Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5084230, 10 pages
https://doi.org/10.1155/2022/5084230

Research Article

Application of Cloud Service-Oriented Heterogeneous Execution Scheduling and VR Technology in Dance Video Teaching

Liu Min

Sport Art College of Guangzhou Sports University, Guangzhou 510500, Guangdong, China

Correspondence should be addressed to Liu Min; 11123@gzsport.edu.cn

Received 14 April 2022; Revised 20 May 2022; Accepted 28 May 2022; Published 24 June 2022

Academic Editor: Vijay Kumar

Cloud computing is the result of the revolution in information technology and a massively complex computing system. As a model for business computing, cloud computing aims to facilitate resource sharing and collaborative work, meet the service needs of users, and generate revenue for cloud service providers. How to reasonably allocate cloud resources, efficiently manage and schedule massive application tasks in real time, reduce the cost of users, and increase the income of cloud service providers on the basis of ensuring the load balance of the cloud computing system and enhancing the utilization of cloud resources is, therefore, one of the research hotspots in the current cloud computing environment. Simultaneously, with the rapid development of human motion simulation and virtual reality technologies, the natural cooperation between humans and computers has become the primary focus of computer science research. The motion capture system is able to track, detect, capture, and record real-time human motion. By analyzing the captured three-dimensional data, we can determine the various characteristics of human motion posture at various times. Due to the potential research and practical value of motion capture technology, it is predominantly used in cutting-edge fields such as animation video production, rehabilitation medicine, sports training, and game software development, thereby effectively realizing the connection between the three-dimensional world and the real world. However, the combination of motion capture technology and educational activities is not universally applicable. This research proposes an approach to dance posture analysis based on matching feature vectors, which can be applied to dance teaching and significantly improves the quality of education and teaching activities. For the purpose of this study, it will be determined whether the combination of dance instruction-based motion capture technology and education is effective and feasible.

1. Introduction

In the past two or three decades, the rapid growth of the social economy has propelled the rapid development of the computer technology industry and improved the needs of the animation production industry [1–3]. Computer graphics, computer virtual reality technology, and computer vision have been applied to a variety of fields, including film and television production, physical training, medical rehabilitation, and other social frontier fields. It offers new hope for the future by combining virtual computer reality with supplementary training to enhance productivity and effectiveness. According to the authors, a new computer system to support teaching and evaluate human postural motion characteristics could better meet educational and teaching objectives, improve the teaching environment and basic equipment, and make digital and scientific teaching a reality.

As a business computing model, cloud computing aims to realize resource sharing and collaborative work, and simultaneously meet the service requests of users and the service revenue of service providers. However, due to the large scale and heterogeneous diversity of cloud computing system resources and the different types of users in the cloud environment, including not only individual users but also enterprise users and service providers themselves, this is not always possible. Therefore, for various types of users, the service request types and QoS target constraints for each user's task are distinct [4]. The aforementioned factors make the task scheduling problem in a cloud environment significantly more difficult than in a traditional distributed

computing environment. Although the problem of task scheduling in the cloud is similar to that in a distributed environment, there are significant differences [5]. In the cloud computing environment, the software and hardware resources of a cloud computing system are of a very large scale. In contrast to conventional data centers and centralized server clusters, a cloud computing system includes not only multiple data centers and server clusters of this scale, but also personal computers, storage devices, network facilities, various mobile terminal devices, etc., and these devices' resources are geographically dispersed and connected to the internet. Therefore, it is extremely difficult to reasonably allocate such vast amounts of resources. Due to the dynamic change in resource status in the cloud computing system, for instance, new resources may connect to the system at any time and some resources may lose connection at any time. Consequently, the task scheduling strategy in a cloud computing environment must track resource changes in real time. Second, a cloud computing system employs virtualization technology to conceal the heterogeneous characteristics of physical resources, allowing the cloud computing environment's task scheduling strategy to be more universal and to support multiple application types. A distributed system's internal resources are isomorphic, and its scheduling strategy can only be designed for specific applications. Lastly, the traditional distributed environment is solely concerned with system performance indicators such as task completion time and system throughput. In the cloud environment, we must not only meet the performance, reliability, resource utilization, and other systematic indicators of the cloud computing system, but we must also consider the resource cost and service income indicators of cloud service providers. Concurrently, we must also meet the multiobjective QoS user task scheduling requirements. Consequently, it makes it more difficult and complex to find a reasonable task scheduling strategy in the cloud computing environment [6].

2. Multiagent Autonomous Scheduling Algorithm Based on Prisoner's Dilemma Game

Through virtualization technology, cloud service providers abstract the physical resources in their data center into virtual resource pools to provide users with virtual resources so as to isolate the physical resources in security. In resource scheduling in the cloud computing environment, a large number of users submit task requests at the same time, and multiple users compete for resources. On the one hand, due to the different QoS requirements of different users, there are different requirements for various types of resources; on the other hand, in the cloud environment, large-scale resources are heterogeneous [7]. In order to meet the QoS constraints of user tasks, such as service time, service reliability, and service cost, it is necessary to evaluate the corresponding service quality indicators of different types of heterogeneous resource sets. Therefore, in order to meet the QoS needs of users and improve the resource utilization of cloud service providers, it is necessary to design a more efficient and reasonable scheduling mechanism.

In recent years, the concept of game theory has been introduced into many fields of task resource scheduling. Game theory is an important economic tool, which is usually used to solve the competitive relationship between multiple individuals, restrict each other, and achieve a certain balance through the process of the game. In the cloud computing environment, due to the conflict and competition between multiple users requesting the same resource, the game theory is just suitable for this type of problem [8].

This study assumes that users have individual rationality and always tend to maximize their own interests. Each user will tend to choose the resources with the best computing power, which makes some resources with strong computing power selected by multiple users, which makes the resource load too large and reduces the overall performance. At this time, the overall resource allocation of the system is not optimal. Due to the conflict of resource allocation, the overall service capacity of the system is reduced, resulting in a prisoner's dilemma between users. Based on this, this chapter proposes a multiobjective scheduling algorithm based on the prisoner's dilemma game, which enables each user to find the optimal resource allocation strategy in a competitive environment [9–11]. Users' different QoS requirements make users have different preferences. For example, some users expect to obtain the minimum service time, while some users expect to obtain the highest reliability and minimum service time. Therefore, the QoS requirements of users are actually multiobjective optimization problems.

In this study, S_k is used to represent the scheduling policy set of a user k. The average service time T_{avg} and service reliability $R(\infty)$ of each scheduling policy can be obtained through the tree scheduling evaluation algorithm, and then, the overall service cost Cost of each scheduling policy can be calculated through the service cost model. This section considers the following three different objectives:

(1) To minimize service T_{avg}, that is, the time when all tasks of user k are successfully completed.

(2) To maximize the service reliability $R(\infty)$, that is, the probability that all tasks of user k are successfully completed.

(3) To minimize the service cost Cost, that is, the sum of the computing resource cost and the transmission channel cost of all tasks of user k.

The purpose of resource scheduling is to allocate all tasks of user k to the resources of various cloud service providers so as to minimize the service time, service cost, and service reliability of the whole task [12]. Therefore, the user's multiobjective optimization problem (MOP) can be calculated by equation (1) as follows:

$$\text{Minimize}\left(T_{\text{avg}}, 1 - R(\infty), \text{Cost}\right). \tag{1}$$

Due to the high complexity of the cloud computing environment, large scale of resources, and scattered geographical location, it is basically unrealistic to design a global

resource management system to centrally control and schedule resources and meet the multiobjective optimization needs of all users. Therefore, aiming at the multiobjective optimization problem of users, this study proposes an autonomous scheduling framework based on game theory. Without central control, according to the competitive game process between users, the prisoner's dilemma theory of game theory is introduced to optimize the allocation of resources among cloud service providers on the premise of meeting the QoS objective constraints of users' tasks. The cloud agent autonomous game process is shown below. The cloud computing agent is responsible for binding the cloud computing system with a high level of service available to each user so that the agent can obtain a reasonable level of service for each cloud computing system [13].

The main process of autonomous game scheduling algorithm includes four steps as follows:

(1) Using cloud agent technology, the cloud agent system assigns a cloud agent B_k to each user k in the cloud environment. The responsibility of the cloud agent is to reasonably allocate the task set J_k submitted by users to the available resources of cloud service providers to form a task resource mapping relationship.

(2) Each cloud agent B_k calculates the Pareto optimal solution set of user k's task scheduling strategy through the NSGA-II algorithm. In the process of computing, it is assumed that the resources in the whole cloud computing environment are exclusive to B_k.

(3) Cloud agent B_k selects a specific scheduling strategy from the Pareto optimal solution set according to the QoS demand preference of user k, that is, the benefits valued by user k.

(4) Because each cloud agent B_k selects the resources most suitable for its own interests according to its preferences, it achieves individual optimization. In order to achieve global optimization, each cloud agent searches for a competitive global optimal scheduling strategy to achieve Nash equilibrium through the prisoner's dilemma game in the process of maximizing their own interests.

The above algorithm flow is actually a game model between cloud agents. As a game participant, the cloud agent takes its nondominated solution set s^* as the strategy space, the user's q0s target preference as its income function, and the game result of the final global optimal strategy to form a complete resource scheduling game model. Among them, the design of the income function is the key point of the resource scheduling game. Different objective optimization of users leads to different income functions, which has an important impact on the results of the game. The following chapters will analyze and design the above steps in detail [14].

In this study, a complete mapping of all tasks of users to the resources of various cloud service providers is defined as a scheduling policy S_i (which can be calculated by equation (2)) as follows:

$$S_i = \left\{\left(J_{i,1}, R_p\right), \left(J_{i,2}, R_q\right), \ldots, \left(J_{i,t}, R_z\right)\right\}, \tag{2}$$

where $J_{i,j}$ represents the jth task of user i, and $p, q, \ldots, z$ represent resources from different cloud service providers. The cloud agent B_k acts in a completely selfish way and tries to choose the resources that maximize its own benefits, regardless of the overall benefits of the overall system. The noncooperative game model between cloud agents can be formally defined as a triple $(B, \overline{S}, \Xi)$, where B is the set of cloud agents, $B = \{B_1, B_2, \ldots, B_n\}$. For any cloud agent $B_k\{B_k \in BV\}$, we select a feasible solution $S_k\,(S_k \in S_k^*)$ in its Pareto optimal solution set, and the set $\overline{S} = (S_1, \ldots, S_k, \ldots, S_n)$ composed of these feasible solutions constitutes the global scheduling strategy combination of the whole system. The agent has a utility $\Xi\,(\mathrm{S_k} \in \overline{\mathrm{S}}) \longrightarrow \mathrm{R}$ for any state of the strategy $B_k\,(B_k \in B)$.

Among them, the scheduling policy $S_k \in S_k^*$ of an individual constituting the global scheduling policy S_k^* represents the set of all available scheduling policies of the agent B_k, which describes the behavior that the agent will choose in various scenarios. For each agent B_k, the result of the game is determined by its utility function. The utility function of a cloud agent is a function of its own scheduling strategy S_k and the scheduling strategy $\overline{S} = (S_1, \ldots, S_k, \ldots, S_n)$ of other agents.

For users, service reliability, service time, and service cost have an important impact on the successful completion of their requested tasks. On the other hand, for cloud service providers, resource utilization, the conflict between resources, and system load balancing have an important impact on their service capabilities. For cloud agents, the reliability of services and the cost of task scheduling are very important to maximize their benefits [15].

Therefore, this study takes service reliability and service cost as cloud agent utility function parameters to calculate the income of each cloud agent. At the same time, in order to achieve the global balance of the system, the model encourages each agent to make appropriate concessions so as to realize the overall load balance of the system, and this study designs a reasonable reward principle to encourage those agents who make more contributions to the overall load balance of the system. Therefore, the utility function $\Xi\,(S_k)$ of the scheduling policy S_k selected by cloud agent B_k is defined as follows in equation (3):

$$\Xi\left(S_i\right) = \frac{\mathrm{Cost}_i}{R_i\,(\infty)} - G_i. \tag{3}$$

In the global scheduling strategy combination s of each game, all cloud agents obtain reward factors by adjusting corresponding scheduling strategies according to users' preferences to improve their own benefits [16]. If the reward factors of all cloud agents are not enough to make adjustments, the whole game process reaches an equilibrium state, which is called Nash equilibrium (NE). The Nash equilibrium defines a basic stable state in the system. When this state is reached, all participants in the system cannot unilaterally change their behavior to improve their own income. If we can find a Nash equilibrium state in the

process of the cloud agent game and the utility function of each cloud agent is within its acceptable range, we can find a globally optimal system state.

We suppose there is a global scheduling policy combination S: $S = \{S_1, \ldots, S_i, \ldots, S_n\}$. For any cloud agent B_i, we replace its scheduling policy S_i with any S_i' to form a new global scheduling policy S': $S' = \{S_1, \ldots, S_i', \ldots, S_n\}$, where only the ith element $S_i' \in S_i^*$ is different between S' and S. If for any S': $\Xi(S) \leq \Xi(S')$, then the global scheduling policy S is in Nash equilibrium. Nash equilibrium state is the result of each agent user's pursuit of maximizing its own benefits. Although it can reach the equilibrium state, such an equilibrium state may not reach the overall optimization, that is, to maximize the overall benefits as shown in equation (4) as follows:

$$\text{Opt} = \min_{S \in S^*} \Xi(S). \tag{4}$$

In fact, to achieve the ideal global benefit maximization, Opt is an ideal situation. For the whole system, among these possible Nash equilibria, we are always interested in the optimal Nash equilibrium [17]. However, finding such a Nash equilibrium is not always feasible, at least in an acceptable calculation time. Therefore, in view of the above situation, this study tries to explore the approximation of an optimal equilibrium state through the limitation of the quality of service of the system. Through iteration, the performance or quality-of-service indicators of the corresponding scheduling strategy combination is gradually refined until a threshold acceptable to the agent user (the threshold of the utility function) is found so that the overall system reaches an approximate optimal equilibrium state. Generally, the threshold is associated with a reasonable number of steps $T_{\max}$ of successful iteration obtained from experience. In $T_{\max}$ iteration steps, the overall global benefit of the system will not continue to improve so as to reach a stable state.

Generally, in a specific game context, the overall performance of Nash equilibrium is measured by the PoA (price of anarchy) index. The index is defined as the ratio of the benefit of the worst Nash equilibrium scheme to the benefit of the optimal solution. This indicator (in equation (5)) reflects the cost of the lack of a unified coordination mechanism.

$$PoA = \frac{\left(\max_{S \in N} \Xi(S)\right)}{\text{Opt}}, \tag{5}$$

where N is the number of all Nash equilibrium states in the game process. On the other hand, scholars introduced the PoS (price of stability) index, which is defined in equation (6), which is the ratio of the benefit of the best Nash equilibrium scheme to the benefit of the optimal solution.

$$PoS = \frac{\min_{S \in N} \Xi(S)}{\text{Opt}}. \tag{6}$$

The PoS is very important in the game process where there are some internal control mechanisms that have a direct or indirect impact on the behavior of participants. This index can be regarded as an optimistic PoA index to measure the impact of these control mechanisms on the game results.

3. Dance Pose Analysis Method Based on Feature Vector Matching

An effective posture analysis approach based on feature plane similarity matching is suggested to suit the demands of fast and accurate human pose identification. Real-time human motion data collection allows for accurate skeleton extraction and human feature plane extraction. Then, taking the plane eigenvector and its included angle as the discrimination basis of attitude analysis, an efficient matching mechanism is established. This method is combined with dance teaching. After experimental verification, it not only provides a stable and accurate analysis of human posture but also can effectively obtain the differences in human movements, which provides good theoretical support for dancers' scientific dance training [18].

This study will utilize an optical motion capture device to collect motion data to construct a library of human motion posture and skeletal models that will be released later. As illustrated in Figure 1, the fundamental procedure is as follows.

According to Figure 2, the performer first puts on monochrome clothes with 21 marker identification points pasted on key parts, then stands inside the preset motion space, starts the high-precision 3D motion capture software, sets the time, and performs the required dance action. The collected and modified human motion data of the selected 21 marker identification sites are matched with the actor model to complete the human motion posture database. Then, the 21 marker identification points are activated to finish the actor model data matching.

The application of a motion capture system to analyze human motion posture is to estimate the motion posture characteristics of the human body from different perspectives. In this study, the key points of motion characteristics are used to mark the key parts of the human body, and the lines between the key points represent rigid bodies, so their shapes will not change at any time. The main research on motion posture is to build a human motion posture database based on the head, body, hip, and limbs, as shown in Figure 3.

In this paper, moving images are used to analyze the characteristics of human poses to derive useful motion pose metrics. By combining motion analysis and instruction, the teaching system may become more individualized and unique. It may also deconstruct the dancers' movements and individually demonstrate each step. The generated parameters can be used to quantitatively analyze action posture and aid in the scientific and intelligent teaching of dance.

A method for analyzing human motion posture based on feature plane similarity matching is provided. This method converts the standard Euclidean distance calculation based on multiple identification points to a

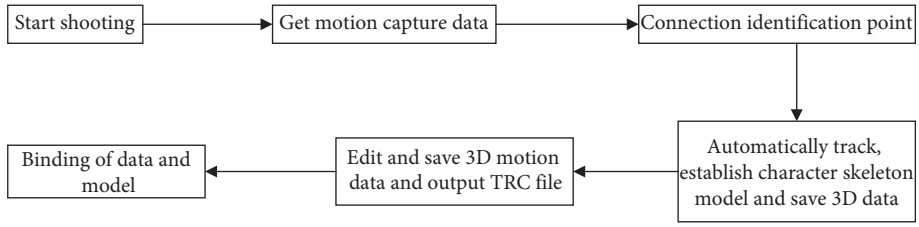

Figure 1: 3D data acquisition flowchart.

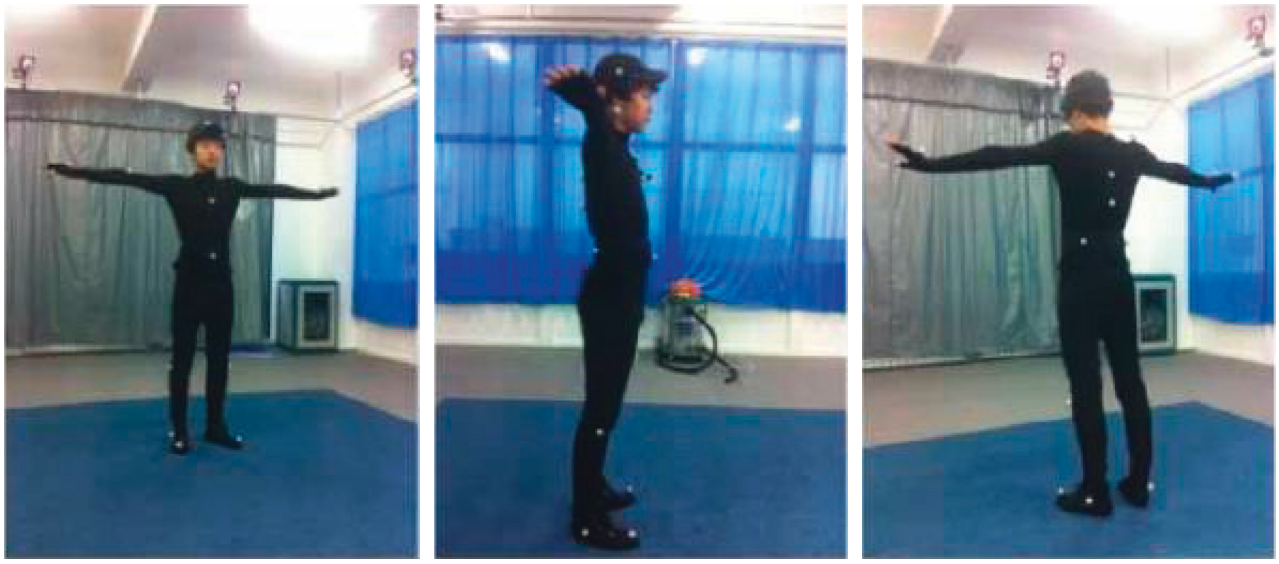

Figure 2: The actor dressed posted marker.

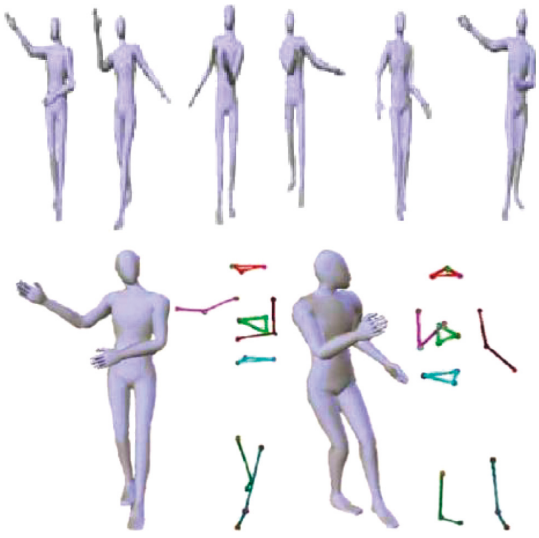

Figure 3: Dance posture.

calculation based on the feature plane and a feature vector. Using the identifying points of 21 primary pieces, the motion difference and correlation are computed. This method can rapidly evaluate human motion posture and increase the efficiency of dance instruction by applying the results to dance education. Figure 4 depicts the procedure in its entirety.

Among the primary steps of the analysis are the following:

Step 1: Real-time skeletal data acquisition: optical motion capture of dancing movement sequences and storing of coordinates of each manikin identification point in the spatial coordinate system.

Step 2: Pose analysis: based on the motion characteristics of the important components of the dance movement, the feature correlation coefficient of human posture is estimated.

Step 3: we compare students' dancing motions to standard movements using the correlation coefficient of the feature vector and its included angle.

Human motion is a complicated process that may be simplified as a chain system connected by a rigid body component without addressing muscle and brain activation. The upper limb is made up of the upper arm and forearm, two rigid bodies joined by the elbow joint, whereas the lower limb is made up of the thigh and calf, joined by the hip joint and the knee joint. A line linking nodes represents the head, body, and hip.

Human posture similarity is a measure of human posture difference or similarity. The most often used approach nowadays is Euclidean distance.

The Euclidean distance is computed using equations (7) and (8) as follows:

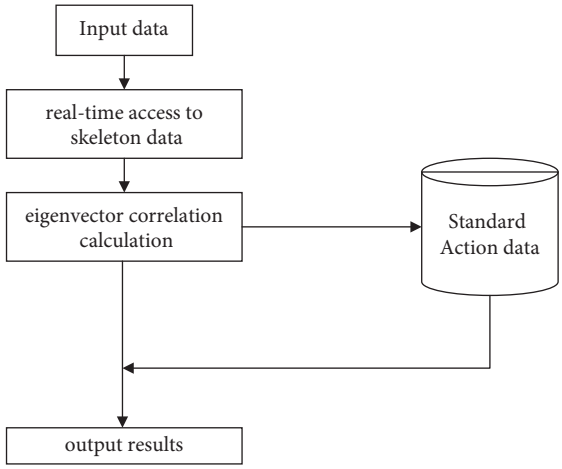

Figure 4: Method flowchart.

$$D = \mathrm{sqrt}\left((x_1 - x_2)^2 + (y_1 - y_2)^2 \right), \tag{7}$$

$$D = \mathrm{sqrt}\left((x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \right), \tag{8}$$

where $x_1 = (x_{11}, x_{12}, x_{13}, \ldots, x_{1n}), x_2 = (x_{21}, x_{22}, x_{23}, \ldots, x_{2n})$ are n-dimensional data. Through the use of a calculation method based on Euclidean distance, the difference between any two identification points may be calculated for any pair of identification points. If the difference between the two identification points is smaller than the threshold, which is determined by the coach, it is assumed that the two identification points are comparable. If the difference between the two identification points is more than this threshold, the two points are regarded to be not comparable. The track of the same identification point of the standard action and the action to be tested is compared, as shown in Figure 5.

This method is based on traditional Euclidean distance direct comparison: this method simultaneously compares two moving targets' moving trajectories, calculates the corresponding distance difference during alignment of each action sequence, and calculates the data matching degree based on a predefined threshold. In addition to requiring an excessive amount of calculation, this method also relies on the intrinsic qualities of the item under consideration. When the body ratio of an object changes such as when it becomes taller, shorter, fatter, or thinner, the distance between the identification points must be recalculated and reset. In order to meet the stringent criteria for moving objects, traditional methods are restricted to specific situations, which not only reduce their computational efficiency but also reduce their applicability across the board.

The motion capture technology is employed in this research to extract the human skeleton model, which is shown in detail. First, three feature points may be used to define the shape of a feature plane, and the skeleton's 21 identifying points may be converted into seven feature planes, which serve as the basis for the basic calculation plane. As seen in Figure 6, each feature plane represents a different component of the human body. Seven normal vectors of the feature plane $(V_1 \sim V_7)$ are extracted from the feature plane in

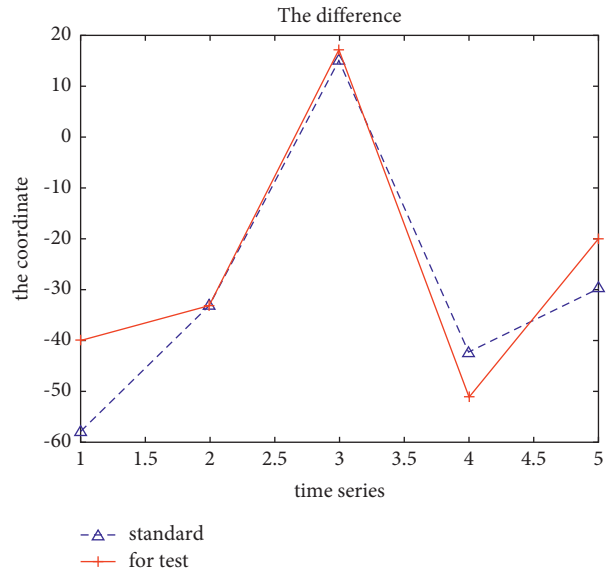

Figure 5: A single identity between unilateral upward motion path.

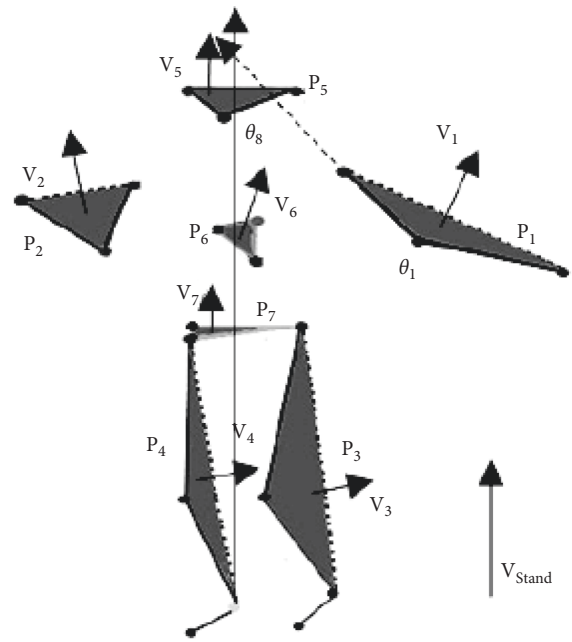

Figure 6: Human skeletal features' planar graph.

order to judge the difference in the overall direction of the motion attitude; second, the local normalization of the motion attitude plane is judged based on the angle of the edge vector $(\theta_1 \sim \theta_7)$ of the feature plane that has been included in the motion attitude plane during the extraction process. In addition, the included angle $(\theta_8 \sim \theta_{12})$ of the edge vector of the feature plane and the vertical direction of the trunk are taken into account when determining the local relationship between the limb and the trunk, respectively. In this manner, a binary is utilized as the input to the model similarity calculation, and the correlation parameter is employed as the result of that computation. In this approach, the calculation error caused by the intrinsic properties of the

item to be measured is successfully solved, and the complexity of the calculation is reduced, while the efficiency and stability of human posture analysis are improved. The human skeletal features' planar graph is shown in Figure 6.

According to the standardization of dance movement requirements and the relative activity range of human skeleton movement, the feature vectors and included angles of main action parts of the human body are specified as follows, as shown in Tables 1 and 2:

3.1. Limbs. Using the inner product of the limb feature plane, it is possible to calculate the movement direction of limbs $P_m\,(m = 1, 2, 3, 4)$, normal vector $V_m\,(m = 1, 2, 3, 4)$, and trunk vertical direction vector V_{stand}, and the normalization of limb movement can be accurately judged by the included angle $\theta_m\,(m = 1, 2, 3, 4)$.

3.2. Trunk. The trunk is comprised of the chest and hips. It is mostly composed of rotational and bending movements. Allowing for rotational movement, the angle between the spine direction vector and the vertical standing vector is altered. The normal vector of the hip plane is perpendicular to the vertical direction when the human body is vertically standing. This is important for the bending motion.

The cosine similarity approach is utilized as the similarity function in this work, and it can be used to assess not only the difference in the direction of vectors but also the similarity and difference in angles between two vectors. Using the cosine of the angle formed by the product of two vectors in space, it is possible to calculate the difference between two feature vectors in a given space. The following equation (9) illustrates the computation method as follows:

$$\text{similarity}\left(V_i, V_j\right) = \frac{V_i \times V_j}{\sqrt{\left(V_i\right)^2} \times \sqrt{\left(V_j\right)^2}}, \tag{9}$$

where $\overrightarrow{V}_i, \overrightarrow{V}_j$ are the corresponding attitude eigenvectors. Cosine similarity is a more accurate measure of the angular separation between two vectors than Euclidean distance. In this way, the difference in feature vector directions may be assessed, and the calculation mistake caused by various Euclidean distance target positions can be avoided. The estimated cosine value falls inside the [0, 1] range. For example, a result near 1 implies that the dance action to be evaluated is in line with the standard action; if it is less than 1, it means there is a significant divergence from the standard movement equation (10).

$$\theta_{\langle i,j\rangle} = \arccos\left(\text{similarity}\left(V_i, V_j\right)\right). \tag{10}$$

The effective motion range $\theta_{\langle i,j\rangle}$ and value range $[\theta_{\min}, \theta_{\max}]$ between feature vectors are used to judge the standardization and normalization of dance action amplitude. If $\theta_{\langle i,j\rangle} < \theta_{\min}$ or $\theta_{\langle i,j\rangle} > \theta_{\max}$, the next procedure cannot be performed because the matching limb movement is incorrect. If $\theta_{\min} \le \theta_{\langle i,j\rangle} \le \theta_{\max}$, it is then possible to perform the following procedure, as the matching limb movement is within a discernible range of motion.

Table 1: Features in the plane of discriminant vectors.

Feature plane	Feature vector
Left arm ($P1$)	$V_1 = V_{\text{Llarm}} \times V_{\text{Lfarm}}$
Right arm ($P2$)	$V_2 = V_{\text{Rlarm}} \times V_{\text{Rfarm}}$
Left leg ($P3$)	$V_3 = V_{\text{Lthigh}} \times V_{\text{Lcrus}}$
Right leg ($P4$)	$V_4 = V_{\text{Rthigh}} \times V_{\text{Rcrus}}$
Head ($P5$)	$V_5 = V_{\text{Lhead}} \times V_{\text{Rhead}}$
Chest ($P6$)	$V_6 = V_{\text{Lchest}} \times V_{\text{Rchest}}$
Hip ($P7$)	$V_7 = V_{\text{Lhip}} \times V_{\text{Rhip}}$

Table 2: Discriminable angle on feature plane.

Feature plane	Geometric relation	$\theta_{\max}$	$\theta_{\min}$
Left arm ($P1$)	$\theta_1 = \langle V_{\text{Llarm}}, V_{\text{Lfarm}}\rangle$	180°	40°
Right arm ($P2$)	$\theta_2 = \langle V_{\text{Rlarm}}, V_{\text{Rfarm}}\rangle$	180°	40°
Left leg ($P3$)	$\theta_3 = \langle V_{\text{Lthigh}}, V_{\text{Lcrus}}\rangle$	180°	35°
Right leg ($P4$)	$\theta_4 = \langle V_{\text{Rthigh}}, V_{\text{Rcrus}}\rangle$	180°	35°
Head ($P5$)	$\theta_5 = \langle V_{\text{Lhead}}, V_{\text{Rhead}}\rangle$	—	—
Chest ($P6$)	$\theta_6 = \langle V_{\text{Lchest}}, V_{\text{Rchest}}\rangle$	—	—
Hip ($P7$)	$\theta_7 = < \langle V_{\text{Lhip}}, V_{\text{Rhip}}\rangle$	—	—

Because the percentage of the human body is fixed, even though there are individual variances in height, weight, arm length, and so on, as a result, the similarity of the angle must be used to determine whether the limb's motion amplitude matches the criterion. It is in equation (11) that you'll find the calculating method as follows:

$$\text{Corr}\left(\theta_{\langle i,j\rangle}\right) = 1 - \left(\frac{\arccos\left(\text{similarity}\left(V_i, V_j\right)\right)}{\pi}\right). \tag{11}$$

A dance trainer's left arm movement may be used as an example, and its characteristic plane can be used as the primary computation plane. $\{\text{Sim}\,(V_1, V_{\text{stand}}), \text{Corr}\,(\theta_1), \textit{Corr}\,(\theta_8)\}$ can be obtained, respectively, on the basis of the above three parameters to determine the overall motion posture of the left arm. Table 3 shows the experimental findings, which demonstrate that the calculation error has been substantially minimized.

4. Experiment and Results

4.1. Experimental Verification and Analysis of Multiagent Autonomous Scheduling Algorithm Based on Prisoner's Dilemma Game. This study demonstrates the efficacy of this algorithm by simulating a multiagent competition utilizing cloud service provider resources. The experiment established 16 separate users. Each user submitted to the cloud agent system a set of randomly generated tasks. Each user's task set includes 1,000 tasks. In the experiment, there are 4 resource allocation scenarios, with 8 resources in scenario 1, 16 resources in scenario 2, 32 resources in scenario 3, and 64 resources in scenario 4. After receiving the task request from the user, each cloud agent calculates the Pareto optimal scheduling policy set for the user using the NSGA-II algorithm. The number of groups (maximum number of iterations) is set to 100, the number of individuals in each generation group is set to 200, the gene crossover probability is set to 0.80, and the gene mutation probability is set to 0.10.

Table 3: Left arm movements and gestures' associated parameters (0-1 s).

Experimental object	$\mathrm{Sim}(V_1, V_{\mathrm{stand}})$	$\mathrm{Corr}(\theta_1)$	$\mathrm{Corr}(\theta_8)$
Standard object	0.7854	0.9612	1.0000
Object to be tested	0.8012	0.8395	0.9887

The number of population and the number of individuals in the population determine the calculation time of the algorithm, while the probability of gene crossover and gene mutation determines the convergence and diversity of the calculation results. In order to ensure the validity of statistical data, each scenario's experiments are repeated 50 times under the same load. In the final Pareto optimal scheduling strategy set, the nondominated solutions from each execution result are compiled.

After searching the solution set space with the preceding algorithm, each cloud agent obtains the Pareto: optimal scheduling policy set for the respective user. All scheduling policies in the set are Pareto optimal only for specific users, that is, when all resources are exclusively shared by the user. Next, each cloud agent will adopt different scheduling policy selection rules based on the specific needs of users and select the scheduling policy from the Pareto optimal scheduling policy set that best meets the specific needs of users.

Next, the multiagent autonomous game scheduling algorithm is used to simulate the process of 16 cloud agents competing to use resources in four resource allocation scenarios using a prisoner's dilemma game. The maximum number of game iterations is set to 200.

To validate the efficacy of the multiagent autonomous game scheduling algorithm, we conduct comparative experiments on four static scheduling policy selection rules for each scenario: (1) optimize reliability policy, (2) optimize cost policy, (3) optimize cost with deadline policy, and (4) optimize balance policy. In the four static rule experiments, each cloud agent uses the same scheduling policy selection rule throughout the entire scheduling cycle; thus, the local scheduling policy chosen by each cloud agent remains constant. In the experiment of a multiagent autonomous game scheduling algorithm, each cloud agent will dynamically adjust its local scheduling strategy based on the local scheduling strategies of other agents in the scheduling cycle. Figure 7 illustrates the service time and service reliability of the obtained global scheduling strategy according to five rules. The first four rules correspond to the aforementioned static scheduling strategy selection rules, while the fifth rule corresponds to the algorithm's dynamic scheduling strategy selection rules.

Figure 7 depicts the experimental findings that, for the same task load, different resource allocation has a decisive effect on task completion time. In the scenario of configuring eight resources, it takes 6972 seconds to complete all tasks for all users, in the scenario of configuring sixteen resources, it takes 4215 seconds to complete all tasks, in the scenario of configuring 32 resources, it takes 3208 seconds to complete all tasks, and in the scenario of configuring 32 resources, it only takes 1412 seconds to complete all tasks. The relationship between the service performance of a cloud service provider's resource collection and the number of resources is essentially linear. In the same resource configuration scenario, the scheduling policy selection rules adopted by each agent also have a substantial effect on the task completion time. Figure 7 demonstrates that regardless of the resource allocation scenario, the service time of the global scheduling strategy obtained by applying the "service cost optimization rule" is the longest. This is because the selection rule tends to allocate tasks to low-cost resources for execution, and does not prioritize execution speed and reliability. The dynamic selection rules adopted by the autonomous game scheduling algorithm have obvious advantages over the other four static selection rules. In a scenario of resource competition, it is difficult to reach a compromise between scheduling strategies with identical optimization objectives. Multiple cloud agents will assign different tasks to the same set of resources, resulting in resource congestion and an imbalanced overall resource load. As a result of each agent's adoption of static selection rules, they will not alter their decisions in response to an imbalance in resource allocation, resulting in a decline in the system's overall service performance.

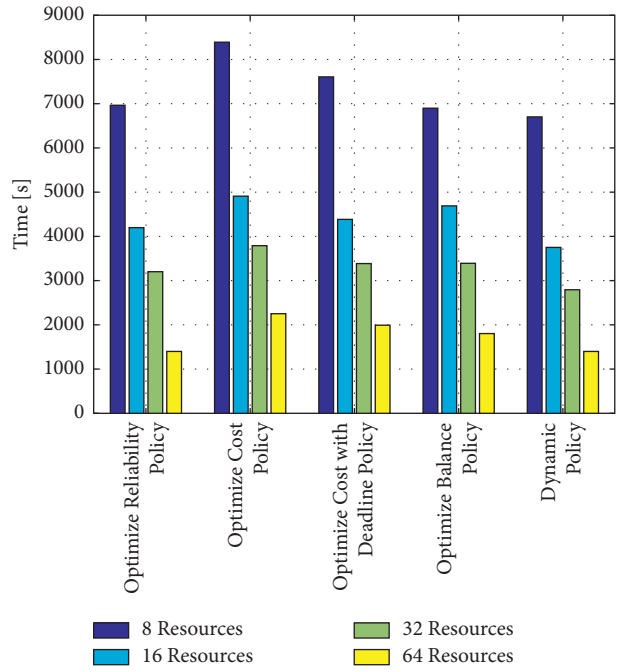

Figure 7: Service completion time.

4.2. Experimental Verification of Dance Pose Analysis Method Based on Feature Vector Matching. In order to fully verify the practical and effective impact of the use of capture technology on the effect of dance teaching, this study adopts the method of comparative experiment, takes the students of 2015 Elective Aerobics in our college as the research object, and carries out teaching verification in the form of grouping. The specific methods are as follows: Table 4

Table 4: Comparison of experimental results.

Group	Action amplitude	Action strength	Movement coherence	Action standardization
Routine group	86.1 ± 5.3	79.1 ± 7.3	80.6 ± 7.3	77.4 ± 6.3
Experience group	83.31 ± 5.9	82.1 ± 4.3	84.4 ± 3.3	81.3 ± 4.3
Significance (P)	0.03	0.024	0.021	0.047

P stands for the significance of the experimental effect, generally $P < 0.05$. According to the experimental results, the students who study with motion capture have a better mastery of dance movements than those who study according to routine.

Step 1 grouping of experimental objects: 60 students who choose aerobics are divided into routine group and experimental group by drawing lots. First, through a simple questionnaire survey, we can understand the two groups of students' learning motivation, initiative, degree of interest, basic movement mastery, physical fitness, and learning efficiency of aerobics. The experimental results show that there is no significant difference between the two groups of students in terms of learning motivation, learning interest, and learning efficiency ($P > 0.05$). Then, the students who take this course are tested for their theoretical knowledge. The results show that there is no significant difference between the two groups ($P > 0.05$). In conclusion, it can be determined that the selection of experimental objects is representative and meets the requirements of this experiment.

Step 2 experimental process: this experiment is conducted from October 2015 to December 2015, and the course learning is 16 class hours. When it comes to studying aerobics, the students in the routine group prefer to learn in an outside environment. The teachers explain, illustrate, identify difficulties, and correct errors in the students' actions. The action sequence taught by the teachers is followed by the pupils throughout their learning. The students in the experimental group report to the assigned laboratory, where they first see a three-dimensional dance animation movie created using motion capture technology, followed by an aerobic lesson using the motion capture equipment. By using the capture equipment and computer to process the learning data, the students' learning situation can be obtained and played back in real time, and the students' nonstandard actions can be corrected in time.

Verification of Step 3 experimental results: after completing all teaching hours, the learning situation of the two groups of students is verified by examination. The college physical education teacher scores the range, strength, consistency, and technical standardization of aerobic actions of the two groups of students. The results are determined by SPSS19.0 analysis. The test results are shown in Table 4.

5. Conclusions

In recent years, as the size of the internet network has continued to grow, the business volume that the cloud computing system must manage has also significantly increased. In recent years, research into cloud computing task scheduling has been one of the most active and technically challenging areas. Cloud computing system resources have the characteristics of large scale, heterogeneous resources, diverse application types and service quality requirements, huge data processing capabilities, diverse user groups, and strong dynamics. Aiming at the aforementioned characteristics of the cloud computing system, this study systematically investigates how to rationally allocate massive resources in cloud computing systems, how to efficiently execute user tasks, how to improve the resource utilization of cloud computing systems based on load balancing, and how to reduce the cost of task scheduling and increase the revenue of cloud service providers.

In the investigation of dance education in colleges, motion capture technology is employed. The dance movements are demonstrated in sections through the tracking, capturing, inspecting, and recording of human motion, which eliminates the need for repeated demonstrations in traditional teachers' teaching, eliminates interference from students or teachers due to personal differences, and psychological and physical factors, and finds and corrects problems in real time through the effective analysis of computer data. It significantly affects the efficacy of education and instruction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] X. Liu, C. Wang, B. B. Zhou, J. Chen, T. Yang, and A. Y. Zomaya, "Priority-based consolidation of parallel workloads in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1874–1883, 2013.

[2] L. Cherkasova, D. Gupta, and A. Vahdat, *When Virtual Is Harder than Real: Resource Allocation Challenges in Virtual Machine Based it Environments*, Hewlett Packard Laboratories, England, UK, 2007.

[3] J. Koomey, *Estimating Total Power Consumption by Servers in the U.S. And the World*, Lawrence Berkeley National Laboratory, Stanford University, California, CA, USA, 2007.

[4] H. Zhang, G. Jiang, K. Yoshihira, and C. Haifeng, "Intelligent workload factoring for a hybrid cloud computing model," in *Proceedings of the 2009 IEEE World Conference on Services-I*, pp. 701–708, IEEE, Los Angeles, CA, USA, July 2009.

[5] K. Li, X. Tang, and Q. Yin, "Energy-aware scheduling algorithm for task execution cycles with normal distribution on heterogeneous computing systems," in *Proceedings of the 41st International Conference on Parallel Processing*, pp. 40–47, IEEE, Pittsburgh, PA, USA, December 2012.

[6] V. Shestak, J. Smith, H. J. Siegel, and A M. Anthony, "A stochastic approach to measuring the robustness of resource allocations in distributed systems," in *Proceedings of the 41st International Conference on Parallel Processing*, pp. 459–470, IEEE, Columbus, OH, USA, August 2006.

[7] D. Poola, S. K. Garg, R. Buyya, and Y. Yun, "Robust scheduling of scientific workflows with deadline and budget constraints in clouds," in *Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications*, pp. 1–8, IEEE, Victoria, Canada, May 2014.

[8] J. Dejun, G. Pierre, and C. H. Chi, "EC2 performance analysis for resource provisioning of service-oriented applications," in *Proceedings of the 2009 ICSOC/ServiceWave Workshops on Service-Oriented Computing*, pp. 197–207, DBLP, Stockholm, Sweden, January 2010.

[9] K. R. Jackson, L. Ramakrishnan, and K. Muriki, "Performance analysis of high performance computing applications on the amazon web services cloud," in *Proceedings of the Second IEEE International Conference on Cloud Computing Technology and Science*, pp. 159–168, IEEE, Indianapolis, IN, USA, January 2010.

[10] C. Allison and E. S. Redhead, "Factors influencing orientation within a nested virtual environment: external cues, active exploration and familiarity," *Journal of environment psychology*, vol. 51, 2017.

[11] Y.-H. Hung, C.-H. Chen, and S.-W. Huang, "Applying augmented reality to enhance learning: a study of different teaching materials," *Journal of Computer Assisted Learning*, vol. 33, 2017.

[12] Xi Zhang, S. Jiang, and Ordonez de Pablos, "Patricia. How virtual reality affects perceived learning effectiveness: a task-technology fit perspective," *Behaviour & Information Technology*, vol. 36, 2017.

[13] M. Slater, Sanchez-Vives, and V. Maria, "Enhancing our lives with immersive virtual reality," *Frontiers in robotics and AI*, vol. 3, 2016.

[14] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[15] Q. Chen, A. Albarakati, and L. Gui, "Research on motion capture of dance training pose based on statistical analysis of mathematical similarity matching," *Applied Mathematics and Nonlinear Sciences*, vol. 3, 2021.

[16] R. Collins, A. Lipton, and T. Kanade, *A System for Video Surveillance and Monitoring*, Carnegie Mellon University, the Robotics Institute, Pittsburgh, PA, USA, 2000.

[17] R. Okada, B. Stenger, and N. Kondoh, "A video motion capture system for interactive games," in *Proceedings of the MVA*, pp. 186–189, DBLP, Tokyo, Japan, January 2007.

[18] J. Chai and J. K. Hodgins, "Performance animation from low-dimensional control signals," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 686–696, 2005.