

UDC 004.7

**Dmytro Zhyzhko,
Simonenko Valery**

**THE ALGORITHM OF DYNAMIC DISTRIBUTION
OF TASKS FOR A CLUSTER SYSTEM**

**Жижко Дмитро,
Сімоненко Валерій**

**АЛГОРИТМ ДИНАМІЧНОГО РОЗПОДІЛУ ЗАДАЧ
ДЛЯ КЛАСТЕРНОЇ СИСТЕМИ**

The article describes the principle of the algorithm for dynamic loading of cluster nodes and compares its work with existing solutions. Was analyzed it work and efficiency in various types of tasks, the ability to balance and distribute the flow of tasks between the system's kernels. A software model for testing was created. The results of the analysis are presented.

Key words: cluster, distribution of tasks, planning algorithms.

Fig.: 6. Bibl.: 5.

У статті наведено принцип роботи алгоритму динамічного завантаження вузлів кластера та порівняно його роботи відносно існуючих рішень. Було проаналізовано його роботу та ефективність при різних типах задач, здатність балансувати та розподіляти потік завдань між ядрами системи. Створено програмну модель для тестування. Наведено результати аналізу.

Ключові слова: кластер, розподіл задач, алгоритми планування.

Рис.: 6. Бібл.: 5.

Relevance of research topic. In our time, computer technology is in almost all spheres of human activity. This, in turn, forces the specialists in the field of information and computer technologies to create new tools that would be able to perform tasks quickly and qualitatively. Every year the software becomes more flexible and perfect. Such programs consume more hardware resources. Initially, such a task was solved by an increase in clock speed of the processor, but now in most cases it is already impossible to do this because a certain limit is reached, or it is very expensive. An alternative is to use several identical processors that can perform different parts of one task in parallel. This approach makes it possible to significantly reduce the time of the task. Systems that have many computing modules are called clusters or grid systems. They have many advantages over noncommercial computers. The main advantage is that they allow you to implement parallel execution and

multitasking. Modern datacenters have thousands and millions of processors in their structure, which allows them to serve a large number of users. On the other hand, the end user is able to use these resources and not spend money on the purchase of personal machines [1, 2, 3].

Formulation of the problem. The increase in the number of computing units and the load on them on the one hand and the price of components on the other hand compel system software developers to create task schedulers that would allow them to achieve significant efficiency. The most important component of the task manager is the planning algorithm. For a few reasons, the purpose of this article is to develop such a planning algorithm that would enable a sufficient level of efficiency of the cluster system. An existing algorithm based on static scheduling gives average performance values of 70% - 90%, the value of which can be significantly reduced with considerable granularity of the computational task. The proposed algorithm is designed to solve this problem.

Analysis of recent research and publications. Since there has been a tendency in the last years to increase the number of cores in processors, as well as the growing popularity of technology, which provides the basis for scientific research on the topic of multi-threaded computing. A number of algorithms and strategies have been created that allow efficient allocation of resources of the cluster system. For an example, the DRF strategy can be called [4]. It makes it possible to distribute resources among users depending on their quotas. But this algorithm works at a higher level of abstraction, which does not allow it to influence the distribution of tasks between processor cores [5].

Identification of unexplored parts of the general problem. The main disadvantage of existing algorithms is that they have static scheduling using queues for each computing unit. With great granularity of tasks it leads to significant losses of efficiency. Also, the use of additional queues of tasks requires more system resources.

Setting objectives. The purpose of the research is to develop a new planning algorithm that would increase the efficiency of the cluster for different types of tasks.

Presentation of the main material. The algorithm of dynamic distribution of tasks for a cluster system.

For experiments and comparisons, a static planning algorithm will be considered. It is based on the following approach. The scheduler transmits a task that consists of n -th number of subtasks. We also have a computing system that includes k processors or cores. The algorithm will distribute all available subtasks between the kernels equally. Until all subtasks are executed, the system will not start the next task. Schematically, the operation of this algorithm can be seen in Fig. 1.

Its main disadvantage is that scheduling occurs only at the stage of cluster loading. If each of the subtasks has a different execution time, then after a certain period of time, one kernel will complete its work, while others do not. This results in lower efficiency. Also, the use of local queues for each core requires certain resources, and if, with this number of these cores are large, then we will have significant losses of RAM.

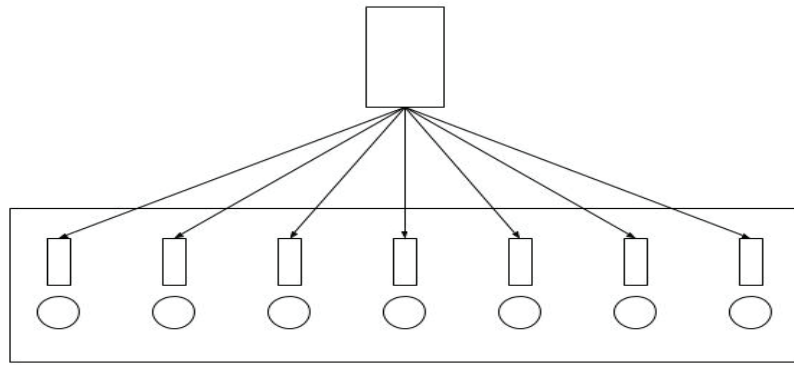


Fig. 1. The structure of the static planning algorithm

The proposed dynamic planning algorithm is devoid of these disadvantages. Planning occurs throughout the time of the task. It does not use local queues of subtasks, but has one common queue. The distribution of tasks occurs dynamically. This allows you to effectively distribute cluster resources. With this approach, the scheduler saves the queue of tasks locally. The kernels have no queues. When the subtask is completed, the scheduler gives the kernel a new task to execute. Schematically it can be depicted as in Fig. 2.

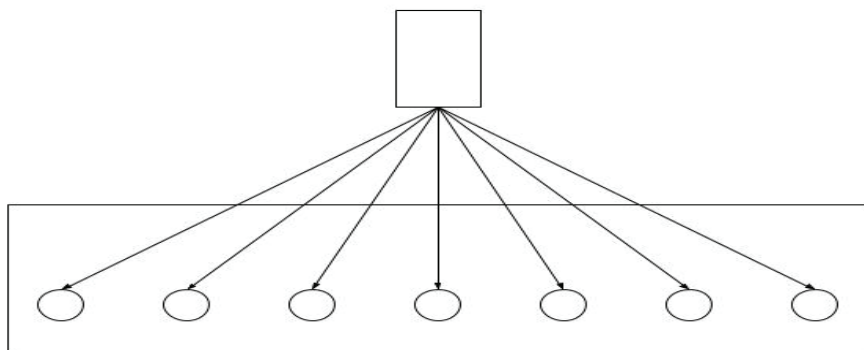


Fig. 2. The structure of the dynamic planning algorithm

To compare the algorithm's work, a number of experiments were carried out, among which measurements of efficiency at different types. In the first experiment, 1000 tasks were generated with execution times from 1 to 10. Efficiency was 0.9 and 0.99 for static and dynamic planning methods, respectively. This is explained by the fact that the static algorithm greatly depends on the granularity of the tasks, and the dynamic is devoid of this disadvantage.

For the second experiment, tasks with the same execution time were generated. The result of the planning for both algorithms was the same, and the efficiency reached 1.

In the third experiment, the dependence of efficiency on the number of processors for each of the algorithms will be analyzed. The results showed that, with increasing number of kernels, the efficiency of each algorithm decreases, but the performance of the second algorithm is better.

In the fourth experiment, the dependence of efficiency from the runtime range on each subtask was analyzed. The test system had 10 cores and 1000 subtasks. The results can be seen in Fig. 3 and fig. 4. Based on them, one can conclude that the efficiency of each of the algorithms does not depend on the range of the time of the task.

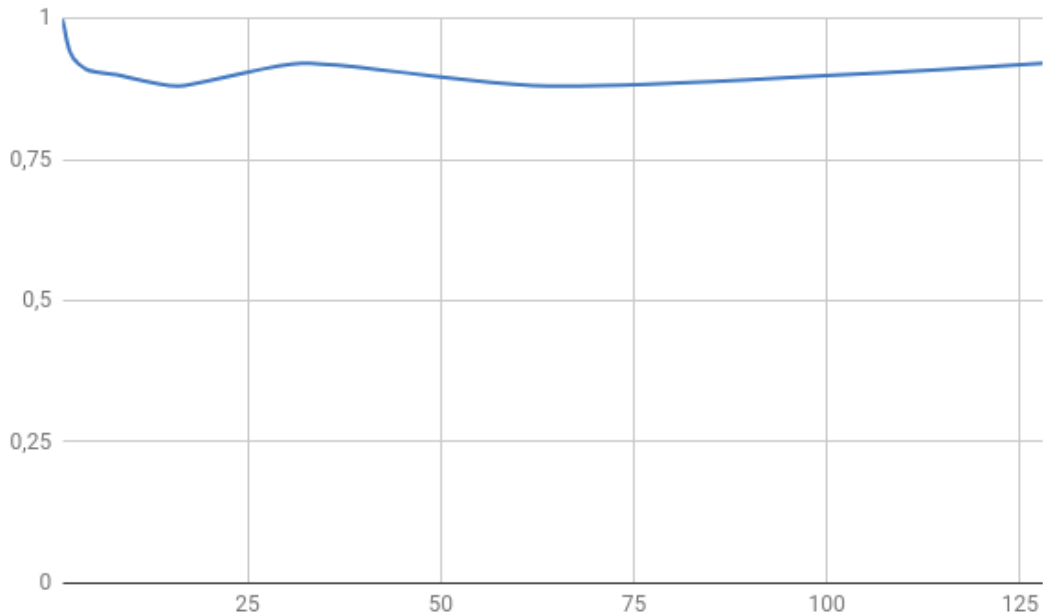


Fig. 3. The results of the fourth experiment for the first algorithm

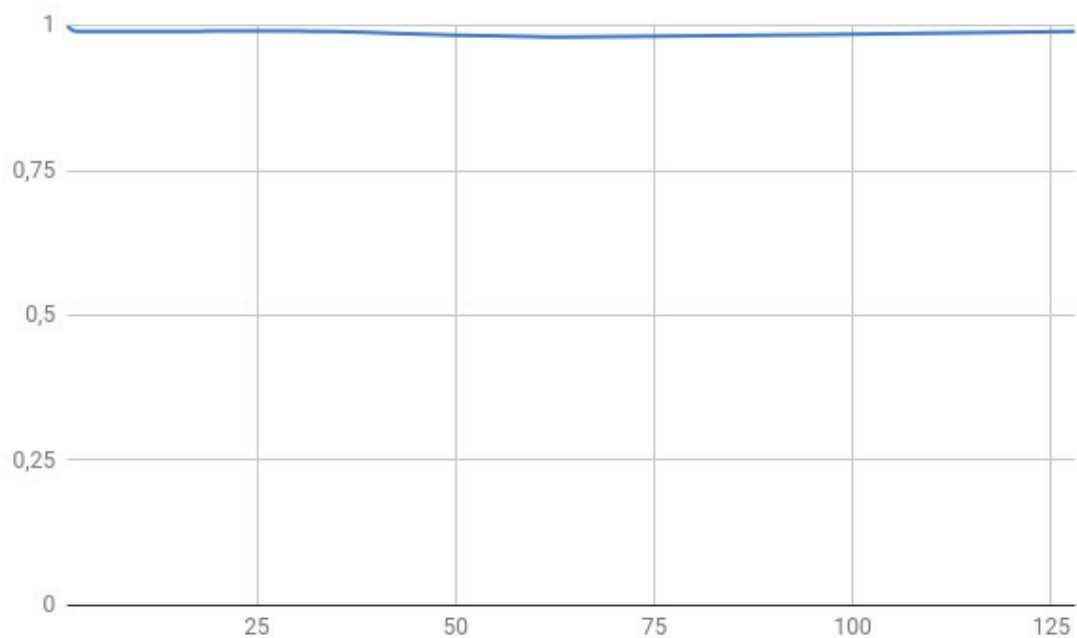


Fig. 4. The results of the fourth experiment for the second algorithm

In the fifth experiment, the dependence of efficiency on the number of tasks in the system was compared. The system has 10 nuclei, the time range for one task is from 1 to 10. The results can be seen in Fig. 5 and rice 6. If the number of tasks is less than the number of kernels in a cluster, then the efficiency is independent of the

algorithm and has a rather low value. When increasing the number of tasks, the efficiency increases for both algorithms.

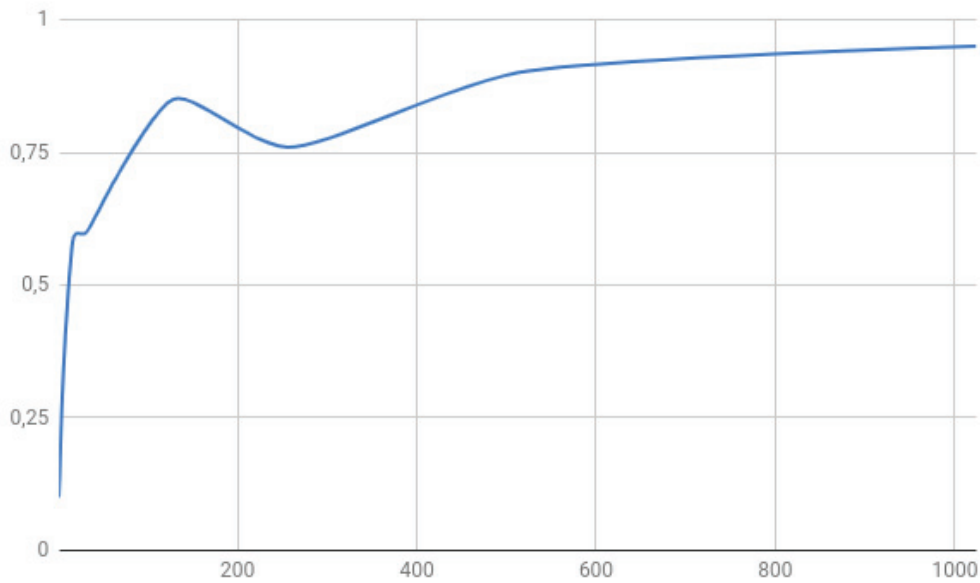


Fig. 5. Results of the fifth experiment for the first algorithm

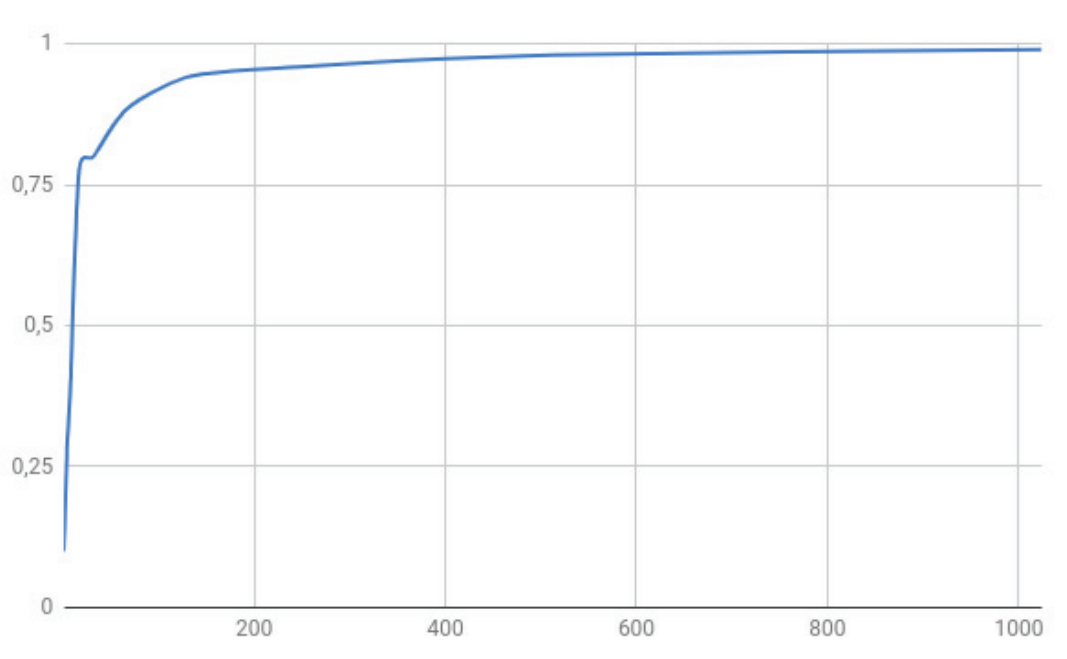


Fig. 6. The results of the fifth experiment for the first algorithm

Conclusions. Nowadays, multithreading calculations, and therefore algorithms for distributing tasks between them, are used more and more often, which suggests that this topic is very relevant. Two algorithms of task planning were analyzed in the work. The proposed dynamic planning algorithm allows you to significantly increase efficiency, so it can be used in modern systems. The average efficiency of the static algorithm was 0.8 - 0.9, and the dynamic - 0.9 - 1, that is, you can get an increase to 20%.

Список використаних джерел

1. Computer cluster [електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Computer_cluster
2. Distributed computing [електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Distributed_computing
3. Introduction to Parallel Computing [електронний ресурс]. — Режим доступу: https://computing.llnl.gov/tutorials/parallel_comp/
4. Maui Cluster Scheduler [електронний ресурс]. — Режим доступу: <http://www.adaptivecomputing.com/products/maui/>
5. Moab HPC Suite – Basic Edition [електронний ресурс]. — Режим доступу: <http://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-basic-edition-download/>

ДОВІДКА ПРО АВТОРІВ

Сімоненко Валерій Павлович — професор, доктор технічних наук, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

Simonenko Valery Pavlovich — Professor, Doctor of Technical Sciences, Department of Computer Engineering, National Technical University of Ukraine «Kiev Polytechnic Institute. Igor Sikorsky»

E-mail: svp@comsys.kpi.ua

Жижко Дмитро Сергійович — студент 6 курсу кафедри обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

Dmytro Zhyzhko Serhiiovych — 6th student of the Department of Computer Engineering, National Technical University of Ukraine «Kiev Polytechnic Institute. Igor Sikorsky»

E-mail: dmytro110101@gmail.com

РОЗШИРЕНА АНОТАЦІЯ

Жижко Дмитро,
Сімоненко Валерій

АЛГОРИТМ ДИНАМІЧНОГО РОЗПОДІЛУ ЗАДАЧ ДЛЯ КЛАСТЕРНОЇ СИСТЕМИ

Актуальність теми дослідження. У наш час комп'ютерні технології є майже у всіх сферах людської діяльності. Це, у свою чергу, змушує фахівців в області інформаційних та комп'ютерних технологій створювати нові інструменти, які могли б швидко і якісно виконувати завдання. Щороку програмне забезпечення стає більш гнучким і досконалим. Такі програми споживають більше апаратних ресурсів. Спочатку таке завдання вирішувалося збільшенням тактової частоти процесора, але тепер у більшості випадків це вже неможливо зробити, тому що досягнута певна межа, або це дуже дорого. Альтернативою є використання декількох ідентичних процесорів, які можуть виконувати різні частини однієї задачі паралельно. Такий підхід дає можливість значно скоротити час виконання завдання. Системи, які мають багато обчислювальних модулів, називаються кластерами або сітковими системами. Вони мають багато переваг перед некомерційними комп'ютерами. Основна перевага полягає в тому, що вони дозволяють здійснювати паралельне виконання і багатозадачність. Сучасні центри обробки даних мають тисячі і мільйони процесорів у своїй структурі, що дозволяє їм обслуговувати велику кількість користувачів. З іншого боку, кінцевий користувач може використовувати ці ресурси і не витратити гроші на придбання особистих машин.

Цілі дослідження. Збільшення кількості обчислювальних одиниць і навантаження на них, з одного боку, і ціни компонентів, з іншого боку, змушують розробників системного програмного забезпечення створювати планувальники завдань, що дозволить їм досягти значної ефективності. Найважливішим компонентом менеджера завдань є алгоритм планування. З кількох причин, метою цієї статті є розробка такого алгоритму планування, який дозволить забезпечити достатній рівень ефективності кластерної системи. Існуючий алгоритм, заснований на статичному плануванні, дає середні показники продуктивності 70% -90%, величина яких може бути значно зменшена при значній деталізації обчислювального завдання. Запропонований алгоритм призначений для вирішення цієї проблеми.

Аналіз актуальних наукових досліджень та питань. Оскільки в останні роки спостерігається тенденція до збільшення кількості ядер у процесорах, а також зростаючої популярності технологій, що є основою для наукових досліджень на тему багатопотокових обчислень. Створено ряд алгоритмів і

стратегій, що дозволяють ефективно розподіляти ресурси кластерної системи. Наприклад, можна назвати стратегію ДРФ. Це дає можливість розподіляти ресурси між користувачами залежно від їх квот. Але цей алгоритм працює на більш високому рівні абстракції, що не дозволяє впливати на розподіл завдань між ядрами процесорів.

Невивчені частини загальних питань визначають. Основним недоліком існуючих алгоритмів є те, що вони мають статичне планування з використанням черг для кожного обчислювального блоку. З великою деталізацією завдань це призводить до значних втрат ефективності. Крім того, використання додаткових черг завдань вимагає більшої кількості системних ресурсів.

Мета дослідження. Метою дослідження є розробка нового алгоритму планування, який би підвищив ефективність роботи кластера для різних типів завдань.

Виклад основних матеріалів. Проведено порівняння алгоритмів статичного та динамічного планування. Розроблено модель для тестування. Проведено моделювання для різних типів завдань. Динамічний алгоритм показав хороші показники.

Висновки. Сьогодні все частіше використовуються багатопоточні розрахунки і, отже, алгоритми розподілу задач між ними, що говорить про те, що ця тема дуже актуальна. У роботі проаналізовано два алгоритми планування завдань. Запропонований алгоритм динамічного планування дозволяє істотно підвищити ефективність, тому його можна використовувати в сучасних системах.

Ключові слова: кластер, розподіл завдань, алгоритми планування.