base}\ (C_1 K_{prof} - C_1 K_1 + 1), & K_1 > K_{prof} \geq K_0 \\ Pr_{base}\ (C_2 K_{prof} - C_2 K_2 + C_1 K_2 - C_1 K_1 + 1), & K_{prof} \geq K_1 \end{cases} \tag{14}$$

where Pr_{dyn} is dynamic application priority;

Pr_{base} – basic application priority;

K_{prof} – coefficient derived from application execution profiling;

K_1, K_2 – expert profiling coefficients;

C_0, C_1, C_2 – expert change factors.

Figure 2 shows an example of the dependence of dynamic priority on the values of **K** and **C** with $\mathbf{Pr_{base}}$ =1.

Thus, when obtaining the values of the profiling coefficient below K_1, a linear decrease in priority relative to the base value is made; when K1 is exceeded, a linear increase in priority is obtained. If K_2 is exceeded, the priority growth increases. The recommended profiling coefficients K and coefficients C are determined by an expert method, based on the characteristics of the functioning and loading of the computing cluster.

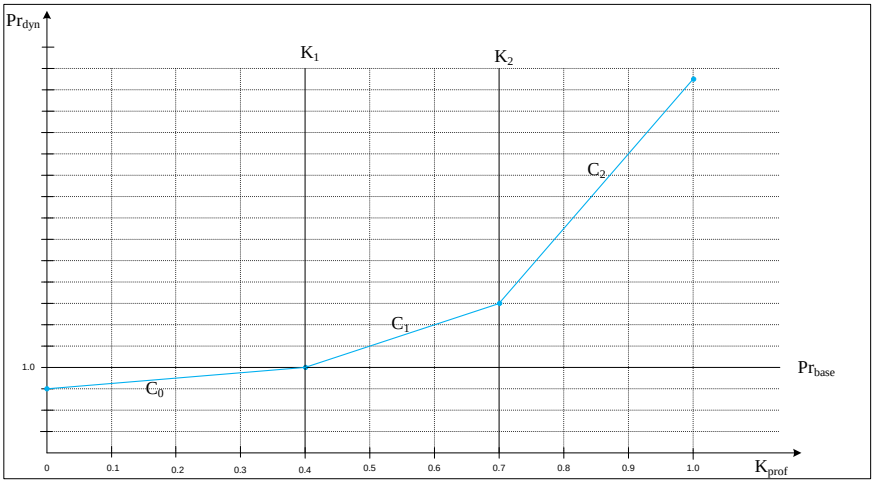

Figure 2: Changing the dynamic priority of the computing task

The choice of the recommended parameters K and C should be made by the owner of the HPC cluster and proceed from the following.

In the early stages of cluster operation, when technologies and algorithms are being debugged, the priority should be changed to the minimum extent. The requirements for grading factors should not be too high. Therefore, the values of **C**, which determine the slope of the straight lines, should be chosen closer to zero, this provides a slight change in priorities.

With increasing workload on the computing cluster and the need for more accurate task management, the values of C can be increased. This leads to a more significant change in priority compared to the baseline with a significant deviation of $\mathbf{K}_{\mathbf{prof}}$ from $\mathbf{K}_{\mathbf{1}}$.

4 Conclusion

The proposed methodology for estimating the workload on the hybrid HPC cluster allows to determine how fully and efficiently the resources of the hybrid cluster are used. On the basis of the results obtained, it is possible to determine indicators of ROI, plan the work of the cluster, and determine the need for modernization.

The system of dynamic priorities will allow to control the quality of resource utilization of hybrid high-performance computing clusters when they perform different types of applications from various fields of science and technology.

Acknowledgements

The research is partially supported by the Russian Foundation for Basic Research (project 18-29-03100).

References

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

2. Abramov, S.M., Lilitko, E.P.: The state and prospects of development of ultra-high-performance computing systems // Information technologies and computing systems Moscow. №2. p. 6-22. (2013)

3. Klinov, MS, Lapshina, S.Yu., Telegin, PN, Shabanov, B.M.: Features of the use of multi-core processors in scientific computing Bulletin of Ufa State Aviation Technical University. V. 16. № 6 (51). p. 25-31. (2012)

4. Abramov, S.M.: True, distorting the truth. how to analyze top500? // Bulletin of the South Ural State University. Series: Computational Mathematics and Computer Science. Chelyabinsk. V. 2, № 3, p. 50-31 (2013)

5. 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. T. 793. p. 80-94. (2017)