Research on load balancing of parallel component programs based on quantum particle swarm optimization algorithm

Yunfeng Peng1*, Guowei Gao1, Congming Shi1, Hai Liu1, Jianan Wang1

1 School of software engineering, Anyang Normal University, Anyang, Henan, 455000, China
*Corresponding author’s e-mail: 01698@aynu.edu.cn

Abstract. Parallel component applications are often deployed on heterogeneous clusters. Load balancing is very important for their performance requirement. Existing load balancing methods have high performance cost and poor balance effect. Based on the analysis of structures of parallel component applications, we established the mathematical model of load balancing for parallel components on heterogeneous clusters. We use the quantum particle swarm optimization algorithm to search the optimal solution of the proposed mathematical model and determine the best load balancing scheme. Comparing with the methods based on real-time detection and other swarm intelligence optimization algorithms, our method has lower balance cost, less number of iterations and better performance.

1. Introduction
In the field of scientific computing, the high productivity of parallel computing software has become the most challenging problem. On the one hand, the development of large-scale parallel application is still in the manual stage. With the increasing complexity and scale of problems, the complexity of parallel software is increasing and the development cycle is prolonged. The software developed is difficult to modify, maintain and reuse. On the other hand, from the perspective of the development of parallel architecture, not only the architecture is becoming more diversified, but also the life cycle of new architecture is becoming shorter. This challenges the portability and development efficiency of parallel programs. It is a problem that must be faced to ensure the correctness and high performance of the applications. Although component-based development is the mainstream of serial software development, serial component specifications such as CORBA[1], COM/DCOM[2] and EJB[3] are not applicable to high-performance computing. Based on the traditional serial component technology, researchers have carried out a series of research on the parallel component technology suitable for the field of high-performance computing. Performance is a major challenge for parallel component technology. Parallel applications often have high performance requirements, and the use of component technology often has to bear the performance overhead caused by component connection and management. Parallel component programs are often deployed on heterogeneous clusters, and load balancing has become an important means to improve program performance[4]. The existing load balancing algorithms for parallel component programs often dynamically detect the real-time load on the computing nodes and migrate the components on the high load nodes to the low load nodes. This method will bring high detection and scheduling overhead[5]. For load balancing on large heterogeneous clusters, some researchers also propose to use swarm intelligence optimization algorithm. However, the existing methods based on tabu search algorithm, bat algorithm and ant
colony algorithm[6-8] often have problems such as low search efficiency, stagnation of local optimal solution search and low solution accuracy.

In order to solve the load balancing problem of parallel component programs running on large heterogeneous clusters, we propose a parallel component load balancing method based on quantum particle swarm optimization algorithm[9]. We proved the feasibility and superiority of the proposed method comparing with other existing load balancing methods. Quantum particle swarm optimization algorithm can avoid the detection and scheduling overhead of dynamic detection methods and the defects of other swarm intelligence optimization algorithms in the solution process, improve the utilization of computing node resources in heterogeneous clusters, make the node load more balanced, and obtain a more ideal load scheme.

2. Mathematical model of load balancing for parallel components

The component connection and deployment used in this paper is based on the parallel component architecture and operational framework proposed in reference[10]. As shown in Figure 1, a parallel component program consists of six components that need to be generated and connected on a management node (root node). After the component program is assembled, CCAFFEINE parallel component running framework is responsible for scheduling components to run on different computing nodes. In order to better manage computing resources, CCAFFEINE framework abstracts resources of a heterogeneous cluster platform, extracts resources such as CPU and memory provided by physical computing nodes to form virtual computing nodes, deploy components on virtual nodes, and take charge of mapping virtual nodes to physical nodes.

Suppose we have a set of parallel components \( C = \{ c_1, c_2, \ldots, c_m \}, c_i \) is the i-th component. The set of all the virtual computing nodes is \( R = \{ r_1, r_2, \ldots, r_l \}, r_j \) is the j-th virtual node. The set of all the physical nodes on a heterogeneous cluster is \( D = \{ d_1, d_2, \ldots, d_n \}, d_k \) is the k-th physical node. The interrelated set of the above three parameters can be expressed as:

\[
S = \{ C, R, D, M_{cr}, M_{rd} \}
\]

In equation (1), \( M_{cr} \) represents the mapping between components and virtual nodes. \( M_{rd} \) represents the mapping between virtual nodes and physical nodes.

Suppose the i-th component is deployed on the j-th virtual node. CCAFFEINE framework maps this virtual node to the k-th physical node. The execution time of this parallel component on the k-th physical node is \( E(c_i, M_{cr}, d_k) \). The beginning time of this execution is \( ST(d_k) \). The finish time of this execution is:

\[
F(c_i, M_{cr}, d_k) = ST(d_k) + E(c_i, M_{cr}, d_k)
\]

The time needed for all the components on physical node \( d_k \) is:

\[
SA(d_k) = \sum_{i=1}^{m} b_{ik} F(c_i, M_{cr}, d_k)
\]
In equation (3), $b_{ik} = 1$ means that the $i$-th component is executed on physical node $d_k$. $b_{ik} = 0$ means the $i$-th component is executed on other nodes. As show in equation (4), the load balance of parallel component application is to find a scheme that the total execution time of all the components is minimal.

$$T = \min \sum_{k=1}^{n} SA(d_k)$$

(4)

3. Load balance algorithm

Standard particle swarm optimization algorithm suppose that for particle $i$, the optimal position searched at time $t$ is $P^t_i = (p_{i,1}^t, p_{i,2}^t, ..., p_{i,N}^t)$. $N$ represents the dimension of the search space. The flying speed of particle $i$ at time $t$ is $V^t_i = (v_{i,1}^t, v_{i,2}^t, ..., v_{i,N}^t)$. The position is $X^t_i = (x_{i,1}^t, x_{i,2}^t, ..., x_{i,N}^t)$. At time $t$, the optimal position searched by the whole particle swarm is $P^*_t = (X^t_{g,1}, X^t_{g,2}, ..., X^t_{g,N})$. The update expression of particle velocity and position is:

$$v_{i,j}^{t+1} = w v_{i,j}^t + \tau_1 (p_{i,j}^t - x_{i,j}^t) + \tau_2 (P^*_t - x_{i,j}^t)$$

(5)

$$x_{i,j}^{t+1} = x_{i,j}^t + \beta v_{i,j}^{t+1}$$

(6)

In equation (4) and (5), $w$ is the weight. $\tau_1$, $\tau_2$ is the acceleration factor. $\beta$ is the search interval.

Like other intelligent optimization algorithms, standard particle swarm optimization algorithm has some limitations, such as low search efficiency and low probability of finding the optimal solution. In order to overcome the defects of standard particle swarm optimization algorithm, quantum particle swarm optimization algorithm is proposed. In the classical mechanical theory, the state of particles can be described by position and velocity, but in the quantum mechanical theory, particles have the characteristics of wave-particle, and the position and velocity of particles contain probability characteristics, that is, the state of particles can be described by wave function $\psi(X, t)$, $X = (x, y, z)$. The particle is a position variable in three-dimensional space. The intensity of the wave function at a certain point is directly proportional to the probability of the particle appearing at that point, that is:

$$\iiint |\psi(X, t)|^2 dx dy dz = 1$$

(7)

In quantum mechanics, the dynamic equation of particle motion is:

$$\hbar \frac{\partial}{\partial t} \psi(X, t) = \hat{H} \psi(X, t)$$

(8)

In equation (8), $\hbar$ is Planck constant. $\hat{H}$ is Hamiltonian operator.

By analyzing the particle convergence behavior of quantum particle swarm optimization algorithm, it can be seen that there is a certain form of attraction potential centered on point $P$, and point $P$ is called the attractor of particles. By establishing a one-dimensional potential well in the attractor, the steady-state Schrodinger equation of the particle in the potential well is derived. It is solved that the probability density function of the particle appearing at the position $Y$ of the relative attractor is:

$$\Phi(Y) = \frac{1}{L} e^{-|Y|/L}$$

(9)

In equation (9), $L$ is the distance between the particle and the average optimal solution of the population. We introduced Monte Carlo stochastic simulation method to measure the position of particles. Particles move in a one-dimensional potential well centered on point $P$, and their position can be expressed as:

$$X = X_p + \frac{u}{\pi}$$

(10)

In equation (10), $u$ is a random number. $X_p$ is the current position of $P$.

According to equations (5), (6) and (10), the particle evolution equation with quantum behavior can be obtained, that is:

$$x_{i,j}^{t+1} = x_{i,j}^t + \alpha |P^t_{i,j} - x_{i,j}^t| \ln \frac{1}{u_{i,j}}$$

(11)

In equation (11), $\alpha$ is the search expansion factor. $u_{i,j}$ is a random number at time $t$.

Based on the above algorithm, our load balance method consists of three steps:

- Build a parallel component application system. Determine the number of virtual computing resources that can be used, the number of parallel components, and the processing capacity of computing resources.
Construct the mathematical model of parallel component program load balancing problem according to relationship among parallel components, virtual computing resources and physical resources, taking the shortest completion time of all components as the objective function.

Introduce the quantum particle swarm optimization algorithm to search the solution of the mathematical model of load balancing of parallel component programs, and find the optimal load balancing scheme.

4. Experiments and results
The tested program used in the experiment is based on the medical information system component application Medical Control developed in reference[10]. The application is developed using java + OpenMPI and has 11 components. The experimental platform is a heterogeneous computer cluster, which contains 32 SMP servers (CPU Intel j3060, memory 8 GB), one 8-core multi-core server (CPU Intel i7-9700, memory 16 GB) and a 16 core multi-core server (CPU Intel e5-2682v4, memory 32 GB). Each server is installed with Linux operating system (Fedora 32 server), and the servers are connected through Ethernet. For the parameters of the quantum particle swarm optimization algorithm, we set the number of quantum particles as 30, and the search expansion coefficient as 0.15. The maximum number of iterations is 400. We also tested the method based on dynamic detection proposed in reference[10] and the method based on improved tabu search proposed in reference[11] as a comparison.

Figure 2 shows the performance tests results. To increase the reliability, we took four tests for each version of load balance method. From the results, we can see that our method based on quantum particle swarm optimization algorithm can get better performance than other two methods.

In the above tests, we also recorded the load distribution among different nodes. From Figure 3, we can see the number of iterations to find the optimal solution for our method and the improved tabu search version. Our method can use less number of iterations to get the optimal solution. Our solution is faster.
5. Conclusion
Load balancing is an important means to improve the performance of parallel component programs on heterogeneous clusters. This paper presents a method to optimize the performance of parallel component programs based on quantum particle swarm optimization algorithm. Compared with the existing methods based on dynamic detection and other intelligent optimization algorithms, it can find the optimal solution with less iterations, avoid the overhead of real-time detection, better balance the load on heterogeneous clusters and improve the performance of parallel component programs.

Acknowledgments
This work was supported by the Key Science and Technology Program of Henan Province, China(No.202102210152, No.212102210611); Key R&D and Promotion Project in Henan Province, China(No.192102310212); Research and Practice of Higher Education Teaching Reform in Henan Province, China(No.2019SJGLX386); Science and Technology Plan Project of Anyang City(No.2021C01GX015, No.2022C01NY020); Research and Cultivation Fund Project of Anyang Normal University, China(No.AYNUKPY-2019-24, No. AYNUKPY-2020-25).

References
[1] Object Management Group. (2021) CORBA Web Site. https://www.corba.org/.
[2] Al Muktadir, A.H., Siahaan, I. S. R., Boakye-Boateng, K., et al. (2021) Recommending Location for Placing Data Collector in the OPC Classic. In: 4th IEEE International Conference on Industrial Cyber-Physical Systems. Victoria, BC, Canada. pp. 158-165.
[3] Späth, P. (2021) Java MVC and EJBs. Apress, Berkeley, CA.
[4] Peng, Y., Wang, R. (2014) Research of Parallel Component Non-functional Attributes. Computer Science, 41(8): 213–218.
[5] Peng, Y., Liu, J., Guo, L. (2021) Research of parallel component composition and performance optimization based on agent technology. Application Research of Computers, 38(6): 1819–1824.
[6] Sun, L., Leng, M., Zhu, P., et al (2015) Load Balancing Task Scheduling Algorithm Based on Tabu Search in Cloud Computing. Journal of Chinese Computer Systems, 36(9): 1948–1952.
[7] Ren, S., Zheng, K., Guan, D., et al (2019) The dynamic load balancing method of cloud computing network based on intuitionistic fuzzy time series. System Engineering–Theory & Practice, 39(5): 1298–1307.
[8] Wei, Y., Zhao, K., Zhang, S., et al (2017) Cloud Computing Task Scheduling Based on Improved Ant Colony Algorithm. Fire Control & Command Control, 42(5): 130–133.
[9] Zhang, Y., Jin, Z. (2018) Quantum particle swarm optimization algorithm for high-dimensional multi-modal optimization. Journal of Harbin Institute of Technology, 50(11): 50–58.

[10] Peng, Y., Wei, S. (2014) Extending OpenMP for Optimization of CCA Parallel Component Application. Journal of Chinese Computer Systems, 35(9): 2034–2038.

[11] Lu, J., Li, J., Zhang, Y., et al (2018) Research on the Load Balancing Scheduling Policy of Tasks in Cloud Computing Environments Based on Improved Tabu Search. Journal of Chinese Computer Systems, 39(10): 2254–2259.